Что такое Test IT?

О Руководстве пользователя Test IT

Это руководство описывает работу с Test IT в следующих поставках:

- **Test IT Про** (серверная поставка) — последняя версия
- **Test IT Базовый** и **Стандарт** (Облако) — текущая версия

Версии различных поставок выходят в разное время, поэтому возможны незначительные различия между интерфейсом на изображениях и в вашей версии Test IT.

Test IT — это платформа управления тестированием ПО (TMS), созданная, чтобы сделать процесс тестирования ПО более эффективным и удобным для тестировщиков. Test IT представляет собой единое пространство для ручных и автоматизированных тестов, позволяющее оптимизировать работу тестировщиков, выстроить прозрачные процессы в отделе QA и обеспечивающее детальную отчетность по качеству тестируемого ПО.

Ручное тестирование

Для инженеров ручного тестирования

- Создавайте тест-кейсы и чек-листы в библиотеке тестов.
- Генерируйте тест-кейсы с помощью ИИ-моделей.
- Выделяйте повторяющиеся в тестах действия в общие шаги.
- Создавайте версии тест-кейсов и актуализируйте их в любое время.
- Создавайте пользовательские атрибуты и наполняйте тестовую документацию нужными вам данными.
- Используйте параметры для масштабирования тест-кейсов.
- Импортируйте тесты из других систем управления тестированием (TMS) с помощью мигратора.

Автоматизированное тестирование

Для AQA-инженеров

- Запускайте автотесты из UI с помощью вебхуков с нужными параметрами.
- Загружайте результаты автотестов с помощью Test IT CLI или адаптера.
- Отслеживайте результаты прогонов.
- Настройте автоматический перезапуск автотестов и автоотметку нестабильных автотестов.
- Анализируйте причины падения автотестов и группируйте типовые ошибки.
- Создавайте шаблоны автотестов из ручных тестов с помощью плагина Test IT Менеджмент.

Управление проектами и командами

Для тест-менеджеров и тимлидов

- Создавайте тест-планы и добавляйте в них статические и динамические тестовые наборы для различных конфигураций устройств, ОС и браузеров.
- Подключите систему к таск-трекеру и создавайте задачи из интерфейса Test IT.
- Назначайте роли и настраивайте права доступа.
- Получайте системные уведомления при назначениях и упоминаниях.
- Повышайте вовлеченность вашей команды с помощью геймификации.

Аналитика

Для всех, кто хочет видеть результаты

- Получайте отчет по тест-планам, ручным и автоматизированным тестам в едином формате.
- Визуализируйте аналитику с помощью проектных и персональных дашбордов.
- Настраивайте дашборды под нужные метрики и показатели.

Не знаете, с чего начать? — Вам поможет страница Быстрый старт.

Это руководство можно скачать в PDF.

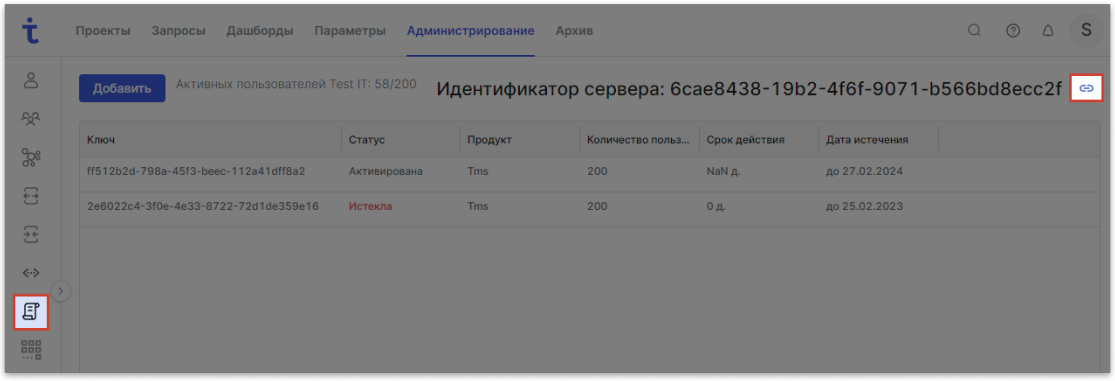
Обновлено: 10/19/26, 5:51 PM

Добавление лицензий

Чтобы добавить лицензию, вам нужно получить ключи лицензий для ваших пользователей и добавить их в ваш аккаунт Test IT.

Чтобы получить ключи лицензий:

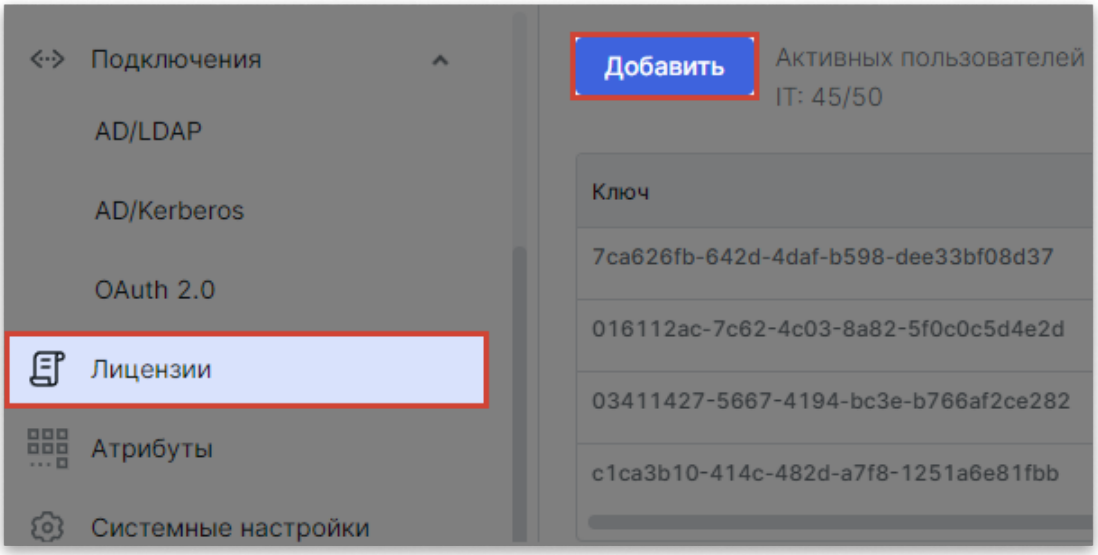
1. Откройте раздел **Администрирование → Лицензии**.
2. Скопируйте идентификатор сервера, нажав значок копирования.



3. Отправьте сообщение на электронный адрес отдела продаж Test IT: order@testit.software . При обращении укажите идентификатор сервера в теме письма или в его тексте.
   Ключи лицензий будут отправлены в ответном сообщении.

Чтобы добавить ключ:

1. Откройте раздел **Администрирование → Лицензии**.
2. Нажмите **Добавить**.



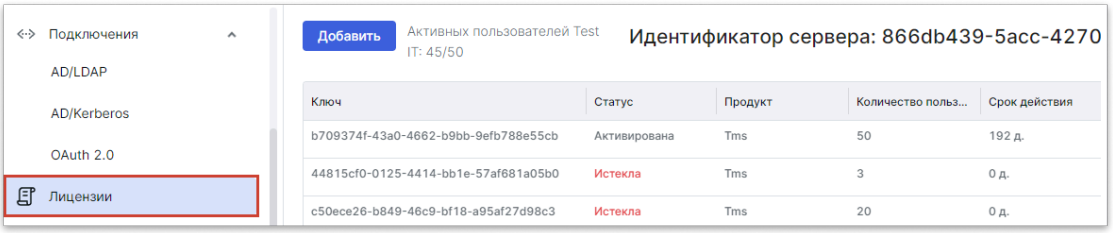
3. Скопируйте ключ в соответствующее поле в открывшемся окне и нажмите **Добавить**.

> **Важно**
>
> Вы можете использовать систему Test IT для работы 25 пользователей в течение 30 дней после установки. При добавлении лицензии все данные в системе сохранятся. При входе в аккаунт администратора вы получите уведомление об оставшемся сроке пробного периода.

Просмотр информации о добавленных лицензиях

Чтобы найти информацию о лицензиях:

1. Используя аккаунт администратора Test IT, в верхней навигационной панели нажмите **Администрирование**.
2. В левом навигационном меню перейдите в раздел **Лицензии**.



После добавления лицензии на экране отобразится информация о ней:

- Ключ лицензии
- Статус (**Активирована/Истекла/Неправильный Server ID**)
- Количество пользователей в рамках данной лицензии
- Срок действия лицензии (обновляется автоматически при добавлении нового ключа и продлении лицензии)

Над таблицей с информацией по лицензиям отображается количество активных пользователей в системе — сумма активных пользователей всех лицензий со статусом **Активирована**.

Если активная лицензия отсутствует

Для доступа ко всем возможностям Test IT требуется активная лицензия. При отсутствии активной лицензии (например, если она истекла), система Test IT работает преимущественно **в режиме просмотра** с возможностью поиска и фильтрации. При этом доступ к отдельным опциям сохраняется в целях безопасности и доступа к ранее созданным материалам.

Опции, доступные при отсутствии активной лицензии

- Импорт проектов
- Комментирование и экспорт рабочих элементов
- Загрузка файлов из модуля фоновых задач
- Создание тест-плана и экспорт конфигураций из модуля тест-планов
- Экспорт результатов прогона в отчете по тест-плану
- Печать отчета по тест-плану
- Создание, редактирование и удаление категорий ошибок
- Редактирование виджетов и фильтров в модуле дашбордов
- Создание, удаление и редактирование интеграций
- Создание и редактирование пользователей
- Создание, редактирование и удаление проектных ролей
- Создание, редактирование и удаление пользовательских групп
- Восстановление и удаление архивированных рабочих элементов, тест-планов и конфигураций
- Изменение системных настроек
- Генерация API-токена

Обновлено: 8/12/26, 4:43 PM

Быстрый старт

Добро пожаловать в Test IT! Следуйте этим шагам, чтобы быстрее освоиться и приступить к работе:

1. Создание проекта

- Создайте новый проект для тестирования вашего продукта.

2. Формирование команды и настройка ролей

- **Тарифы Базовый и Стандарт:** пригласите коллег прямо из интерфейса.
- **Тариф Test IT Про:** добавьте пользователей локально. Также можно настроить LDAP-подключение либо OpenID Connect.
- Присвойте пользователям системные и проектные роли.

3. Подключение таск-трекера и AI-модели

- Интегрируйте вашу систему управления задачами.
- Подключите нейросеть для генерации тест-кейсов.
- Добавьте в проект интеграции с таск-трекером и нейросетью.
- Создавайте задачи прямо из интерфейса Test IT.

4. Наполнение библиотеки тестов

- Создайте тесты вручную.
- Сгенерируйте тесты с помощью искусственного интеллекта.
- Импортируйте существующие тест-кейсы из Excel-файлов.
- Автоматически перенесите тесты из других систем с помощью мигратора.

5. Планирование и выполнение

- Создайте тест-план.
- Объедините тесты в наборы.
- Назначьте исполнителей.

- Приступайте к выполнению.

6. Интеграция автотестов

Test IT поддерживает интеграцию с различными фреймворками автоматизации, а также гибкие способы их запуска:

- Быстрая подготовка стартовой сборки для вашего языка программирования и тестового фреймворка
- Запуск из CI/CD (Jenkins, GitLab, Bamboo и других систем)
- Запуск вручную через интерфейс Test IT
- Загрузка результатов в JUnit XML из любых внешних систем, поддерживающих этот формат
- Связывание ручных и автоматизированных тестов для комплексного тестирования
- Создание шаблонов автотестов в IDE-системе из ручных тестов в Test IT с помощью плагина Менеджмент

7. Анализ результатов

- Отслеживайте аналитику по проектам и тест-планам, находите узкие места и улучшайте процессы.

Дополнительные возможности

Бесплатный тариф **Базовый** включает базовые функции. В расширенных тарифах доступны:

- Неограниченное количество проектов
- Версионирование тестов
- Test IT MCP — ИИ-помощник для широкого круга задач
- Подключение любой AI-модели, совместимой с OpenAI
- Гибкое планирование и отчет по тест-плану
- Параметризованные запуски автотестов (разные наборы данных, окружения, конфигурации)
- Автоматическая отметка и перезапуск нестабильных автотестов
- Геймификация

Смотрите также видеокурс по Test IT на нашем сайте.

Обновлено: 8/19/26, 10:55 AM

Администрирование Test IT

Данный раздел описывает работу администратора Test IT по настройке платформы. Все процессы, описанные в данном разделе доступны пользователю с системной ролью Администратор. Данный раздел описывает следующие процессы:

- Вход в систему после установки
- Настройка подключений
- Добавление пользователей и работа с ними
- Работа с группами пользователей
- Присвоение системных ролей
- Настройка проектных ролей
- Настройка очистки хранилища
- Настройка атрибутов
- Настройка интеграций с таск-трекерами
- Настройка интеграций с ИИ-моделями

Обновлено: 8/12/26, 4:43 PM

Вход в Test IT после установки

> **Функциональность Test IT Про**
>
> Данная инструкция описывает вход в систему после установки **Test IT Про**. Если вы используете тарифы **Test IT Базовый** или **Стандарт**, обратитесь к соответствующей инструкции для Личного кабинета.

Вход

При установке в Test IT Про создается служебный пользователь с именем `admin` и паролем `Qwerty123`. У этого пользователя есть права установки системы.
При первом входе в Test IT:

1. В браузере откройте Test IT, используя адрес сервера в локальной сети: `http://<IP>` или `https://<домен>`.
2. Введите данные служебного пользователя (`admin` / `Qwerty123`).
3. Нажмите **Войти** или `Enter`.

Смена пароля

После входа в систему рекомендуется сменить пароль. Чтобы сменить пароль:

1. Перейдите на вкладку **Администрирование**.
   По умолчанию откроется раздел **Пользователи**.
2. Выберите пользователя **System Administrator**.
   Откроется окно редактирования пользователя.
3. Введите новый пароль в соответствующем поле.
4. Нажмите **Сохранить**.

После смены пароля вы можете:

- Настроить подключения, чтобы добавить в систему других пользователей через AD/LDAP, AD/Kerberos, и настроить вход через OpenID Connect.
- Добавить пользователей локально.

Изменение цвета интерфейса (темная и светлая темы)

Чтобы переключить цветовую тему интерфейса:

- Нажмите значок переключения цветовой темы слева от значка профиля в правом верхнем углу окна.

Обновлено: 10/9/26, 6:25 PM

Внешние подключения в Test IT

> **Функциональность Test IT Про**
>
> Данная инструкция описывает настройку подключений в **Test IT Про**.
> Если вы используете облачный тарифы **Test IT — Базовый** или **Стандарт**, пропустите этот раздел.

Test IT Про позволяет добавить пользователей из служебных каталогов и входить в систему с помощью следующих утилит:

- AD/LDAP
- AD/Kerberos
- OpenID Connect (OAuth 2.0)

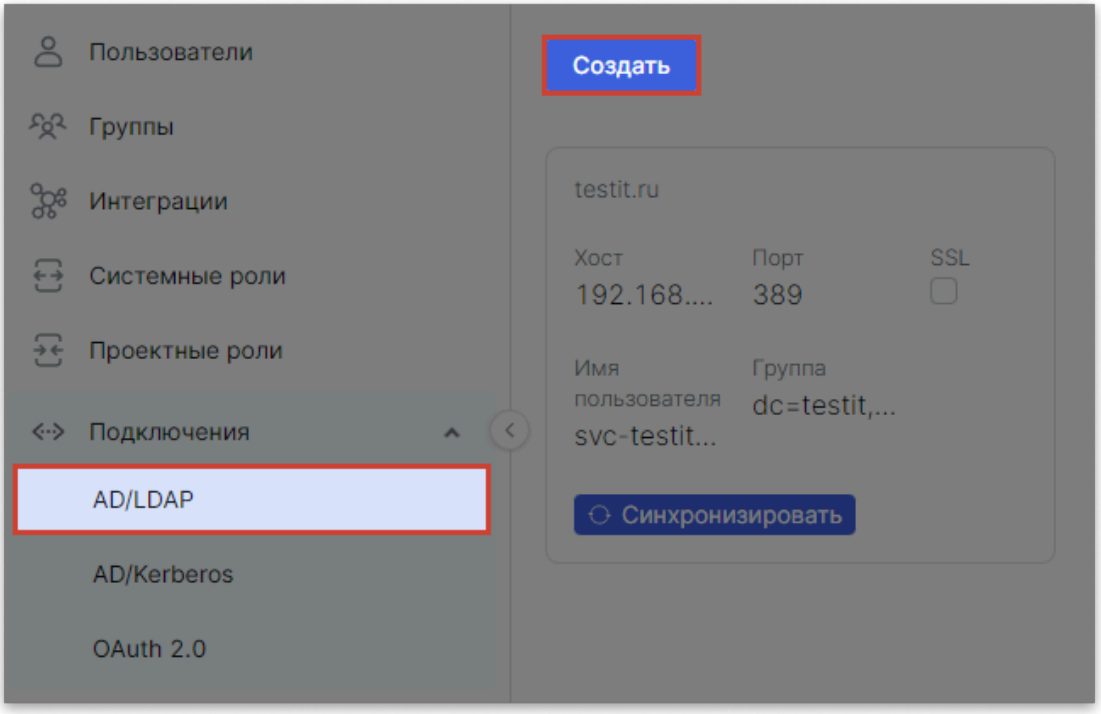
Обновлено: 10/9/26, 6:25 PM

AD/LDAP

Настройка подключения

После того, как системный администратор добавил пользователей в активную директорию (AD), вам необходимо настроить подключение по LDAP в системе Test IT.

1. Используя аккаунт с правами администратора, нажмите **Администрирование** на верхней навигационной панели.
2. В навигационном меню слева выберите **Подключения**. Убедитесь, что вы находитесь на вкладке **AD/LDAP** и нажмите **Создать**.



3. Заполните обязательные поля, нажмите **Протестировать**. После успешного тестирования сохраните подключение. Чтобы узнать больше о полях, смотрите Поля настроек подключения.

На странице подключений AD/LDAP можно создать несколько подключений, а также выбрать подключение по умолчанию. Выбранное подключение по умолчанию будет отображаться на странице авторизации как основное.

Добавление пользователей из группы AD

В настройках AD/LDAP можно указывать схему пользователей и групп для поддержки большинства служб.

User Schema

Пример фильтра для пользователей из одной группы:

```text
(&(objectCategory=Person)(sAMAccountName=*)
(memberOf=CN=Group-Sonya,OU=Sonya-
test,OU=podrazdelenie2,OU=Podrazdelenie1,DC=mtest,DC=ru))
```

Пример фильтра, который добавляет пользователей из вложенных групп:

```text
(&(objectCategory=Person)(sAMAccountName=*)
(memberOf:1.2.840.113556.1.4.1941:=CN=Group-Sonya,OU=Sonya-
test,OU=podrazdelenie2,OU=Podrazdelenie1,DC=meistertest,DC=ru
```

Group Schema

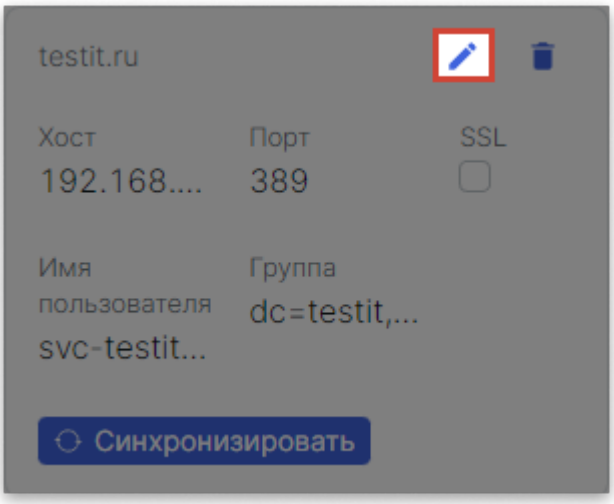
Фильтр для одной группы (имя начинается с):

```text
(&(objectCategory=group)(name=Group-Sonya*))
```

Редактирование подключения

На вкладке **AD/LDAP** доступно редактирование соединения и кнопка для принудительной синхронизации. Чтение и синхронизация происходят каждые 10 минут. Чтобы внести изменения в подключение:

1. Нажмите значок редактирования на карточке подключения.



2. Внесите изменения.
3. Нажмите **Протестировать**. В случае успешного тестирования нажмите **Сохранить**.
4. Нажмите **Синхронизировать** на карточке подключения.

Обновлено: 10/9/26, 6:25 PM

Поля настроек подключения AD/LDAP

Внимание

При настройке LDAP-подключения с Windows AD отключите опции **Follow Referral** и **Bind Referral requests** для корректной работы.

При подключении AD/LDAP используются следующие поля и их значения:

#	Поле	Значение	Пример / пояснение
1	Имя	Имя соединения в разделе **Администрирование** Test IT	`My connection`
Settings			
2	Base DN	Узел в LDAP для поиска пользователей и групп	`dc=testit,dc=software`
3	Host	Имя хоста сервера, на котором запущен LDAP	`198.162.43.21`
4	Port	Порт сервера LDAP	`389`
5	Distinguished Name	Имя пользователя для подключения к серверу LDAP	`admin@testit.software`
6	Password	Пароль пользователя	`AncientStars1999`
7	SSL	Подключение с помощью защищенного протокола	-

8	Paging Mode	Режим пагинации (выпадающее меню)	• **NoPaging**: без пагинации. Все результаты запроса передаются одним пакетом (макс. 1000 записей) • **SimplePaging**: использует стандарт RFC 2696. Передает результаты по количеству записей, указанному в поле Page Size (например, если записей 8500, в ответ придет 8 списков по 1000 записей и 1 по 500) • **VirtualListView (VLV)**: виртуальный список. Используется для запроса диапазона записей из общего списка (например, с 1000 по 1500) Для больших каталогов (более 3000 записей) рекомендуем использовать `page size = 1000` , для каталогов менее 1000 рекомендуем указать `100` или выбрать `NoPaging` .
9	Page Size	Количество результатов на странице	1000
10	Follow Referral	Позволяет серверу каталогов перенаправлять директорию на другие серверы. Используется для предотвращения ошибок на серверах активных каталогов (active directory), настроенных без надлежащего DNS.	• **Выбрано**: автоматически следовать перенаправлениям (рефералам) на другие серверы LDAP • **Не выбрано**: не следовать рефералам, возвращать ошибку если запрос требует перенаправления

11	Bind Referral Requests	Включает повторное связывание реферальных запросов (referral requests)	• **Выбрано**: использовать учетные данные из исходного запроса для аутентификации на целевом • **Не выбрано**: использовать анонимный доступ при переходе по рефералам
User Schema			
12	User Name	Атрибут имени пользователя (sAMAccountName)	`RomanSidorov`
13	Distinguished Name	Атрибут уникального имени пользователя	`CN=Роман Сидоров, OU=Test IT, DC=testit,DC=software`
14	Display Name	Атрибут для загрузки полного имени пользователя	`Роман Сидоров`
15	First Name	Атрибут для загрузки имени пользователя	`Роман`
16	Last Name	Атрибут для загрузки фамилии пользователя	`Сидоров`
17	Email	Атрибут для загрузки email	`r.sidorov@testit.software`
18	Membership	Атрибут принадлежности пользователя к группе	`CN=MyGroup, OU=Testit, DC=testit,DC=software`
19	Id	Атрибут для идентификации пользователя при его переименовании	`878ce8b7-2713-41a9-a765-5e3905ab5ef2`
20	Common Name	Атрибут для загрузки имени пользователя	`cn`
21	Filter	Фильтр, используемый при поиске объектов пользователя	`(&(userAccountControl=*)(!(objectClass=computer)))`
Group Schema			
22	Members	Атрибут для загрузки членов группы из группы AD	`member`
23	Name	Атрибут имени группы	`LittleMembers`

24	Id	ID группы	`878ce8b7-2713-41a9-a765-5e3905ab5ef2`
25	Common Name	Атрибут для загрузки имени группы	`cn`
26	Filter	Фильтр, используемый при поиске объектов групп	`(groupType=*)`

Обновлено: 10/9/26, 6:25 PM

AD/Kerberos

Вы можете авторизоваться в системе с помощью протокола Kerberos.

Настройка подключения

Подключение от имени системного администратора

> **Используйте надежные типы шифрования**
>
> Рекомендуем указывать только сильные, современные типы шифрования. Например, `crypto AES256-SHA1,AES128-SHA1` (Шаг 2).

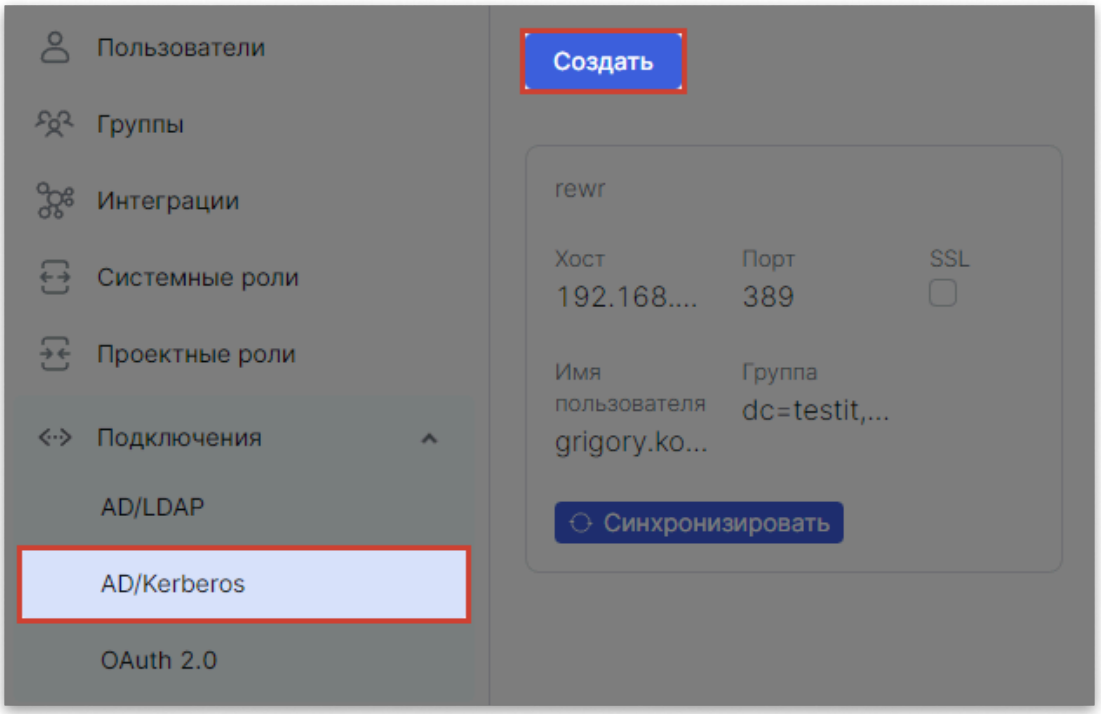
1. Добавьте пользователя в контроллер домена, от имени которого будет происходить синхронизация с системой дистрибуции ключей (KDC).
2. Сгенерируйте файл `.keytab` на контроллере домена, используя утилиту ktpass:

```text
ktpass.exe /princ HTTP/application_url@YOURDOMAIN /mapuser KDCAdmin@YOURDOMAIN /crypto AES256-SHA1,AES128-SHA1 /ptype KRB5_NT_PRINCIPAL /pass * /out C:\example.keytab
```

Чтобы узнать больше об утилите ktpass и ее параметрах, смотрите ktpass .

Подключение от имени администратора Test IT

1. Используя аккаунт с правами администратора, нажмите **Администрирование** на верхней навигационной панели.
2. В навигационном меню слева выберите **Подключения**, перейдите во вкладку **AD/Kerberos**, нажмите **Создать**.



3. Заполните все обязательные поля, загрузите созданный ранее файл `.keytab` и нажмите **Протестировать**. В случае успешного тестирования, сохраните подключение. Детальное описание полей можно найти на странице Поля настроек подключения.
4. Синхронизируйте список пользователей в контроллере домена со списком пользователей внутри системы Test IT.

> **Важно**
>
> Если вы меняете логин пользователя внутри контроллера домена, вам необходимо выполнить синхронизацию повторно. В противном случае новый логин в контроллере домена не будет сопоставляться с логином в интерфейсе Test IT.

При успешной синхронизации списков пользователей, на экране авторизации появится возможность войти через Kerberos.

Обновлено: 10/9/26, 6:25 PM

OpenID Connect

Test IT позволяет авторизоваться с помощью сервисов аутентификации Keycloak и Auth0, основанных на протоколе OpenID Connect.

1. Используя аккаунт с правами администратора, нажмите **Администрирование** на верхней навигационной панели.
2. В навигационном меню слева выберите **Подключения**.
3. Перейдите на вкладку **OpenID Connect**.
4. Нажмите **Создать**.
5. В открывшемся окне укажите:
   а. Название подключения. При необходимости можете добавить значок провайдера, для которого настраиваете подключение.
   б. В поле **Authority** укажите адрес провайдера, через которого вы настраиваете подключение.
   в. В поле **Endpoint для получения конфигурации OpenID** укажите адрес провайдера с путем до конфигурационного файла OpenID.
   г. Укажите идентификатор приложения провайдера в поле **Client_ID**.
   д. Укажите секретный ключ приложения в поле **Client_secret**.
   е. В поле **Scope** укажите данные о пользователях, которые необходимо получить из внешнего провайдера для входа в систему. Минимальный набор данных: **OpenID**, **email**, **profile**.
6. Нажмите **Сохранить**.
7. Во внешнем сервисе укажите callback URL.
   Ссылка должна иметь следующий формат:
   `https://{test_it_domain}/api/auth/signin-oidc` , где `{test_it_domain}` необходимо заменить на адрес вашей системы Test IT.

После добавления OAuth 2.0 провайдера на странице логина появляется возможность авторизации с помощью него. Пользователь, который авторизуется с помощью OAuth 2.0 провайдера, регистрируется в системе и наделяется системной ролью **Пользователь** при наличии свободных лицензий.

Обновлено: 10/9/26, 6:25 PM

Добавление пользователей и работа с ними

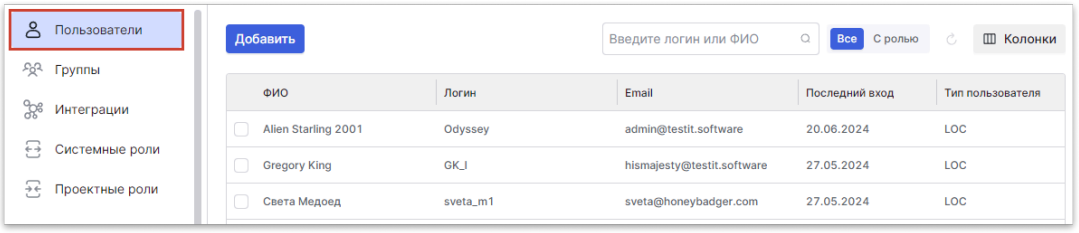
Процесс добавления пользователей для облачной (тарифы Базовый и Стандарт) и серверной (тариф Про) поставок различается. Выберите инструкцию для своей формы поставки.

Test IT Про (серверная версия)

- Добавление локальных пользователей в Test IT Про
- Редактирование данных локальных пользователей в Test IT Про
- Удаление локальных пользователей в Test IT Про
- Создание пользователей с привязкой к провайдеру OpenID

Вы можете добавлять, удалять и редактировать данные пользователей, добавленных в Test IT Про локально, через активную директорию (AD) или через OpenID Connect. Чтобы открыть окно информации о пользователях:

1. Используя аккаунт администратора Test IT, в верхней навигационной панели нажмите **Администрирование**.
2. В навигационном меню слева выберите **Пользователи**.

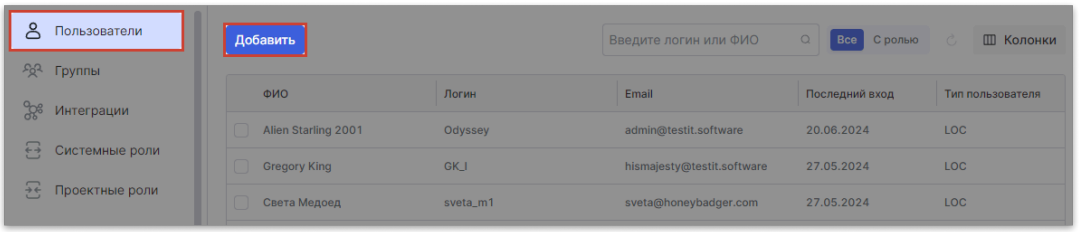


Чтобы узнать о **работе с группами** пользователей, смотрите статью Работа с группами пользователей.

Добавление локальных пользователей в Test IT Про

Вы можете добавлять локальных пользователей в Test IT Про:

1. В разделе **Пользователи** нажмите **Добавить**.



2. В открывшемся окне укажите:
   - Логин пользователя для входа в Test IT
   - Пароль пользователя
   - Email пользователя
   - ФИО — имя, которое будет отображаться в Test IT
3. Нажмите **Добавить**.
4. Присвойте пользователю системную роль.
   Добавленный пользователь сможет просматривать данные в Test IT.
5. Присвойте пользователю проектную роль, чтобы он смог совершать действия в системе.

Редактирование данных локальных пользователей в Test IT Про

После добавления пользователей в Test IT Про вы можете просматривать всю информацию о них в разделе **Пользователи**. Для точного поиска используйте Фильтр. Настройки отображения сохраняются для текущего браузера.

Вы можете менять данные только для локальных пользователей. Данные пользователей, добавленных через AD необходимо менять в AD.

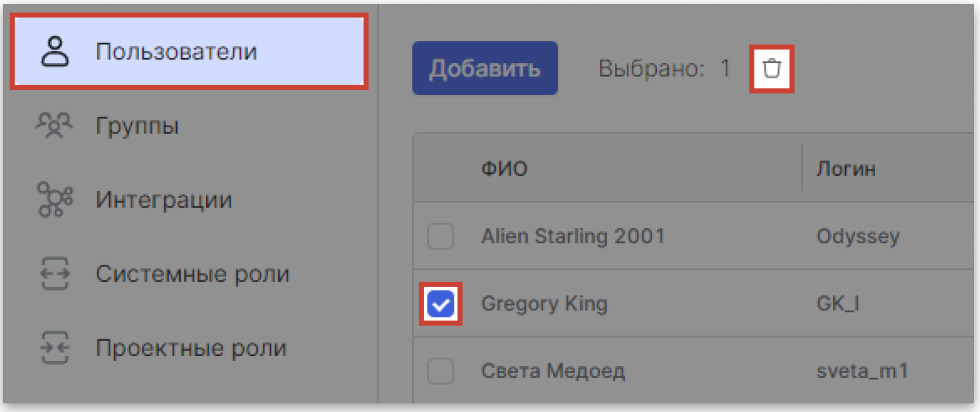
Чтобы редактировать данные пользователя:

1. Нажмите на строку пользователя, данные которого хотите отредактировать.
2. Внесите изменения.
3. Нажмите **Сохранить**.

Удаление локальных пользователей в Test IT Про

Чтобы удалить локальных пользователей из Test IT Про:

1. В списке пользователей выберите одного или нескольких локальных пользователей.
2. Нажмите значок удаления (корзина).
3. Подтвердите удаление в диалоговом окне.



Создание пользователей с привязкой к провайдеру OpenID

Вы можете создавать учетные записи пользователей OpenID с помощью API-метода `POST api/v2/users`. Способ позволяет создавать пользователей, не ожидая их первого входа в систему.

Этот способ требует активации переменной `Application__AllowCreateExternalUsersManually`. При этом авторизация через OpenID пользователей, которые не были добавлены в систему, будет невозможной.

Чтобы использовать метод `POST api/v2/users`, в конфигурации Test IT требуется переменная `Application__AllowCreateExternalUsersManually`, разрешающая создание внешних пользователей со значением "true" (см. Подготовка в Docker Compose и Подготовка в Kubernetes).

Подготовка в Docker Compose

1. В `.env`-файл добавьте переменную и укажите для нее значение "true":
   - `Application__AllowCreateExternalUsersManually=true`
2. Сохраните файл.
3. Выполните пересоздание контейнеров с помощью команды:

```sh
docker compose -f docker-compose.yml --project-name testit up --detach --timeout 120
```

   Создание пользователей будет доступно.

Подготовка в Kubernetes

1. В файл `values-override.yaml` или `values.yaml` в раздел `webapi` добавьте переменную и укажите для нее значение "true":
   - `Application__AllowCreateExternalUsersManually=true`

   Пример:

```sh
webapi:
  enabled: true
  ...
  config:
    Application__AllowCreateExternalUsersManually: "true"
```

2. Сохраните файл.
   После сохранения файла будет доступно создание пользователей.

Создание пользователей

1. Откройте Swagger Test IT. Если вы не авторизованы в Test IT, авторизуйтесь с помощью API-токена.
2. В группе **Users** откройте метод `POST /api/v2/users`.
3. В запросе укажите значения полей:

▼ **Значения полей (развернуть / свернуть)**

#	Поле	Значение	Тип	С
1	`userName`	Имя пользователя	Строка	Д
2	`firstName`	Имя	Строка	Д
3	`lastName`	Фамилия	Строка	Д
4	`displayName`	Отображаемое имя пользователя	Строка	Д
5	`password`	Пароль	Строка	Д у
6	`email`	Электронная почта	Электронная почта	Д
7	`providerId`	Идентификатор сконфигурированного провайдера в Test IT. Доступен в Swagger Test IT в группе **OpenIDConnections** по запросу `GET /api/v2/openid-connections`	UUID	Н
8	`externalId`	Идентификатор пользователя из системы OpenId (подробности см.в конфигурации провайдера)	Строка	Д у

4. Отправьте запрос вида:

```json
{
  "userName": "der.golem",
  "firstName": "Roman",
  "lastName": "Sidorov",
  "displayName": "RomanSidorov",
  "password": "TechnoHouseIsAShrimp1994!",
  "email": "dergolem@mosswitch.com",
  "providerId": "923d8bc2-5fe1-4b69-9538-fa2d541e5b5d",
  "externalId": "876ce373-172e-4d36-bdbb-5dd5a67b3a8a"
}
```

Пользователь будет создан в системе при корректно указанных данных.

Test IT Базовый и Стандарт (облачная версия)

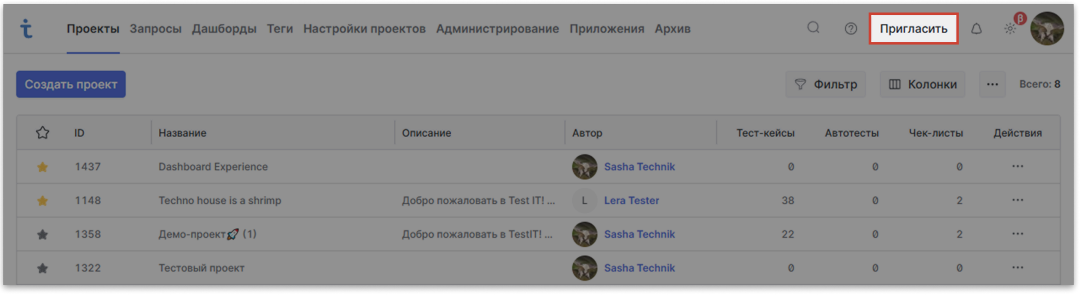
- Приглашение пользователей в Test IT Базовый и Стандарт
- Ограничение доступа к рабочему пространству

Приглашение пользователей в Test IT Базовый и Стандарт

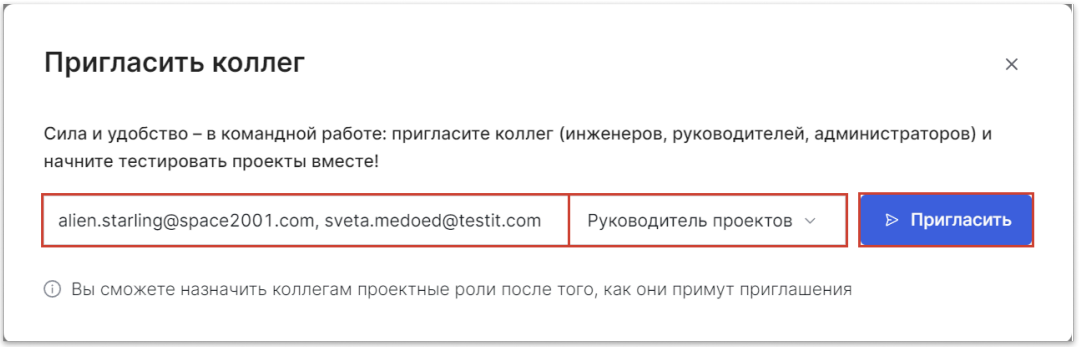
Владелец пространства может приглашать новых пользователей в Test IT, присваивать им системные роли прямо из пространства и ограничивать доступ к рабочему пространству. Также вы можете приглашать пользователей из Личного кабинета.

Чтобы пригласить пользователя:

1. Откройте рабочее пространство, в которое хотите пригласить пользователей.
2. В верхнем левом углу нажмите **Пригласить**.



3. В открывшемся окне укажите адрес электронной почты пользователя и выберите системную роль в выпадающем меню (**Администратор**, **Руководитель проектов** или **Пользователь**). Больше информации о системных ролях можно найти в разделе Присвоение системных ролей.
4. **Опционально:** Чтобы пригласить нескольких пользователей, введите их электронные адреса через запятую.



5. Нажмите **Пригласить**.

Отправленные приглашения отображаются в разделе **Приглашение пользователей** в Личном кабинете. Вы можете отслеживать информацию о приглашении:

- Электронный адрес, на который было отправлено приглашение
- Системная роль приглашенного пользователя
- Статус приглашения (**Отправлено** или **Истекло**)
- Оставшийся срок действия приглашения

Приглашение действительно в течение *трех дней после отправления*. Если пользователь не принял приглашения за это время, вы можете отправить приглашение из Личного кабинета. Подробную информацию смотрите в разделе Приглашение пользователей из Личного кабинета.

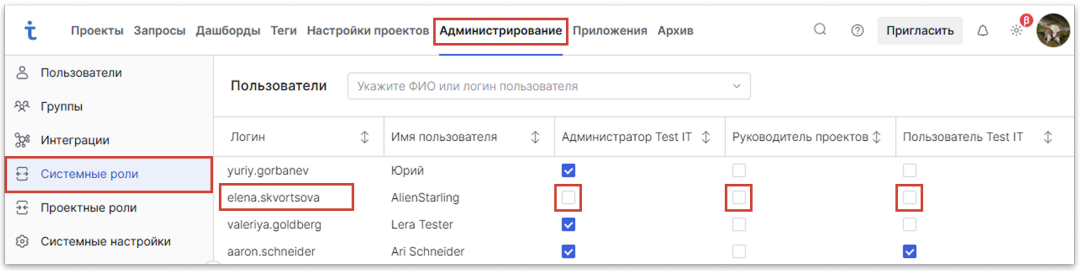
Чтобы узнать о **работе с группами** пользователей, обратитесь к смотрите статью Работа с группами пользователей.

Ограничение доступа к рабочему пространству

Вы можете ограничить доступ пользователя к рабочему пространству Test IT, сняв с него все системные роли. При необходимости вы можете вернуть пользователю системные роли и доступ к Test IT.

Чтобы ограничить доступ пользователя:

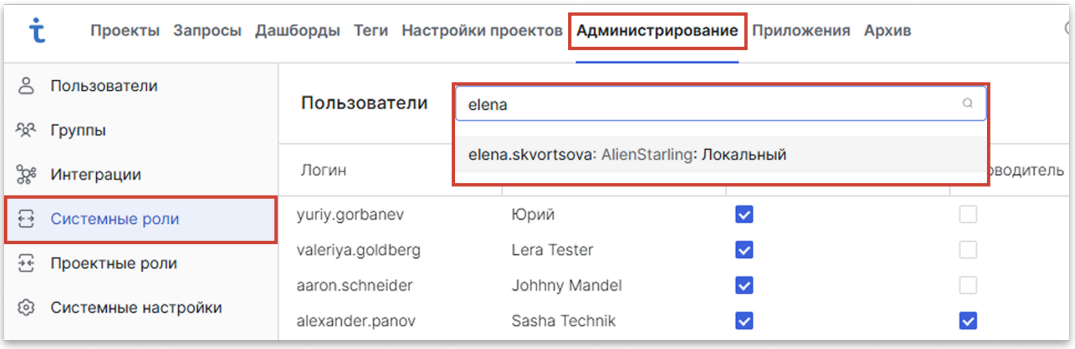
1. В верхней навигационной панели откройте **Администрирование**, затем выберите **Системные роли**.
2. Удалите все системные роли у выбранного пользователя, сняв галочки в соответствующих столбцах таблицы системных ролей.



При снятии ролей отобразится сообщение о снятии системных ролей. При удалении всех системных ролей пользователь не будет иметь доступ в систему. Пользователи без системных ролей не отображаются в списке после перезагрузки страницы.

Чтобы вернуть пользователю системные роли:

1. В поле **Пользователи** введите логин или имя пользователя. Отобразится список пользователей без системных ролей.



2. Выберите пользователя из списка, затем назначьте ему системные роли.
   Пользователь получит доступ к рабочему пространству в соответствии с назначенными ролями.

Обновлено: 8/12/26, 4:43 PM

Работа с группами пользователей

- **Для этого раздела мы подготовили видео:**

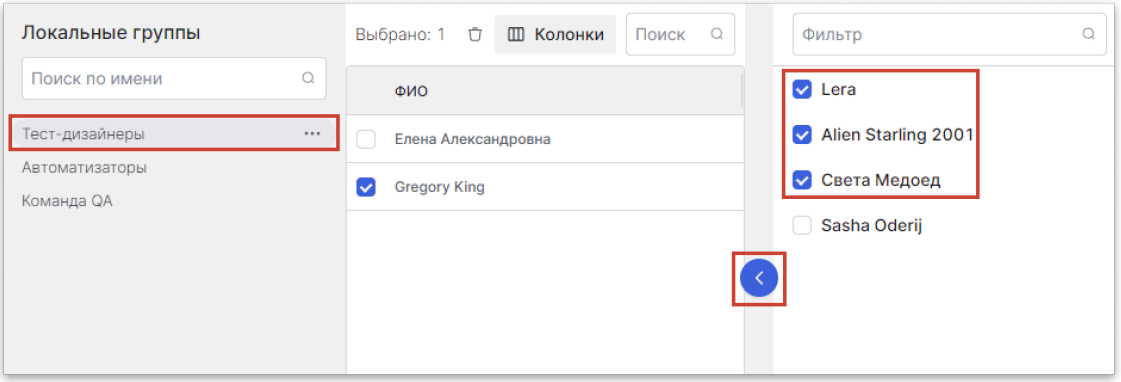
Вы можете объединять пользователей в группы, чтобы массово добавлять их на созданные проекты. Чтобы добавить группу, ей необходимо присвоить системную роль. Для просмотра информации о группах:

1. Используя аккаунт администратора Test IT, в верхней навигационной панели нажмите **Администрирование**.
2. В навигационном меню слева выберите **Группы**.

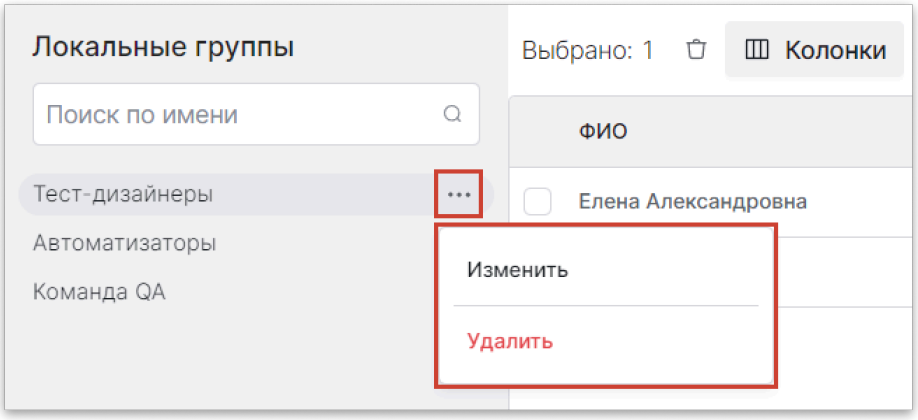
Добавление, редактирование и удаление локальных групп

1. Нажмите **Создать группу**.
2. Укажите название группы, при необходимости добавьте описание.
3. Нажмите **Сохранить**. Новая группа будет отображаться в списке групп.
4. В колонке справа выберите пользователей, которых хотите добавить в созданную группу.

5. Нажмите значок стрелки.



   Выделенные пользователи добавятся в группу.
6. **Опционально:** Чтобы удалить пользователей из текущей группы, поставьте флажки напротив пользователей, которых хотите удалить, и нажмите **Удалить**.
7. **Опционально:** Чтобы изменить название и описание группы или удалить группу, используйте меню действий справа от названия группы.



Присвоение группам системных ролей

Для добавления в проект локальным группам, как и пользователям, требуется системная роль. Чтобы присвоить группе системную роль:

1. На странице **Администрирование** в левом навигационном меню выберите **Системные роли**.
2. Если группе не присвоена системная роль, она не отобразится в списке. В секции Группы под списком пользователей в поле **Добавить группу** начните вводить название группы, затем выберите ее.
3. Присвойте группе системную роль, сделав отметку в нужном столбце.

Обновлено: 8/12/26, 4:43 PM

Присвоение системных ролей

- **Для этого раздела мы подготовили видео:**

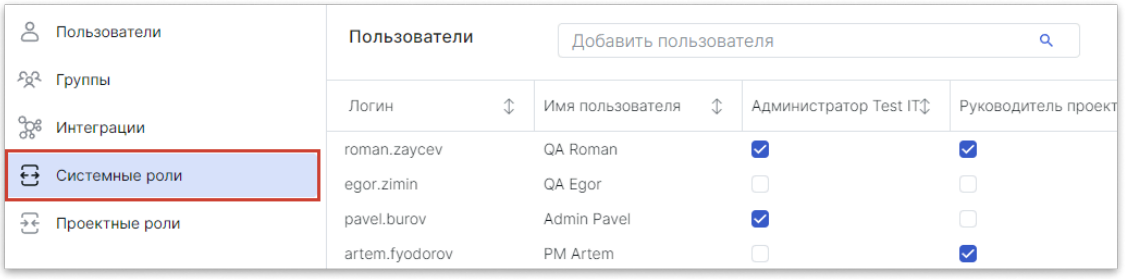
Для входа в систему пользователю необходима системная роль. Test IT предусматривает следующие системные роли, отличающиеся уровнем прав в системе:

- **Администратор** — данная роль предусматривает самые полные права в системе. Администратору доступно добавление лицензий, пользователей (в версии **Test IT Про**, так как в тарифах **Базовый** и **Стандарт** приглашать пользователей может только владелец пространства) и групп, создание подключений, интеграций, проектных ролей, глобальных атрибутов и проектов.
- **Руководитель проектов** — данная роль предусматривает права на создание и редактирование проектов, создание проектных атрибутов, присвоение проектных ролей.
- **Пользователь** — данная роль предусматривает минимальный уровень прав в системе. По умолчанию пользователь может только войти в свою учетную запись. Чтобы получить более полные права доступа, пользователю необходима проектная роль, в которой указаны соответствующие права.
- **Наблюдатель** (также **просмотровая лицензия**) — роль, позволяющая просматривать данные в системе без возможности внесения изменений. Плата за просмотровые лицензии не взимается, свободных лицензий для них не требуется. При назначении пользователю роли **Наблюдатель** остальные

системные роли снимаются. В тарифах **Базовый** и **Стандарт** (ранее Test IT Cloud) роль **Наблюдатель** недоступна.

Чтобы назначать системные роли пользователям:

1. Используя аккаунт администратора Test IT, в верхней навигационной панели нажмите **Администрирование**.
2. В навигационном меню слева выберите **Системные роли**. В открывшемся окне отобразятся все пользователи и группы, добавленные в систему.



3. Поставьте флажок роли, которую вы хотите дать определенному пользователю или группе пользователей. Роль применится и сохранится автоматически.

Пользователям Test IT Про: При добавлении пользователя через AD/LDAP к группе, у которой есть права в системе, синхронизация может занять около 20 минут.

> **Важно**
>
> Вы не можете снять роль сами с себя, так как без системной роли вы **не сможете войти в систему**.

Обновлено: 8/12/26, 4:43 PM

Настройка проектных ролей

- **Для этого раздела мы подготовили видео (о проектных ролях смотрите с 1:25):**

Все обладатели системной роли **Пользователь** по умолчанию имеют только права на просмотр проектов и рабочих элементов в Test IT. Администратор или менеджер может выдать им проектные роли, расширяющие права доступа. Создание проектных ролей доступно только администратору. Проектная роль присваивается пользователю в настройках проекта (раздел **Пользователи**). Подробнее: Добавление пользователей и групп в проект.

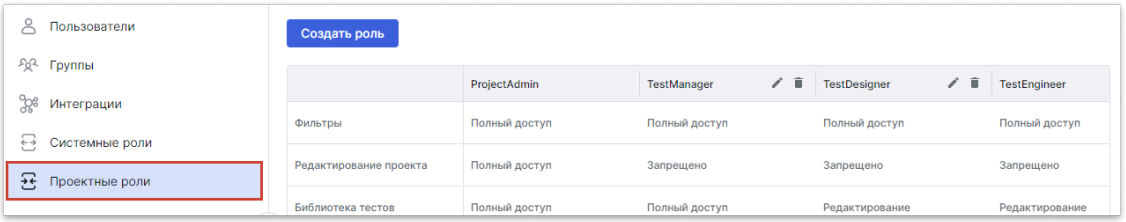
В Test IT доступны следующие преднастроенные проектные роли:

- **Администратор проекта** — полный доступ ко всей функциональности проекта, включая его настройку
- **Менеджер тестирования** — доступны все опции проекта кроме его настроек
- **Тест-дизайнер** — роль для пользователей, которые работают с библиотекой тестов. У роли нет прав удаления тест-кейсов, только изменение. Закрыт доступ к остальной функциональности, кроме раздела **Запросы**.
- **Инженер тестирования** — доступ ко всем разделам без права удаления сущностей
- **Исполнитель тестов** — роль для прохождения ручных тестов и запуска автотестов. Зона ответственности — раздел **Тест-планы**, вкладка **Выполнение**.

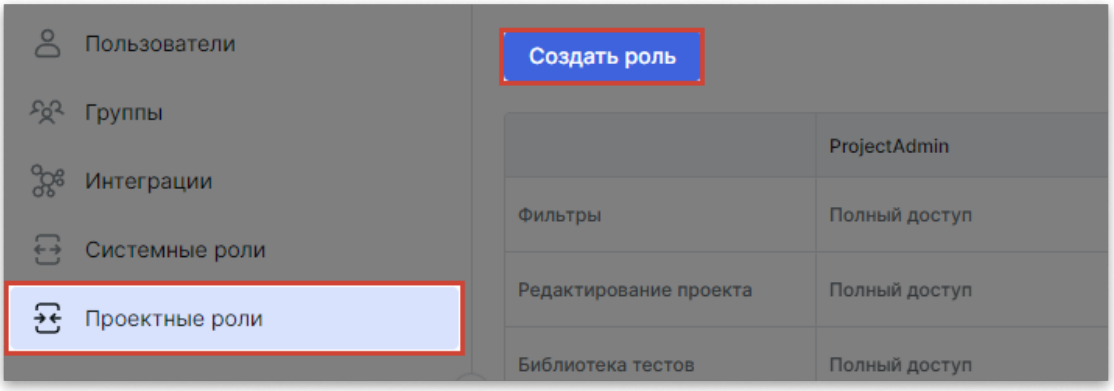
- **Читатель** — минимальный доступ к проекту, только просмотр элементов
  Вы можете удалить все преднастроенные роли, кроме **Администратор проекта** — эта роль обязательна для любого проекта.

Создание проектных ролей

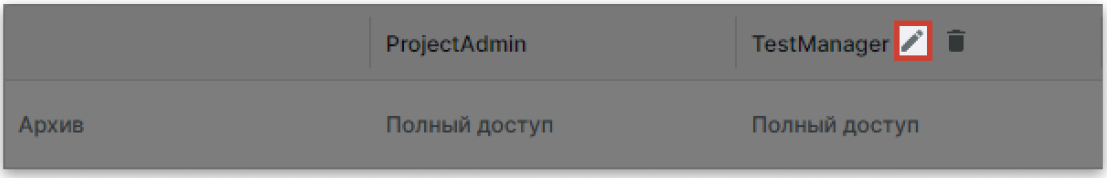
1. Используя аккаунт администратора Test IT, в верхней навигационной панели нажмите **Администрирование**.
2. В навигационном меню слева выберите **Проектные роли**. В открывшемся окне отобразятся проектные роли.



3. Нажмите **Создать роль**.



4. В открывшемся окне укажите название пользовательской проектной роли и нажмите **ОК**. Пользовательская роль отобразится рядом с преднастроенными ролями.
5. Включите режим редактирования роли.



   В режиме редактирования вы можете дать или ограничить доступ к следующим разделам системы:
   - **Архив** — возможность архивировать, восстанавливать и удалять из архива рабочие элементы, тест-планы и конфигурации
   - **Библиотека автотестов** — раздел автотестирования, в котором хранятся карточки автотестов, история прогонов, аналитика по запускам, а также производятся настройки запуска

- **Конфигурации** — раздел создания и настройки конфигураций для рабочих элементов
- **Фильтры** — раздел для создания отфильтрованных запросов для поиска среди различных рабочих элементов
- **Импорт** — доступ к импорту тестов в настройках проекта
- **Редактирование проекта** — возможность создания и изменения описания проекта, добавление пользователей в проект, создания проектных атрибутов, настройки интеграции проекта с внешними сервисами (например Jira)
- **Дашборды** — раздел сводной отчетности по проекту
- **Библиотека тестов** — раздел, где хранятся рабочие элементы проекта (тест-кейсы, чек-листы, общие шаги)
- **Тест-планы** — раздел, позволяющий создавать наборы тестов и распределять задачи между пользователями
- **Отчет по тест-плану** — раздел категоризированных отчетов о результатах тест-плана.
- **Выполнение тестов** — раздел, позволяющий записывать результаты прохождения тестов, запускать автотесты и анализировать причины падения автотестов
- **Вебхуки** — раздел для установки параметров вебхуков

6. Используйте выпадающие меню в строках разделов, чтобы дать роли права доступа (**Запрещено**, **Просмотр**, **Редактирование** и **Полный доступ**) к этим разделам в зависимости от ваших требований к роли. Вы можете менять права доступа к разделам для всех преднастроенных ролей, кроме **ProjectAdmin** — у этой роли по умолчанию полный доступ ко всем разделам.
7. Сохраните изменения.
8. **Опционально:** Чтобы удалить роль, нажмите значок удаления справа от ее названия и подтвердите действие.

> **Пример из практики**
>
> Вам может потребоваться ограничить права доступа подрядчику, который пишет тесты для вашего проекта. Вы можете создать роль **Подрядчик**, которая будет ограничивать доступ подрядчика к редактированию всех разделов, кроме библиотеки тестов, библиотеки автотестов и выполнения тестов.

После того, как вы добавили проектную роль, пользователи с системными ролями **Администратор** и **Менеджер проектов** могут присваивать ее пользователям, добавленным в проект.

Уровни прав доступа

В Test IT предусмотрены четыре уровня прав доступа к рабочим элементам:

- **Запрещено** — ограничивает доступ пользователя к API и UI, связанным с определенным разделом. В UI этот раздел будет скрыт и будет недоступен по прямой ссылке. При обращении к разделу через API возвращается код ошибки доступа.
- **Просмотр** — позволяет просматривать содержимое раздела без прав на редактирование и удаление. Возможности редактирования и удаления скрыты в UI или недоступны по прямой ссылке. При обращении к разделу через API (кроме GET запросов) возвращается код ошибки доступа.
- **Редактирование** — дает возможность видеть и изменять элементы Test IT, но ограничивает возможность удалять элементы. Например, можно изменить текст в шаге тест-кейса, но удалять тест-кейс полностью запрещено. Для настроек проекта это означает, что проект нельзя архивировать и восстанавливать. При обращении через API-запросы типа `DELETE` возвращают код ошибки доступа.
- **Полный доступ** — предоставляет неограниченный доступ к разделу.

> **Важно**
>
> Некоторые права на элементы нельзя изменить. Например, нельзя запретить доступ к разделу **Библиотека тестов** или ограничить доступ к разделу **Запросы**.

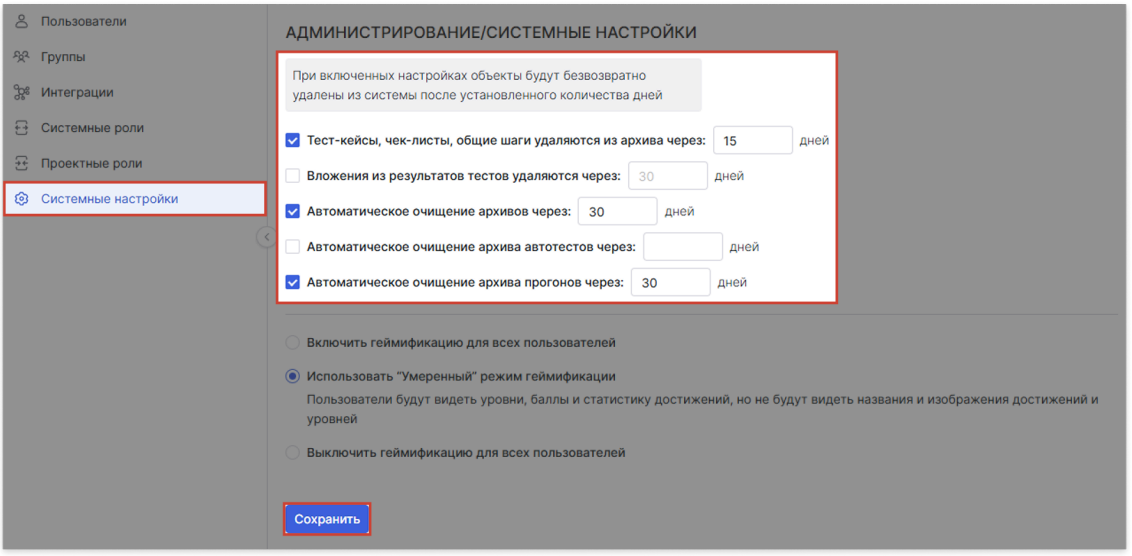
Обновлено: 8/12/26, 4:43 PM

Настройка очистки хранилища

Вы можете настроить удаление архивных рабочих элементов и вложений из результатов тестов. Test IT позволяет установить период времени, по истечении которого архивные элементы безвозвратно удаляются.

> **Необратимое действие**
>
> Объекты, удаленные из архива, невозможно восстановить в системе.

1. Войдите в систему с аккаунта администратора и перейдите в раздел **Администрирование**.
2. Перейдите в раздел **Системные настройки**.



3. Выберите элементы, которые хотите удалять из хранилища — рабочие элементы, вложения результатов тестов, системные архивы, автотесты и прогоны.
4. Установите количество дней, по истечении которого элементы будут безвозвратно удаляться.
5. Нажмите **Сохранить**.

Максимальные сроки очистки хранилища для тарифов Test IT

Для тарифов Test IT сроки очистки хранилища различны.

Объект / действие	Тариф Про	Тарифы Базовый и Стандарт
Тесты, чек-листы, общие шаги	180 дней	30 дней

Вложения из результатов тестов (включая неархивированные тесты)	180 дней	30 дней
Автоматическое очищение архива	90 дней	30 дней
Остальные архивируемые объекты	30 дней	30 дней

Обновлено: 4/15/26, 7:08 PM

Настройки геймификации

Тарифные ограничения

Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

- **Для этого раздела мы подготовили видео:**

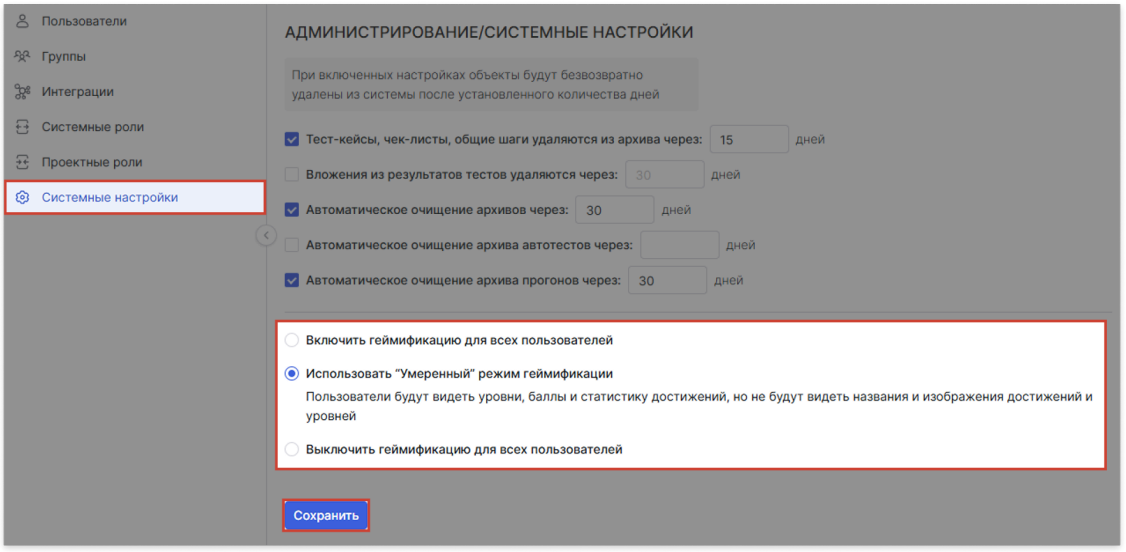
Пользователи с ролью **Администратор** могут:

- Включать и отключать уровни пользователей (геймификацию) для всех пользователей. Система учитывает внесенные пользователем данные независимо от того, включена геймификация или нет. Поэтому, если вы отключили, а затем снова включили геймификацию, прогресс и уровень пользователей будут учитываться и отображаться в системе.
- Включать умеренный режим геймификации. Пользователи будут видеть уровни, баллы и статистику достижений, но не будут видеть названия и изображения достижений и уровней.

Чтобы включить, отключить или настроить геймификацию:

1. На верхней навигационной панели нажмите **Администрирование**.
2. Выберите **Системные настройки**.

3. Внесите изменения, выбрав нужные опции.



4. Нажмите **Сохранить**.

Обновлено: 8/18/26, 4:54 PM

Настройка атрибутов

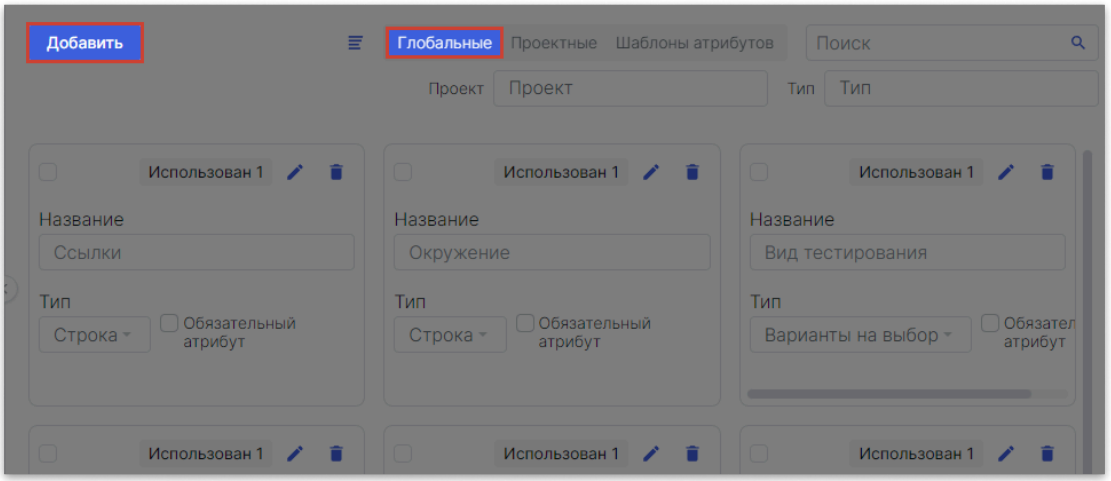
Вы можете задать для тестов и тест-планов нужные вам значения, используя атрибуты и их шаблоны. Атрибуты также помогают сделать поиск с помощью Фильтра более точным.

> **Пример из практики**
>
> Вы можете создать атрибут **Тип поставки**, предполагающий множественный выбор: *Серверная* и *Облачная* и сделать его обязательным, чтобы пользователи указывали тип поставки вашего продукта при создании тест-кейсов и/или тест-планов.

Атрибуты соответствуют столбцам в таблице тест-кейсов (например приоритет, статус) или тест-планов (например версия продукта). Администратор может создавать глобальные атрибуты и объединять их в шаблоны — готовые наборы атрибутов для массового использования в проектах. В отличие от проектных атрибутов, глобальные атрибуты доступны сразу для всех проектов в Test IT.

Создание глобальных атрибутов

1. Используя аккаунт с правами администратора, нажмите **Настройки проектов** на верхней навигационной панели.
2. В навигационном меню слева выберите **Атрибуты**, убедитесь, что вы находитесь на вкладке **Глобальные**, и нажмите **Добавить**.

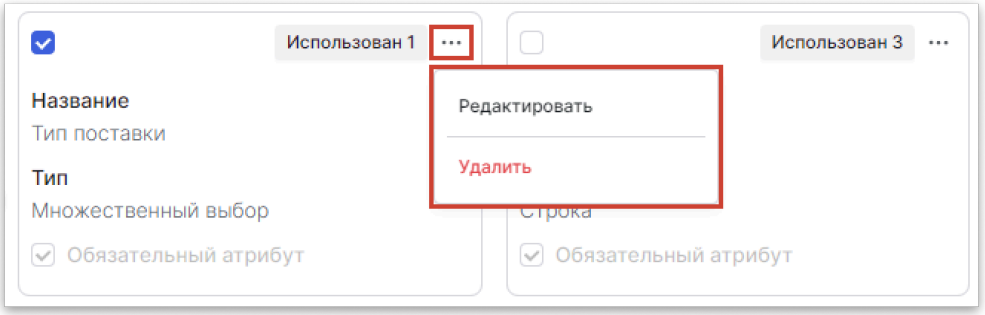


3. В открывшемся окне введите название нового атрибута, выберите его тип: **Строка**, **Дата**, **Варианты на выбор**, **Множественный выбор**, **Пользователь** и **Чекбокс**. Для типов **Варианты на выбор** и **Множественный выбор** введите необходимые значения. Например, для атрибута "Команда" значениями могут быть команды, ответственные за разработку определенной функциональности.
4. Вы можете сделать атрибут обязательным для всех проектов, поставив флажок **Обязательный атрибут**.
5. Нажмите **Сохранить**.

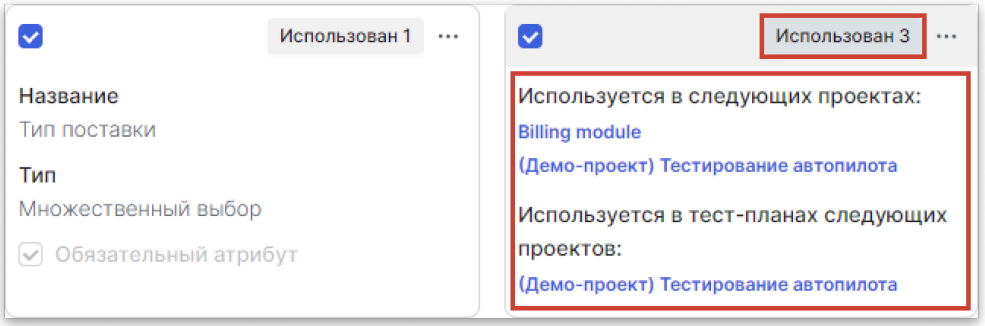
Чтобы использовать глобальные атрибуты в тестах проекта, их нужно добавить в проект. Для использования в тест-планах их необходимо добавить в атрибуты тест-плана.

Вы можете:

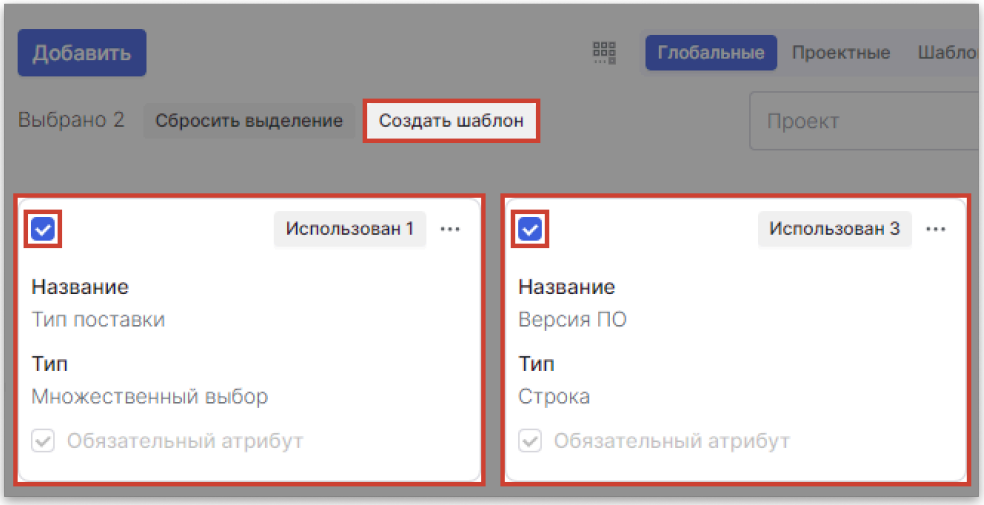
- Отредактировать атрибут или удалить его из системы, используя меню действий в карточке или строке атрибута.



- Отслеживать, в каких проектах используется атрибут. В карточке или строке атрибута нажмите **Использован [N]**. Отобразится список проектов, в которых использован атрибут.

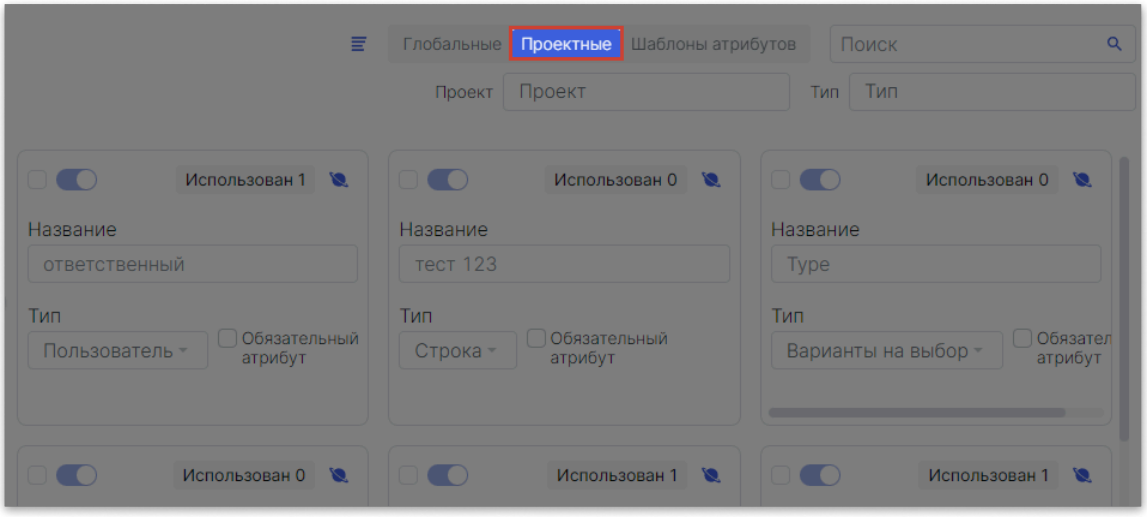


- Создать шаблон атрибутов, выбрав несколько карточек глобальных атрибутов и нажав **Создать шаблон** в левом верхнем углу окна атрибутов.



Работа с проектными атрибутами

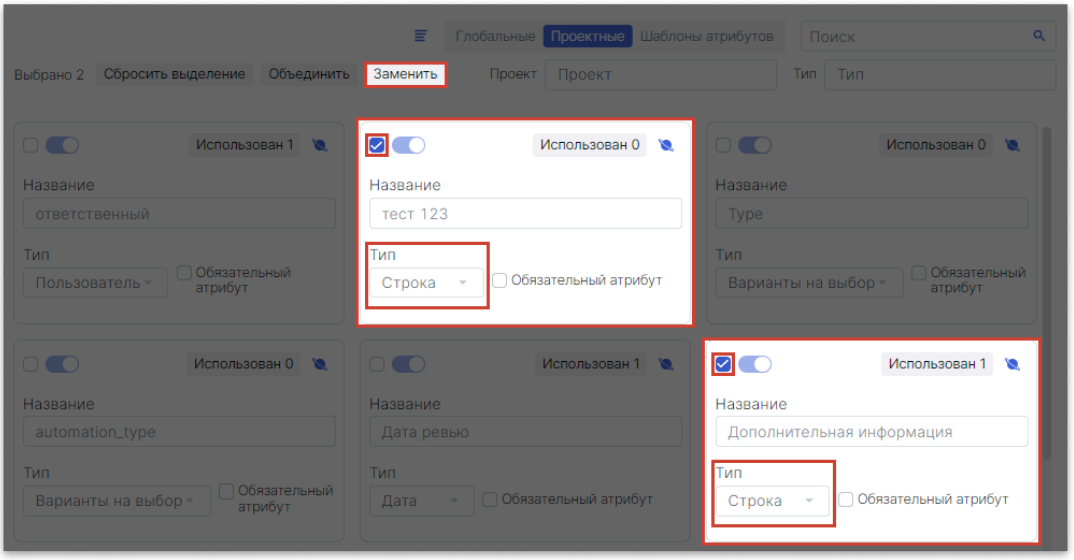
В разделе **Атрибуты** отображаются карточки всех проектных атрибутов, созданных в Test IT. Для этого необходимо перейти на вкладку **Проектные**.



Вы можете:

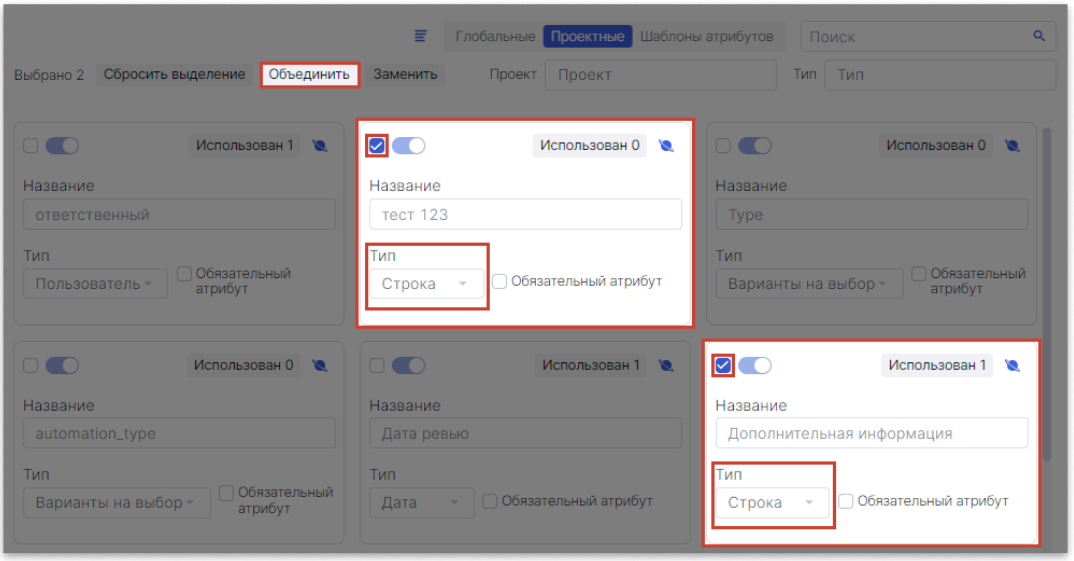
- Сделать проектный атрибут глобальным. В этом случае аттрибут будет доступен не только в проекте, где его создали, а во всей системе Test IT.
- Заменить проектный атрибут или группу проектных атрибутов на глобальные атрибуты *такого же типа*. Для этого:
  1. Выберите карточки атрибутов одного типа.

2. Нажмите **Заменить** в левом верхнем углу окна атрибутов.



3. Выберите глобальный атрибут из открывшегося списка и нажмите **Сохранить**.
4. **Опционально:** Сопоставьте значения проектного атрибута типа **Варианты на выбор** с значениями глобального атрибута, на который вы хотите его заменить. Если вы решите этого не делать, значения проектного атрибута будут утеряны, а сам атрибут удалится после замены на глобальный.

- Объединить несколько проектных атрибутов *одного типа*. Для этого:
  1. Выберите карточки атрибутов одного типа.
  2. Нажмите **Объединить** в левом верхнем углу окна атрибутов.



  3. Отредактируйте получившийся атрибут, затем нажмите **Сохранить**.

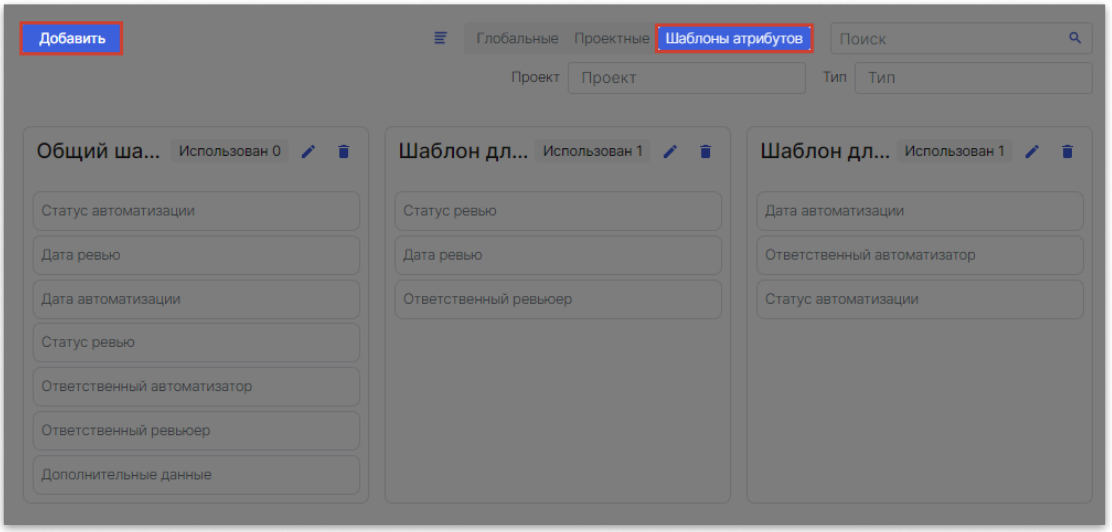
Подробнее: Создание проектных атрибутов.

Создание шаблонов атрибутов

Шаблон атрибутов — это готовый набор глобальных атрибутов, которые можно массово использовать в проектах.

> **Пример из практики**
>
> Вы можете создать шаблон атрибутов **Информация о продукте**, предполагающий множественный выбор: *Серверная* и *Облачная* и сделать его обязательным, чтобы пользователи указывали тип поставки вашего продукта при создании тест-кейсов и/или тест-планов.

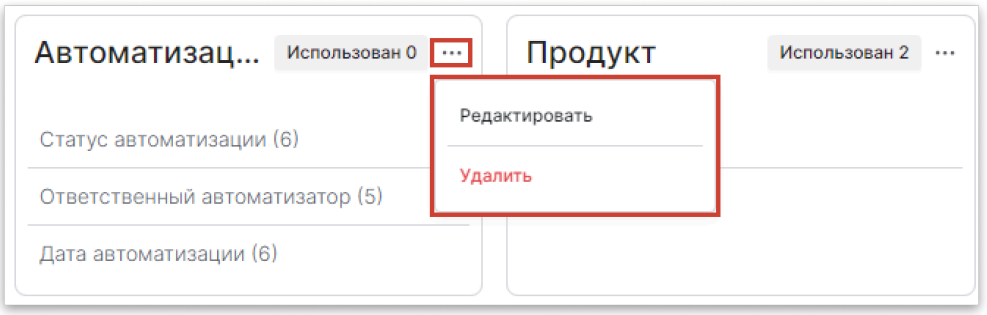
1. Используя аккаунт с правами администратора, нажмите **Настройки проектов** на верхней навигационной панели.
2. В навигационном меню слева выберите **Атрибуты**, перейдите на вкладку **Шаблоны атрибутов** и нажмите **Добавить**.



3. В открывшемся окне введите название шаблона и добавьте в него необходимые атрибуты. Вы также можете создать шаблон атрибутов из вкладки Глобальные атрибуты.

После создания карточка с шаблоном атрибута отобразится в окне. Вы можете:

- Отредактировать шаблон или удалить его из системы, используя меню действий в карточке или строке атрибута.



- Отслеживать, в скольких проектах используется шаблон. Нажмите кнопку **Используется [N]**. Отобразится список проектов, в которых использован шаблон.

Чтобы использовать шаблоны атрибутов в проекте, их нужно добавить в проект.

Обновлено: 4/15/26, 7:08 PM

Интеграция с таск-трекерами

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

- **Для этого раздела мы подготовили видео:**

Интеграция Test IT с таск-трекерами позволяет:

- Автоматически создавать задачи с копиями тестовых сценариев
- Создавать дефекты в таск-трекерах прямо из системы Test IT
- Связывать рабочие элементы с задачами с помощью ссылок
- Отслеживать статусы связанных задач и дефектов из Test IT

Пользователям серверных решений

Интеграция разрабатывается и тестируется с учетом текущих версий таск-трекеров. С некоторыми версиями (например устаревшими) интеграция может работать некорректно.

Этот раздел содержит следующие инструкции:

- Интеграция с Azure DevOps
- Интеграция с ClickUp
- Интеграция с Easy Redmine
- Интеграция с EvaProject
- Интеграция с Kaiten
- **Интеграция с Jira:**
  - Jira Server
  - Jira Cloud
  - Настройка подключения OAuth 1.0 в Jira
- Интеграция с Monday
- Интеграция с Redmine
- Интеграция с TeamStorm
- Интеграция с YouTrack
- Интеграция с Яндекс Трекер

Обновлено: 8/12/26, 4:43 PM

Интеграция с Azure DevOps

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Информация для пользователей серверного решения

Интеграция с серверным решением протестирована на версии **Azure DevOps 2022 (RC1); #19.205.32728.1**.

Интеграция с серверной версией протестирована на версии Azure DevOps 2022 (RC1); #19.205.32728.1.

Вы можете настроить интеграцию с Azure DevOps и создавать задачи прямо из Test IT.

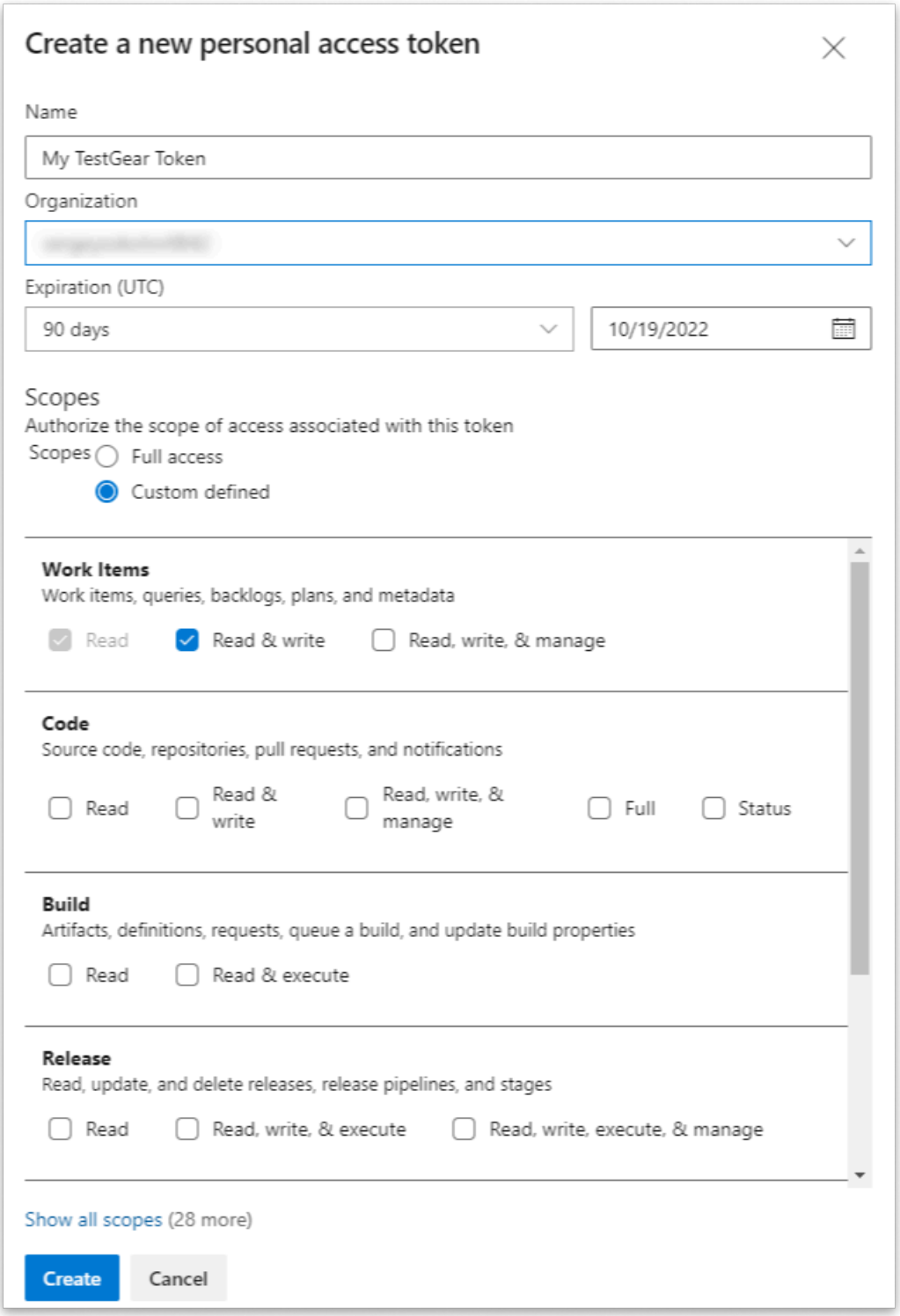
Для этого потребуется:

- Получить токен Azure DevOps
- Создать интеграцию в Test IT

Создание токена в Azure DevOps

1. Откройте ваш профиль Azure DevOps.
2. Откройте настройки профиля.
3. В навигационном меню слева выберите **Personal access tokens**.
4. Нажмите **New Token**.
5. В открывшемся окне укажите:
   - Название токена (**Name**)
   - Вашу организацию (**Organization**)
   - Срок действия токена (**Expiration**)
   - Права доступа к следующим сущностям:

- **Graph**: `Read`
- **Work Items**: `Read & Write`

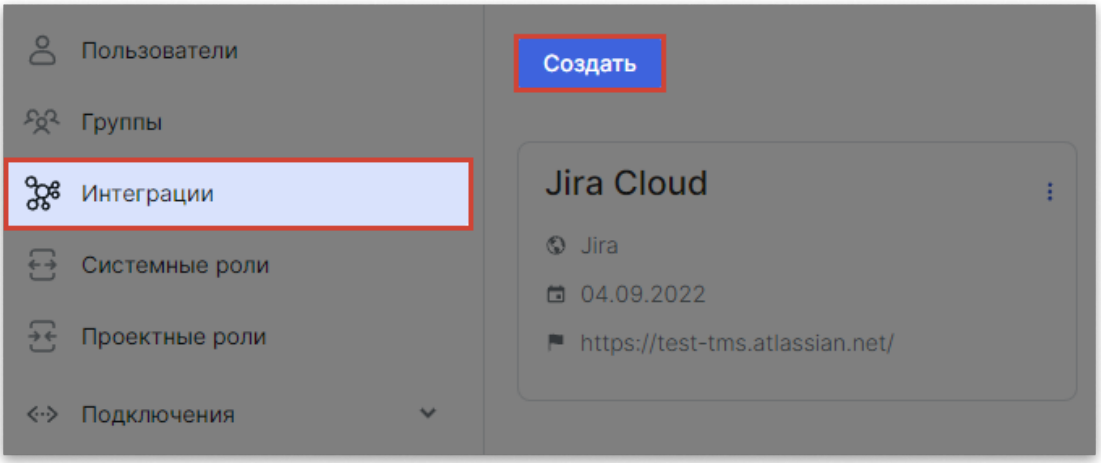


6. Нажмите **Create**.
7. Скопируйте сгенерированный токен. Убедитесь, что вы сохранили токен, так как Azure DevOps не хранит его. Вы можете потерять токен при закрытии окна создания.

Создание интеграции в Test IT

> **Внимание**
>
> Поддерживаются версии Azure DevOps начиная с Azure DevOps Server 2020 и Azure DevOps Services.

1. Войдите в систему с аккаунта администратора и перейдите в раздел **Администрирование**, затем в левом навигационном меню выберите **Интеграции**.
2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **AzureDevOps**.
4. Выберите тип установки: **Cloud** (для облачной версии) или **Self-hosted** (для версии, развернутой на сервере)
5. Заполните данные для подключения:
   - Имя интеграции
   - URL системы Azure DevOps (для облачной версии используйте https://dev.azure.com/).
   - Название организации. Убедитесь, что название организации совпадает с тем, которое вы указали при создании токена в Azure DevOps.
   - Токен. Вставьте скопированный ранее токен из Azure DevOps.
6. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с Azure DevOps.

Обновлено: 5/18/26, 12:34 PM

Интеграция с ClickUp

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Пользователям серверного решения ClickUp

Интеграция с серверным решением протестирована на версии **ClickUp 2.198**.

Вы можете настроить интеграцию с ClickUp и создавать задачи прямо из Test IT.

Внимание

- Интеграция может быть недоступна, т.к. сервис ClickUp прекратил обслуживание пользователей на территории Российской Федерации. Для уточнения информации обратитесь в техническую поддержку: support@testit.software .
- При настройке интеграции указывается URL системы ClickUp: https://app.clickup.com/ .
- Интеграция с ClickUp позволяет создавать задачи одного типа: *Task*.

Для настройки интеграции с ClickUp вам потребуется:

- Получить токен ClickUp
- Создать интеграцию в Test IT

Получение токена доступа к API в ClickUp

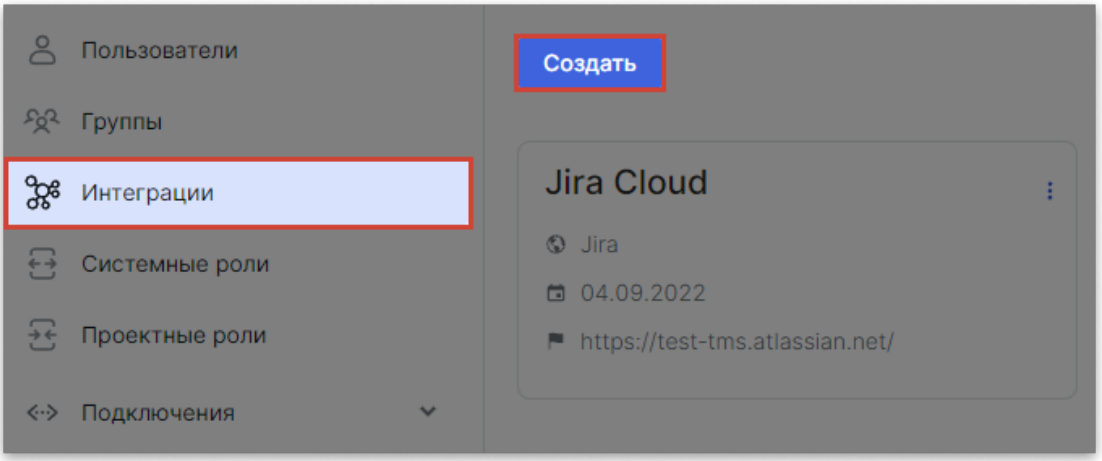
1. Войдите в ClickUp по вашей учетной записи.

2. Если вы используете ClickUp 2.0, нажмите на изображение вашего профиля в левом нижнем углу, затем выберите **Apps**. Если вы используете ClickUp 3.0, нажмите изображение вашего профиля в правом верхнем углу, выберите **Settings**, затем пролистайте станицу вниз и нажмите **Apps** на боковой панели.
3. В секции **API Token** нажмите **Generate**, чтобы создать API-токен.
4. Скопируйте API-токен.

Подробности читайте в официальной документации ClickUp .

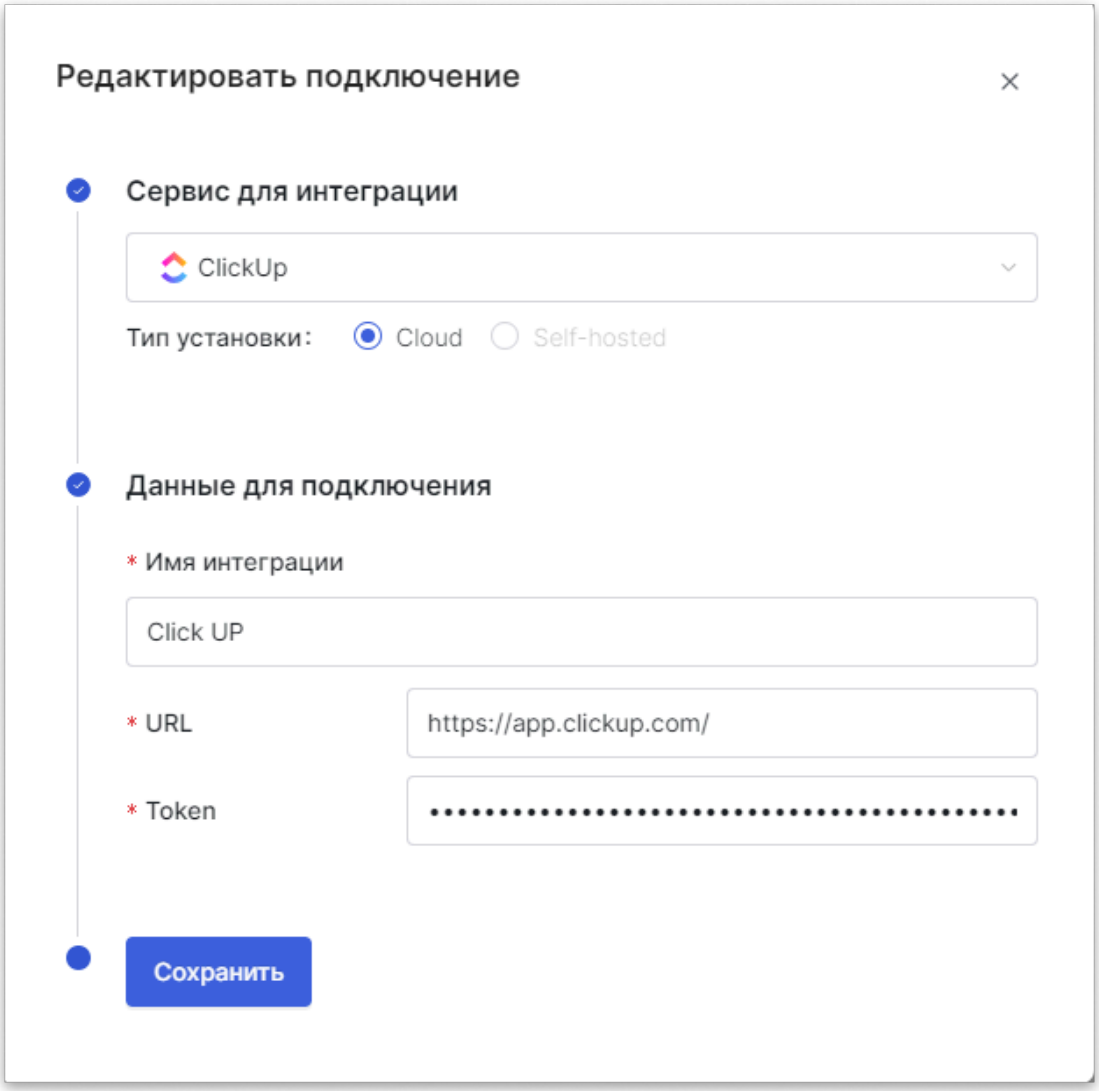
Создание интеграции в Test IT

1. Войдите в систему, используя учетную запись администратора и перейдите в раздел **Администрирование**.
2. Перейдите в раздел **Интеграции**.
3. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
4. В меню **Сервис для интеграции** выберите **ClickUp**.
5. Заполните следующие поля:
   - Имя интеграции
   - URL системы ClickUp: https://app.clickup.com/
   - Токен. Вставьте скопированный ранее **API-токен** из ClickUp.

6. Нажмите **Сохранить**.



Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с ClickUp.

Обновлено: 7/7/26, 2:09 PM

Интеграция с Easy Redmine

Доступна интеграция с облачной версией

Вы можете подключить Test IT к облачной версией Easy Redmine. Интеграция с серверной версией находится в разработке.

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Вы можете настроить интеграцию с Easy Redmine и создавать задачи прямо из Test IT. Для настройки интеграции с Easy Redmine вам потребуется:

- Получить ключ доступа к API в Easy Redmine
- Создать интеграцию в Test IT

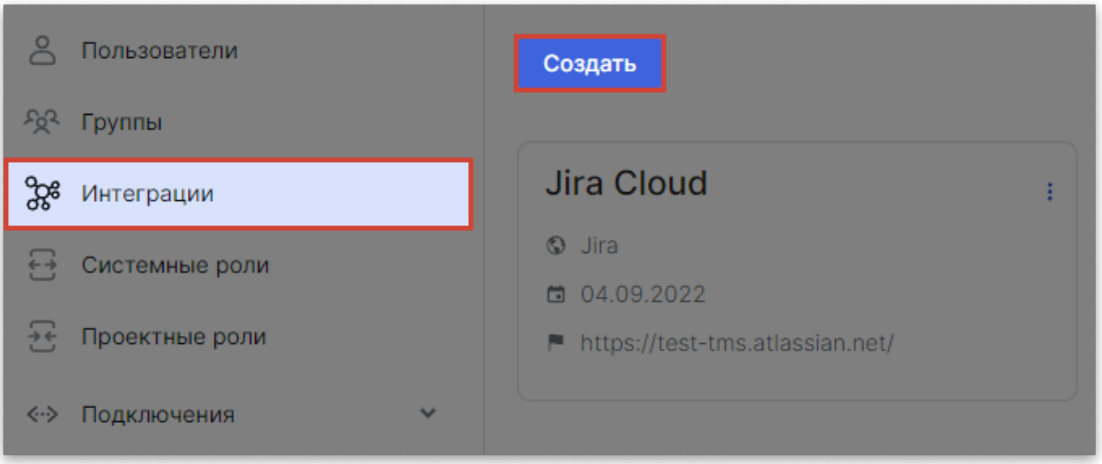
Получение ключа доступа к API в Easy Redmine

Для пользователей с правами администратора:

1. Войдите в аккаунт Easy Redmine.
2. Нажмите на значок профиля пользователя в правом верхнем углу окна, затем нажмите **Редактировать**. Откроется окно редактирования профиля.
3. На вкладке **Общее** пролистайте страницу. Ключ расположен в правой нижней части страницы.
4. Скопируйте ключ доступа в поле **Ключ доступа к API**.

Создание интеграции в Test IT

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.
2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **Easy Redmine**.
4. Заполните данные для подключения:
   - Имя интеграции
   - URL вашей системы Easy Redmine
   - Токен. Вставьте скопированный ранее токен из Easy Redmine.
5. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с Easy Redmine.

Обновлено: 5/18/26, 12:34 PM

Интеграция с EvaProject

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Вы можете настроить интеграцию с EvaProject и создавать задачи прямо из Test IT.

Для настройки интеграции с EvaProject вам потребуется:

- Получить API-токен EvaProject
- Создать интеграцию в Test IT

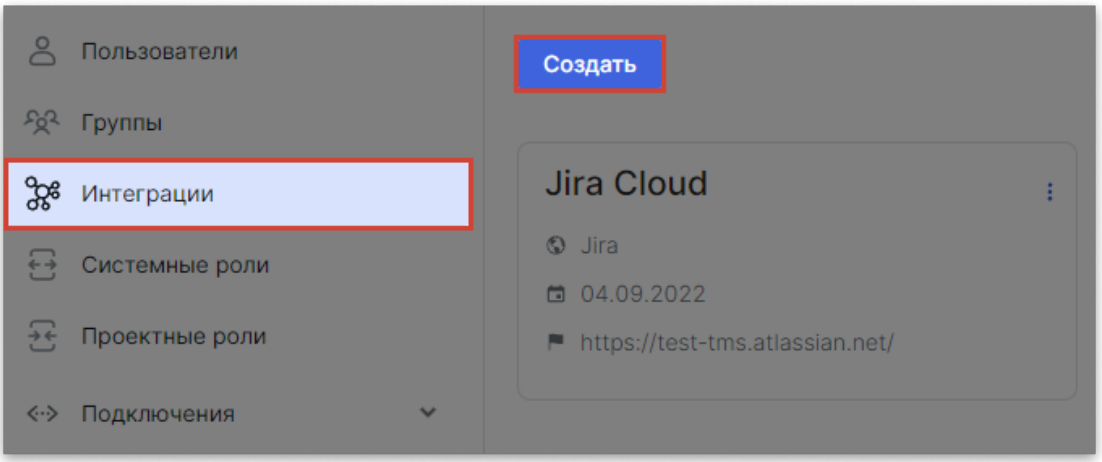
Получение API-токена в EvaProject

1. Создайте специального пользователя для работы с API, например apiuser@example.com.
2. Авторизуйтесь под созданной учетной записью.
3. Нажмите **Моя страница**.
4. Откройте вкладку **Безопасность**.
5. Нажмите **Сгенерировать API-токен**.
6. Сохраните полученный токен. Подробности — в официальной документации EvaProject .

Создание интеграции в Test IT

1. Войдите в систему, используя учетную запись администратора и перейдите в раздел **Администрирование**.
2. Перейдите в раздел **Интеграции**.

3. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
4. В меню **Сервис для интеграции** выберите **EvaProject**.
5. Заполните следующие поля:
   - Имя интеграции
   - URL системы EvaProject
   - Токен. Вставьте скопированный ранее **API-токен** из EvaProject.
6. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с EvaProject.

Обновлено: 5/18/26, 12:34 PM

Интеграция с Jira Cloud

Внимание

Интеграция с Next Gen проектами (проекты, управляемые командой) не поддерживается.

- **Для этого раздела мы подготовили видео:**

Подготовка Jira

Перед созданием интеграции внутри системы Test IT вам необходимо настроить пользовательские поля и новые типы задач в Jira Cloud.

Создание задач типа TestCase

1. Перейдите в настройки системы Jira Cloud.
2. В открывшемся меню выберите **Задачи**.
3. В открывшемся окне убедитесь, что вы находитесь в разделе **Типы задач**, а затем нажмите **Добавить тип задачи**.
4. В открывшемся окне назовите новый тип задачи **TestCase** и убедитесь, что в разделе **Тип** поставлен флажок **Стандартный тип задачи**. Данный тип задач необходим при создании дубликатов тест-кейсов в Jira.

5. Нажмите **Добавить**.
6. Откройте проект, который хотите интегрировать с Test IT. Перейдите в **Настройки проекта → Задачи → Типы**.
7. В колонке слева выберите раздел **Схемы типов задач**.
8. Откройте меню действий для схемы задач проекта и нажмите **Редактировать**.
9. Перетащите созданный тип задачи из поля **Доступно** в поле **Типы задач для текущей схемы**.
10. Нажмите **Сохранить**.

> **Проверьте новый тип задачи**
>
> Убедитесь, что новый тип задачи присутствует во всех необходимых схемах, иначе дублирование работать не будет.

Получение токена Jira Cloud

1. Нажмите на значок профиля и выберите в открывшемся меню **Настройки профиля**.
2. Перейдите в раздел **Безопасность**.
3. В открывшейся секции **Приватный токен** нажмите **Сгенерировать**.
4. В открывшемся окне подтвердите действие.

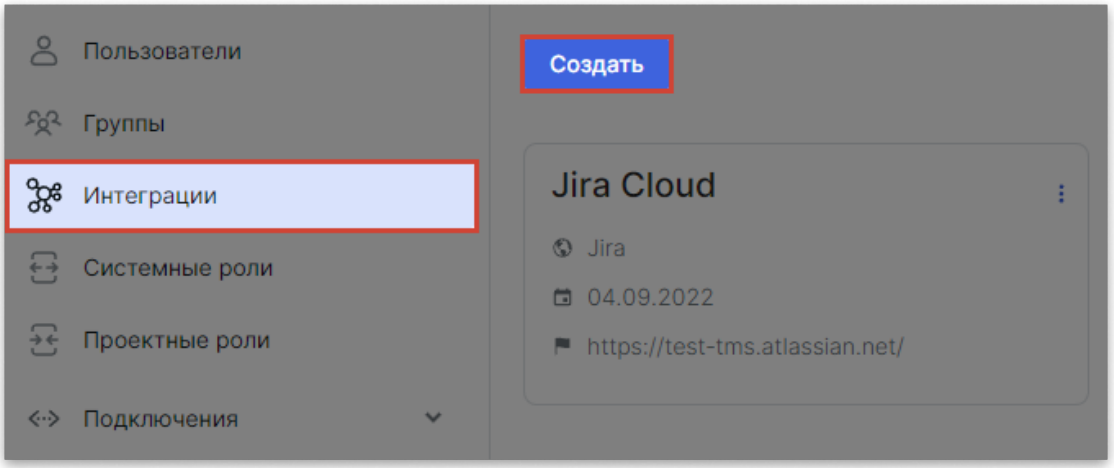
Система сгенерирует токен API, который вы можете копировать и использовать как пароль для интеграции с системой Test IT.

Создание интеграции с Test IT

> **Требуется переменная FRONTEND_URL**
>
> Для корректной работы интеграции с Jira нужно задать в `.env` -файле переменную `FRONTEND_URL` .

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.

2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **Jira**, затем выберите тип установки: **Cloud**.
4. Заполните данные для подключения:
   - Имя интеграции
   - URL вашей системы Jira с протоколом и портом
   - Логин аккаунта Jira с правами администратора
   - Токен API для Jira Cloud, сгенерированный ранее
5. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее.

> **Внимание**
>
> Полное подключение новой интеграции на стороне Jira может занять некоторое время (до нескольких часов после создания интеграции).

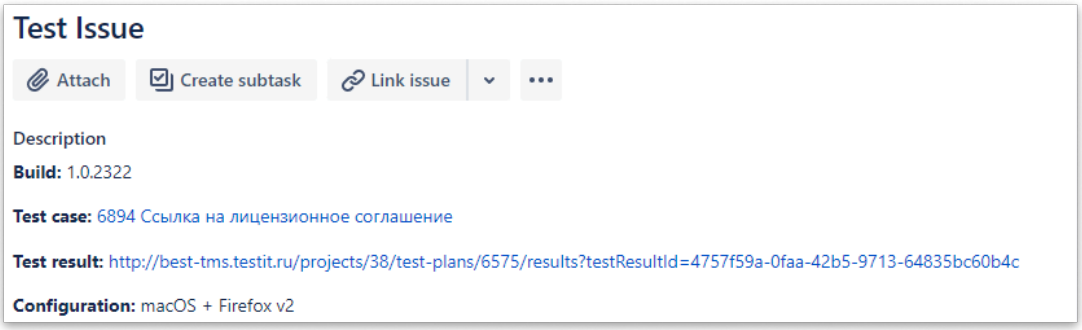
Интеграцию с Jira в дальнейшем можно использовать для подключения проекта Test IT к Jira.

Проверка работы интеграции

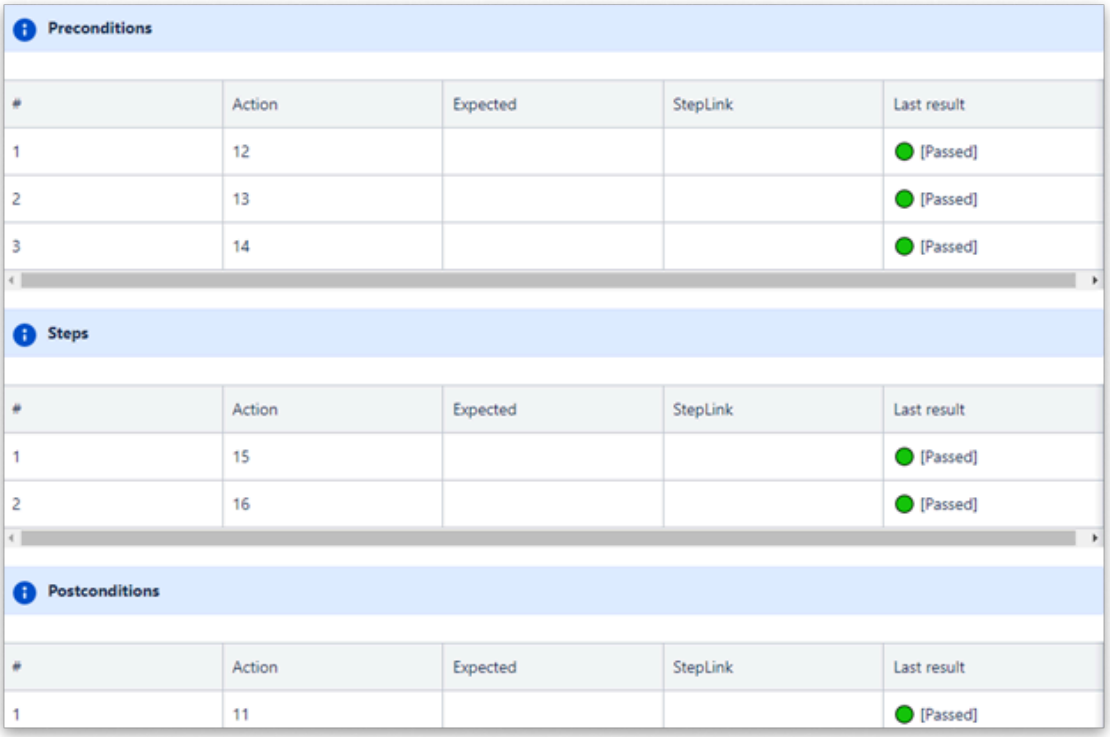
1. Создайте тестовый проект.
2. Добавьте в него пользователей.
3. Подключите проект к созданной интеграции с Jira.
4. Создайте тест-кейс.
5. Перейдите в раздел **Тест-планы** и создайте новый тест-план.
6. Откройте созданный тест и поставьте ему результат **Провален**. В верхней части карточки теста должна появиться кнопка **Сохранить и создать дефект**.
7. Нажмите **Сохранить и создать дефект** и выберите проект для создания задачи. Откроется окно создания дефекта в Jira. Убедитесь, что поле

Description содержит информацию о тест-кейсе.

8. Нажмите **Create**. В описании дефекта должно появиться следующее:
   - Ссылка на тест-кейс в поле **Test Case**.
   - Ссылка на результат теста, для которого создавался дефект, в поле **Test Result**.



9. Перейдите по ссылке в поле **Test Result**. Должен открыться результат теста, для которого вы создавали дефект. В поле **Ссылки** должна появиться ссылка на созданный вами дефект. Проверьте ее.
10. Перейдите в раздел **Тесты**.
11. Откройте созданный тест-кейс. В правой части карточки тест-кейса должно появиться поле **Копия теста в Jira** со ссылкой на дубликат тест-кейса.
12. Перейдите по ссылке. В открывшемся дубликате теста должны отображаться те же шаги, что и в созданном тест-кейсе. Последний результат прохождения каждого шага должен отображаться в таблице в поле **Last Result**.



Если все шаги выше пройдены успешно, ваша интеграция успешно настроена.

Обновлено: 8/12/26, 4:43 PM

Настройка подключения Jira через OAuth 1.0

> **Внимание**
>
> Для корректной работы интеграции задайте значение переменной `FRONTEND_URL` в `.env` -файле.

Настройка подключения к Jira через Oauth 1.0 включает в себя 3 этапа:

1. Получение ключей в Test IT
2. Настройка учетной записи в Jira
3. Установка подключения с Jira в Test IT

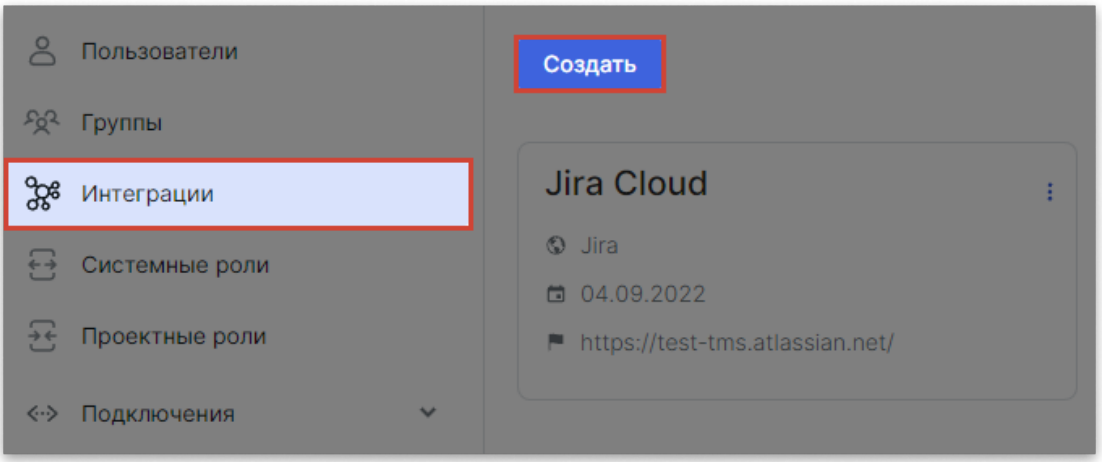
Получение ключей в Test IT

На данном этапе создается Consumer Key (в Jira — *Ключ покупателя, Ключ потребителя*) и копируется Публичный ключ, а также указывается URL учетной записи Jira, с которой создается подключение.

> **Сохраните Consumer Key!**
>
> Сохраните Consumer Key. Он потребуется для получения ссылки на приложение в Jira.

1. Авторизуйтесь в Test IT.
2. В верхней навигационной панели выберите **Администрирование**, затем в открывшемся меню слева выберите **Интеграции**.

3. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
4. В меню **Сервис для интеграции** выберите **Jira**, затем выберите тип установки: **Cloud** или **Self-hosted**.
5. Заполните данные для подключения:
   - Введите имя интеграции.
   - Выберите опцию **OAuth 1.0** как тип подключения
   - В поле **URL** введите URL учетной записи Jira, с которой хотите настроить подключение
   - Придумайте Consumer Key, введите его в поле Consumer Key и сохраните в отдельном файле. Если вы начали настройку в Jira, и Consumer Key у вас уже есть, скопируйте его в поле **Consumer Key**.
   - Скопируйте публичный ключ из соответствующего поля.

> **Совет: не закрывайте вкладку.**
>
> Не закрывайте вкладку, в которой открыта система Test IT. После получения ссылки и токенов в Jira, вы сможете продолжить работу, не выполняя лишних шагов.

Настройка учетной записи Jira

Для подключения необходимо ввести данные учетной записи Test IT.

> **Внимание**
>
> Порядок ввода данных и названия кнопок могут различаться в зависимости от версии Jira.

1. В новой вкладке откройте учетную запись Jira, с которой хотите настроить подключение.
2. В левом верхнем углу выберите:
   - Для Jira Server — **Администрирование → Приложения** (*Administration → Applications*)
   - Для Jira Cloud — **Настройки → Продукты** (*Settings → Products*)
3. В левом навигационном меню выберите **Ссылки на приложения** (*Application links*), затем нажмите **Создать ссылку** (*Create link*).
4. В открывшемся окне **Создать ссылку** заполните необходимую информацию (см. Данные для подключения).

Данные для подключения

Укажите данные для подключения:

- **[Для Jira Server]:** В секции **Тип приложения** (*Application Type*) выберите **Продукты Atlassian** (*Atlassian Produts*).
- В поле **URL приложения** (*Application URL*) вставьте URL вашей учетной записи Test IT.
- В поле **Название приложения** (*Application Name*) введите название приложения (может быть произвольным, например *Test IT*).
- Поле **Тип приложения** (*Application Type*) оставьте значение **Базовое приложение** (*Generic Application*), которое используется по умолчанию.
- В поле **Имя поставщика услуг** (*Service Provider Name*) введите имя поставщика услуг, которое может быть произвольным.
- В поле **Ключ покупателя** (также **Ключ потребителя** / *Consumer Key*) введите *Consumer key*, созданный в Test IT. Если вы начали настройку в Jira, придумайте Consumer key и сохраните его для использования в Test IT.
- В поле **Общий секретный ключ** (*Shared secret*) введите произвольное значение или оставьте поле пустым.
- В поля **URL запроса маркера** (*Request Token URL*), **URL точки доступа** (*Access Token URL*) и **Авторизация URL** (*Authorize URL*) вставьте URL вашей учетной записи Test IT.
- Отметьте флажком опцию **Создать входящую ссылку** (*Create Incoming Link*).
- Заполните поле **Имя пользователя** (*Consumer Name*), которое может быть произвольным, например имя администратора Test IT.
- В поле **Открытый ключ** (*Public Key*) вставьте Публичный ключ из Test IT. :::

5. Нажмите **Продолжить**.

Установка подключения

На этапе установки подключения создаются два токена: **Access token** и **Access token secret**. Токены вводятся в Test IT, после чего устанавливается соединение с Jira.

1. Вернитесь в учетную запись Test IT.
2. Если вы закрывали вкладку, откройте окно создания подключения. Для этого выполните шаги 2-4, описанные в разделе Подготовка к настройке подключения.
3. Нажмите **Создать подключение**. Вы будете перенаправлены в Jira на форму **Подтверждение доступа**, которая откроется в новой вкладке.
4. В форме подтверждения доступа нажмите **Allow**. Отобразится сообщение, содержащее два уникальных токена — **Access token** и **Access token secret**: *Access is allowed! Access token: `<access_token>` Access token secret: `<access_token_secret>`* .
5. В форме **Новое подключение** введите полученные значения **Access token** и **Access token secret**.
6. Нажмите **Сохранить**.

Добавление администратора

Чтобы подключить другого администратора Jira (изменить автора задачи), потребуется заново создать **Ссылку на приложение** и **токены** в Jira с новым публичным ключом, авторизоваться в Jira от имени пользователя, которого нужно подключить к Test IT, в рамках одной сессии браузера, затем повторить шаги, описанные в разделе Настройка учетной записи в Jira. :::

Обновлено: 8/13/26, 12:57 PM

Интеграция с Kaiten

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Вы можете настроить интеграцию с Kaiten и создавать задачи прямо из Test IT. Для настройки интеграции с Kaiten вам потребуется:

- Получить API-ключ Kaiten
- Создать интеграцию в Test IT

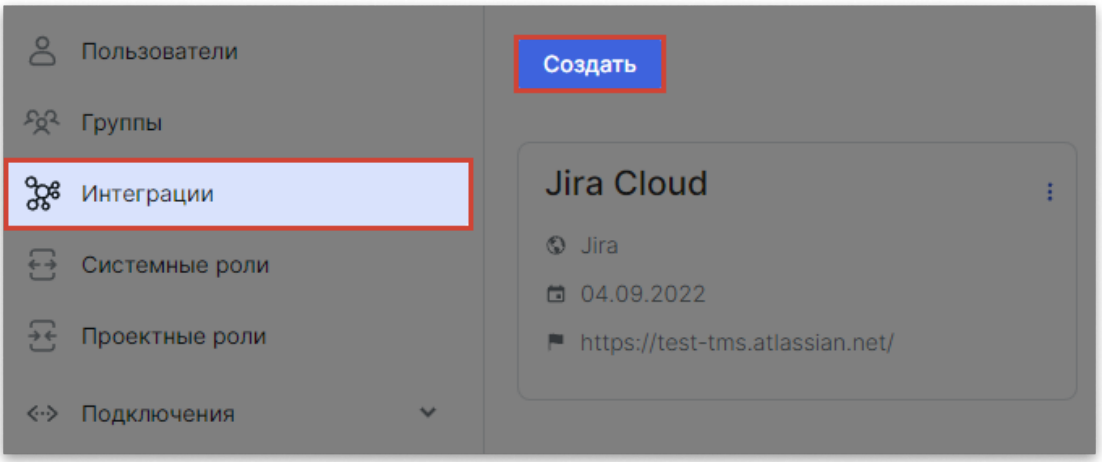
Получение API-ключа в Kaiten

1. Откройте Kaiten.
2. В правом верхнем углу окна нажмите значок вашего профиля и выберите **Настройки профиля**.
3. В меню слева выберите **Ключ доступа API**.
4. Нажмите **Создать ключ**.
5. Нажмите значок **Копировать**, чтобы скопировать API-ключ.
6. Сохраните API-ключ в безопасном месте для настройки интеграции с Test IT.

Создание интеграции в Test IT

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.

2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **Kaiten**.
4. Выберите форму поставки вашей системы Kaiten:
   - Для облачной версии выберите **Cloud**.
   - Для серверной версии выберите **Self-hosted**.
5. Заполните данные для подключения:
   - Имя интеграции
   - URL вашей системы Kaiten
   - Токен. Вставьте скопированный ранее API-ключ из Kaiten.
6. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с Kaiten.

Обновлено: 8/13/26, 12:57 PM

Интеграция с Monday

Доступна интеграция с облачной версией

Вы можете подключить Test IT к облачной версии Monday. Интеграция с серверной версией находится в разработке.

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Возможны ограничения на стороне Monday

Интеграция может быть недоступна, т.к. сервис Monday прекратил обслуживание пользователей на территории Российской Федерации. Для уточнения информации обратитесь в техническую поддержку: support@testit.software .

Вы можете настроить интеграцию с Monday и создавать задачи прямо из Test IT. Для настройки интеграции с Monday вам потребуется:

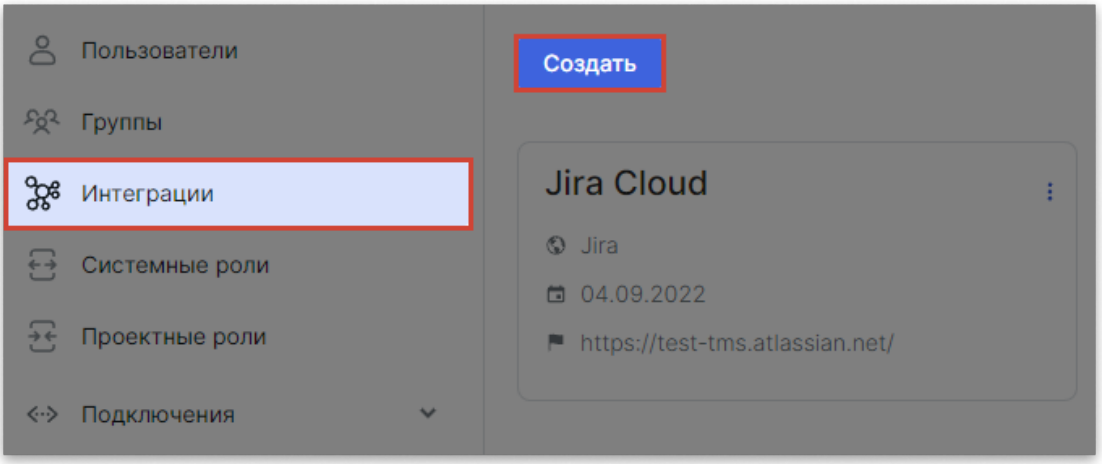
- Получить API-токен в Monday
- Создать интеграцию в Test IT

Получение API-токена в Monday

1. Войдите в аккаунт Monday.
2. Нажмите на значок профиля пользователя в правом верхнем углу окна, затем выберите **Администрирование** (**Administration**). От кроется окно администрирования.
3. В меню, расположенном слева, выберите **API**.
4. Нажмите значок копирования, чтобы скопировать API-токен.

Создание интеграции в Test IT

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.
2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **Monday**.
4. Заполните данные для подключения:
   - Имя интеграции
   - URL вашей системы Monday
   - Токен. Вставьте скопированный ранее токен из Monday.
5. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с Monday.

Обновлено: 9/10/26, 6:52 PM

Интеграция с Redmine

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Пользователям серверного решения Redmine

Интеграция с серверным решением протестирована на версии **Redmine 4.1.1**.

Вы можете настроить интеграцию с Redmine и создавать задачи прямо из Test IT. Для настройки интеграции с Redmine вам потребуется:

- Получить ключ доступа к API в Redmine
- Создать интеграцию в Test IT

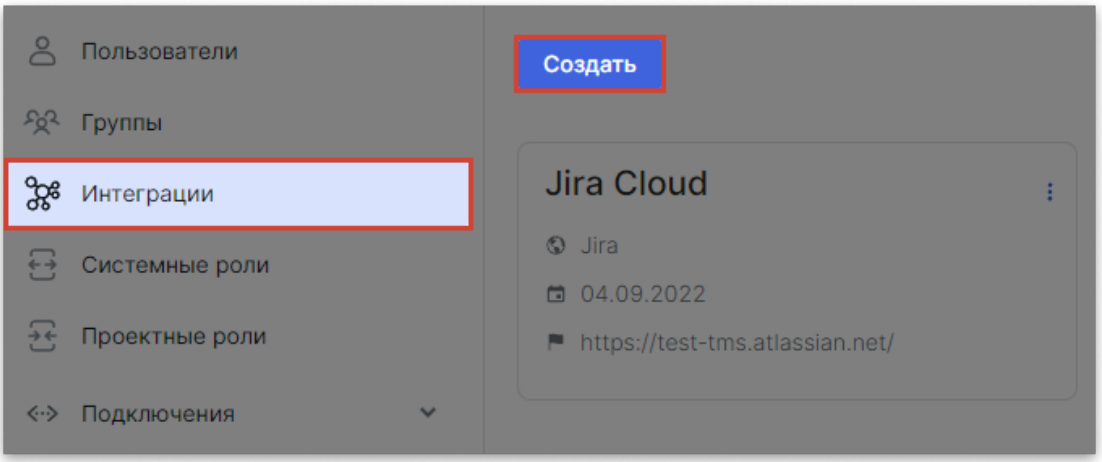
Получение ключа доступа к API в Redmine

1. Войдите в Redmine по вашей учетной записи.
2. Войдите в ваш профиль Redmine. Для этого нажмите **Моя учетная запись** (*My account*) в правой верхней части окна.
3. Скопируйте **Ключ доступа к API** (*API access key*).

Создание интеграции в Test IT

1. Войдите в систему, используя учетную запись администратора и перейдите в раздел **Администрирование**.
2. Перейдите в раздел **Интеграции**.

3. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
4. В меню **Сервис для интеграции** выберите **Redmine**.
5. Заполните следующие поля:
   - Имя интеграции
   - Тип форматирования текста: HTML или Markdown
   - URL системы Redmine
   - Токен. Вставьте скопированный ранее **Ключ доступа к API** из Redmine.
6. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее.
7. **Опционально:** Чтобы изменить настройки, откройте меню **Действия** справа от названия интеграции, нажмите **Редактировать**, затем внесите необходимые изменения.

После успешной установки соединения вы сможете привязывать проекты к интеграции с Redmine.

Проверка работы интеграции

После создания интеграции рекомендуем вам проверить корректность ее работы. Это позволит убедиться, что задачи создаются в соответствии с вашими требованиями.

1. Добавьте интеграцию в проект, следуя инструкции.
2. В библиотеке тестов создайте тест-кейс для проверки интеграции.
3. Используя тест-кейс, создайте тест-план.
4. Откройте созданный тест и поставьте ему результат **Провален**.
   В верхней части карточки теста должна появиться кнопка **Сохранить и**

создать дефект.

5. Нажмите **Сохранить и создать дефект**, затем выберите проект Redmine для создания дефекта и тип задачи. Откроется окно создания дефекта в Redmine. Убедитесь, что все поля содержат информацию о проваленном тесте.
6. Заполните поле **Status**, выбрав нужную опцию из выпадающего меню. Убедитесь, что статусы задач соответствуют статусам в Redmine.
7. Нажмите **Сохранить**.
   В Redmine должна автоматически создаться новая задача, а на странице теста в меню слева должна быть ссылка на нее.
8. Из меню слева перейдите по ссылке на задачу.
9. Убедитесь, что содержание задачи соответствует содержанию тест-кейса, а форматирование текста не нарушено.
10. Если форматирование текста в задаче некорректно, измените тип форматирования в настройках интеграции.
    Если все данные отображаются корректно, интеграция настроена успешно. Вы можете переходить к созданию дефектов.

Обновлено: 5/18/26, 12:34 PM

Интеграция с TeamStorm

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

TeamStorm и Test IT образуют единую экосистему, что обеспечивает простоту интеграции, повышает надежность вашей ИТ-инфраструктуры и независимость от сторонних сервисов. Вы можете настроить интеграцию с TeamStorm и создавать задачи прямо из Test IT.

Для настройки интеграции с TeamStorm вам потребуются:

- Доступ к учетной записи TeamStorm с системными ролями **Системный администратор** и **Пользователь** для получения токена
- Учетная запись Test IT с доступом к модулю **Администрирование** для создания интеграции

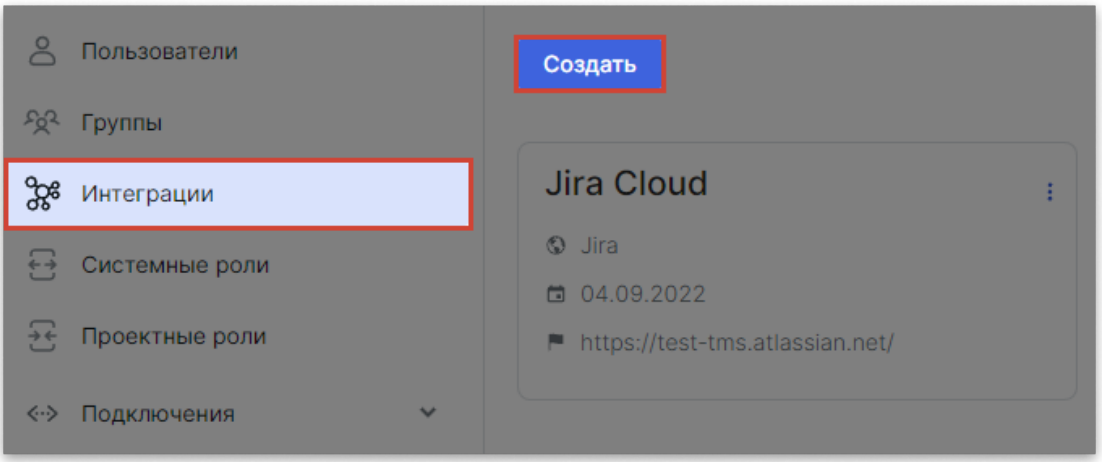
Получение токена TeamStorm

1. Войдите в учетную запись TeamStorm.
2. Получите токен TeamStorm, следуя инструкции .

Создание интеграции в Test IT

1. Войдите в Test IT, используя учетную запись, от имени которой будут создаваться дефекты в системе TeamStorm.
2. Откройте раздел **Администрирование**.

3. Откройте вкладку **Интеграции** и нажмите **Создать**.



   Откроется окно **Новое подключение**.
4. В меню **Сервис для интеграции** выберите **TeamStorm**.
5. Заполните данные для подключения:
   - В поле **Имя** введите имя новой интеграции.
   - В поле **URL** укажите адрес сервера, на котором развернута система TeamStorm.
   - В поле **Токен** вставьте токен TeamStorm, скопированный ранее.
6. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с TeamStorm.

Отслеживание созданных дефектов

После настройки интеграции вы сможете создавать дефекты в TeamStorm из системы Test IT. Ссылки на созданные дефекты будут отображаться в следующих разделах Test IT:

- **Тест-план** (колонка **Ссылки**)
- **История** выполнения задач (тест-план, страница тест-кейса, колонка **Ссылки**)
- **Отчет** по тест-плану (секция **Дефекты тест-плана**, колонка **Дефекты**)
- **Ссылки** на странице тест-кейса (колонка **Ссылки**)
- **История результатов** на странице тест-кейса (колонка **Ссылки**)

Обновлено: 5/18/26, 12:34 PM

Интеграция с YouTrack

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Возможны ограничения на стороне YouTrack

Интеграция может быть недоступна, т.к. сервис YouTrack прекратил обслуживание пользователей на территории Российской Федерации. Для уточнения информации обратитесь в техническую поддержку: support@testit.software .

Пользователям серверного решения YouTrack

Интеграция с серверным решением протестирована на версии **YouTrack 2022.3**.

Вы можете настроить интеграцию с YouTrack и создавать задачи прямо из Test IT. Для настройки интеграции с YouTrack вам потребуется:

- Получить ключ доступа к API в YouTrack
- Создать интеграцию в Test IT

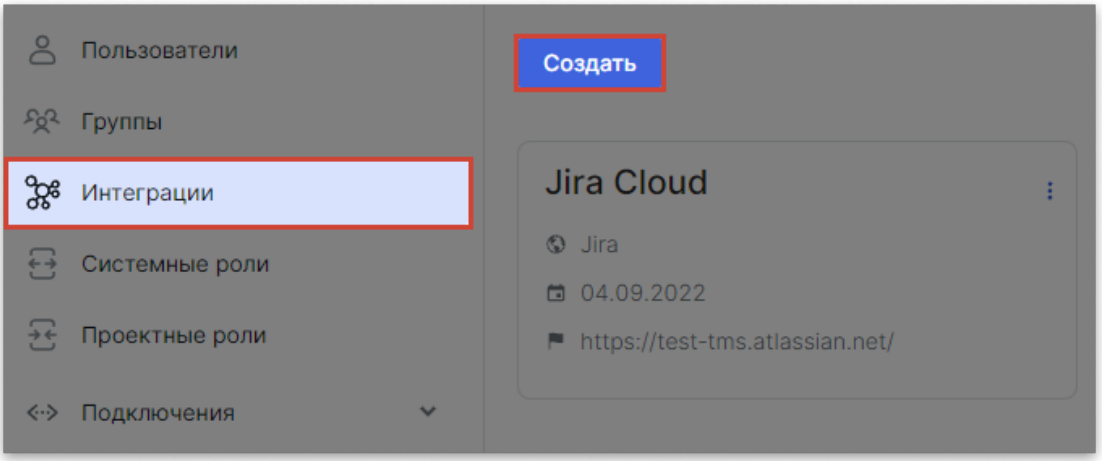
Получение ключа доступа к API в YouTrack

1. Войдите в аккаунт YouTrack.
2. Нажмите на значок профиля пользователя, затем из выпадающего меню выберите **Профиль**.
3. Перейдите во вкладку **Безопасность аккаунта**.
4. Нажмите **Новый токен**.

5. В окне **Новый постоянный токен** заполните поля **Имя** (название токена) и **Область доступа**.
6. Нажмите **Создать**.
7. Скопируйте токен.

Создание интеграции в Test IT

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.
2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите YouTrack.
4. Заполните данные для подключения:
   - Имя интеграции
   - Тип установки: выберите **Cloud** для облачной версии или **Self-hosted** для серверной версии YouTrack.
   - URL системы YouTrack
   - Токен. Вставьте скопированный ранее токен из YouTrack.
5. Нажмите **Сохранить**. Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.

После успешной установки соединения вы сможете привязывать проекты к интеграции с YouTrack.

Обновлено: 7/7/26, 2:09 PM

Интеграция с Битрикс24

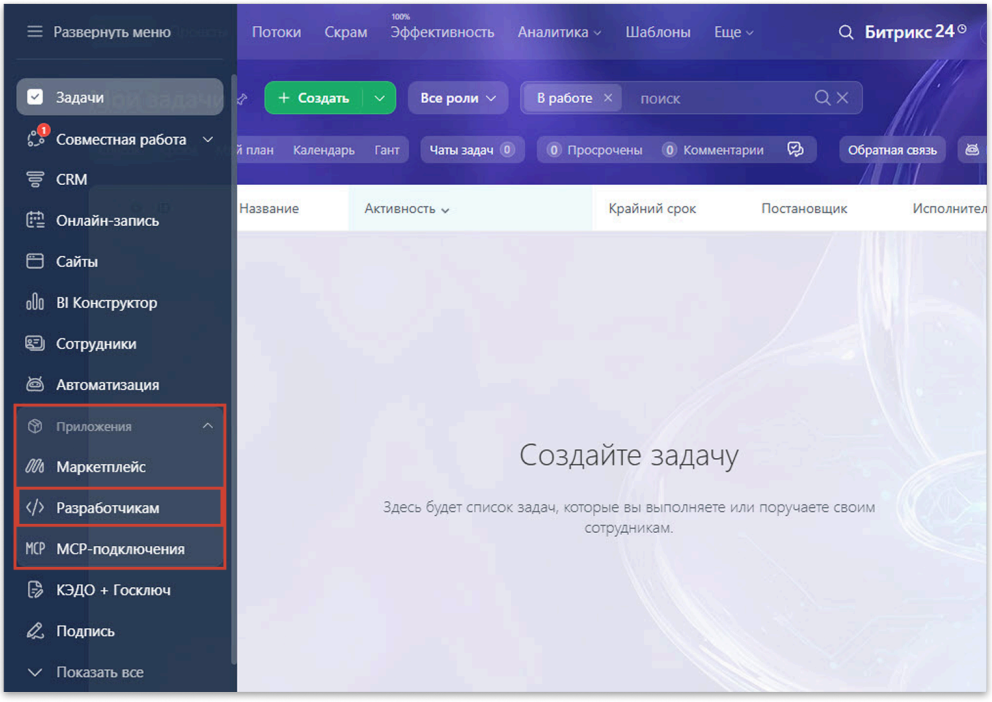
> **Ограничения функциональности полей**
>
> При создании дефекта для заполнения будут доступны только обязательные поля. Для множественных пользовательских полей можно заполнить только 1 значение.

Вы можете настроить интеграцию с Битрикс24 и создавать задачи прямо из Test IT. Для настройки интеграции вам потребуется:

- Получить входящий вебхук в Битрикс24
- Создать интеграцию в Test IT

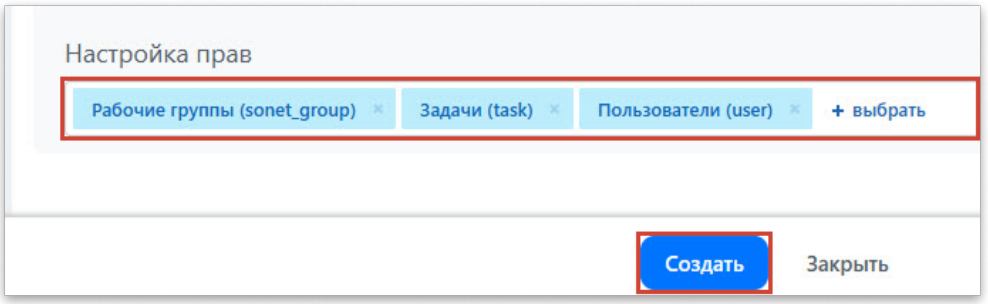
Получение входящего вебхука в Битрикс24

1. Войдите в аккаунт Битрикс24.
2. В левой навигационной панели откройте **Приложения** → **Разработчикам**.



3. В открывшемся окне нажмите **Другое**, затем выберите **Входящий вебхук**.
   Откроется страница настройки вебхука.

4. На странице настройки вебхука в меню **Настройка прав** выберите необходимые права:
   - **Рабочие группы (sonet_group)**
   - **Пользователи (user)**
   - **Задачи (task)**



5. Нажмите **Создать**.
   Откроется список созданных вебхуков.
6. Чтобы открыть страницу вебхука и скопировать его данные, дважды нажмите на вебхук в списке.
7. Скопируйте данные из поля **Вебхук для вызова rest api**.

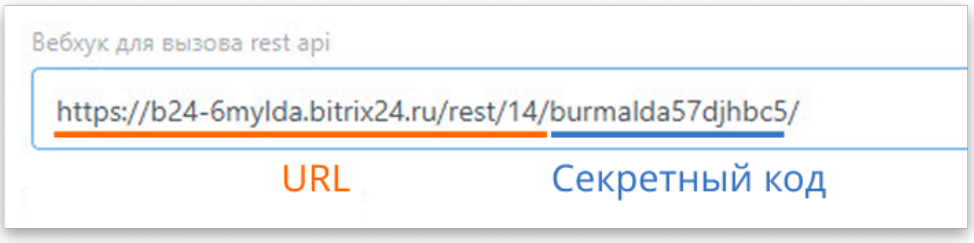
Подробнее: Официальная документация Битрикс24

URL и секретный код вебхука Bitrix24

> **В URL вебхука укажите** `https`
>
> Битрикс24 требует защищенное HTTPS-соединение к REST API по соображениям безопасности. Поэтому при настройке в URL вебхука указывайте `https` (а не `http`).

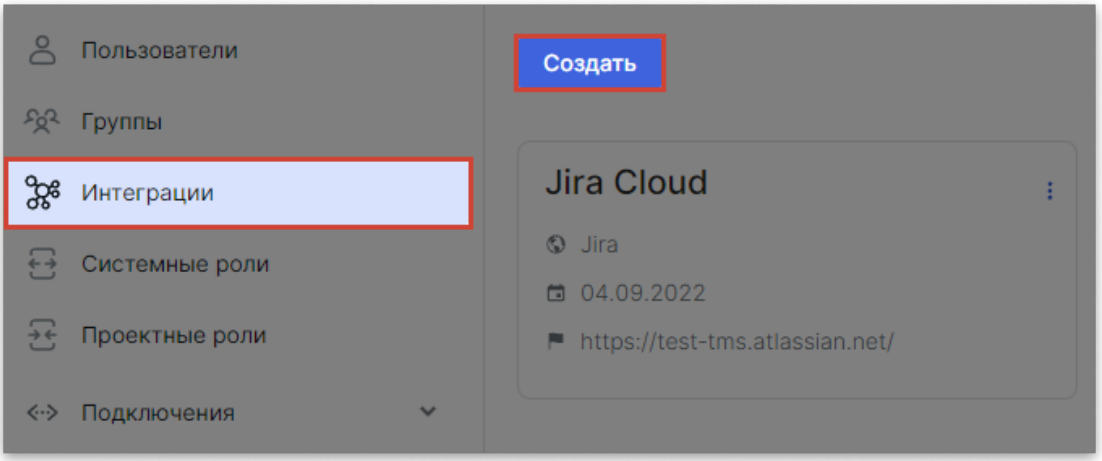
Скопированные данные вебхука содержат URL и Секретный код. При настройке интеграции в Test IT они указываются отдельно друг от друга. Например, вебхук `https://b24-6mylda.bitrix24.ru/rest/14/burmalda57djhbc5/` содержит:

- URL: `https://b24-6mylda.bitrix24.ru/rest/14/`
- Секретный код (`/` в конце не используется): `burmalda57djhbc5`

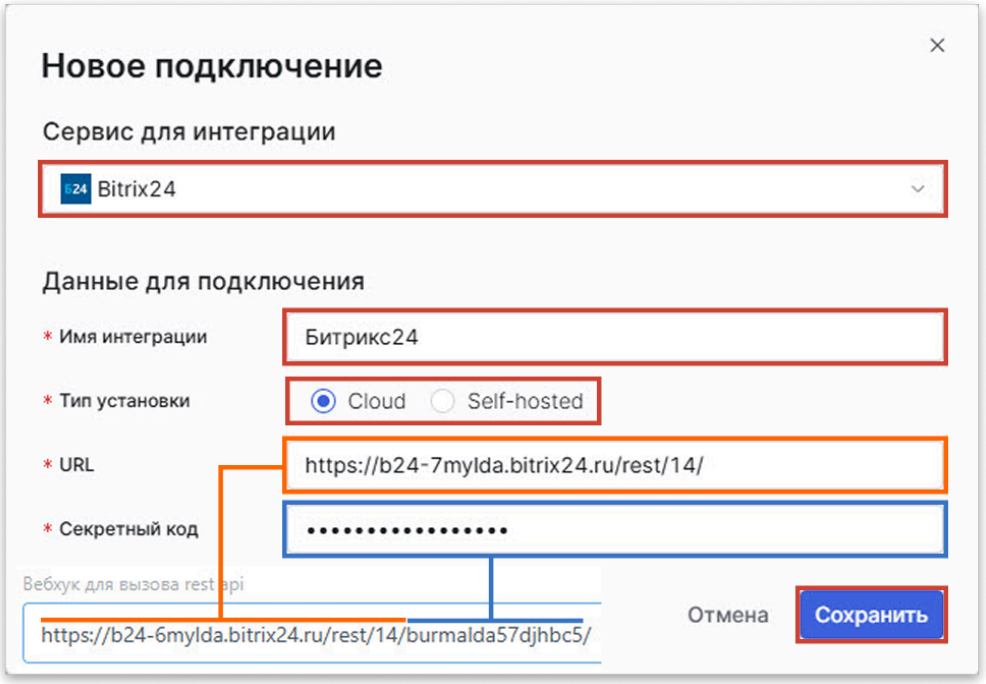


Создание интеграции в Test IT

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.
2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **Bitrix24**.
4. Заполните данные для подключения:
   - Имя интеграции
   - Тип установки
   - URL входящего вебхука Битрикс24 (`https` , а не `http`)
   - Секретный код. Вставьте часть вебхука для вызова rest api, содержащую секретный код.
5. Нажмите **Сохранить**. Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время.



После успешной установки соединения вы сможете привязывать проекты к интеграции с Битрикс24.

Обновлено: 10/29/26, 1:42 PM

Интеграция с Яндекс Трекер

> **Может потребоваться добавление адресов Yandex Cloud в белый список**
>
> **Пользователям Test IT Облако:** Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.
>
> Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Вы можете настроить интеграцию с Яндекс Трекер и создавать в нем задачи прямо из Test IT.

Настройка включает в себя этапы:

1. Получение OAuth-токена:
   - Получение идентификатора приложения (Client ID) на портале Яндекс ID
   - Получение Oauth-токена с использованием Client ID
2. Получение ID организации в Яндекс Трекер
3. Создание интеграции в Test IT

Получение OAuth-токена в Яндекс ID

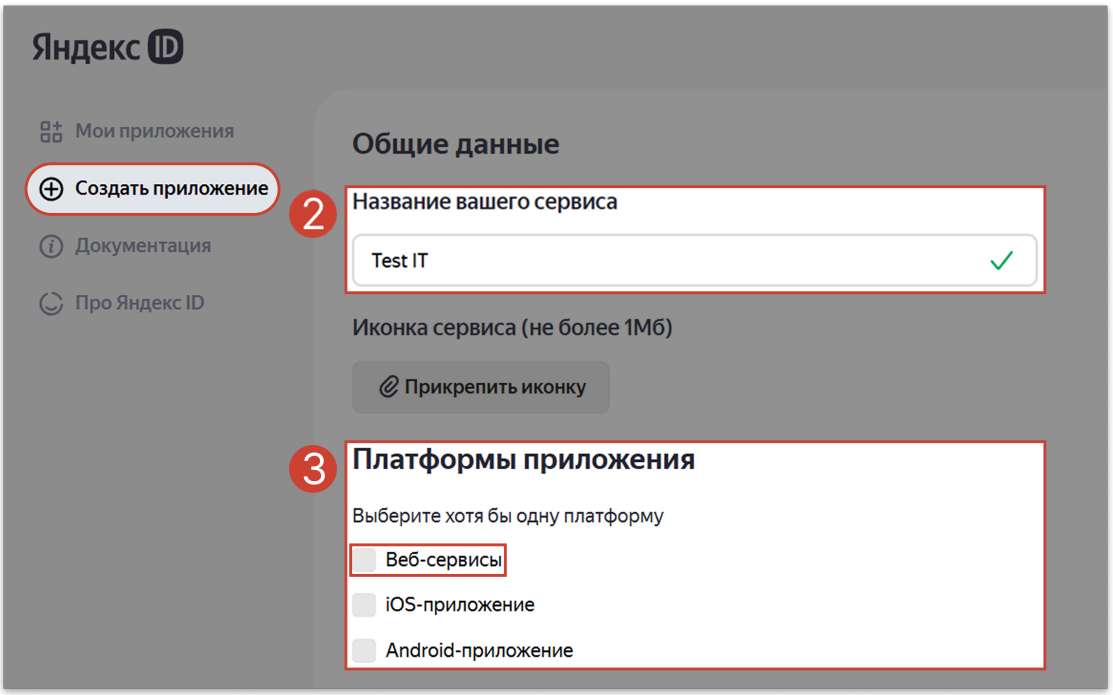
Для настройки интеграции нужно получить OAuth-токен в сервисе Яндекс ID. Подробности смотрите в официальной документации Яндекс .

Получение Client ID на портале Яндекс ID

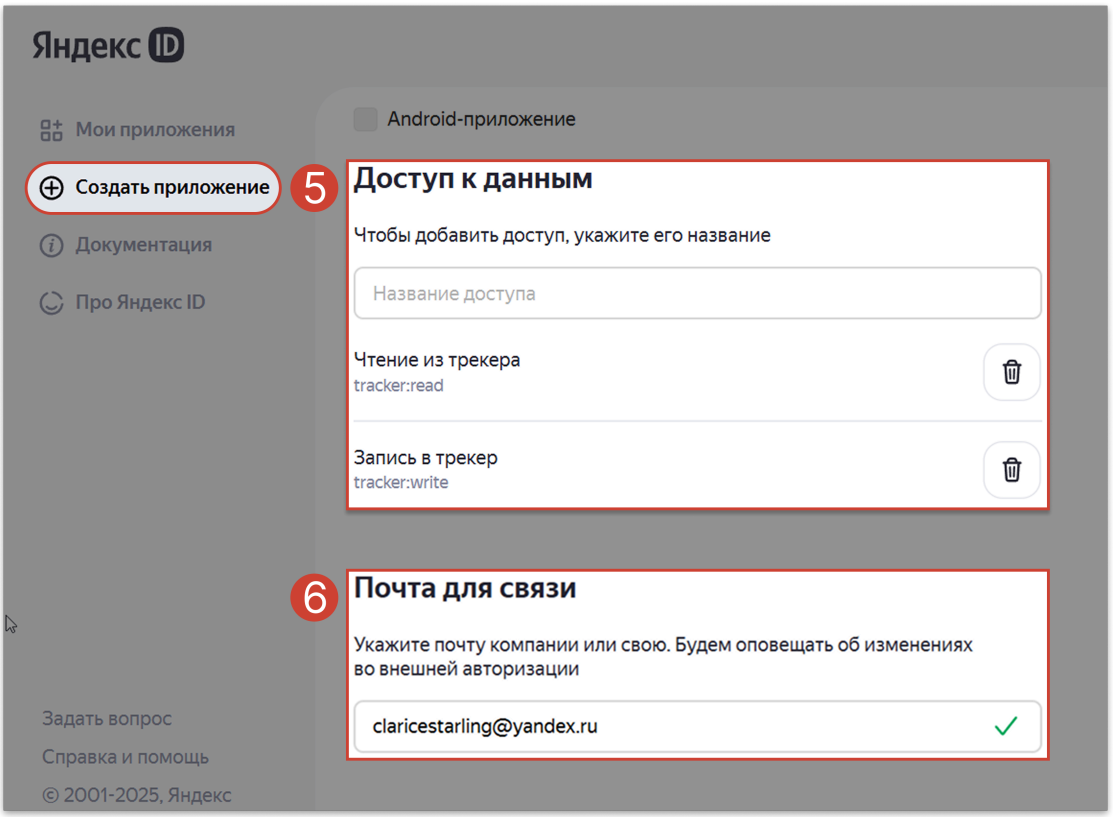
Чтобы получить идентификатор приложения (Client ID):

1. В Яндекс ID откройте страницу регистрации приложения, пройдя по ссылке: `https://oauth.yandex.com/client/new` . При необходимости пройдите регистрацию. Откроется сервис Яндекс ID. По умолчанию откроется страница **Создать приложение**.

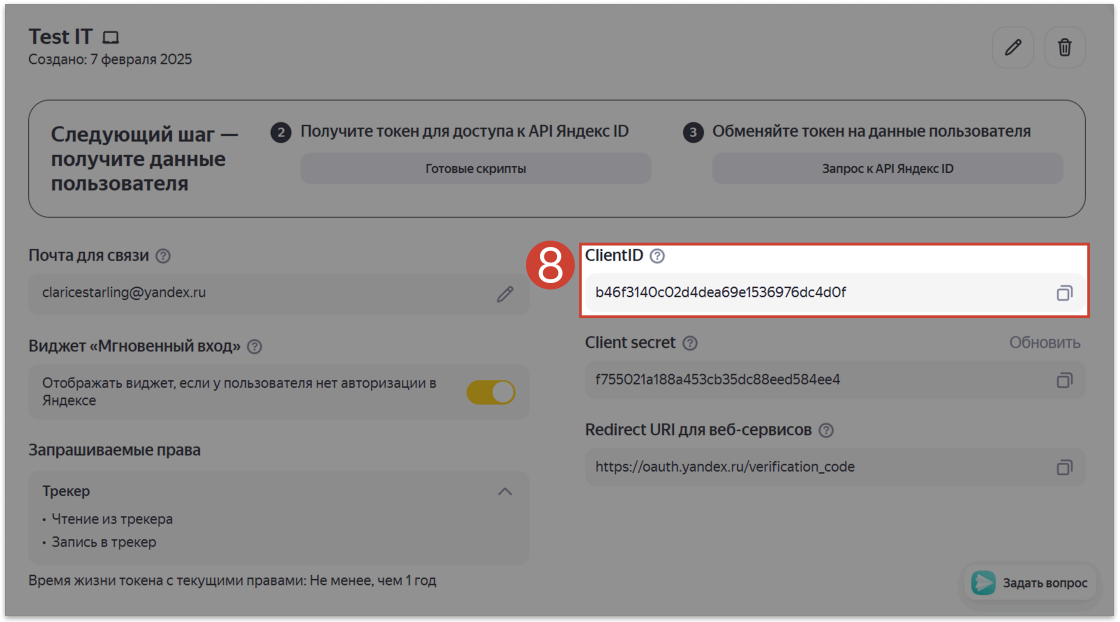
2. В поле **Название вашего сервиса** укажите название приложения (например, Test IT).
3. В разделе **Платформы приложения** выберите **Веб-сервисы**.



4. В поле **Redirect URI** вставьте URL: https://oauth.yandex.ru/verification_code.
5. В выпадающем меню **Доступ к данным** выберите типы доступа: **Чтение из трекера** и **Запись в трекер**. Чтобы найти тип доступа, в поле начните вводить его название.
6. В поле **Почта для связи** укажите адрес электронной почты для получения уведомлений.



7. Нажмите **Создать приложение**. Откроется окно приложения.
8. В поле **Client ID** скопируйте значение идентификатора. Идентификатор потребуется для получения OAuth-токена.



Получение OAuth-токена в Яндекс ID

Для настройки интеграции нужно получить OAuth-токен. Подробности смотрите в официальной документации Яндекс .

Чтобы получить OAuth-токен:

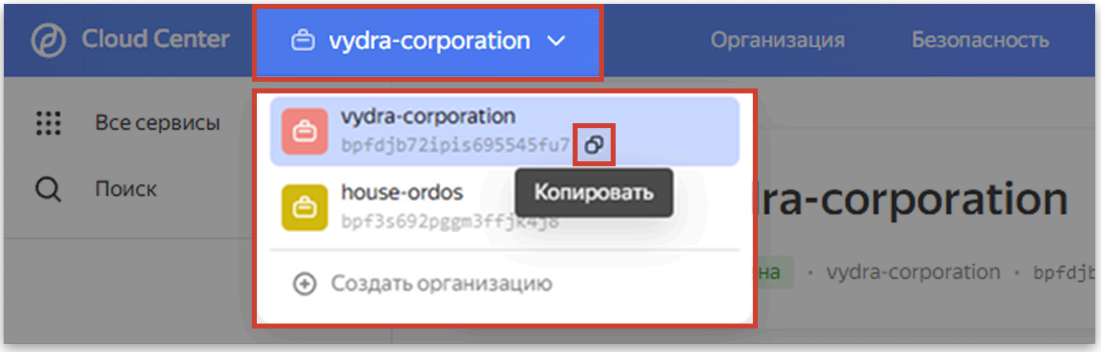
1. В Яндекс ID откройте страницу получения токена, пройдя по ссылке: `https://oauth.yandex.ru/authorize?response_type=token&client_id=<идентификатор_приложения>` . Откроется страница с сообщением об ошибке (404), так как в ссылке не указан идентификатор приложения.
2. В адресной строке замените текст <идентификатор_приложения> на идентификатор, полученный ранее. Например: `https://oauth.yandex.ru/authorize?response_type=token&client_id=b46f3140c02d4dea69e1536976dc4d0f` . Перейдите по ссылке. Откроется окно с OAuth-токеном.
3. Скопируйте OAuth-токен.

Получение идентификатора организации в Яндекс Трекер

Идентификатор организации можно скопировать в Yandex Center. Если вы используете Yandex Cloud, идентификатор можно получить в Cloud Center .

Чтобы скопировать идентификатор организации:

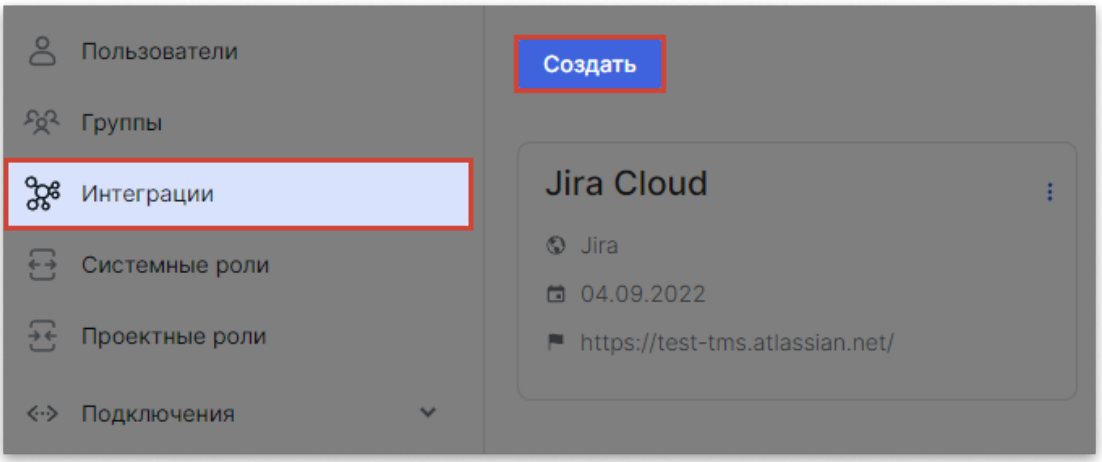
1. Войдите в ваш аккаунт Яндекс Трекер.
2. Откройте раздел **Администрирование** и выберите **Организации**.
3. Скопируйте идентификатор вашей организации.



Создание интеграции в Test IT

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.

2. Нажмите **Создать**.



   Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **Яндекс Трекер**.
4. Заполните данные для подключения:
   - Имя интеграции
   - URL системы Яндекс Трекер. Он общий для всех аккаунтов: https://api.tracker.yandex.net
   - Токен. Вставьте скопированный ранее OAuth-токен из Яндекс Трекер.
   - Тип организации: **Yandex Cloud Organization** или **Яндекс 360**
   - Идентификатор организации. Вставьте идентификатор, скопированный из Яндекс Трекер.
5. Нажмите **Сохранить**.

Система Test IT автоматически протестирует интеграцию и в случае успешного тестирования сохранит ее. Это может занять некоторое время. После успешной установки соединения вы сможете добавить интеграцию в проект.

Проверка работы интеграции

После создания интеграции рекомендуем вам проверить корректность ее работы. Это позволит убедиться, что задачи создаются в соответствии с вашими требованиями.

1. Добавьте интеграцию в проект, следуя инструкции.
2. В библиотеке тестов создайте тест-кейс для проверки интеграции.
3. Используя тест-кейс, создайте тест-план.
4. Откройте созданный тест и поставьте ему результат **Провален**.
   В верхней части карточки теста должна появиться кнопка **Сохранить и**

создать дефект.

5. Нажмите **Сохранить и создать дефект**, затем выберите проект Яндекс Трекер для создания дефекта и тип задачи. Откроется окно создания дефекта в Яндекс Трекер. Убедитесь, что все поля содержат корректную информацию о проваленном тесте.
6. Нажмите **Сохранить**.
   В Яндекс Трекер должна автоматически создаться новая задача. В Test IT на странице теста в меню слева должна появиться ссылка на нее.
7. Из меню слева перейдите по ссылке на задачу.
8. Убедитесь, что содержание задачи соответствует содержанию тест-кейса.
   Если все данные отображаются корректно, интеграция настроена успешно. Вы можете переходить к созданию дефектов.

Обновлено: 8/13/26, 11:57 AM

Интеграция Test IT с ИИ-моделями

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

В Библиотеке Test IT вы можете создавать тесты с помощью искусственного интеллекта (ИИ). Тесты создаются по задачам из вашего таск-трекера. Из одной задачи создается несколько тестовых сценариев. ИИ-генерация платная, оплата производится провайдеру ИИ-модели.

Рекомендации по формулированию задач

Чтобы создание тестов с помощью ИИ-модели было продуктивным и безопасным, перед началом работы убедитесь, что:

- Задачи, по которым создаются тест-кейсы, не содержат чувствительных данных, подлежащих NDA. Все данные передаются нейросети и обрабатываются ей.
- Задачи, по которым создаются тест-кейсы, имеют полное и развернутое описание и соответствуют спецификации. Изображения, вложенные файлы и комментарии к задачам в настоящее время не обрабатываются: нейросеть генерирует тест-кейсы по текстовому описанию в задаче. Специального формата для описания задач в таск-трекере нет.

Подробнее читайте в блоге Test IT .

Для генерации тестов вам потребуется:

- Настроить интеграцию с таск-трекером и подключить ее к проекту
- Настроить интеграцию с ИИ-моделью и добавить ее в проект

Для интеграции с Test IT доступны следующие сервисы:

- ГигаЧат за исключением эмбеддинг-моделей (embedding)

- YandexGPT за исключением речевых моделей (speech-моделей)
- Сервисы, совместимые с OpenAI

Обновлено: 8/12/26, 4:43 PM

Интеграция с ГигаЧат

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Эмбеддинг-модели недоступны

Интеграция с эмбеддинг-моделями от ГигаЧат недоступна.

Для ИИ-генерации тестов с помощью ГигаЧат (GigaChat) требуется аккаунт в Сбер Developer Studio (Сбер Studio) с правами администратора, содержащий проект GigaChat API Business.

В Личном кабинете Studio необходимо получить:

- Client Secret — секретный ключ клиента
- Client ID — идентификатор клиента

Также потребуются настройки на стороне Test IT.

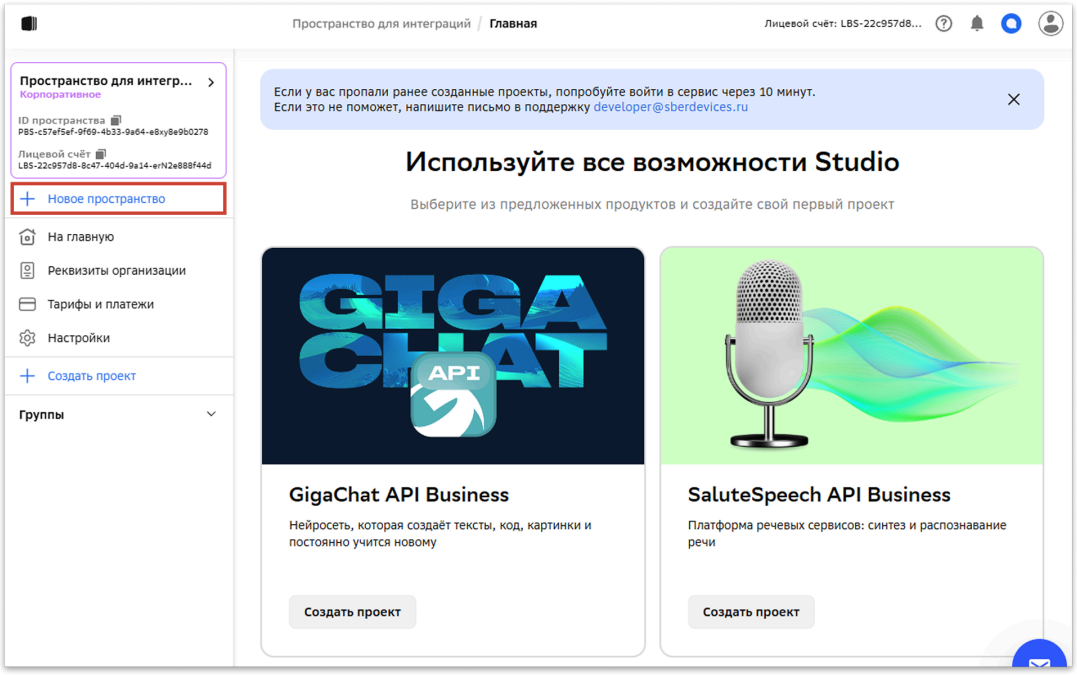
Функциональность Сбер Studio может допускать различные сценарии. Подробности смотрите в официальной документации Сбер Studio .

Использование API ГигаЧат платное.

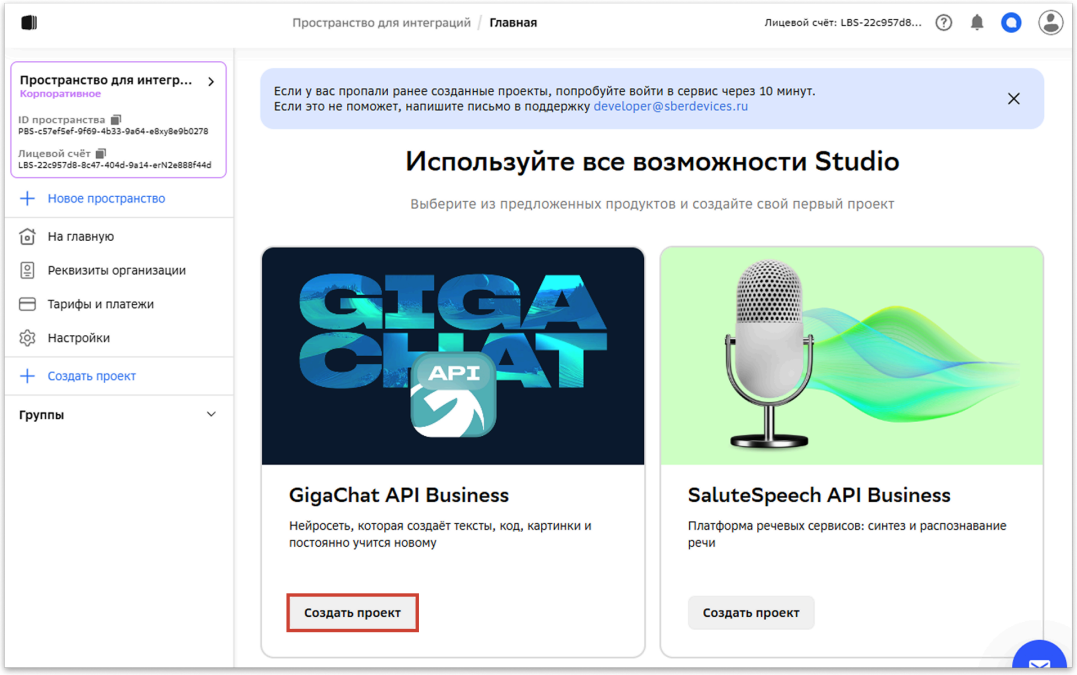
Получение Client Secret и Client ID в Сбер Studio

В эту инструкцию входит описание регистрации в Сбер Studio, создания пространства и проекта GigaChat API Business. Если у вас уже есть аккаунт и пространство с проектом, можно переходить к шагу 4.

1. Если у вас нет аккаунта Сбер Developer Studio , зарегистрируйтесь, используя Сбер ID . Следуйте инструкциям на экране. После регистрации будет первое пространство будет создано по умолчанию.
2. **Опционально:** Создайте новое пространство :
   - Нажмите **+ Новое пространство**.



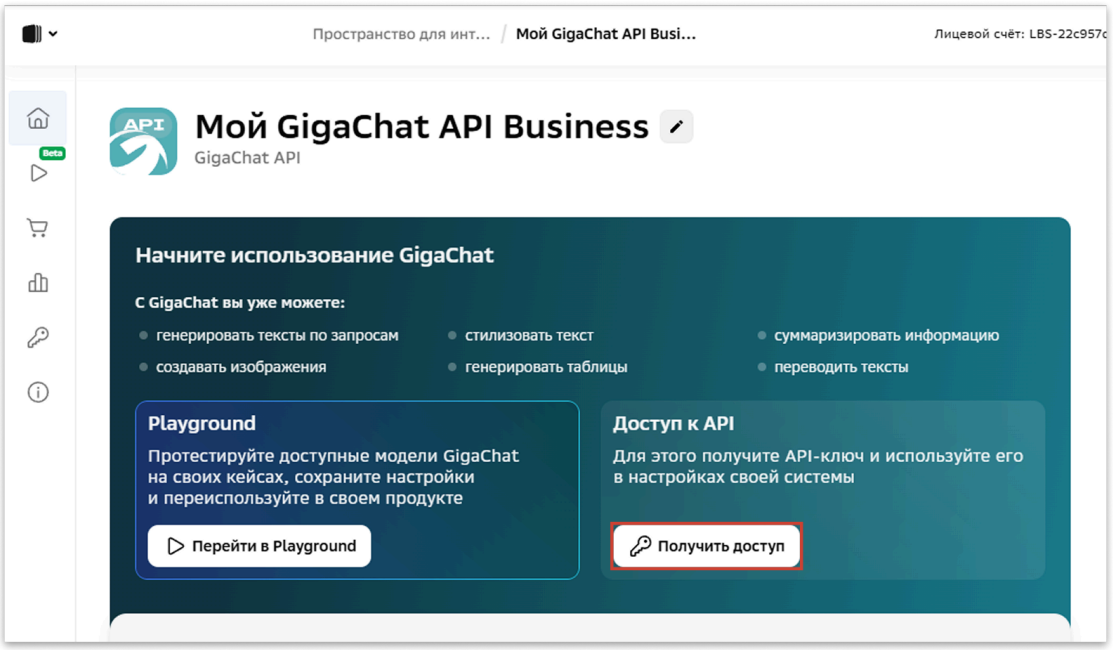
   - В окне создания пространства введите имя пространства, выберите его тип, затем нажмите **Создать**.
3. Создайте проект:
   - В секции **GigaChat API Business** нажмите **Создать проект**.



   - В окне создания проекта введите имя проекта, при необходимости сделайте дополнительные настройки, затем нажмите **Создать**

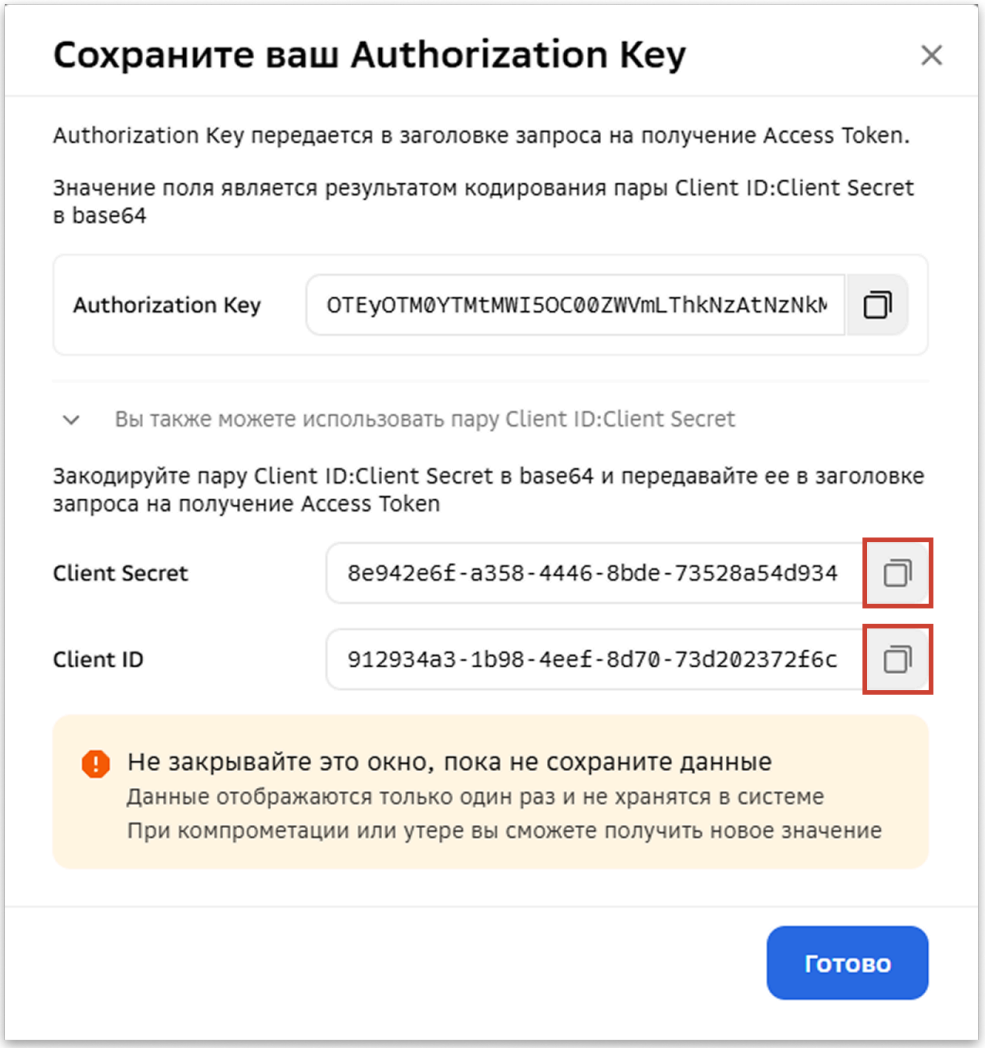
проект. Откроется окно проекта.

4. В секции проекта **Доступ к API** нажмите **Получить доступ**.



Откроется окно настройки API.

5. Нажмите **Получить новый ключ**. Обратите внимание, при создании ключа текущий *Authorization key* будет потерян. Откроется окно с обновленными данными авторизации.
6. Разверните окно данных авторизации, нажав **>** ниже поля **Authorization Key**. Отобразятся значения **Client Secret** и **Client ID**.
7. Скопируйте значения **Client Secret** и **Client ID** и сохраните.



8. Нажмите **Готово**.

Создание интеграции в Test IT

1. Используя аккаунт с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.
2. Нажмите **Создать**. Откроется окно **Новое подключение**.
3. В меню **Сервис** для интеграции выберите **GigaChat**.
4. Заполните данные для подключения:
   - Имя интеграции
   - Client Secret
   - Client ID
   - Api version: выберите **personal** для физических лиц; **paid** для юридических лиц, использующих платные пакеты ; **corporate** для юридических лиц, использующих план pay-as-you-go
5. Нажмите **Сохранить**.
   Созданная интеграция с ГигаЧат отобразится на странице интеграций.

Для создания тестов с помощью ИИ потребуется добавить интеграцию в проект.

Обновлено: 8/13/26, 12:57 PM

Интеграция с YandexGPT

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Речевые модели недоступны

Интеграция с речевыми моделями (speech-моделями) от YandexGPT недоступна.

Для генерации тестов из таск-трекера с помощью ИИ-модели YandexGPT потребуются:

- Folder id — идентификатор каталога, в котором размещается ресурс в Yandex Cloud
- API key — API-ключ Yandex Cloud

Получение этих значений включает этапы:

- Получение идентификатора каталога в Yandex Cloud:
  - Регистрация в Yandex Cloud
  - Копирование идентификатора каталога в Yandex Cloud
- Получение API-ключа в Yandex Cloud:
  - Создание сервисного и платежного аккаунтов Yandex Cloud
  - Получение API-ключа

Также потребуются настройки на стороне Test IT.

Функциональность Yandex Cloud допускает различные сценарии. Подробности смотрите в официальной документации Yandex Cloud .

Использование YandexGPT платное. Оплата производится после создания платежного аккаунта.

Получение идентификатора каталога в Yandex Cloud

Регистрация в Yandex Cloud

Если вы уже зарегистрированы в Yandex Cloud, пропустите этот раздел.

1. Перейдите на сайт Yandex Cloud и нажмите **Подключиться**.
2. Зарегистрируйтесь, используя ваш Yandex ID (если у вас его нет, создайте его).
3. После авторизации войдите в консоль Yandex Cloud, нажав **Консоль** в правом верхнем углу окна справа от значка аккаунта.
4. Создайте организацию, следуя инструкции Yandex Cloud . При создании организации будет автоматически создано облако , содержащее каталог Yandex Cloud.

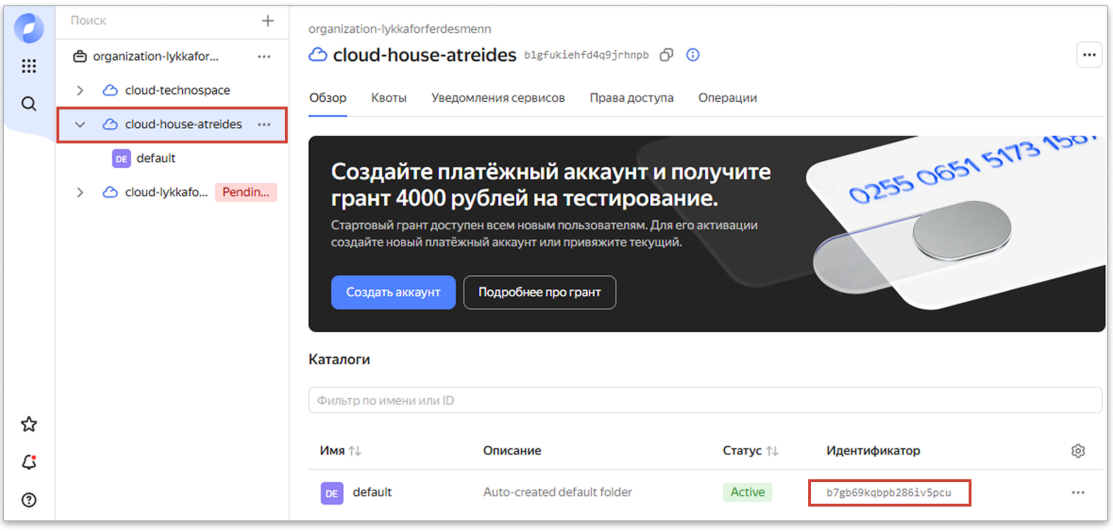
Копирование идентификатора каталога в Yandex Cloud

Каталог (*folder*) создается по умолчанию при создании облака. Вы можете использовать каталог по умолчанию или создать новый, следуя инструкции Yandex Cloud .

Чтобы получить идентификатор каталога:

1. В консоли управления в списке слева выберите облако. В центральной части экрана откроется список каталогов.

2. В колонке **Идентификатор** скопируйте идентификатор каталога.



При настройке интеграции в Test IT вставьте его в поле **folder_id**.

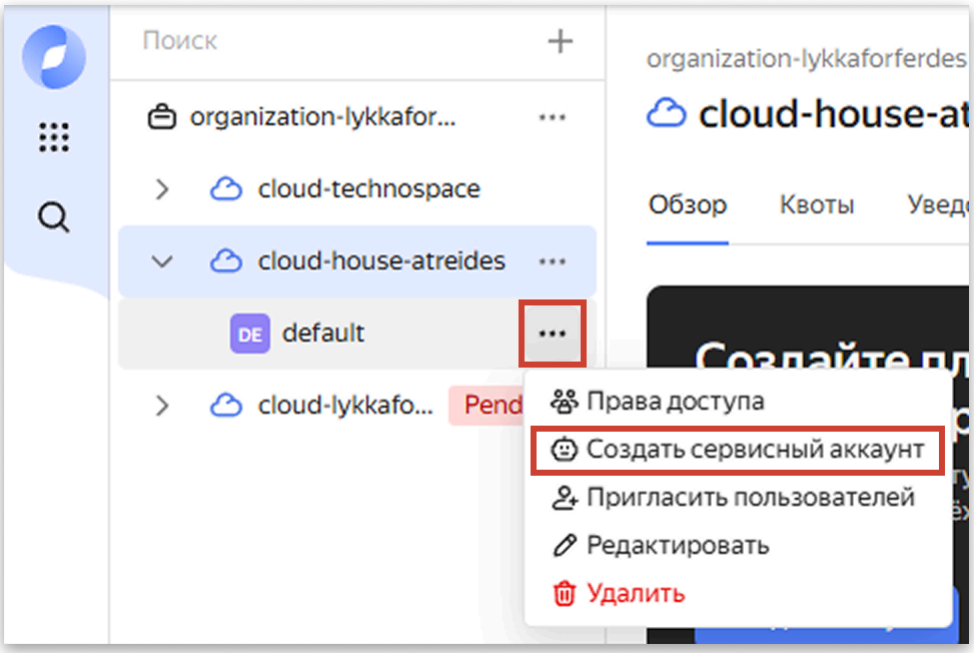
Получение API-ключа в Yandex Cloud

Создание сервисного и платежного аккаунтов Yandex Cloud

Для доступа к YandexGPT в облаке необходимо создать сервисный аккаунт с ролью `playground.user`, позволяющей использовать YandexGPT. После создания сервисного аккаунта система предложит вам создать платежный аккаунт, необходимый для оплаты YandexGPT. Вы также можете узнать в документации YandexCloud о создании сервисного и платежного аккаунтов.

Чтобы создать сервисный аккаунт:

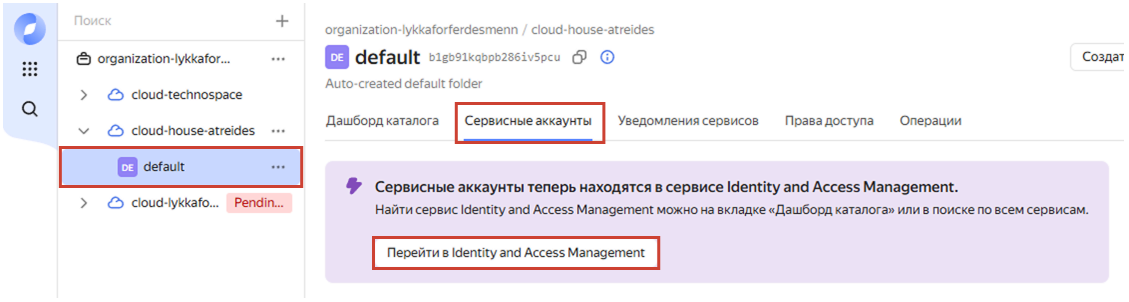
1. В консоли управления в списке слева выберите облако. Откроется выпадающий список каталогов.
2. В требуемом каталоге нажмите значок **Действия** и выберите **Создать сервисный аккаунт**. Откроется окно создания аккаунта.



3. Введите имя сервисного аккаунта.
4. Нажмите **Добавить роль**. В поле добавления роли начните вводить `playground.user` и выберите нужную роль.
5. Нажмите **Создать**. После создания сервисного аккаунта откроется окно создания платежного аккаунта. Заполните данные для оплаты (например, банковской карты), и завершите создание платежного аккаунта, следуя инструкциям на экране.

Получение API-ключа

1. В консоли управления в списке слева выберите облако. Откроется выпадающий список каталогов.
2. Выберите каталог, перейдите во вкладку **Сервисные аккаунты**, из нее перейдите в модуль **Identity and Access Management**.



   Откроется список сервисных аккаунтов.
3. Выберите сервисный аккаунт.
4. В верхней панели сервисного аккаунта нажмите **Создать новый ключ**, затем выберите **Создать API-ключ**.

5. В открывшемся окне укажите:
   - **Опционально:** описание ключа
   - Область действия ключа: yc.ai.languageModels.execute
   - **Опционально:** дату окончания действия ключа
6. Нажмите **Создать**. Откроется окно с идентификатором и значением ключа.
7. Скопируйте и сохраните значение **Ваш секретный ключ**.

Создание интеграции в Test IT

1. Используя аккаунт Test IT с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.
2. Нажмите **Создать**. Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **YandexGPT**.
4. Заполните данные для подключения:
   - Имя интеграции
   - API Key: API-ключ Yandex Cloud
5. Нажмите **Сохранить**.
   Созданная интеграция с YandexGPT отобразится на странице интеграций.
   Для создания тестов с помощью ИИ потребуется добавить интеграцию в проект.

Обновлено: 8/7/26, 11:21 AM

Интеграция с сервисами, совместимыми с OpenAI

₽ Тарифные ограничения

Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Может потребоваться добавление адресов Yandex Cloud в белый список

Пользователям Test IT Облако: Для корректной работы интеграций с сервисами, размещенными за закрытым сетевым контуром, может потребоваться добавление в белый список IP-адресов Test IT Облака.

Test IT Облако размещено на Yandex Cloud, соответственно, потребуется добавить в белый список весь пул адресов Яндекса .

Об интеграции с сервисами, совместимыми с OpenAI

Для генерации тест-кейсов на основе задач из вашего таск-трекера вы можете настроить интеграцию с ИИ-сервисами, совместимыми с OpenAI и соответствующие стандарту API OpenAI . Например, вы можете подключить Test IT к сервисам (некоторые зарубежные продукты могут быть недоступны):

- Anthropic
- Deepseek
- OpenRouter
- Qwen

В этой инструкции описывается интеграция с OpenRouter. Интеграции со всеми сервисами настраиваются аналогичным образом.

Для настройки интеграции требуются следующие данные подключаемого сервиса:

- **Base URL**: директория API-модели. Она доступна в документации ИИ-сервиса, который вы подключаете. Например, для OpenAI — **https://api.openrouter.com/v1**. Директория часто заканчивается символами **/v1**, но возможны исключения.
- **API Key**: API-ключ ИИ-сервиса, который вы подключаете.
- **Модель AI**: используемая языковая модель. Например, **qwen/qwen3-coder:free**. В зависимости от выбранной модели стоимость использования сервиса может различаться. Подробности доступны в документации ИИ-сервиса.

Создание интеграции в Test IT

Чтобы настроить интеграцию:

1. Используя аккаунт Test IT с ролью администратора, перейдите в раздел **Администрирование** и выберите вкладку **Интеграции**.
2. Нажмите **Создать**. Откроется окно **Новое подключение**.
3. В меню **Сервис для интеграции** выберите **OpenAI compatible**.
4. Заполните данные для подключения:
   - Имя интеграции
   - Base URL
   - API Key
5. Нажмите **Сохранить**.
   Созданная интеграция отобразится на странице интеграций.
   Для создания тестов с помощью ИИ потребуется добавить интеграцию в проект.

Обновлено: 8/18/26, 4:54 PM

Авторизация

> **Функциональность Test IT Про**
>
> Данная инструкция описывает авторизацию в **Test IT Про**.
> Если вы используете тарифы **Базовый** или **Стандарт**, обратитесь к соответствующей инструкции.

Для авторизации необходимо перейти на вашу страницу Test IT.
Для этого:

- Введите в адресной строке браузера адрес системы Test IT, используемый вашей компанией.

Авторизация локального пользователя

Для пользователей, добавленных в систему локально, возможен вход в систему по логину и паролю, хранящимся внутри системы Test IT.

1. Выберите способ авторизации **Локальный аккаунт**.
2. Введите имя пользователя Test IT. Если вы не знаете точное имя пользователя, обратитесь к администратору системы.
3. Введите пароль. Если вы не знаете пароль, обратитесь к администратору системы. Поставьте флажок **запомнить меня**, чтобы не вводить учетные данные при следующем открытии браузера.
4. Нажмите **Войти** или Enter .

Авторизация через AD

Для входа в систему через AD необходимо настроить данный тип подключения.

1. В поле выбора типа подключения выберите подключение по домену. Уточняйте название подключения у администратора системы. Как правило, оно совпадает с доменом компании.
2. Введите имя пользователя и пароль, к которым привязана ваша учетная запись. Имя пользователя необходимо ввести без домена. Например,

для почты `roman.sidorov@testit.software` нужно ввести имя пользователя `roman.sidorov`. Поставьте флажок **запомнить меня**, чтобы не вводить учетные данные при следующем открытии браузера.
3. Нажмите **Войти** или `Enter`.

Авторизация через OpenID Connect

Для авторизации через OpenID Connect необходимо настроить данный тип подключения.

Чтобы авторизоваться через OpenID Connect:

1. Нажмите на кнопку авторизации через сервис, к которому подключен OpenID Connect, например GitLab.
2. В открывшемся окне выполните вход в аккаунт сервиса, например GitLab.

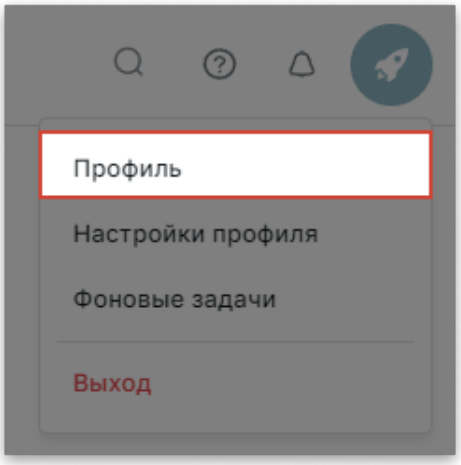
Обновлено: 10/9/26, 6:25 PM

Данные профиля пользователя

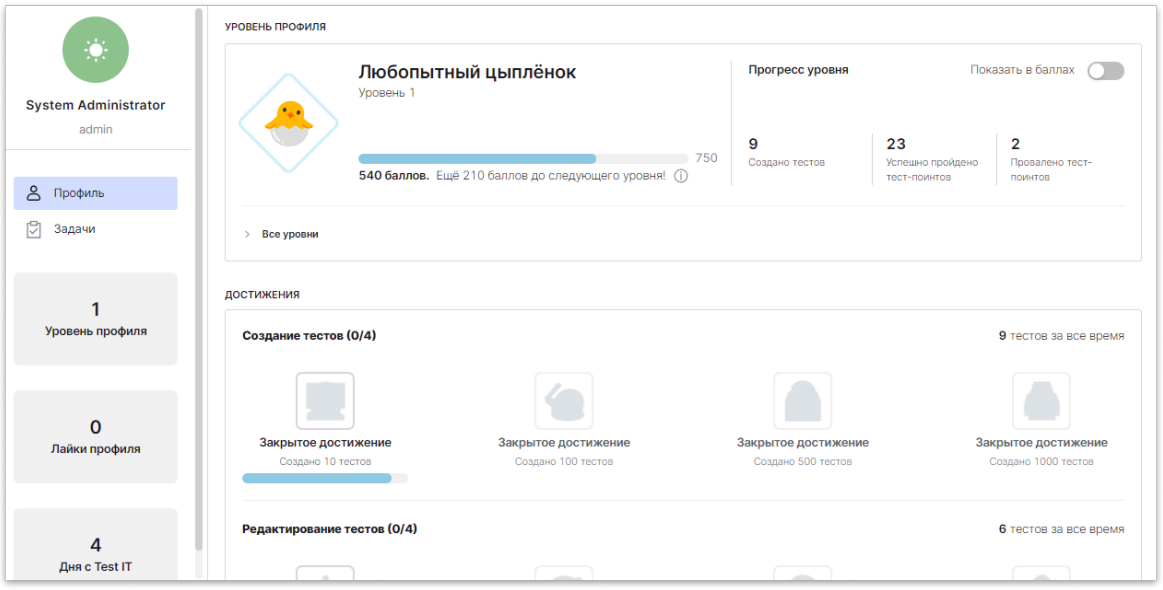
Просмотр профиля

После авторизации в системе вы можете открыть профиль пользователя. Для этого:

- Нажмите значок пользователя в правом верхнем углу экрана и выберите **Профиль** в выпадающем меню.

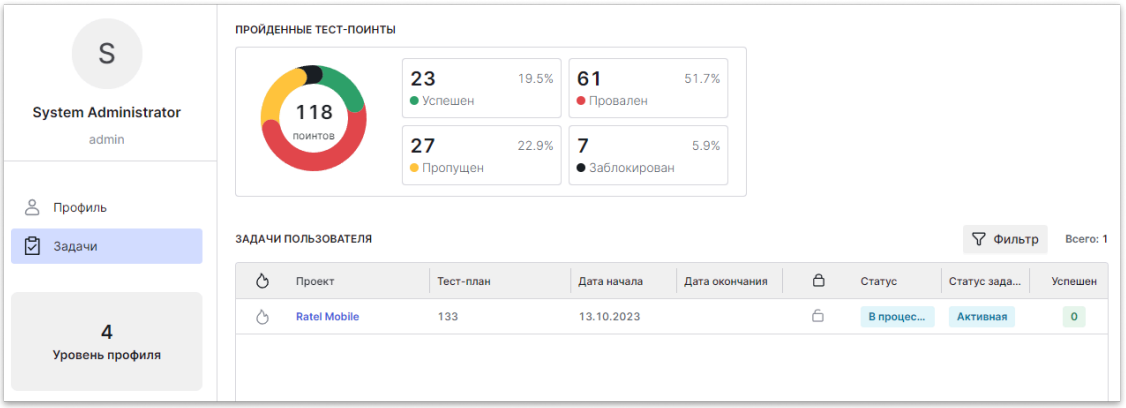


В профиле отображается информация о вашей активности в Test IT и актуальных задачах.



Вы можете:

- Просматривать количество пройденных вами тест-поинтов с разбиением по результатам (Успешен, Провален, Пропущен, Заблокирован) в абсолютном значении и процентах от общего количества (вкладка **Задачи**)



- Исследовать уровень своего профиля и отслеживать прогресс вашей работы на проектах (вкладка **Профиль**)
- Отслеживать актуальные задачи по тестированию. Вы можете отфильтровать содержимое таблицы. Если дата окончания тест-плана уже прошла, задача помечается значком огня в колонке **Срочность**.
- Редактировать профиль в настройках пользователя и генерировать приватный API-токен

Уровень профиля (геймификация)

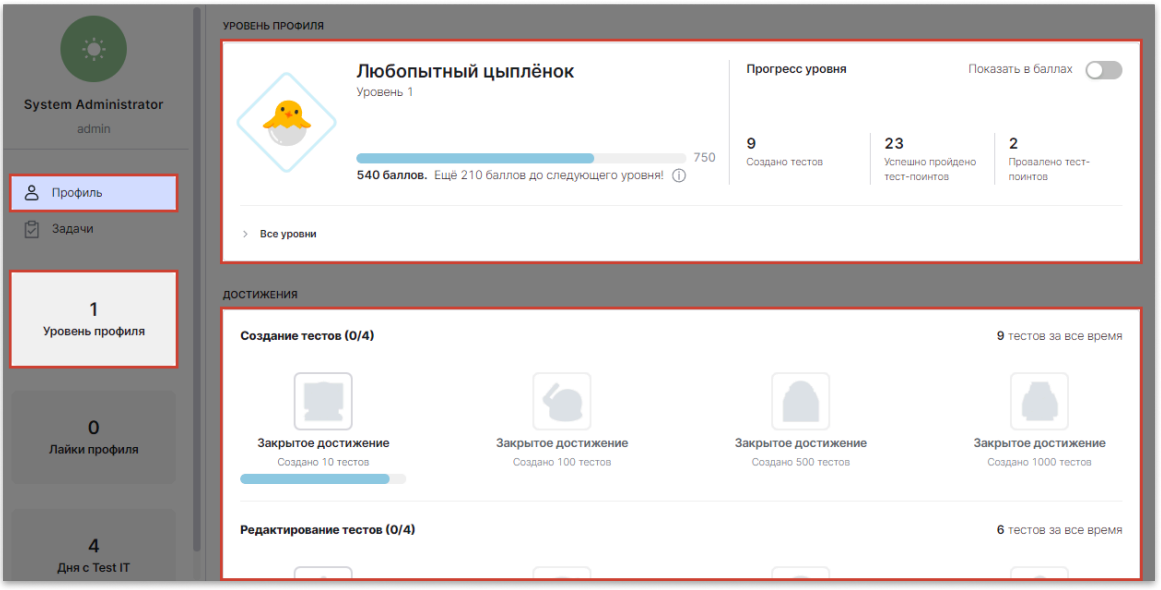
> **Важно**
>
> Вы можете отключить геймификацию для всех пользователей. Для этого требуется системная роль **Администратор.** Прогресс пользователей будет сохранен.

- **Для этого раздела мы подготовили видео:**

В Test IT доступна система геймификации, которая позволяет пользователям:

- Получать уровни за набранные баллы
- Открывать игровые достижения за выполнение действий в системе (при получении нового достижения пользователь получает оповещение в системе)

Здесь вы можете видеть свой прогресс, лайки, поставленные другими пользователями за ваши тест-кейсы и количество дней использования Test IT.



За работу на проектах пользователям начисляются баллы:

- За создание теста — 30 баллов
- За пройденный тест-поинт — 10 баллов
- За проваленный тест-поинт — 20 баллов

По мере накопления баллов уровень профиля повышается. Администратор Test IT может отключить геймификацию для всех пользователей (прогресс пользователей будет сохранен).

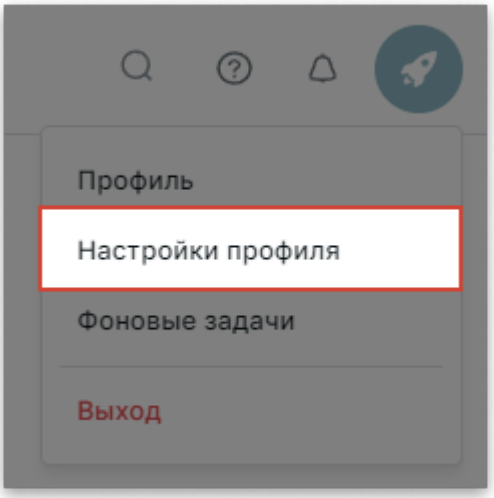
Обновлено: 8/12/26, 4:43 PM

Настройки профиля и создание API-токена

В настройках профиля Test IT вы сможете внести необходимые изменения в свой профиль, изменить язык UI, а также получить приватный токен для работы с API.

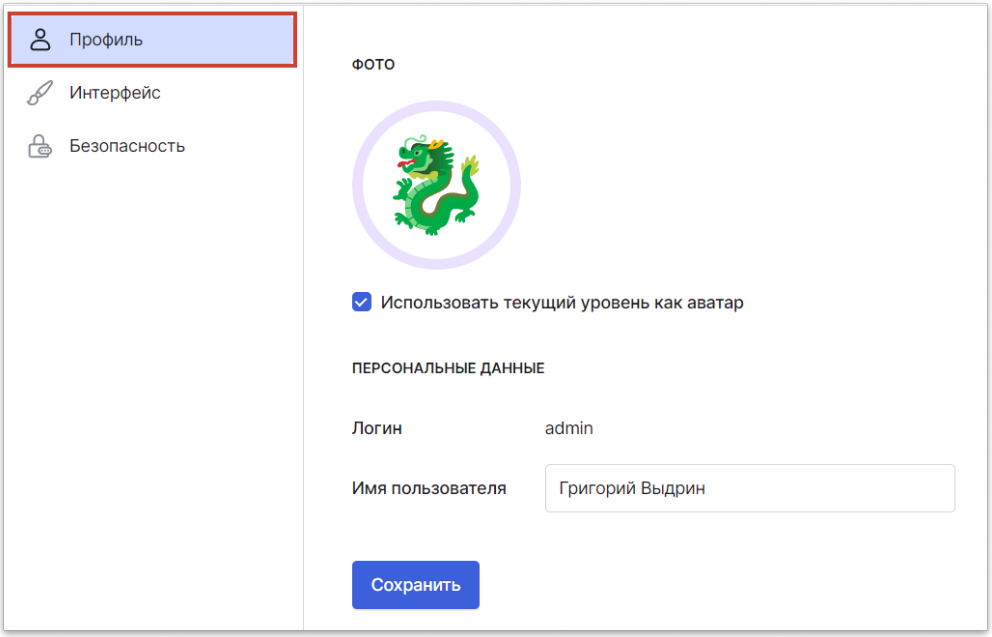
Чтобы перейти к настройкам профиля:

1. Войдите в профиль пользователя, нажав на значок в правом верхнем углу экрана.
2. Перейдите в **Настройки профиля**.



Настройки профиля

1. Перейдите на вкладку **Профиль**.



2. Внесите необходимые изменения. Вы можете редактировать аватар профиля и отображаемое имя пользователя, а также установить в качестве аватара значок вашего пользовательского уровня.
3. Нажмите **Сохранить**.

Настройки языка

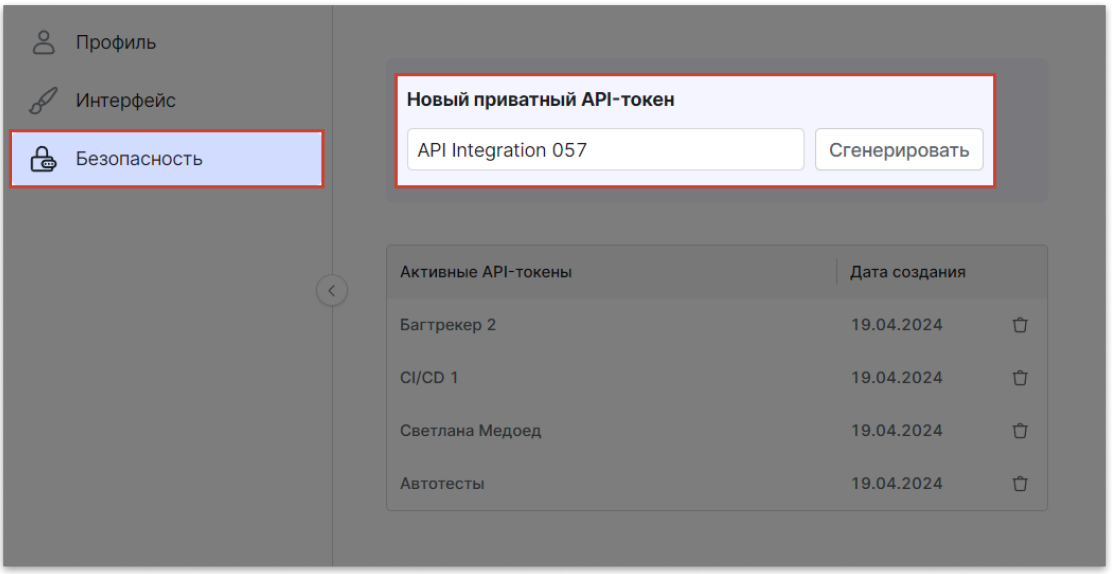
1. Перейдите на вкладку **Интерфейс**.
2. Задайте необходимые настройки языка.
3. Нажмите **Сохранить**.

Создание приватного API-токена

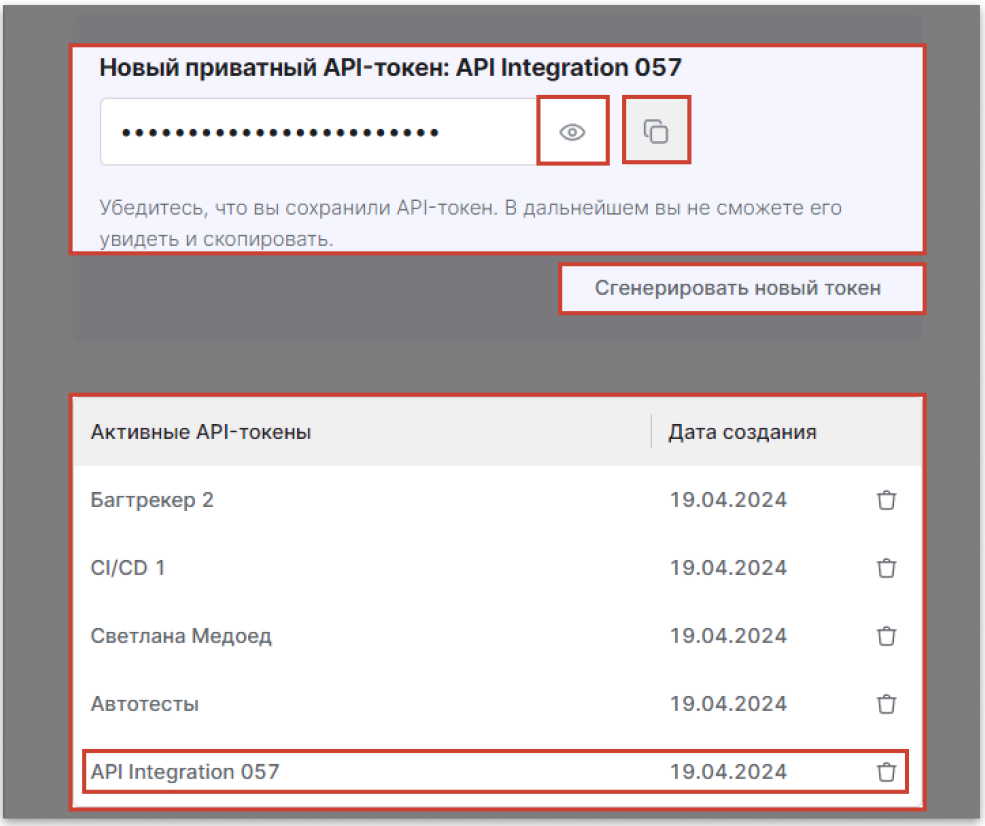
> **Внимание**
>
> - Сохраняйте каждый созданный API-токен. После закрытия или перезагрузки страницы вы не сможете его увидеть или скопировать. В списке токенов отображаются только их названия и даты создания, но не сами токены.
> - Приватные токены действуют бессрочно и утрачивают действие после удаления.

Для работы с API вам необходим приватный API-токен. Вы можете создать несколько токенов. Чтобы сгенерировать приватный токен:

1. Нажмите значок профиля в правом верхнем углу, выберите **Настройки профиля**, затем в левом навигационном меню перейдите во вкладку **Безопасность**.
2. Введите название нового токена, затем нажмите **Сгенерировать**.
   Название токена должно быть уникальным.



3. Подтвердите действие в диалоговом окне.
   Система сгенерирует новый токен.
4. Скопируйте API-токен, нажав значок копирования справа от поля **Приватный API-токен**. Для просмотра токена нажмите значок просмотра.
5. Сохраните полученный токен.
   Новый API-токен отобразится в списке токенов.
6. **Опционально:** Чтобы создать новый токен, нажмите **Сгенерировать новый токен** и выполните действия, описанные в **шагах 1-5**.



7. **Опционально:** Чтобы удалить токен, нажмите значок удаления в списке токенов и подтвердите действие в диалоговом окне.
   Удаленный токен утратит действие.

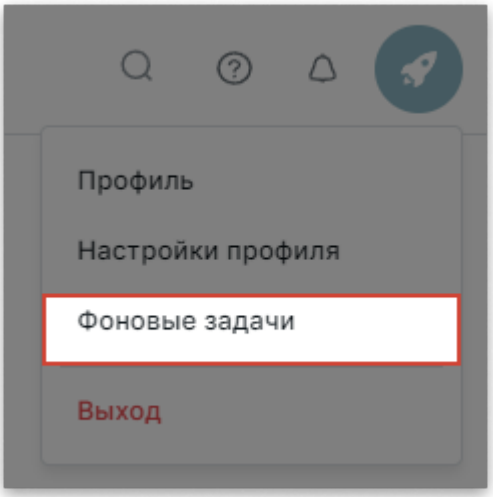
Обновлено: 4/15/26, 7:08 PM

Фоновые задачи

Фоновые задачи — это задачи по импорту и экспорту .xlsx-, .xml- и .json-файлов. Вы можете просматривать фоновые задачи, отслеживать их статус и процент выполнения, дату начала и окончания и выявленные ошибки, сортировать задачи по типу (импорт или экспорт), отменять их, а также загружать файлы фоновых задач.

Для просмотра фоновых задач:

- Нажмите значок **Действия пользователя** (изображение профиля) в правом верхнем углу страницы, затем в выпадающем меню выберите



  Фоновые задачи.
  Список фоновых задач отобразится в виде таблицы. Для задач, которые находятся в процессе выполнения, в колонке **Статус** отобразится индикатор, показывающий процент выполнения задачи.

Вам доступны следующие действия:

- Чтобы отменить фоновую задачу в статусе *В процессе*, нажмите **Отменить** в колонке **Статус**.
- Чтобы отсортировать фоновые задачи по типу, нажмите на заголовок колонки **Тип**.
- Чтобы загрузить файл задачи, в колонке **Название** нажмите название задачи.

Обновлено: 4/15/26, 7:08 PM

Использование тегов

> **Тарифные ограничения**
>
> В тарифе **Базовый** вы можете создавать теги без возможности редактирования и удаления. Библиотека тегов в **Базовый** недоступна.
> Подробнее: о тарифах Test IT .

Теги — это слова-метки, сообщающие дополнительную информацию о:

- Рабочем элементе
- Тест-плане
- Автотесте
- Прогоне

Например, вы можете указать, что элемент относится к регрессионному тестированию с помощью тега *regression*. Вы можете сортировать рабочие элементы по тегам и использовать их в Фильтре. Одни и те же теги могут быть использованы на разных проектах Test IT.

Чтобы добавить теги:

1. Откройте ручной тест, тест-план или автотест, в который хотите добавить теги.
2. На странице объекта в правой панели в поле **Теги** введите теги, которые хотите добавить. Если тег с таким названием уже существует, он отобразится в выпадающем списке. Если тег новый, нажмите **Enter**, чтобы добавить его и сохранить в системе.

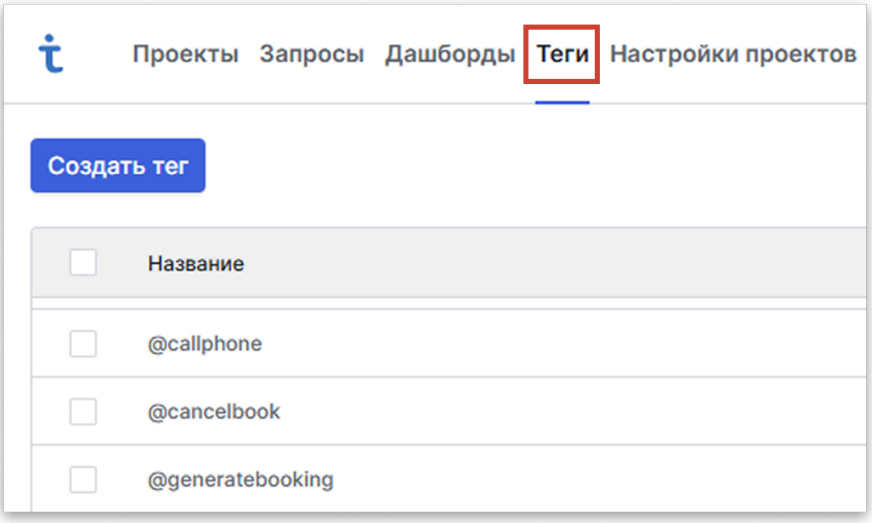


Работа с библиотекой тегов

Все теги хранятся в библиотеке тегов. Здесь вы можете просматривать, создавать, редактировать и удалять все теги.

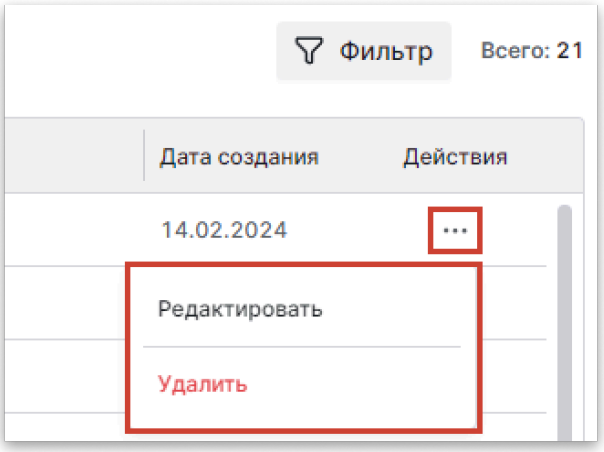
Чтобы открыть библиотеку тегов:

- В верхней навигационной панели выберите **Теги**.



Вам доступны следующие действия:

- Чтобы создать тег, нажмите **Создать тег**. В открывшемся окне введите название тега, затем нажмите **Сохранить**.
- Чтобы изменить тег, откройте меню действий справа от тега и выберите **Редактировать**.
- Чтобы удалить один или несколько тегов, отметьте их галочками в левой части списка. Чтобы удалить из библиотеки все теги, поставьте галочку слева от колонки **Название**. Нажмите значок **Корзина**, который появится рядом с кнопкой **Создать тег**. Для удаления отдельного тега воспользуйтесь меню **Действия** в правой части списка. Выберите **Удалить** и подтвердите действие в диалоговом окне.



  Удаленные теги невозможно восстановить.
- Чтобы найти тег по названию, воспользуйтесь **Фильтром** справа над списком тегов.
- Чтобы упорядочить теги по названию или дате изменения, нажмите на заголовки соответствующих колонок в списке тегов.

Доступ к функциям библиотеки тегов

Доступ к функциям библиотеки тегов ограничен системными ролями.

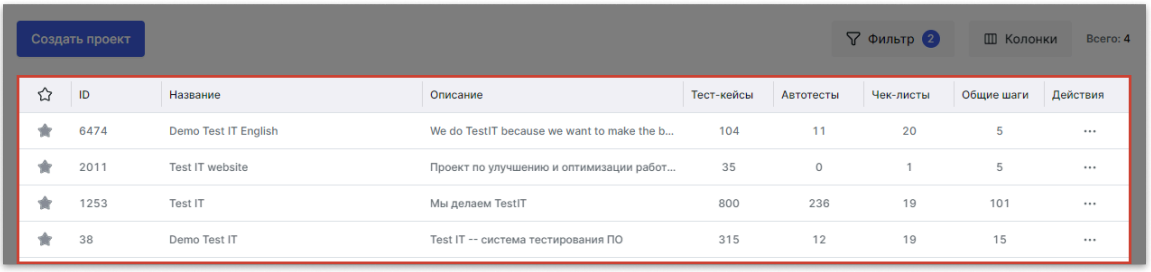
Функция	Администратор	Руководитель проектов	Пользователь
Просмотр библиотеки тегов	✓	✓	✓
Создание тегов	✓	✓	✓
Редактирование тегов	✓	✓	✗
Удаление тегов	✓	✗	✗
Массовое удаление тегов	✓	✗	✗

Обновлено: 8/12/26, 4:43 PM

Структура и компоненты проекта Test IT

Проект

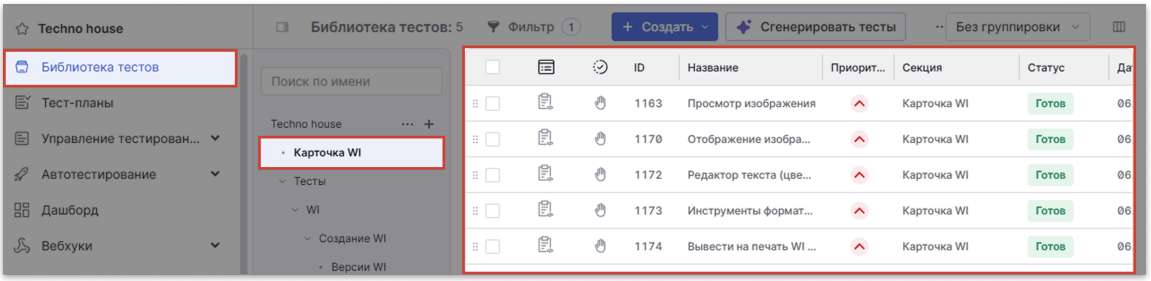
Проект — совокупность всех тестов, связанных с отдельным IT-продуктом (например приложением). Тесты, входящие в проект (библиотека тестов), могут быть ручными и автоматизированными. Проект — первый уровень группировки тестов, внутри него выделяются секции, тест-планы, тестовые наборы и прогоны автотестов. Список проектов доступен в разделе **Проекты** и открывается при входе в систему по умолчанию.



При открытии проекта отображается список рабочих элементов: тест-кейсов, чек-листов и общих шагов.

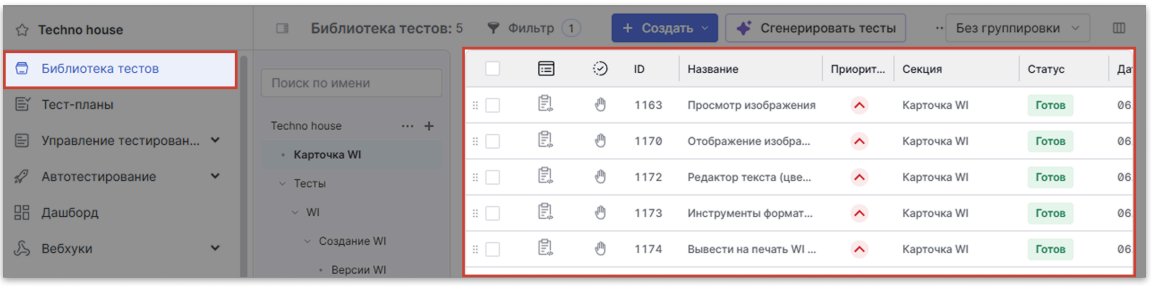
Ручное тестирование

Секция — группа тестов (рабочих элементов) внутри проекта, создаваемая для удобства навигации. Например, вы можете выделить в отдельную секцию тесты, относящиеся к модулю вашего приложения. Секции доступны в библиотеке тестов.



Тест-кейс — тестовый сценарий для определенной функциональности (запуск приложения, авторизация, и т. д.). Тест-кейс содержит шаги, которые требуется пройти для проверки, а также может содержать

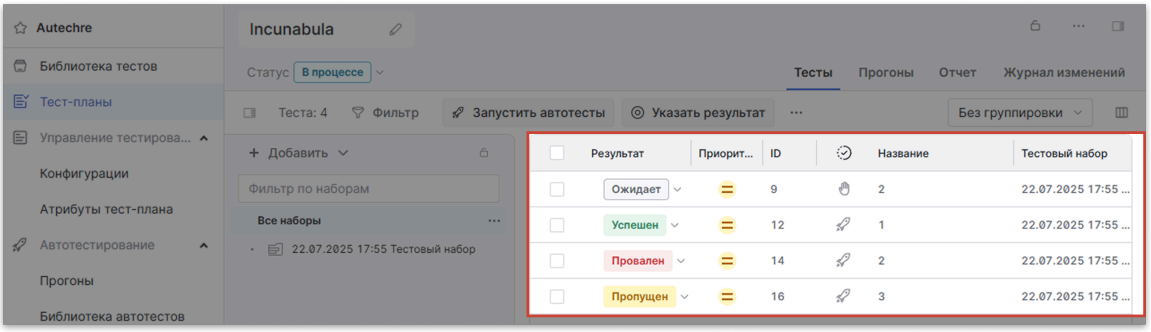
предусловия и постусловия. Тест-кейсы могут связываться с автоматизированными тестами. Тест-кейсы, входящие в проект, доступны на странице проекта во вкладке **Библиотека тестов**.



Чек-лист — список проверок, позволяющий добавить шаги теста, но не описывать предусловия и постусловия и ожидаемые результаты. Чек-лист можно использовать на раннем этапе разработки. Вы можете конвертировать чек-лист в тест-кейс. Чек-листы, входящие в проект, доступны на странице проекта во вкладке **Библиотека тестов**.

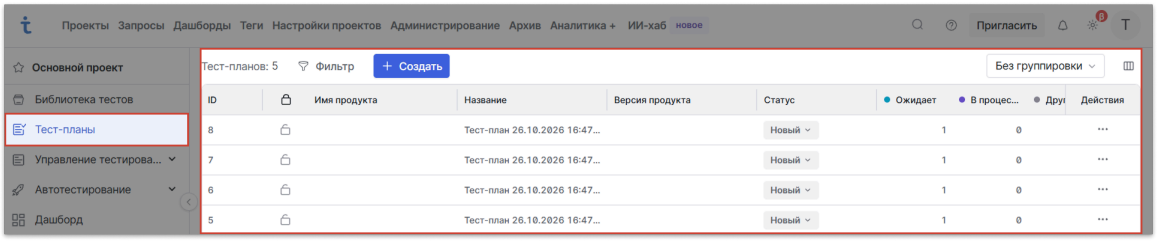
Общий шаг — шаг, который можно использовать в нескольких тест-кейсах, что позволяет сэкономить время на создание тестовых сценариев с типичными шагами. Общие шаги, входящие в проект, доступны на странице проекта во вкладке **Библиотека тестов**.

Тест-поинт — ручной тест, проводимый по сценарию, описанному в тест-кейсе с определенными значениями параметров или конфигурациями. Тест-поинт отражает результат тест-кейса и его отдельных шагов. Например, после завершения тест-кейса “Запуск приложения”, выполненного в двух различных браузерах, создаются два тест-поинта, которые могут показать одинаковые или различные результаты. Тест-поинты могут связываться с автоматизированными тестами. Список тест-поинтов доступен на странице тест-плана.

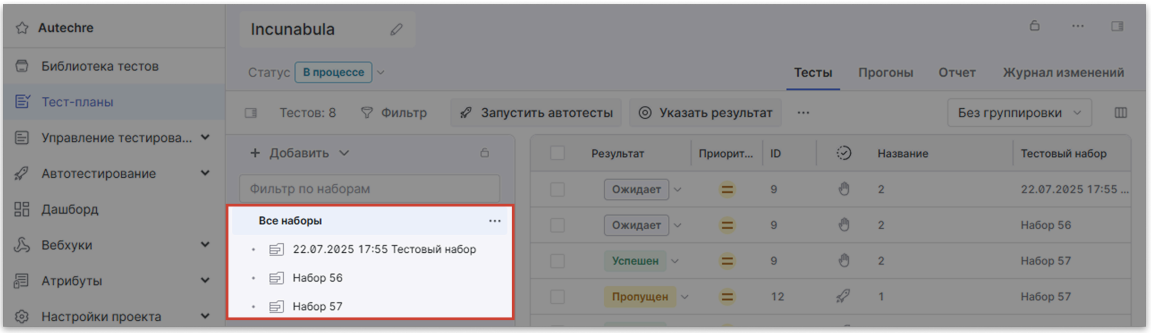


Тест-поинты, совершенные в рамках одного тест-кейса, доступны на странице тест-кейса во вкладке **История результатов**.

Тест-план — группа ручных тестов проекта, на которую назначаются тестировщики и в которой указываются результаты тестов и добавляются атрибуты. В тест-план также могут входить автотесты, привязанные к ручным тестам. Список тест-планов доступен в разделе **Тест-планы**.



Тестовый набор — часть тест-плана, как правило включающая совокупность тестов (тест-кейсов и тест-поинтов) одного компонента системы. В системе различаются пользовательский набор, набор из секций библиотеки и динамический набор. В тестовых наборах сохраняется группировка по секциям. Список тестовых наборов доступен на странице тест-плана.



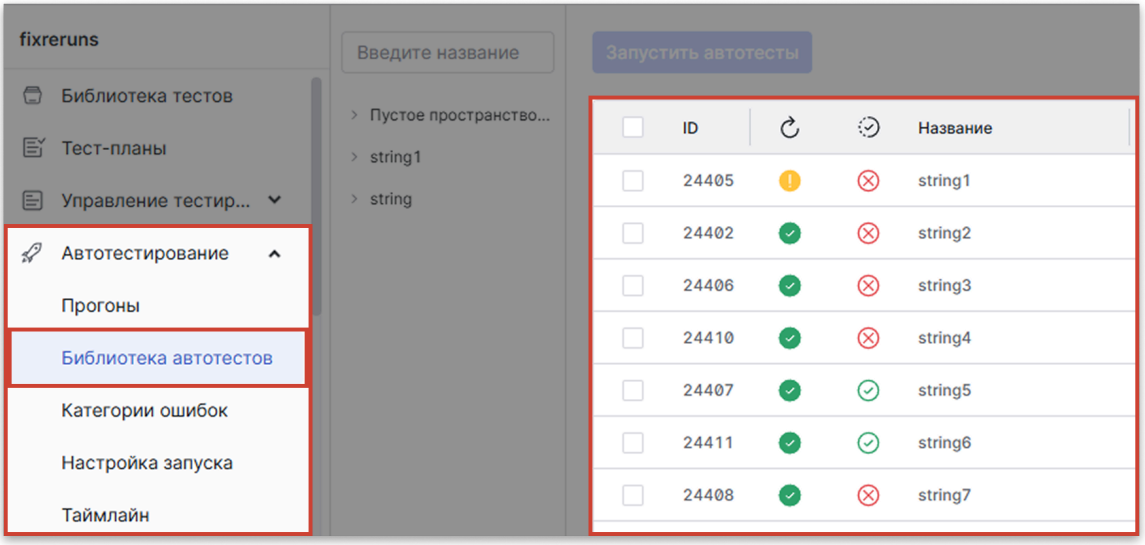
Практический пример

Распространенная практика — создание набора, в котором постусловие предыдущего тест-кейса является предусловием последующего. Например, тестовый набор может включать в себя последовательность тестов: “Запуск приложения”, “Авторизация”, “Внесение изменений” и “Сохранение изменений“.

Автоматизированные тесты

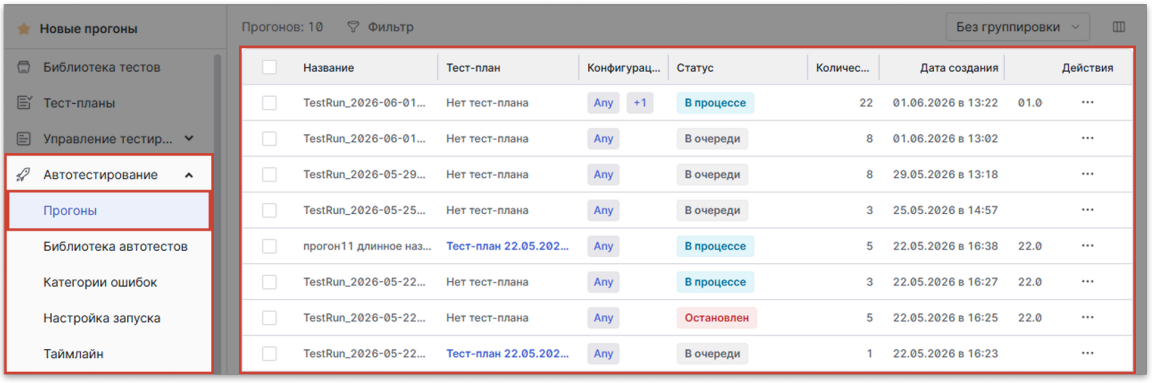
Автотест — автоматизированный тест, который создается и проводится в сторонних системах. Аналогично тест-кейсу, автотест содержит сценарий, но не для ручного, а для автоматизированного тестирования. Test IT позволяет запускать автотесты из UI или клиентской библиотеки, помечать их как стабильные или нестабильные, отслеживать их прогоны (тест-

раны), историю результатов, связывать с ручными тестами и выполнять другие задачи. Список автотестов доступен в разделе **Автотестирование → Библиотека автотестов**.



Автотесты, связанные с ручными тестами, доступны на странице ручного теста во вкладке **Связанные автотесты**.

Прогон (тест-ран) — запуск одного или нескольких автотестов при определенной конфигурации с указанием результата. При прогоне одного автотеста на двух различных конфигурациях появляется два результата, которые могут совпадать или различаться. Список всех прогонов, входящих в проект, доступен в разделе **Автотестирование → **Прогоны**.



Список прогонов всех автотестов, входящих в тест-план, доступны на странице тест-плана во вкладке **Прогоны**. Результаты отдельного прогона доступны на странице прогона во вкладке **Результаты тестов**. Результаты прогонов отдельного автотеста доступны на странице автотеста во вкладке **История результатов**. Прогоны, связанные с отдельным тест-поинтом, доступны на странице тест-поинта в разделе **Тест-планы**. Прогоны автотестов, связанных с отдельным тест-кейсом, доступны на странице тест-кейса (вкладка **Связанные автотесты)**.

Результаты тестов

Результат теста — отметка об итоге проведения тест-поинта или прогона автотеста. Тест может иметь следующий результат: *Ожидает*, *Успешен*, *Заблокирован*, *Пропущен*, *Провален*. Результаты тест-поинтов указываются вручную на странице тест-поинта во вкладке Выполнение раздела Тест-планы с возможностью указать результат для каждого отдельного шага. Результаты прогонов автотестов проставляются автоматически.

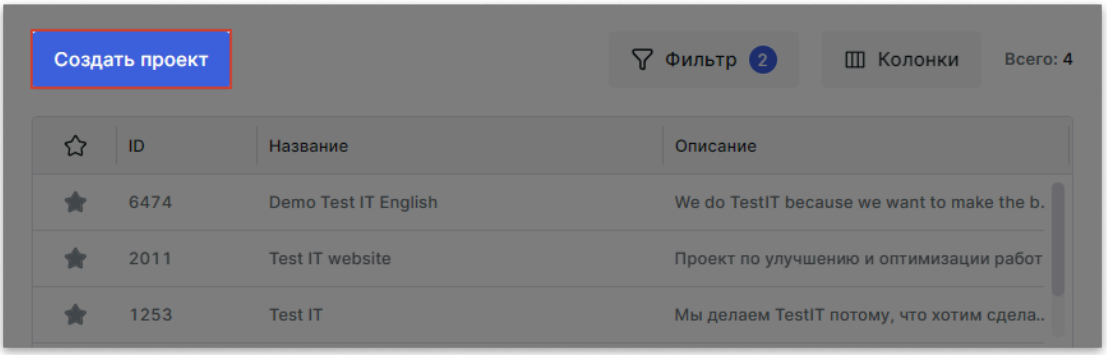
Обновлено: 7/24/26, 4:01 PM

Создание и редактирование проекта

> **₽ Тарифные ограничения**
>
> В тарифе **Базовый** вы можете вести 1 рабочий проект (демо-проект не учитывается). Первый проект в рабочем пространстве Test IT создается автоматически.
> Подробнее: о тарифах Test IT .

Создание проекта

1. Используя аккаунт с системной ролью **Администратор** или **Руководитель проектов**, перейдите на вкладку **Проекты**.
2. Нажмите **Создать проект**.



3. В открывшемся окне введите название нового проекта и, при необходимости, добавьте его описание.
4. Нажмите **Сохранить**.

После сохранения проекта вы будете перенаправлены в созданный проект в раздел **Библиотека тестов**.

Чтобы перейти к списку проектов, перейдите на вкладку **Проекты**. Список всех проектов в системе отображается в виде таблицы.

После создания проекта, перейдите в настройки проекта, чтобы добавить в него пользователей и завершить настройку.

Демонстрационный проект в тарифах Базовый и Стандарт

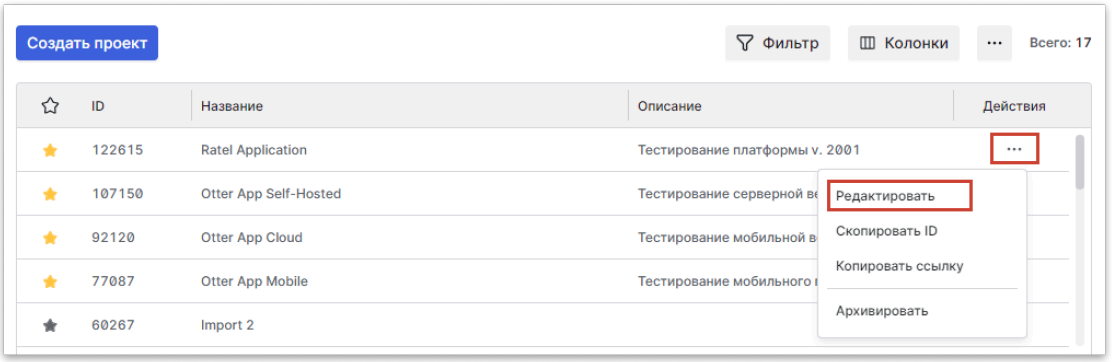
Если вы начинающий пользователь и работаете в тарифах **Базовый** или **Стандарт**, вы можете создать демо-проект с готовыми примерами тестов, тест-планов и дашбордов, чтобы быстро познакомиться с возможностями системы. Для этого:

- Нажмите **Создать проект**, затем в нижней части окна создания проекта нажмите **Создать демо-проект** и продолжите создания проекта. Созданный вами проект будет содержать предварительно загруженные тесты, тест-план и дашборды.

Редактирование проекта

Вы можете отредактировать название и описание проекта. Чтобы отредактировать проект:

1. В списке проектов откройте выпадающее меню в колонке **Действия** для требуемого проекта и нажмите **Редактировать**.



2. Внесите изменения.
3. Нажмите **Сохранить**.

Копирование идентификатора (ID) и другие действия с проектом

Идентификатор (ID) проекта Test IT требуется для работы с некоторыми интеграциями и автоматизации тестирования. Чтобы скопировать идентификатор проекта:

- В списке проектов откройте выпадающее меню в колонке **Действия** для требуемого проекта и нажмите **Скопировать ID**.

Также в меню **Действия** вы можете:

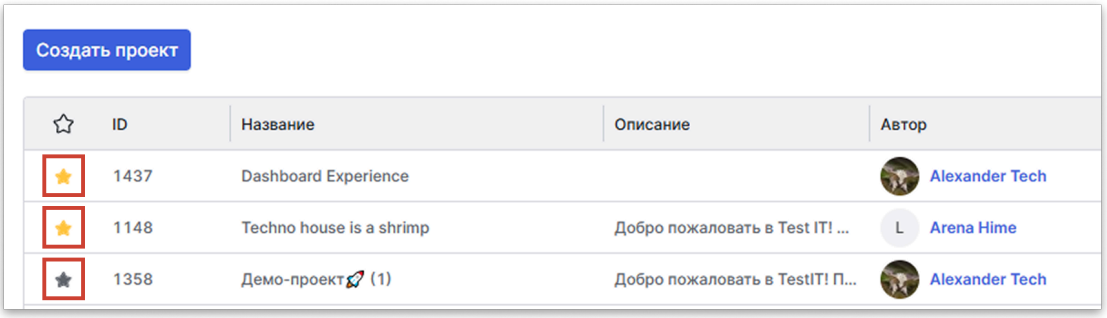
- Отредактировать проект
- Скопировать ссылку на проект
- Архивировать проект

Работа с избранными проектами

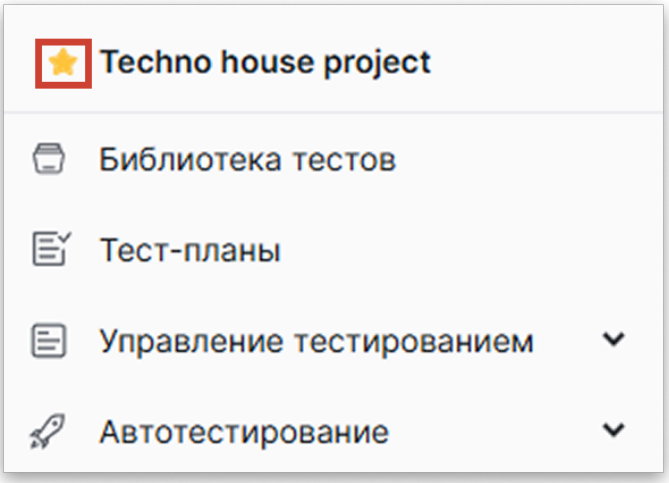
Вы можете отметить ваши приоритетные проекты как избранные. По умолчанию избранные проекты отображаются в верхней части списка. Каждый созданный проект отмечается как избранный. Чтобы отсортировать содержимое таблицы, вы можете воспользоваться фильтром.

Отметить проекты как избранные или снять отметку избранности можно:

- Из списка проектов, нажав значок **Звезда** в колонке избранных проектов.



- В открытом проекте, нажав значок **Звезда** слева от названия проекта.



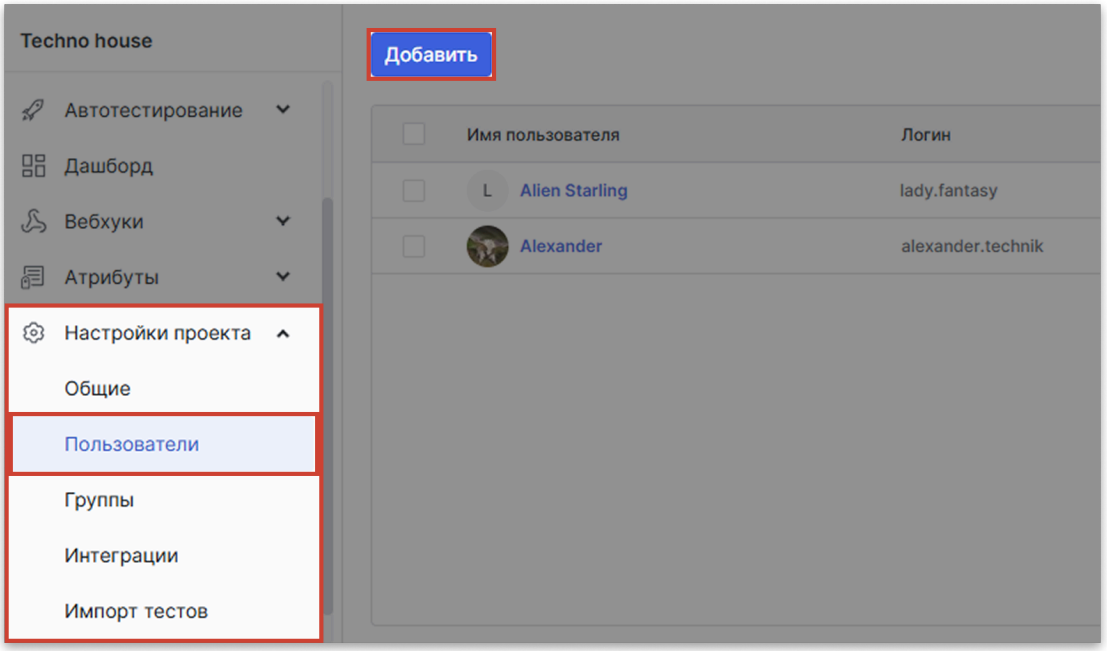
  Также вы можете перейти к другим проектам, наведя курсор на название открытого проекта.

Обновлено: 8/17/26, 2:21 PM

Добавление пользователей и групп в проект

Добавление пользователей

1. Откройте проект.
2. В меню **Настройки проекта** выберите раздел **Пользователи**.
3. Нажмите **Добавить**.



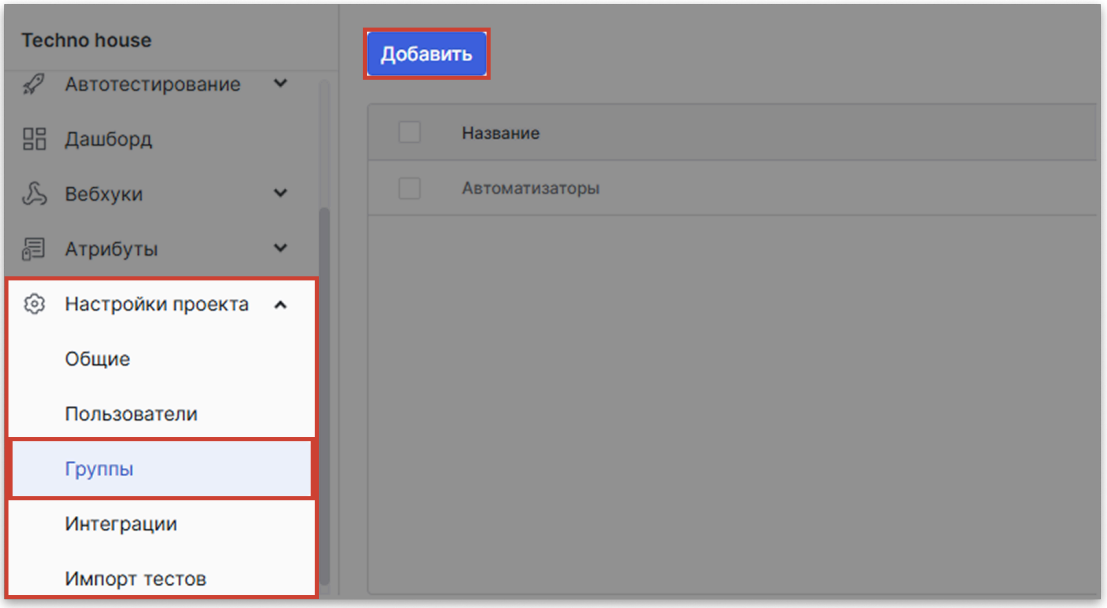
4. Введите имя пользователя, которого хотите добавить, или выберите его из выпадающего списка.
5. Выберите проектные роли для нового пользователя.
6. Нажмите **Сохранить**.

Добавленный пользователь отобразится в таблице в разделе **Пользователи**. Воспользуйтесь фильтром, чтобы отсортировать содержимое таблицы.

Добавление групп

Вы можете добавлять в проекты группы пользователей, созданные Администратором.

1. Откройте проект.
2. В меню **Настройки проекта** выберите раздел **Группы**.
3. Нажмите **Добавить**.



4. Введите название группы пользователей, которую хотите добавить, или выберите ее из выпадающего списка.
5. Выберите проектную роль. Проектная роль будет применена ко всем пользователям из группы.
6. Нажмите **Сохранить**.

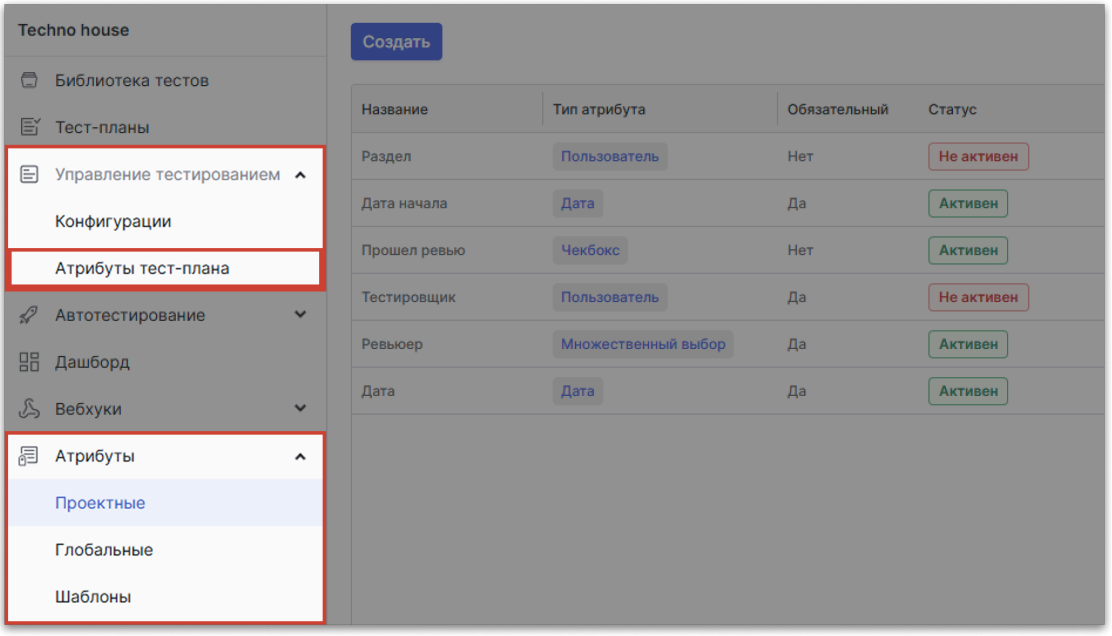
Добавленная группа отображается в таблице в разделе **Группы**. Воспользуйтесь фильтром, чтобы отсортировать содержимое таблицы.

Обновлено: 4/15/26, 7:08 PM

Настройка атрибутов в проекте

Настраивайте поля рабочих элементов и тест-планов с помощью пользовательских атрибутов. Вы можете добавить в проект 3 вида атрибутов и шаблоны глобальных атрибутов:

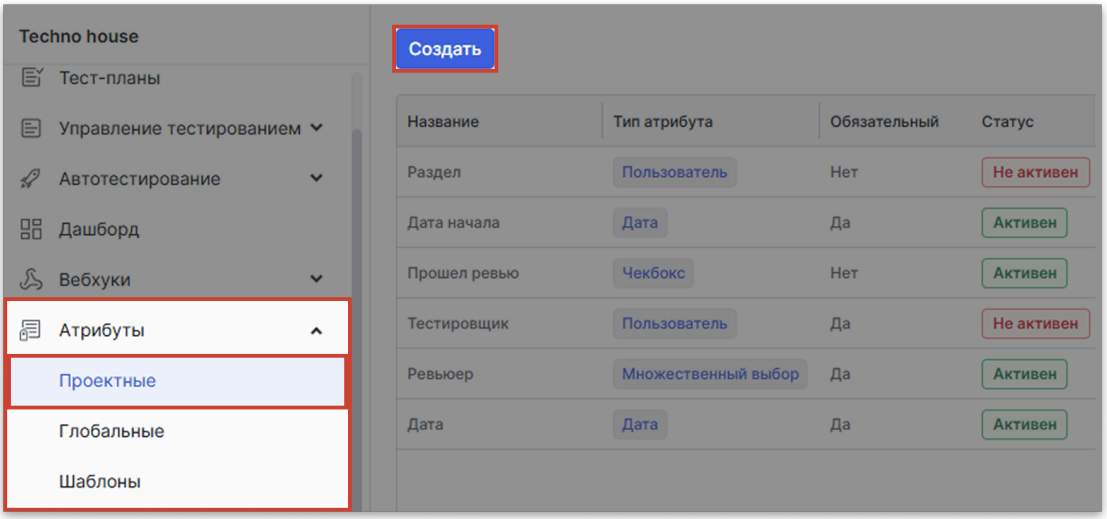
- Атрибуты тест-плана — глобальные атрибуты, применяемые к тест-планам. Добавляют соответствующие им поля во все тест-планы проекта.
- Проектные атрибуты — создаются и используются только внутри вашего проекта. Добавляют соответствующие им поля во все рабочие элементы проекта.
- Глобальные атрибуты — создаются администратором системы и доступны во всех проектах в системе. Добавляют соответствующие им поля во все рабочие элементы проекта.
- Шаблоны атрибутов — создаются администратором системы с помощью объединения нескольких глобальных атрибутов. Добавляют соответствующие им поля во все рабочие элементы проекта.



Создание проектных атрибутов

1. Откройте проект.
2. Перейдите в раздел **Атрибуты → Проектные атрибуты**.

3. Нажмите **Создать**.



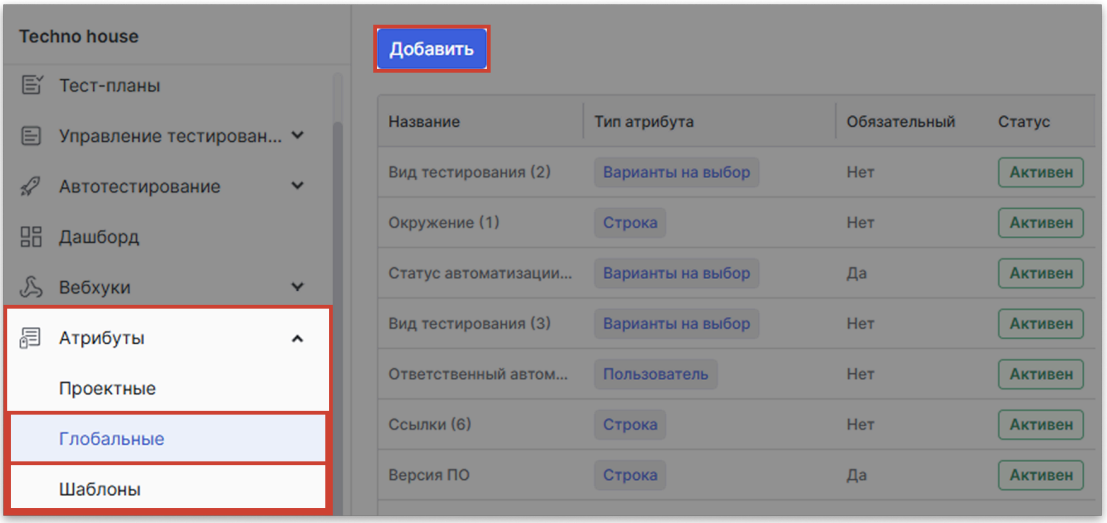
4. Введите название атрибута. Если вы хотите сделать атрибут обязательным для всех создаваемых рабочих элементов, выберите **Обязательный атрибут**.
5. Выберите тип атрибута. Вы можете создать проектные атрибуты следующих типов:
   - **Дата** — поддерживает ввод дат и может быть использован для назначения дат начала тестирования или дедлайнов
   - **Варианты на выбор** — поддерживает ввод нескольких опций. Пользователь может выбрать только один из вариантов.
   - **Множественный выбор** — поддерживает ввод нескольких опций. Пользователь может выбрать несколько вариантов.
   - **Строка** — поддерживает ввод и просмотр многострочного текста
   - **Пользователь** — выбирается из команды проекта. Этот тип атрибута можно использовать для назначения исполнителя, ревьюера или наблюдателя.
6. Для атрибутов типа **Варианты на выбор** и **Множественный выбор введите** возможные значения.
7. **Опционально:** Чтобы сделать атрибут неактивным, отключите опцию **Активный**. Неактивные атрибуты отображаются в настройках проекта, но не могут применяться к рабочим элементам.
8. Нажмите **Сохранить**.
9. **Опционально:** Чтобы отредактировать или удалить атрибут, используйте меню **Действия** справа от строки атрибута.

Добавление атрибутов тест-плана

Вы можете добавлять к тест-планам проекта глобальные атрибуты. Инструкцию смотрите в разделе тест-планов.

Добавление глобальных атрибутов и шаблонов атрибутов

1. Откройте проект.
2. Перейдите в раздел **Атрибуты**.
3. Выберите **Глобальные** или **Шаблоны** атрибутов.
4. Нажмите **Добавить**.



5. Отметьте шаблоны, которые хотите добавить. При необходимости воспользуйтесь поиском.
6. Нажмите **Добавить**.
7. **Опционально:** Чтобы отредактировать или удалить атрибут, используйте меню **Действия** справа от строки атрибута.

Смотрите также: Настройка атрибутов.

Обновлено: 4/22/26, 2:57 PM

Настройка статусов прохождения тестов

₽ Тарифные ограничения

Опция входит в платные тарифы. При возврате к тарифу **Базовый** пользовательские статусы и их наборы, созданные ранее, будут доступны в ваших проектах. Вы можете их безвозвратно удалить, но не сможете изменять или создавать новые. Подробнее: о тарифах Test IT

- Статусы прохождения тестов в Test IT
- Добавление набора статусов тестов
- Действия с пользовательским набором статусов

Статусы прохождения тестов в Test IT

Для указания результата тест-поинтов и автотестов в Test IT имеются предустановленные базовые (системные) статусы: *Ожидает*, *Успешен*, *Пропущен*, *Провален* и *Заблокирован*.

Чтобы отслеживать выполнение плана с большей точностью или с учетом потребностей вашей команды, вы можете настроить пользовательские статусы и изменять их.

Примеры из практики

1. Команде требуется дифференцировать проваленные результаты тестов, чтобы оперативно выявлять причину ошибки и передавать задачу в соответствующий отдел. Статус *Провален* не подходит, так как сообщает только общую информацию о результате.
   Для этого в Test IT создаются пользовательские статусы:
   - *Ошибка: Функциональность* — если тестируемая функциональность не соответствует требованиям
   - *Ошибка: Инфраструктура* — если причина ошибки не в функциональности, а в инфраструктуре или окружении
2. Для испытания высокоточного оборудования компания контролирует каждый шаг тестирования. Статуса *В процессе* для этого недостаточно, т.к. он дает только общую информацию. Чтобы обеспечить необходимый контроль, статус *В процессе* дифференцируется: *Шаг 1 в процессе*, *Шаг 2 в процессе* и т.д.

Статусы тестов располагаются в группах (например **Успешно** или **Провалено**), а группы объединены в наборы.

Наборы статусов

Статусы объединяются в наборы. **Набор статусов** содержит полный список статусов, применяемых в проекте. Одновременно в проекте может применяться только один набор статусов.

Чтобы использовать набор статусов в проектах, требуется выполнить одно из действий:

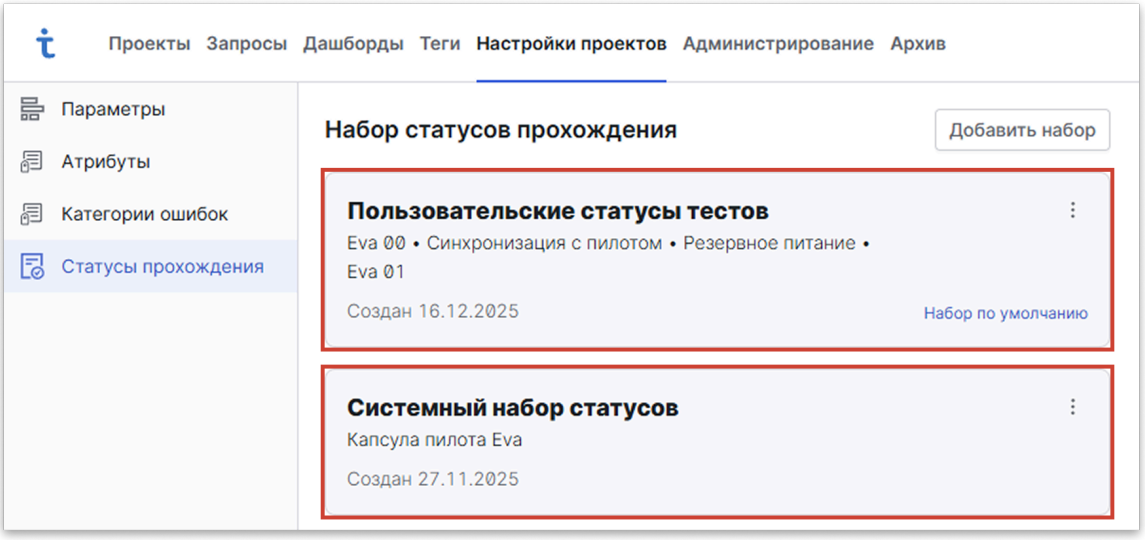
- Указать нужные проекты в настройках набора
- Сделать его **набором по умолчанию** — тогда он будет применяться во всех проектах.

Что произойдет со статусами проекта, когда набор по умолчанию изменится?

Если набор по умолчанию был переназначен, новый набор будет применяться только к проектам, созданным после переназначения. В проектах, созданных ранее, сохранится старый набор.

В Test IT есть два типа наборов статусов:

- **Системный набор** — предустановлен в Test IT, неизменяем, изначально является набором по умолчанию (назначение по умолчанию можно отменить или вернуть)
- **Пользовательский набор** — может быть создан, изменен, наполнен статусами тестов, назначен по умолчанию

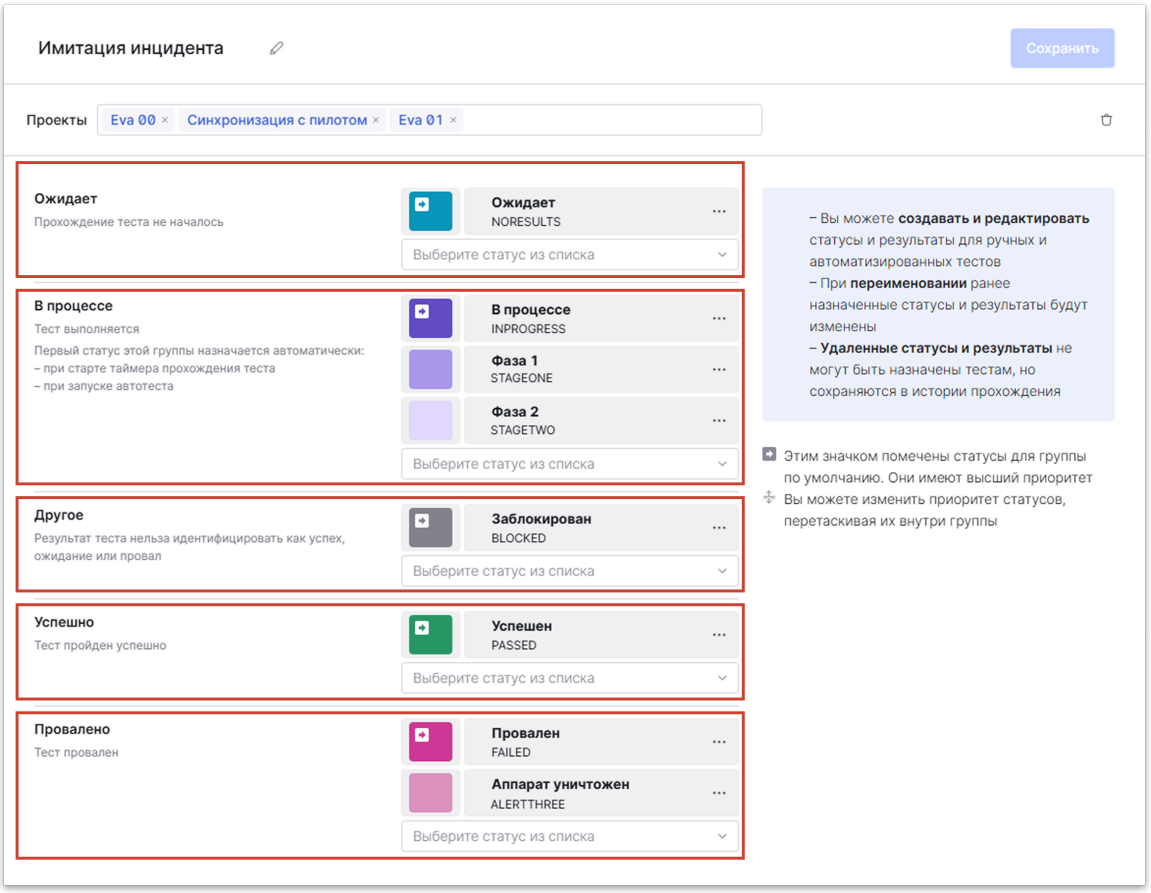


Для создания пользовательских статусов нужно создать пользовательский набор. Изначально пользовательский набор имеет те же статусы, что и системный. Статусы пользовательского набора можно изменить и дополнить.

Группы статусов

Наборы статусов состоят из **групп**, в каждую из которых входят статусы тестов.

В Test IT доступны группы статусов: **Ожидает**, **В процессе**, **Успешно**, **Провалено** и **Другое**.



Группы нельзя переименовывать, удалять или создавать. Внутри групп пользовательского набора можно создавать статусы.

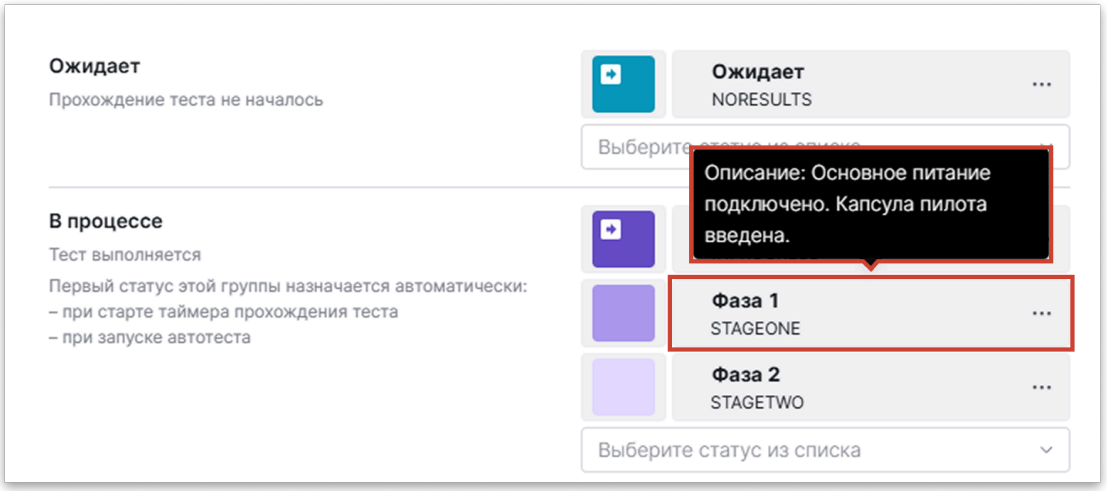
Статусы прохождения

Статусы располагаются внутри групп. Внутри каждой группы имеется **статус по умолчанию** — его нельзя удалить, но можно переименовать.

Статус содержит информацию:

- Название
- Код статуса — уникальное значение, позволяющее идентифицировать статус и избежать повторов.

- Назначение — описание статуса



  Статусы расположены в группе по приоритету. **Приоритет статуса** — это порядок, в котором статусы будут применяться в случае конфликтов между наборами. Приоритет статусов можно изменять, перетаскивая их внутри группы.

Особенности работы с пользовательскими статусами

- Статусы одного набора могут использоваться в других наборах.
- При переименовании ранее назначенные статусы также изменяются в истории прохождения теста. Удаленные статусы не могут быть назначены тестам, но сохраняются в истории прохождения.
- При работе с автотестами должны использоваться коды статусов из набора, используемого в данном проекте.
- Пользовательские статусы отображаются в дашбордах и группируются по категориям.

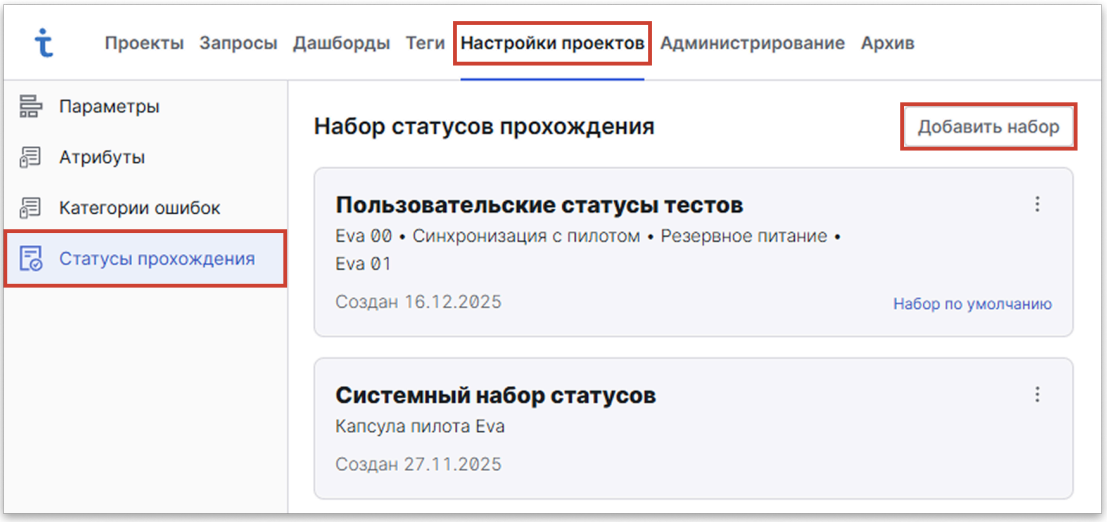
Добавление набора статусов тестов

> **Создайте статусы перед началом тестирования**
>
> Для корректной работы мы рекомендуем создавать пользовательские статусы перед выполнением тест-плана и запуска автотестов.

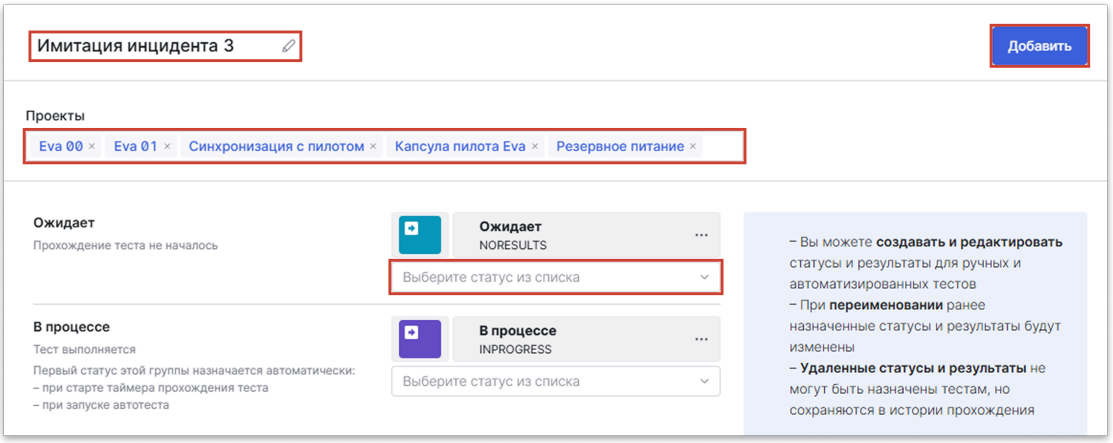
Вы можете создавать, изменять и удалять пользовательские наборы статусов прохождения тестов.

Чтобы добавить набор пользовательских статусов:

1. В верхней навигационной панели откройте **Настройки проектов → Статусы прохождения**.
2. Нажмите **Добавить набор**.

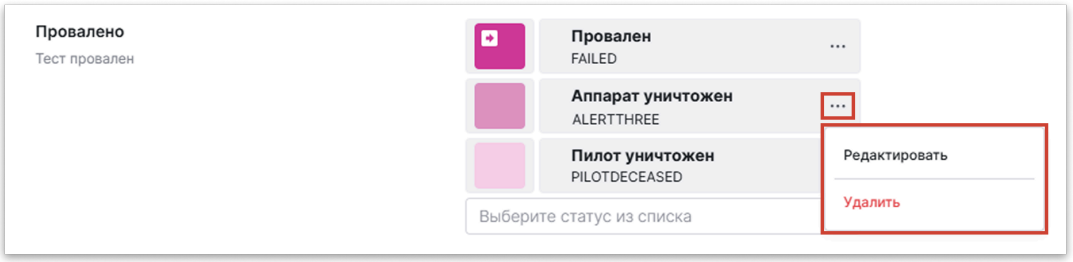


Откроется окно создания набора.



3. **Опционально:** В поле **Проекты** выберите проекты, в которых будет использоваться набор. Если вы оставите поле пустым и не установите набор как набор по умолчанию, он не будет применен к проектам.
4. В группе статусов нажмите в поле выбора статуса.
   Откроется список статусов, созданных ранее, с возможностью создать новый статус.
5. Чтобы **создать новый статус**:
   а. Нажмите **Создать новый статус**. Откроется окно **Новый статус**.
   б. Введите название, код и если нужно — описание статуса, затем нажмите **Добавить**.
6. Чтобы **добавить существующий статус**, нажмите на него в списке статусов.

- **Опционально:** Чтобы отредактировать или удалить статус, откройте меню **Действия** справа от названия статуса и выберите нужную опцию.



7. **Опционально:** Добавьте в набор другие статусы, повторяя последовательности шагов **4** → **5** либо **4** → **6** в требуемых группах.
8. Когда набор статусов готов, в правом верхнем углу окна нажмите **Добавить**.
   Набор отобразится в списке. Пользовательские статусы будут доступны в выбранных проектах для указания результатов тестов.

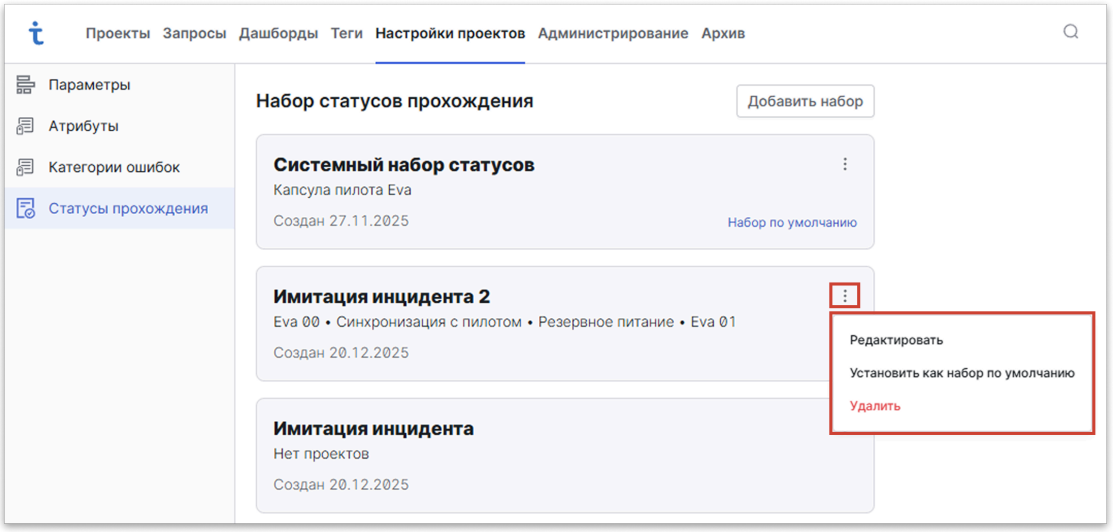
Действия с пользовательским набором статусов

Вы можете:

- Изменить название набора и входящие в него статусы
- Назначить его набором по умолчанию, чтобы он применялся ко всем проектам
- Безвозвратно удалить

Для этого:

1. Справа от названия набора откройте меню **Действия** и выберите нужную опцию.



2. Выполните требуемое действие.

Обновлено: 5/4/26, 10:18 AM

Настройка вебхуков

Тарифные ограничения

В тарифе **Базовый** доступно не более 10 стандартных вебхуков.
Подробнее: о тарифах Test IT .

- **Для этого раздела мы подготовили видео:**

Вебхуки используются для передачи данных о действии или событии между двумя сервисами. Test IT использует вебхуки для интеграций с мессенджерами и CI/CD-системами. Для настройки вебхуков вам необходимо обладать ролью с правами доступа на работу с вебхуками.

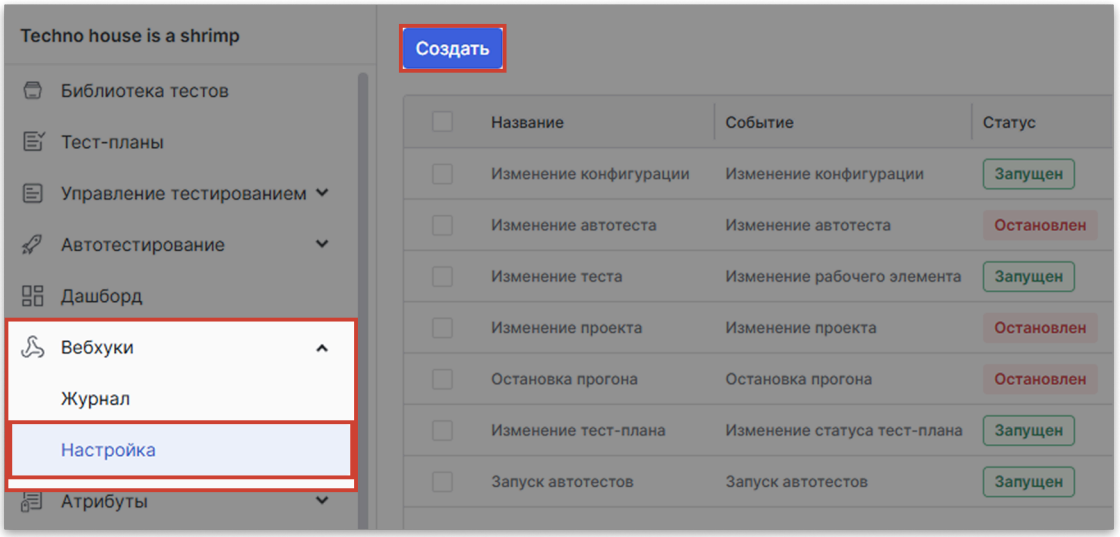
Поддерживаются только HTTP- и HTTPS-запросы

При работе с вебхуками в Test IT поддерживаются только HTTP- и HTTPS-запросы.

Создание вебхуков

1. Откройте проект.
2. В навигационном меню слева нажмите **Вебхуки**.

3. Нажмите **Создать**.



4. Укажите основные параметры нового вебхука:
   - **Событие** — действие в системе, которое будет инициировать отправку вебхука (например **Запуск автотестов** и **Изменение статуса тест-плана**).
   - **Название** — имя вебхука, которое будет отображаться в списке вебхуков.
   - **Описание** — текстовое описание вебхука, для указания краткого комментария. Данное поле необязательно для заполнения.
5. Укажите параметры запроса HTTP:
   - **URL** — адрес внешней системы, на который будет отправлен HTTP-запрос.
   - **Тип запроса** — тип запроса, отправляемого сервисом — **POST**, **PUT** или **DELETE**.
6. Укажите параметры, передаваемые в URL и заголовках запроса HTTP. В значениях параметров URL и заголовка используются специальные переменные, которые отправляют во внешнюю систему данные, связанные с конкретным событием. Список специальных переменных предопределен для каждого события, исходя из его контекста. Чтобы ознакомиться с переменными, смотрите Вебхук для запуска автотестов и Вебхук для изменения статуса тест-плана.
7. **Опционально:** Нажмите **Проверить** справа от поля **URL**, чтобы проверить доступность введенного URL-адреса.
8. Для **POST** и **PUT** запросов настройте передачу тела запроса. В поле **Тип контекста** выберите одну из опций:
   - **Отправлять стандартный контекст** — передает предопределенную для события модель данных. Примеры стандартных контекстов для разных событий:
     - Пример для запуска автотестов
     - Пример для изменения статуса тест-планов
   - **Отправлять пользовательский контекст** — позволяет включить следующие опции:

        - **Заменить системные параметры** — заменяет в теле отправляемого запроса системные параметры, вводимые через `$`, на ассоциирующиеся с ними переменные. Например, при вводе параметра `$BUILD`, в теле запроса выводится номер сборки, для которой произошло событие вебхука.
        - **Экранировать параметры** — экранирует escape-последовательности в теле HTTP. Например, если параметру `$PROJECT_NAME` присваивается значение `dev\new`, `\n` в теле запроса без экранирования параметров будет прочитано как escape-последовательность, и будет отображаться как новая строка. При экранировании параметров escape-последовательности отображаются как текст.
    - **Не отправлять контекст**
9. Нажмите **Сохранить**.
   Созданный вебхук отображается в списке на вкладке **Вебхуки**.

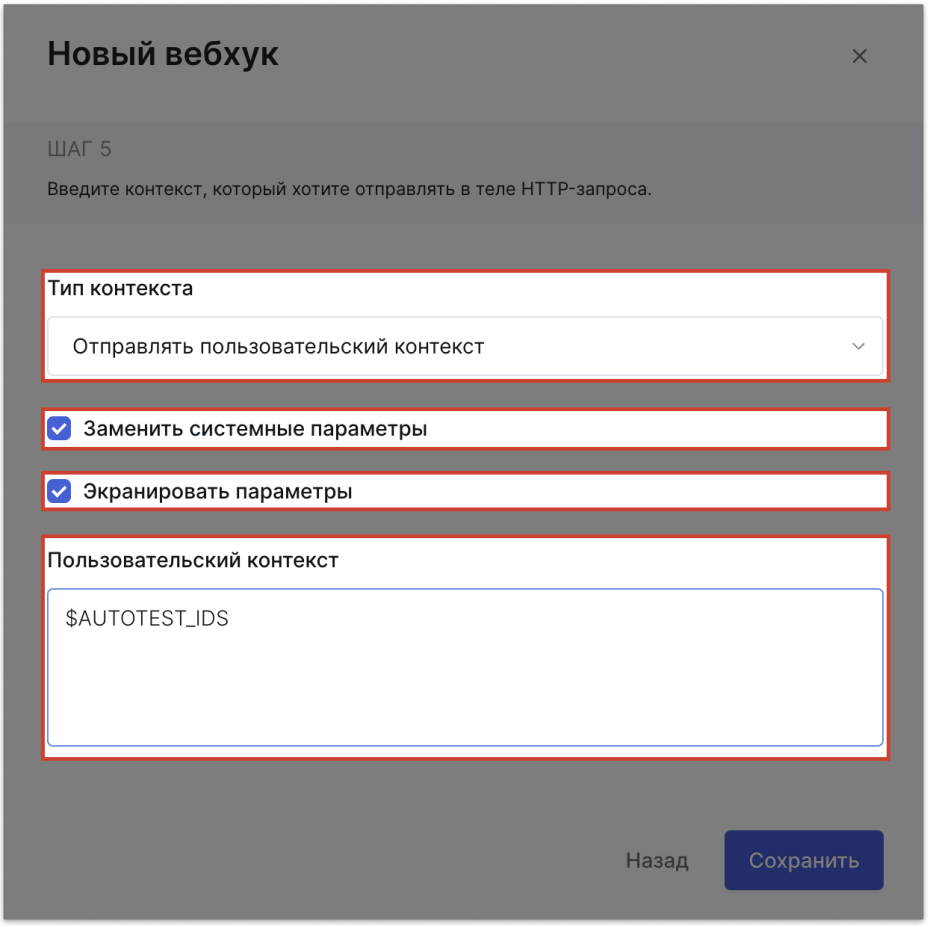
Получение массива данных

Вы можете настроить вебхук на передачу массива данных для событий:

- Остановка прогона
- Прохождение всех автотестов

Чтобы настроить вебхук на передачу массива данных:

1. Начните создавать вебхук. В **Шаге 1** в меню **Событие** выберите **Остановка прогона** или **Прохождение всех автотестов**.
2. В **Шаге 2** в меню **Тип Запроса** выберите **POST** или **PUT**.
3. Заполните данные в шагах 3-4 в соответствии с вашими требованиями.
4. В **Шаге 5** укажите следующее:
    - В меню **Тип** контекста выберите **Отправлять пользовательский контекст**.
    - Выберите пункты **Заменить системные параметры** и **Экранировать параметры**.
    - В поле **Пользовательский контекст** введите символ $, затем в выпадающем меню выберите параметр `AUTOTEST_IDS`.



5. Нажмите **Сохранить**. При наступлении выбранного события вебхук будет передавать массив данных.

Отслеживание логов вебхуков

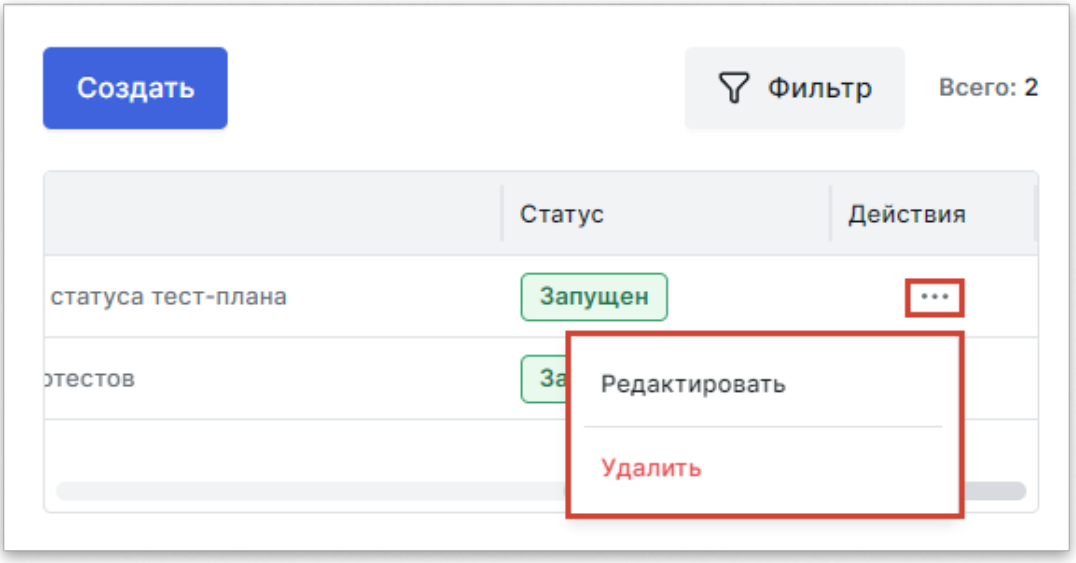
Вы можете посмотреть логи всех запущенных вебхуков. Для этого:

1. Перейдите в раздел **Вебхуки → Журнал**.
2. **Опционально:** Воспользуйтесь фильтром для поиска вебхуков.

По умолчанию отображаются логи за последние 7 дней. Вы можете изменить этот период с помощью фильтра.

Редактирование и удаление вебхуков

1. Откройте проект.
2. В навигационном меню слева перейдите в раздел **Вебхуки → Настройка**.
3. Найдите вебхук, который хотите удалить, в общем списке.
4. В колонке **Действия** откройте меню, нажав значок **Действия** и выберите необходимое действие — **Редактировать** или **Удалить**.



5. Сохраните изменения.

Обновлено: 8/31/26, 7:52 PM

Вебхук для изменения проекта

HTTP-запрос передается при создании, редактировании или архивации проекта.

Доступные специальные переменные

Переменная	Описание
$EVENT_TYPE	Тип события: создание, редактирование, архивация
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится тест-план
$PROJECT_NAME	Название проекта, к которому относится тест-план
$PROJECT_URL	Ссылка на проект, к которому относится тест-план
$PROJECT_GLOBAL_ID	Глобальный идентификатор измененного проекта
$USER_NAME	Имя пользователя, изменившего проект

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "eventType": "UPDATED",
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl": "https://test.testit.ru/projects/1586"
}
```

Обновлено: 4/15/26, 7:08 PM

Вебхуки для действий с рабочими элементами

Рабочие элементы — это тест-кейсы, чек-листы и общий шаги. Для действий с рабочими элементами в системе имеется набор вебхуков.

Вебхук для изменения рабочего элемента

HTTP-запрос передается при создании, редактировании, архивации или восстановлении рабочего элемента.

Доступные специальные переменные

Переменная	Описание
$EVENT_TYPE	Тип события
$WORK_ITEM_ID	Идентификатор рабочего элемента
$WORK_ITEM_NAME	Название рабочего элемента
$WORK_ITEM_TYPE	Тип рабочего элемента
$WORK_ITEM_URL	Ссылка на рабочий элемент
$PROJECT _ID	Идентификатор проекта в Test IT
$PROJECT_NAME	Название проекта
$PROJECT_URL	Ссылка на проект в Test IT
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, рабочий элемент в котором был изменен
$USER_NAME	Имя пользователя, изменившего проект
$WORK_ITEM_DURATION	Продолжительность прохождения теста
$WORK_ITEM_MEDIAN_DURATION	Медианная продолжительность прохождения теста
$WORK_ITEM_PRIORITY	Приоритет теста
$WORK_ITEM_SECTION	Секция, к которой относится рабочий элемент

Пример стандартного тела HTTP

Пример вебхука для создания автотеста. При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "eventType": "CREATED",
  "workItemId": "032d0657-0ac6-42f8-884d-05e8efbac3ae",
  "workItemName": "wi",
  "workItemType": "TestCases",
  "workItemUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1/tes
  "projectId": "7470422c-7db0-4402-8513-fbd7e23d04a4",
  "projectName": "lugg",
  "projectUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1"
}
```

Вебхук для изменения атрибута рабочего элемента

HTTP-запрос передается, когда у атрибута рабочего элемента меняется значение.

Доступные специальные переменные

Переменная	Описание
$WORK_ITEM_ID	Идентификатор рабочего элемента
$WORK_ITEM_NAME	Название рабочего элемента
$WORK_ITEM_URL	Ссылка на рабочий элемент
$ATTRIBUTE_NAME	Название атрибута, значение которого было изменено
$ATTRIBUTE_VALUE	Новое значение измененного атрибута
$USER_NAME	Имя пользователя, изменившего рабочий элемент
$WORK_ITEM_DURATION	Продолжительность прохождения теста
$WORK_ITEM_MEDIAN_DURATION	Медианная продолжительность прохождения теста
$WORK_ITEM_PRIORITY	Приоритет теста
$WORK_ITEM_SECTION	Секция, к которой относится рабочий элемент
$WORK_ITEM_TYPE	Тип рабочего элемента

Пример стандартного тела HTTP

Пример вебхука для создания автотеста. При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "workItemId": "e0bfd5dd-3d93-491c-8213-476beaa8bb1a",
  "workItemName": "Add footnotes to document and verify f
numbering",
  "workItemUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1/tes
  "attributeName": "123",
  "attributeValue": "str"
}
```

Обновлено: 10/11/25, 8:38 PM

Вебхуки для работы с тест-планами

Вебхук для изменения тест-плана

HTTP-запрос передается при создании, редактировании или архивации тест-плана.

Доступные специальные переменные

Переменная	Описание
$EVENT_TYPE	Тип события: создание, редактирование, архивация
$TEST_PLAN_ID	Идентификатор тест-плана в формате GUID
$TEST_PLAN_NAME	Название тест-плана
$TEST_PLAN_URL	Ссылка на тест-план
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится тест-план
$PROJECT_NAME	Название проекта, к которому относится тест-план
$PROJECT_URL	Ссылка на проект, к которому относится тест-план
$BUILD	Номер сборки, указанный при создании тест-плана
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$END_DATE	Дата планируемого окончания тест-плана
$PRODUCT_NAME	Название продукта
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план
$START_DATE	Дата планируемого начала тест-плана
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана

Переменная	Описание
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана
$TEST_PLAN_STATUS	Статус, в который перешел тест-план

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "eventType": "CREATED",
  "testPlanId": "d94f04e7-2cc9-42c1-83a7-
a9ae65e12199",
  "testPlanName": "a",
  "testPlanUrl":
"https://test.testit.ru/projects/1586/test-
plans/118621/plan",
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl": "https://test.testit.ru/projects/1586"
}
```

Вебхук для изменения статуса тест-плана

HTTP-запрос передается при изменении статуса тест-плана.

Доступные специальные переменные

Переменная	Описание
$BUILD	Номер сборки, указанный при создании тест-плана
$PRODUCT_NAME	Название продукта, указанное при создании тест-плана
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится тест-план
$PROJECT_NAME	Название проекта, к которому относится тест-план
$START_DATE	Дата планируемого начала тест-плана
$END_DATE	Дата планируемого окончания тест-плана
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана
$TEST_PLAN_ID	Идентификатор тест-плана в формате GUID
$TEST_PLAN_NAME	Название тест-плана
$TEST_PLAN_STATUS	Статус, в который перешел тест-план
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана
$TEST_PLAN_URL	Ссылка на тест-план

Переменная	Описание
$SERVER_URL	Ссылка на систему Test IT
$USER_NAME	Имя пользователя выполнившего изменение статуса тест-плана
$PROJECT_URL	Ссылка на проект в Test IT, к которому относится тест-план

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "GlobalId": 0,
  "StateName": "string",
  "Name": "string",
  "StartDate": "11/29/2019 14:37:00",
  "EndDate": "11/29/2019 14:37:00",
  "Description": "string",
  "Build": "string",
  "ProjectId": "guid",
  "ProductName": "string",
  "Duration": "",
  "Tags": [],
  "Link": "string",
  "IsDeleted": false
}
```

Вебхук для создания задачи из тест-плана в Jira

HTTP-запрос передается, когда из тест-плана создается задача в Jira.

Использование переменной `$CONFIGURATIONS_PARAMETERS[KEY]`

Переменная `$CONFIGURATIONS_PARAMETERS[KEY]` передает доступ к значениям параметра конфигурации, на которой запускается автотест.
Пример: Для параметра "OS — Windows" укажите:

```json
{
  "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
}
```

В теле вебхука отобразится:

```json
{
  "OperatingSystem": "Windows"
}
```

При указании несуществующего ключа параметра в [KEY] отобразится пустое значение.

Доступные специальные переменные

Переменная	Описание
$RESULT_NAME	Результат, присвоенный тест-поинту: *Успешен*, *Пропущен*, *Провален* или *Заблокирован*
$TEST_POINT_ID	Идентификатор тест-поинта в Test IT
$TEST_POINT_NAME	Название тест-поинта
$TEST_PLAN_ID	Идентификатор тест-плана, в который входит тест-поинт
$TEST_PLAN_NAME	Название тест-плана, в который входит тест-поинт
$TEST_PLAN_URL	Ссылка на тест-план в Test IT, в котором создан тест поинт
$PROJECT_ID	Идентификатор проекта, к которому относится тест-поинт, в формате GUID
$PROJECT_NAME	Название проекта
$PROJECT_URL	Ссылка на проект в Test IT, к которому относится тест-поинт
$BUILD	Номер сборки, указанный при создании тест-плана
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$CONFIGURATION_ID	Идентификатор конфигурации

Переменная	Описание
$CONFIGURATION_NAME	Название конфигурации
$CONFIGURATIONS_PARAMETERS[KEY]	Доступ к значениям параметра конфигурации, на которой запускается автотест
$END_DATE	Дата планируемого окончания тест-плана
$PRODUCT_NAME	Название продукта, указанное при создании тест-плана
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план
$RESULT_URL	Ссылка на результат теста
$START_DATE	Дата планируемого начала тест-плана
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана
$TEST_PLAN_STATUS	Статус тест-плана

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl":
"https://test.testit.ru/projects/1586",
  "testPlanId": "9c294e9c-e79c-44a2-8ba7-
6ee649fc3202",
  "testPlanName": "23",
  "testPlanUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/plan",
  "testPointId": "6c1ff1bd-1f9c-45eb-aa16-
b6deb2832d0c",
  "testPointName": "1",
  "testPointUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/results?testResultId=794e2ed6-2927-4fba-
8824-1b55d6d8b2c9",
  "resultName": "Skipped"
}
```

Обновлено: 4/15/26, 7:08 PM

Вебхук для изменения конфигурации

HTTP-запрос передается при создании, редактировании или архивировании конфигурации.

Использование переменной `$CONFIGURATIONS_PARAMETERS[KEY]`

Переменная `$CONFIGURATIONS_PARAMETERS[KEY]` передает доступ к значениям параметра конфигурации, на которой запускается автотест.
Пример: Для параметра "OS — Windows" укажите:

```json
{
  "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
}
```

В теле вебхука отобразится:

```json
{
  "OperatingSystem": "Windows"
}
```

При указании несуществующего ключа параметра в [KEY] отобразится пустое значение.

Доступные специальные переменные

Переменная	Описание
$EVENT_TYPE	Тип события
$CONFIGURATION_ID	Идентификатор конфигурации
$CONFIGURATION_NAME	Название конфигурации
$CONFIGURATIONS_PARAMETERS[KEY]	Доступ к значениям параметра конфигурации, на которой запускается автотест
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, в котором используется конфигурация

Пример стандартного тела HTTP

Пример вебхука для создания автотеста. При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "eventType": "CREATED",
  "configurationId": "72c8fa17-f19e-46a4-a0db-
dd9ff5b9188b",
  "configurationName": "dw",
  "projectId": "7470422c-7db0-4402-8513-fbd7e23d04a4",
  "projectName": "lugg",
  "projectUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1"
}
```

Обновлено: 7/17/24, 9:57 PM

Вебхуки для работы с тест-поинтами

Вебхук для прохождения всех тест-поинтов

HTTP-запрос передается, когда все тест-поинты, назначенные на пользователя, получают результат (за исключением результатов *В процессе* и *Ожидает*).

Доступные специальные переменные

Переменная	Описание
$USER ID	Идентификатор пользователя в Test IT
$USER_NAME	Имя пользователя в Test IT
$TEST_PLAN_ID	Идентификатор тест-плана
$TEST_PLAN_NAME	Название тест-плана
$TEST_PLAN_URL	Ссылка на тест-план
$PROJECT_ID	Идентификатор проекта в формате GUID
$PROJECT_NAME	Название проекта
$PROJECT_URL	Ссылка на проект в Test IT
$FAILED_COUNT	Количество проваленных тест-поинтов
$SKIPPED_COUNT	Количество пропущенных тест-поинтов
$BLOCKED_COUNT	Количество заблокированных тест-поинтов
$PASSED_COUNT	Количество успешно пройденных тест-поинтов
$BUILD	Номер сборки, указанный при создании тест-плана
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$END_DATE	Дата планируемого окончания тест-плана
$PRODUCT_NAME	Название продукта
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план

Переменная	Описание
$START_DATE	Дата планируемого начала тест-плана
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана
$TEST_PLAN_STATUS	Статус тест-плана

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "userId": "93cbfbd4-9dac-4210-aead-3a72bbace723",
  "userName": "System Administrator",
  "testPlanId": "9c294e9c-e79c-44a2-8ba7-
6ee649fc3202",
  "testPlanName": "23",
  "testPlanUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/plan",
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl":
"https://test.testit.ru/projects/1586",
  "failedCount": "0",
  "skippedCount": "3",
  "blockedCount": "0",
  "passedCount": "0"
}
```

Вебхук для назначения тест-поинта

HTTP-запрос передается, когда пользователя выбирают в тест-плане в поле **Тестировщик**.

Использование переменной `$CONFIGURATIONS_PARAMETERS[KEY]`

Переменная `$CONFIGURATIONS_PARAMETERS[KEY]` передает доступ к значениям параметра конфигурации, на которой запускается автотест.
Пример: Для параметра "OS — Windows" укажите:

```json
{
  "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
}
```

В теле вебхука отобразится:

```json
{
  "OperatingSystem": "Windows"
}
```

При указании несуществующего ключа параметра в [KEY] отобразится пустое значение.

Доступные специальные переменные

Переменная	Описание
$USER ID	Идентификатор пользователя в Test IT
$USER_NAME	Имя пользователя в Test IT
$TEST_POINT_ID	Идентификатор тест-поинта
$TEST_POINT_NAME	Название тест-поинта
$TEST_PLAN_ID	Идентификатор тест-плана, в который входит тест-поинт
$TEST_PLAN_NAME	Название тест-плана, в который входит тест-поинт
$TEST_PLAN_URL	Ссылка на тест-план, в который входит тест-поинт
$PROJECT_ID	Идентификатор проекта в формате GUID
$PROJECT_NAME	Название проекта
$PROJECT_URL	Ссылка на проект в Test IT
$BUILD	Номер сборки, указанный при создании тест-плана
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$CONFIGURATION_ID	Идентификатор конфигурации
$CONFIGURATION_NAME	Название конфигурации

Переменная	Описание
$CONFIGURATIONS_PARAMETERS[KEY]	Доступ к значениям параметра конфигурации, на которой запускается автотест
$END_DATE	Дата планируемого окончания тест-плана
$PRODUCT_NAME	Название продукта
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план
$START_DATE	Дата планируемого начала тест-плана
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана
$TEST_PLAN_STATUS	Статус тест-плана

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "userId": "93cbfbd4-9dac-4210-aead-3a72bbace723",
  "userName": "System Administrator",
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl":
"https://test.testit.ru/projects/1586",
  "testPlanId": "9c294e9c-e79c-44a2-8ba7-
6ee649fc3202",
  "testPlanName": "23",
  "testPlanUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/plan",
  "testPointId": "4ec5aa23-2220-48e4-bff2-
9299ec9be7a0",
  "testPointName": "2"
}
```

Обновлено: 4/15/26, 7:08 PM

Вебхуки для работы с автотестами

Вебхук для запуска автотестов

HTTP-запрос передается при запуске автотестов из интерфейса Test IT. При настройке вебхуков могут использоваться глобальный идентификатор (ID) проекта или тест-плана. Их можно найти в колонке ID в списке в разделах **Проекты** и **Тест-планы**.

Использование переменных $CONFIGURATIONS_PARAMETERS и $CUSTOM_PARAMETERS

Переменные $CONFIGURATIONS_PARAMETERS[KEY] и $CUSTOM_PARAMETERS[KEY]

- Переменная `$CONFIGURATIONS_PARAMETERS[KEY]` передает доступ к значениям параметра конфигурации, на которой запускается автотест.
  Пример: Для параметра "OS — Windows" укажите:

  ```json
  {
    "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
  }
  ```

  В теле вебхука отобразится:

  ```json
  {
    "OperatingSystem": "Windows"
  }
  ```

  При указании несуществующего ключа параметра в [KEY] отобразится пустое значение.
- Переменная `$CUSTOM_PARAMETERS[KEY]` используется аналогичным образом, но значения берутся из параметров, задаваемых при параметризации запуска автотестов.

Переменные $CONFIGURATIONS_PARAMETERS[config_param_my] и $CUSTOM_PARAMETERS[custom_my]

- Переменная `$CONFIGURATIONS_PARAMETERS[config_param_my]` позволяет использовать параметры конфигурации, указанной для запуска автотестов.
- ₽ Переменная `$CUSTOM_PARAMETERS[custom_my]` позволяет выбрать пользовательские параметры в момент запуска автотестов.

`config_param_my` и `custom_my` — это ключи параметров, используемых для запуска автотестов. Они задаются пользователем при создании вебхука.

Доступные специальные переменные

Переменная	Описание
$AUTO_TESTS_COUNT	Количество автотестов, участвующих в запуске
$BUILD	Номер сборки, указанный при создании тест-плана
$PRODUCT_NAME	Название продукта, указанное при создании тест-плана
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относятся запускаемые автотесты
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относятся запускаемые автотесты
$PROJECT_NAME	Название проекта, к которому относятся запускаемые автотесты
$START_DATE	Дата планируемого начала тест-плана
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана, в рамках которого происходит запуск автотестов
$TEST_PLAN_ID	Идентификатор тест-плана в формате GUID, в рамках которого происходит запуск автотестов

Переменная	Описание
$TEST_PLAN_URL	Ссылка на тест-план в Test IT
$TEST_POINTS_COUNT	Количество тестовых случаев, участвующих в запуске
$TEST_RUN_ID	Идентификатор запуска автотестов
$SERVER_URL	Ссылка на систему Test IT
$USER_NAME	Имя пользователя, выполнившего запуск автотестов
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$CONFIGURATION_IDS	Идентификаторы конфигураций
$CONFIGURATION_NAMES	Названия конфигураций
$CONFIGURATIONS_PARAMETERS[KEY]	Доступ к значениям параметра конфигурации, на которой запускается автотест
$CUSTOM_PARAMETERS[KEY]	Доступ к значениям параметра конфигурации, на которой запускается автотест. Значения берутся из параметризации запуска автотеста
$END_DATE	Дата планируемого окончания тест-плана
$PROJECT_URL	Ссылка на проект в Test IT, к которому относятся автотесты

Переменная	Описание
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана
$TEST_PLAN_NAME	Название тест-плана
$TEST_PLAN_STATUS	Статус тест-плана
$TEST_RUN_COMPLETED_DATE	Дата и время завершения прогона
$TEST_RUN_DESCRIPTION	Описание прогона
$TEST_RUN_NAME	Название прогона
$TEST_RUN_STARTED_DATE	Дата запуска прогона
$TEST_RUN_URL	Ссылка на прогон в Test IT

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
    "AutoTests": [
        {
           "Id": "019a9b77-a3c8-76da-9f77-
b74839fd6ce4",
           "ExternalId": "1",
           "ProjectId": "14b18f7f-924c-4c29-8597-
cb05080ccb59",
           "Name": "1",
           "Namespace":
"TestIT.Tests.Tests.Attributes",
           "Classname": "ProjectCustomAttributeTests",
           "ExternalKey": null,
           "GlobalId": 3
        },
        {
            "Classname": "WorkItemServiceTest.cs",
            "ExternalId": "10003",
            "ExternalKey": null,
            "Id": "01893e23-a953-4509-8cac-
650354fa83e3",
            "Name": "WorkItem valid creation test",
            "Namespace": "UnitTest.Core",
            "ProjectId": "915a9144-9803-42a3-8076-
dbc685bfdd1d"
        }
    ],
    "Build": "build",
    "Configurations": [
        {
            "Id": "7faad2ea-8222-4791-8822-
284f0db94649",
            "Name": "Any",
            "Parameters": {
                "Browser": "chrome"
            },
            "ProjectId": "915a9144-9803-42a3-8076-
dbc685bfdd1d"
        }
    ],
    "CustomParameters": {
        "OperatingSystem": "windows"
```

```
    },
    "Name": "TestRun_2024-04-24T11:25:43",
    "ProductName": "my product",
    "Status": "NotStarted",
    "TestPlanGlobalId": 30,
    "TestPlanId": "9dd8429d-f4ee-4fc0-8ed5-
31af29de5cc0",
    "TestPointIds": [
        "4a0663fb-fd47-41f4-9f42-bde8a68d93f6",
        "5358d2a6-7486-4435-944c-f57516cef0f4"
    ],
    "TestRunId": "7823ae8b-92a5-47bd-98bb-
eb09a27807d4"
}
```


Вебхук для создания, редактирования и удаления карточек автотестов

HTTP-запрос передается, когда карточка автотеста была создана, отредактирована или удалена.

Доступные специальные переменные

Переменная	Описание
$EVENT_TYPE	Тип события
$AUTOTEST_ID	Идентификатор автотеста
$AUTOTEST_NAME	Название автотеста
$AUTOTEST_URL	Ссылка на автотест в Test IT
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относятся автотесты
$PROJECT_NAME	Название проекта, к которому относятся автотесты
$PROJECT_URL	Ссылка на проект в Test IT, к которому относятся автотесты
$AUTOTEST_CLASSNAME	Название класса автотеста
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится автотест
$USER_NAME	Имя пользователя, который создал, отредактировал или удалил карточку автотеста

Пример стандартного тела HTTP

Пример вебхука для создания автотеста. При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "eventType": "CREATED",
  "autotestId": "a4f60cc2-3f2b-4949-b779-
7c532485efa1",
  "autotestName": "string",
  "autotestUrl":
"https://test.testit.ru/projects/1586/autotests/1588",
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl": "https://test.testit.ru/projects/1586"
}
```

Вебхук для прохождения всех автотестов

HTTP-запрос передается, когда все автотесты в прогоне пройдены: всем автотестам был проставлен результат *Успешен*, *Провален*, *Пропущен* или *Заблокирован*, и автотестов со статусами *В процессе* или *Ожидает* не осталось.

Доступные специальные переменные

Переменная	Описание
$TEST_RUN_ID	Идентификатор прогона
$TEST_RUN_NAME	Название прогона
$TEST_RUN_URL	Ссылка на прогон в Test IT
$TEST_PLAN_ID	Идентификатор тест-плана (не заполняется, если прогон был запущен не в рамках тест-плана)
$TEST_PLAN_URL	Ссылка на тест-план в Test IT (не заполняется, если прогон был запущен не в рамках тест-плана)
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится прогон
$PROJECT_NAME	Название проекта, к которому относится прогон
$PROJECT_URL	Ссылка на проект в Test IT
$BUILD	Номер сборки, указанный при создании тест-плана
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$END_DATE	Дата планируемого окончания тест-плана
$PRODUCT_NAME	Название продукта
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относятся автотесты
$START_DATE	Дата планируемого начала тест-плана
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*

Переменная	Описание
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана
$TEST_PLAN_STATUS	Статус тест-плана
$TEST_RUN_COMPLETED_DATE	Дата и время завершения прогона
$TEST_RUN_DESCRIPTION	Описание прогона
$TEST_RUN_STARTED_DATE	Дата запуска прогона

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "testRunId": "227f2b6b-5361-4196-bbb3-7859ac88891b",
  "testRunName": "TestRun_2023-06-15T11:38:57",
  "testRunUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/test-runs/227f2b6b-5361-4196-bbb3-
7859ac88891b",
  "testPlanId": "9c294e9c-e79c-44a2-8ba7-
6ee649fc3202",
  "testPlanName": "23",
  "testPlanUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/plan",
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl": "https://test.testit.ru/projects/1586"
}
```

Вебхук для остановки прогона

HTTP-запрос передается при остановке прогона с помощью API-метода `POST/api/v2/testRuns/{id}/stop` .

Доступные специальные переменные

Переменная	Описание
$TEST_RUN_ID	Идентификатор прогона
$TEST_RUN_NAME	Название прогона
$TEST_RUN_URL	Ссылка на прогон в Test IT
$TEST_PLAN_ID	Идентификатор прогона (не заполняется, если прогон запущен не в рамках тест-плана)
$TEST_PLAN_NAME	Название прогона (не заполняется, если прогон запущен не в рамках тест-плана)
$TEST_PLAN_URL	Ссылка на тест-план в Test IT (не заполняется, если прогон запущен не в рамках тест-плана)
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится прогон
$PROJECT_NAME	Название проекта, к которому относится прогон
$PROJECT_URL	Ссылка на проект в Test IT, к которому относится прогон
$BUILD	Номер сборки, указанный при создании тест-плана
$COMPLETED_ON_DATE	Дата и время перевода тест-плана в статус *Завершен*
$END_DATE	Дата планируемого окончания тест-плана
$PRODUCT_NAME	Название продукта
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план

Переменная	Описание
$START_DATE	Дата планируемого начала тест-плана
$STARTED_ON_DATE	Дата и время перевода тест-плана в статус *В процессе*
$TEST_PLAN_DESCRIPTION	Описание тест-плана
$TEST_PLAN_DURATION	Продолжительность тест-плана
$TEST_PLAN_GLOBAL_ID	Глобальный идентификатор тест-плана
$TEST_PLAN_STATUS	Статус тест-плана
$TEST_RUN_COMPLETED_DATE	Дата и время завершения прогона
$TEST_RUN_DESCRIPTION	Описание прогона
$TEST_RUN_STARTED_DATE	Дата запуска прогона

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl":
"https://test.testit.ru/projects/1586",
  "testPlanId": "9c294e9c-e79c-44a2-8ba7-
6ee649fc3202",
  "testPlanName": "23",
  "testPlanUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/plan",
  "testRunId": "7c0400f8-ad43-4e45-9cf7-382047f4044f",
  "testRunName": "TestRun_2023-06-15T11:46:31",
  "testRunUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/test-runs/7c0400f8-ad43-4e45-9cf7-
382047f4044f"
}
```

Вебхук для прохождения автотеста

HTTP-запрос передается, когда автотесту проставляется результат *Успешен*, *Провален*, *Пропущен* или *Заблокирован*.

Доступные специальные переменные

Переменная	Описание
$RESULT_NAME	Результат автотеста: *Успешен*, *Пропущен*, *Провален* или *Заблокирован*
$RESULT_URL	Ссылка на результат автотеста
$AUTOTEST_ID	Идентификатор автотеста
$AUTOTEST_NAME	Название автотеста
$AUTOTEST_URL	Ссылка на автотест
$TEST_RUN_ID	Идентификатор прогона
$TEST_RUN_NAME	Название прогона
$TEST_RUN_URL	Ссылка на прогон в Test IT
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится прогон
$PROJECT_NAME	Название проекта, к которому относится прогон
$PROJECT_URL	Ссылка на проект в Test IT, к которому относится автотест
$AUTOTEST_CLASSNAME	Название класса автотеста
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план
$TEST_RUN_COMPLETED_DATE	Дата и время завершения прогона
$TEST_RUN_DESCRIPTION	Описание прогона
$TEST_RUN_STARTED_DATE	Дата запуска прогона

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "resultName": "Passed",
  "resultUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/results?testResultId=471a0169-e327-4302-
b500-4d68241282ef",
  "autotestId": "a4f60cc2-3f2b-4949-b779-
7c532485efa1",
  "autotestName": "string",
  "autotestUrl":
"https://test.testit.ru/projects/1586/autotests/1588",
  "testRunId": "227f2b6b-5361-4196-bbb3-7859ac88891b",
  "testRunName": "TestRun_2023-06-15T11:38:57",
  "testRunUrl":
"https://test.testit.ru/projects/1586/test-
plans/1590/test-runs/227f2b6b-5361-4196-bbb3-
7859ac88891b",
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl": "https://test.testit.ru/projects/1586"
}
```

Изменение связи между автотестом и рабочим элементом

HTTP-запрос передается, когда связь между автотестом и рабочим элементом создается или удаляется.

Доступные специальные переменные

Переменная	Описание
$EVENT_TYPE	Тип события
$AUTOTEST_ID	Идентификатор автотеста
$AUTOTEST_NAME	Название автотеста
$AUTOTEST_URL	Ссылка на автотест
$WORK_ITEM_ID	Идентификатор рабочего элемента
$WORK_ITEM_NAME	Название рабочего элемента
$WORK_ITEM_URL	Ссылка на рабочий элемент
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относятся запускаемые автотесты
$PROJECT_NAME	Название проекта, к которому относятся запускаемые автотесты
$PROJECT_URL	Ссылка на проект в Test IT, к которому относится автотест
$AUTOTEST_CLASSNAME	Название класса автотеста
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится тест-план
$USER_NAME	Имя пользователя, который создал или удалил связь между ручным и автоматическим тестом
$WORK_ITEM_DURATION	Продолжительность прохождения ручного теста
$WORK_ITEM_PRIORITY	Приоритет ручного теста
$WORK_ITEM_SECTION	Секция, к которой относится ручной тест
$WORK_ITEM_TYPE	Тип рабочего элемента (ручного теста)

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "eventType": "UPDATED",
  "workItemId": "e4428f83-fe7f-4740-9e4f-8a6136360ef6",
  "workItemName": "Test incremental search with multi-pag
document",
  "workItemType": "TestCases",
  "workItemUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1/tes
  "projectId": "7470422c-7db0-4402-8513-fbd7e23d04a4",
  "projectName": "lugg",
  "projectUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1"
}
```

Обновлено: 8/13/26, 12:57 PM

Вебхуки для уведомлений пользователя

Вебхук для выбора пользователя в атрибуте рабочего элемента

HTTP-запрос передается, когда пользователь был выбран в атрибуте рабочего элемента.

Доступные специальные переменные

Переменная	Описание
$USER ID	Идентификатор пользователя в Test IT
$USER_NAME	Имя пользователя в Test IT
$WORK_ITEM_ID	Идентификатор рабочего элемента
$WORK_ITEM_NAME	Название рабочего элемента
$WORK_ITEM_URL	Ссылка на рабочий элемент
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится прогон
$PROJECT_NAME	Название проекта, к которому относится прогон
$PROJECT_URL	Ссылка на проект в Test IT, к которому относится автотест
$ATTRIBUTE_NAME	Название атрибута
$ATTRIBUTE_VALUE	Значение атрибута
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится рабочий элемент
$WORK_ITEM_DURATION	Продолжительность прохождения теста
$WORK_ITEM_MEDIAN_DURATION	Медианная продолжительность прохождения теста
$WORK_ITEM_PRIORITY	Приоритет теста
$WORK_ITEM_SECTION	Секция, к которой относится рабочий элемент

Переменная	Описание
$WORK_ITEM_TYPE	Тип рабочего элемента

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "projectId": "cf76841e-fb42-42ab-ae07-2a5d02b04fc3",
  "projectName": "вебхуки",
  "projectUrl":
"https://test.testit.ru/projects/1586",
  "userId": "93cbfbd4-9dac-4210-aead-3a72bbace723",
  "userName": "System Administrator",
  "workItemId": "3e2b1d07-22f9-4548-976c-
53a699e2dfa5",
  "workItemName": "1",
  "workItemUrl":
"https://test.testit.ru/projects/1586/tests/1591",
  "attributeName": "usr",
  "attributeValue": "93cbfbd4-9dac-4210-aead-
3a72bbace723"
}
```

Вебхук для упоминания пользователя

HTTP-запрос передается, когда пользователя Test IT отмечают в комментарии с помощью символа @.

Доступные специальные переменные

Переменная	Описание
$USER ID	Идентификатор пользователя в Test IT
$USER_NAME	Имя пользователя в Test IT
$WORK_ITEM_ID	Идентификатор рабочего элемента
$WORK_ITEM_NAME	Название рабочего элемента
$WORK_ITEM_URL	Ссылка на рабочий элемент
$PROJECT_ID	Идентификатор проекта в формате GUID, к которому относится прогон
$PROJECT_NAME	Название проекта, к которому относится прогон
$PROJECT_URL	Ссылка на проект в Test IT, к которому относится автотест
$PROJECT_GLOBAL_ID	Глобальный идентификатор проекта, к которому относится рабочий элемент
$WORK_ITEM_DURATION	Продолжительность прохождения теста
$WORK_ITEM_MEDIAN_DURATION	Медианная продолжительность прохождения теста
$WORK_ITEM_PRIORITY	Приоритет теста
$WORK_ITEM_SECTION	Секция, к которой относится рабочий элемент
$WORK_ITEM_TYPE	Тип рабочего элемента

Пример стандартного тела HTTP

При выборе опции **Отправлять стандартный контекст** в поле **Тело HTTP** Test IT передает следующую структуру данных:

```json
{
  "userId": "3c6fdc26-a7a1-4c50-83be-1ea55bd43587",
  "userName": "System Administrator",
  "workItemId": "68d744c8-47d7-4189-a161-e392527f24c2",
  "workItemName": "Test full screen reading on minimal re
system",
  "workItemUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1/tes
  "projectId": "7470422c-7db0-4402-8513-fbd7e23d04a4",
  "projectName": "project",
  "projectUrl":
"https://regress01.qa.enterprise.testit.ru/projects/1"
}
```

Обновлено: 4/15/26, 7:08 PM

Настройка уведомлений в Slack с помощью вебхуков

> **₽ Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Настройка Slack

1. Войдите в рабочее пространство Slack.
2. Откройте Slack App Directory .
3. Найдите в поиске приложение **Incoming webhooks**. Откройте страницу приложения.
4. Нажмите **Add to Slack**.
5. Выберите из выпадающего списка канал, в который хотите отправлять уведомления и нажмите **Add Incoming Webhooks Integration**.
6. Вы можете изменить аватар имя пользователя, от которого Slack будет присылать уведомления. Для этого пролистайте страницу вниз до секции **Integration Settings**, введите имя в поле **Customize Name** и загрузите аватар в поле **Customize Icon**.
7. Скопируйте URL вебхука из поля **Webhook URL**.
8. Нажмите **Save Settings**.

Создание вебхука в Test IT

1. Откройте проект.
2. В навигационном меню слева нажмите **Вебхуки**.
3. Нажмите **Создать**.
4. Укажите название вебхука и выберите тип события — **Запуск автотестов** или **Изменение статуса тест-плана**. В данной инструкции используется пример для событие **Изменение статуса тест-плана**.
5. Вставьте скопированный ранее URL вебхука Slack в поле **URL** и убедитесь, что в поле **Тип запроса** указан тип **POST**.
6. В шагах **Параметры URL** и **Заголовки HTTP** нажмите **Далее**.

7. В шаге **Тело HTTP** поставьте флажок **Отправлять пользовательский контекст**.
8. Поставьте флажок **Заменять системные параметры** над полем ввода текста.
9. Введите текст уведомления для Slack, используя системные параметры. Для ввода системных параметров вводите их значения через `$`.

Пример стандартного тела HTTP (развернуть)

```json
{
 "blocks": [
  {
   "type": "header",
   "text": {
    "type": "plain_text",
    "text": "$USER_NAME changed test plan status to
$TEST_PLAN_STATUS",
    "emoji": true
   }
  },
  {
   "type": "section",
   "fields": [
    {
     "type": "mrkdwn",
     "text": "*Build number:*\n$BUILD"
    },
    {
     "type": "mrkdwn",
     "text": "*Product name:*\n$PRODUCT_NAME"
    }
   ]
  },
  {
   "type": "section",
   "fields": [
    {
     "type": "mrkdwn",
     "text": "*Project name:*\n$PROJECT_NAME"
    },
    {
     "type": "mrkdwn",
     "text": "*Test plan name:*\n$TEST_PLAN_NAME"
    }
   ]
  },
  {
   "type": "section",
   "fields": [
    {
     "type": "mrkdwn",
```

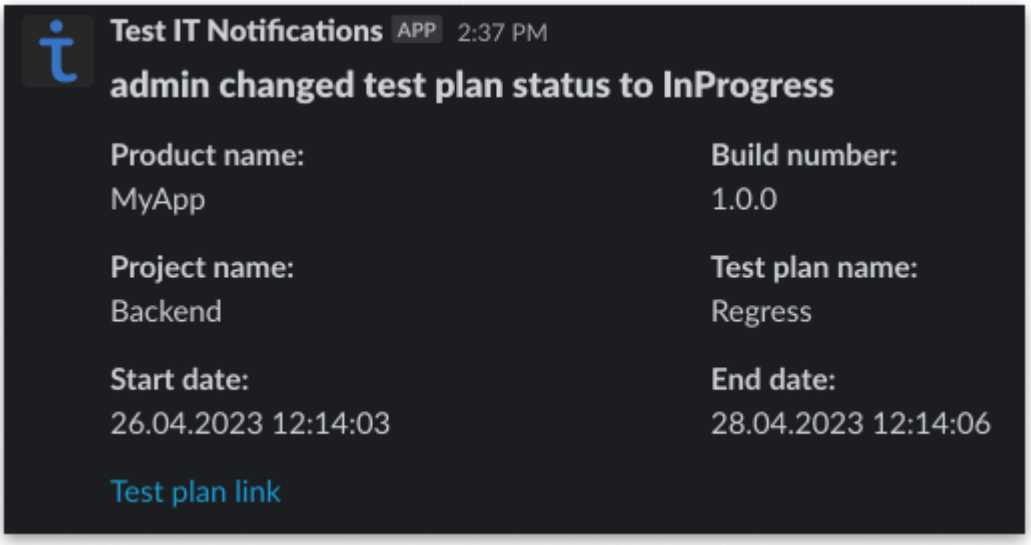
```
      "text": "*Start date:*\n$START_DATE"
     },
     {
      "type": "mrkdwn",
      "text": "*End date:*\n$END_DATE"
     }
    ]
   },
   {
    "type": "section",
    "text": {
     "type": "mrkdwn",
     "text": "<$TEST_PLAN_URL|Test plan link>"
    }
   }
  ]
 }
```

Ознакомьтесь с переменными системных параметров:

- Вебхук для запуска автотестов
- Вебхук для изменения статуса тест-плана

11. Нажмите **Сохранить**.

При изменении статуса тест-планов в проекте вам будут приходить уведомления в указанный вами канал в Slack.



Обновлено: 4/15/26, 7:08 PM

Настройка уведомлений в Telegram с помощью вебхуков

> **₽ Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

- **Для этого раздела мы подготовили видео:**

Вы можете получать уведомления из Test IT в чаты Telegram (группы и личные беседы). Количество чатов, в которые вы можете получать уведомления, не ограничено.

Настройка включает в себя этапы:

1. Регистрация компании в чат-боте Test IT
2. Добавление чат-бота в группу или личную беседу Telegram
3. Создание и редактирование вебхука для чат-бота Telegram

Регистрация компании в чат-боте Test IT

Регистрация компании осуществляется в Telegram. В процессе регистрации используются команды, которые вводятся в чат Telegram (например `/help`). По окончании регистрации вы получите токен, который потребуется при создании

или редактировании вебхука. Токен действует бессрочно.

Чтобы зарегистрировать компанию:

1. В Telegram откройте чат-бота для уведомлений Test IT. Для этого в поле **Поиск** введите идентификатор чат-бота: идентификатор @testit_notifications_bot.
2. **Опционально:** Для просмотра списка доступных команд используйте команду `/help`. Вы сможете задать требуемую команду, нажав на нее.
3. В чате бота зарегистрируйте вашу компанию. Для этого используйте команду `/register`, затем введите название компании. В чате отобразится сообщение, содержащее запрос электронной почты.

   > **Внимание!**
   >
   > С одного аккаунта Telegram вы можете зарегистрировать не более одной компании.

4. Введите адрес электронной почты. В чате отобразится сообщение об успешной регистрации.
5. Получите токен для авторизации бота (используется при создании вебхука) с помощью команды `/getcompany`. Скопируйте токен.

Добавление чат-бота в группу или личную беседу Telegram

Добавление чат-бота осуществляется в Telegram. В процессе используются команды, которые вводятся в чат Telegram (например `/help`).

Чтобы добавить чат-бота в группу или беседу:

1. В чат-боте Test IT получите идентификатор чата с помощью команды `/getchatid`. Отобразится сообщение, содержащее идентификатор. Например, Chat ID: 908664085, где 908664085 — идентификатор.
2. Скопируйте идентификатор чата и добавьте чат в зарегистрированную компанию. Для этого используйте команду `/addchat`, затем через пробел введите идентификатор чат-бота Test IT. Например: `/addchat 908664085`.
3. Добавьте чат-бота в группу или личную беседу. Для этого:
   а. Откройте группу или личную беседу, в которую хотите добавить бота.
   б. Перейдите в настройки группы.
   в. Нажмите **Добавить участников**.
   г. Введите название бота: *@testit_notifications_bot*. Нажмите на него, затем нажмите **OK**.

4. Получите идентификатор группы или личной беседы, в которую хотите получать уведомление с помощью команды `/getchatid`. Отобразится сообщении о добавлении чата. Например: *Chat ID: -918078093*.

   > **Внимание!**
   >
   > Если идентификатор чата содержит специальные символы, скопируйте их. Например, идентификатор "-918078093" содержит знак "-", который необходимо скопировать.

5. Добавьте чат-бота Test IT в группу или личную беседу. Для этого используйте команду `/addchat`, затем через пробел введите идентификатор чат-бота Test IT. Отобразится сообщение о добавлении чата.

Создание и редактирование вебхука для чат-бота Telegram

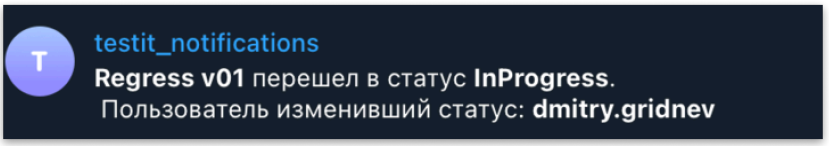
Для работы с Telegram вам потребуется создать новый вебхук в Test IT или отредактировать уже существующий с учетом следующих особенностей:

1. В поле **URL** укажите URL Telegram-сервиса Test IT: **https://webhooks.testit.software/telegram**
2. В меню **Тип запроса** выберите **POST**.
3. В разделе **Заголовки HTTP** укажите **Authorization**, в поле **Значение** введите токен для авторизации бота, полученный в чат-боте Test IT.
4. В меню **Тип контекста** выберите **Отправлять пользовательский контекст**.
5. В поле **Пользовательский контекст** введите запрос:
   а. В поле **"message"**, расположенном в теле запроса, введите текст сообщения, используя язык разметки HTML .
   б. В поле **"parseMode"** укажите режим форматирования текста: *HTML*.
   в. В поле **"chatIds"** добавьте идентификаторы чатов, в которые хотите направлять уведомления. Вы можете добавить несколько идентификаторов. Если вы оставите поле **"chatIds"** пустым, уведомления будут направляться во все чаты, которые добавлены к компании. Если вы добавите идентификатор чата, не добавленного в компанию, уведомления в него приходить не будут.

Пример запроса (развернуть)

```json
{
  "message": "<b>$TEST_PLAN_NAME</b> перешел в статус
<b>$TEST_PLAN_STATUS</b>. \n Пользователь изменивший
статус: <b>$USER_NAME</b>",
  "parseMode": "HTML",
  "chatIds": [
    908664085
  ]
}
```

После отправки сообщения в Telegram будет отправлено уведомление вида:



6. Завершите создание или редактирование вебхука.
   Уведомления будут приходить в чаты, определенные в пользовательском контексте.

Обновлено: 8/13/26, 11:57 AM

Настройка интеграции с GitFlic с помощью вебхуков

- **Вебинар о возможностях интеграции с GitFlic:**

GitFlic — это сервис для хранения исходного кода и работы с ним, основанный на системе контроля версий Git. Вы можете настроить интеграцию с GitFlic для работы с автотестами, расположенными в его репозитории, из интерфейса Test IT.

Для настройки интеграции вам нужно:

- Зарегистрировать агент GitFlic Runner, используемый вашей компанией (для облачной версии GitFlic Runner)
- Настроить репозиторий GitFlic:
  - Подготовить файлы `requirements.txt` и `connection_config.ini`
  - Создать конвейер (пайплайн)
  - Получить токен доступа GitFlic
- Настроить вебхук в Test IT

Чтобы узнать больше о работе с GitFlic API, смотрите официальную документацию GitFlic .

Регистрация агента (для облачной версии)

Для работы с облачной (Cloud) версией агента GitFlic Runner необходима регистрация агента. Чтобы зарегистрировать агент:

- Следуйте инструкции в документации GitFlic .

Настройка репозитория GitFlic

Перед началом работы

При интеграции с Test IT агент GitFlic Runner как правило уже настроен и готов к работе.
В случае необходимости произведите необходимые настройки согласно документации GitFlic .

Подготовка

1. Создайте файл `requirements.txt`, в котором должны быть перечислены следующие пакеты для установки:

```text
pytest
testit-adapter-pytest
```

2. Поместите в репозиторий GitFlic автотесты с декораторами `externalID` и `workitemID`. Данные декораторы должны совпадать с External ID автотестов и Global ID рабочих элементов в Test IT.
3. Создайте файл `connection_config.ini`, в котором необходимо задать переменные окружения.

```text
[testit]
url={URL}
privatetoken={PRIVATE_TOKEN}
testrunId={TEST_RUN_ID}
projectId={TMS_PROJECT_ID}
configurationId={TMS_CONFIGURATION_ID}
```

Идентификатор проекта, конфигурации и другие данные о проекте доступны в консоли. Подробнее: Просмотр идентификатора проекта.

Создание конвейера (пайплайна)

1. Создайте файл `gitflic-ci.yaml` и поместите в корневую папку вашего проекта.
    - Пример файла `gitflic-ci.yaml` :

```yaml
stages:
  - test

run_tests:
  image: python:3.9
  stage: test

  before_script:
      --pip install -r requirements.txt
  script:
    - pytest --testit
```

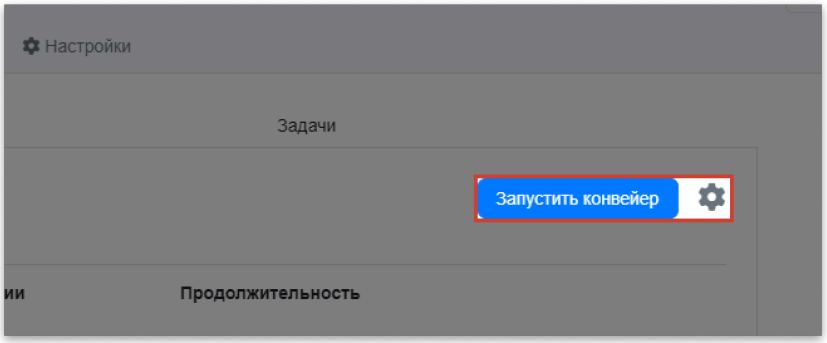
    - Пример `gitflic-ci.yaml` файла без `connection_config.ini` . В данном случае необходимо инициировать использование переменных окружения через параметры командной строки.

```yaml
stages:
  - test

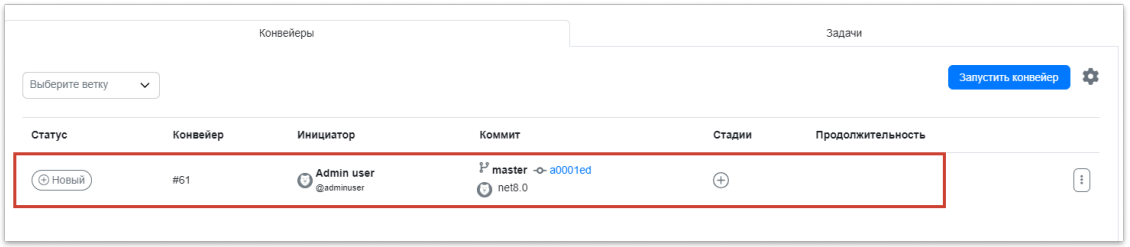
run_tests:
  image: python:3.9
  stage: test

  before_script:
      --pip install -r requirements.txt
  script:
    - pytest --testit --testrunid=${TEST_RUN_ID} --
testit_url=${URL} --privatetoken=${PRIVATE_TOKEN}
```

2. Выполните коммит изменений.
3. Войдите в репозиторий GitFlic и в верхнем навигационном меню выберите **CI/CD**.
4. Нажмите **Запустить конвейер**.



При успешном создании конвейера он отобразится в списке.



Получение токена доступа в GitFlic

Для настройки интеграции вам потребуется токен доступа `accessToken`, получаемый через API GitFlic.

1. Следуйте инструкции в официальной документации GitFlic.
   - Пример токена доступа:

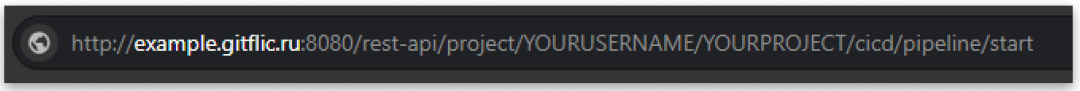
```json
{
  "accessToken":
  "868f3765-aaaa-aaaa-aaaa-13faw8a3ca68",
  "refreshToken":
  "cb4917e7-aaaa-aaaa-aaaa-9bb604b42e0c",
  "expires": "2023-01-12T15:45:01.961001"
}
```

2. Скопируйте значение `accessToken` из полученного объекта .json.

Настройка вебхука в Test IT

1. Откройте проект.
2. В навигационном меню слева откройте секцию **Настройки** и выберите **Вебхуки**.
3. Нажмите **Создать**.
4. В Шаге 1 укажите название нового вебхука и выберите тип события — **Запуск автотестов**.

5. В Шаге 2 убедитесь, что тип передаваемого запроса выставлен как **POST**, в поле **URL** вставьте строку:
   - Для версии **Self-hosted**: `http://{example.gitflic.ru}:8080/rest-api/project/{USERALIAS}/{PROJECTALIAS}/cicd/pipeline/start`

   - Для версии **Cloud**:
     `https://api.gitflic.ru/project/{USERALIAS}/{PROJECTALIAS}/cicd/pipeline/start` ,
     где
   - `{USERALIAS}` — ваш user alias (скопируйте из адресной строки проекта)
   - `{PROJECTALIAS}` — ваш project alias (скопируйте из адресной строки проекта)
6. В Шаге 3 нажмите **Далее**.
7. В Шаге 4 **Заголовки HTTP** укажите следующие данные:
   - В поле **Параметр** введите *Authorization*
   - В поле **Значение** введите *token {accessToken}*, где {accessToken} — ваш токен доступа GitFlic, например: *token 868f3765-aaaa-aaaa-aaaa-13faw8a3ca68*
8. В Шаге 5 в поле **Тип контекста** выберите **Не отправлять контекст**.
9. Нажмите **Сохранить**.

При запуске автотестов из Test IT в GitFlic будет запускаться конвейер. Результаты запуска возвращаются в систему Test IT.

Обновлено: 8/12/26, 4:43 PM

Настройка интеграции с GitLab CI с помощью вебхуков

- **Для этого раздела мы подготовили видео:**

Настройка репозитория в GitLab

Подготовка

1. Создайте файл `requirements.txt` , в котором должны быть перечислены следующие пакеты для установки:

```text
pytest
testit-adapter-pytest
```

2. Поместите в репозиторий GitLab автотесты с декораторами `externalID` и `workitemID` . Данные декораторы должны совпадать с External ID автотестов и Global ID рабочих элементов в Test IT.
3. Создайте файл `connection_config.ini` , в котором необходимо задать переменные окружения.

```text
[testit]
url={URL}
privatetoken={PRIVATE_TOKEN}
testrunID={TEST_RUN_ID}
```

Создание пайплайна

1. Войдите в репозиторий GitLab.
2. В навигационном меню выберите **CI/CD → Editor**.
3. Нажмите **Create New CI/CD Pipeline**. Откроется редактор файла `.gitlab-ci.yml`.
4. Настройте `.gitlab-ci.yml` файл для установки необходимых пакетов из файла **requirements.txt** и запуска автотестов с требуемыми переменными окружения:
   - Пример `.gitlab-ci.yml` файла при наличии `connection_config.ini`.

     ```text
     stages:
       - test

     run_tests:
       image: python:3.9
       stage: test

       before_script:
           --pip install -r requirements.txt
       script:
         - pytest --testit
     ```

   - Пример `.gitlab-ci.yml` файла без `connection_config.ini`. В данном случае необходимо инициировать использование переменных окружения через параметры командной строки.

```text
stages:
  - test

run_tests:
  image: python:3.9
  stage: test

  before_script:
      --pip install -r requirements.txt
  script:
    - pytest --testit --testrunid=${TEST_RUN_ID} --
testit_url=${URL} --privatetoken=${PRIVATE_TOKEN}
```

5. Нажмите **Commit changes**.

Получение токена

1. Войдите в репозиторий GitLab.
2. В навигационном меню выберите **Settings → CI/CD**.
3. Раскройте секцию **Pipeline triggers** и нажмите **Add trigger**. При необходимости добавьте описание.

Настройка вебхука в Test IT

1. Откройте проект.
2. В навигационном меню слева нажмите **Вебхуки**.
3. Нажмите **Создать**.
4. Укажите название нового вебхука и выберите тип события — **Запуск автотестов**.
5. В поле **URL** вставьте строку `https://{DOMAIN}/api/v4/projects/{PROJECT_ID}/trigger/pipeline` , где DOMAIN нужно заменить на адрес вашего GitLab, а PROJECT_ID — на Global ID репозитория в GitLab. Убедитесь, что тип передаваемого запроса выставлен как **POST**.
6. В шаге **Параметры URL** установите следующие передаваемые параметры:
   - ref = название ветки GitLab репозитория с автотестами
   - token = token триггера, созданного на этапе настройки репозитория.
   - variables[URL] = $SERVER_URL
   - variables[TEST_RUN_ID] = $TEST_RUN_ID
   - variables[PRIVATE_TOKEN] = секретный ключ API из профиля Test IT.
7. В шаге **Заголовки HTTP** нажмите **Далее**.
8. В шаге **Тело HTTP** в поле **Тип контекста** выберите **Не отправлять контекст**.

9. Нажмите **Сохранить**.

Теперь при запуске автотестов из Test IT в GitLab будет запускаться пайплайн. Результаты пайплайна возвращаются в систему Test IT.

Обновлено: 8/12/26, 4:43 PM

Настройка запуска автотестов в Jenkins CI с помощью вебхука

- **Для этого раздела мы подготовили видео:**

Настройка включает в себя следующие этапы:

1. Подключение вашего репозитория к Jenkins (настройка пайплайнов). Подробности смотрите в официальной документации Jenkins .
2. Подключение адаптера Test IT или Test IT CLI
3. Настройка Jenkins
4. Настройка вебхука в Test IT
5. Проверка интеграции (запуск автотестов)

Настройка Jenkins

Создание параметров

1. Войдите в Jenkins.
2. Перейдите в **Settings → General**.
3. Добавьте новые параметры с наименованиями следующих переменных окружения:
   - `TEST_RUN_ID`
   - `PRIVATE_TOKEN`

- `URL`

Получение токена авторизации

1. Перейдите в **Профиль → Статус**.
2. Скопируйте содержимое поля **Jenkins User ID**.
3. Перейдите в **Профиль → Настроить**.
4. Скопируйте **API Token**.
5. Закодируйте строку `JENKINS_USER_ID:API_TOKEN` методом Base64. Получившаяся строка — ваш токен авторизации Jenkins.

Настройка вебхука в Test IT

1. Откройте проект.
2. В навигационном меню слева нажмите **Вебхуки**.
3. Нажмите **Создать**.
4. Укажите название вебхука и выберите тип события — **Запуск автотестов**.
5. В поле **URL** вставьте строку `http://{DOMAIN}/job/{ITEM_NAME}/buildWithParameters`, где `DOMAIN` это адрес вашей системы Jenkins, а `ITEM_NAME` — наименование Jenkins Item. Убедитесь, что тип передаваемого запроса выставлен как **POST**.
6. В шаге **Параметры URL** установите следующие передаваемые параметры:
   - **Ключ**: `testRunId`, **Значение**: `$TEST_RUN_ID`.
   - **Ключ**: `testProjectId`, **Значение**: `$PROJECT_ID`.
7. В шаге **Заголовки HTTP** установите следующий параметр:
   - **Ключ:** `Authorization`, **Значение:** `Basic {AUTHORIZATION_TOKEN}`, где `AUTHORIZATION_TOKEN` — токен авторизации, сгенерированный на этапе настройки Jenkins.
8. В шаге **Тело HTTP** в поле **Тип запроса** выберите **Не отправлять контекст**.
9. Нажмите **Сохранить**.

Проверка интеграции

1. Запустите автотесты в Jenkins.
2. Перейдите к логам, нажав **Посмотреть журнал**.

В логах можно увидеть результирующий запрос и ответ от внешней системы:

Запрос:

```http
Method: POST
RequestUrl: 'http://jenkins.mycompanyname.ru/job/testit-webinar
/buildWithParameters?testRunId=a62172c1-f067-4006-97c0-
e93aa8e8f3b5&testProjectId=301'
Version: 2.0
Content: <null>
Headers: { Authorization: Basic ****** }
```

Ответ:

```text
StatusCode: 201
ReasonPhrase: 'Created'
Version: 1.1
Content:
System.Net.Http.HttpConnection+HttpConnectionResponseContent
Headers:
 {
 Date: Fri, 26 Jul 2019 16:32:34 GMT
 X-Content-Type-Options: nosniff
 Location: http://jenkins.mycompanyname.ru/queue/item/1777974/
 Server: Jetty(9.4.z-SNAPSHOT)
 Content-Length: 0
 }
```

В Jenkins запустилась сборка проекта с полученными параметрами.

Обновлено: 8/12/26, 4:43 PM

Импорт и экспорт проектов

₽ Тарифные ограничения

Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Пользователи с системной ролью **Администратор** могут экспортировать или импортировать один или несколько проектов в формате `.zip` (формат архивированных файлов — `.csv`). Эта опция может быть полезна, например при миграции из поставок Test IT Базовый или Стандарт в Test IT Про.

Объем импортируемых и экспортируемых файлов:

- В Test IT Базовый и Стандарт (облачная поставка) — до 1 ГБ
- В Test IT Про (серверная поставка) — до 50 ГБ

Экспортируемые / импортируемые данные

Информация о проекте, **передаваемая** при экспорте и импорте:

- Проект и его описание
- Библиотека тестов и их версии
- Тест-планы
- Автотесты
- Прогоны автотестов за последние 30 суток
- Вебхуки
- Атрибуты, параметры, конфигурации, категории ошибок проекта
- Архив проекта

Информация о проекте, **не передаваемая** при экспорте и импорте:

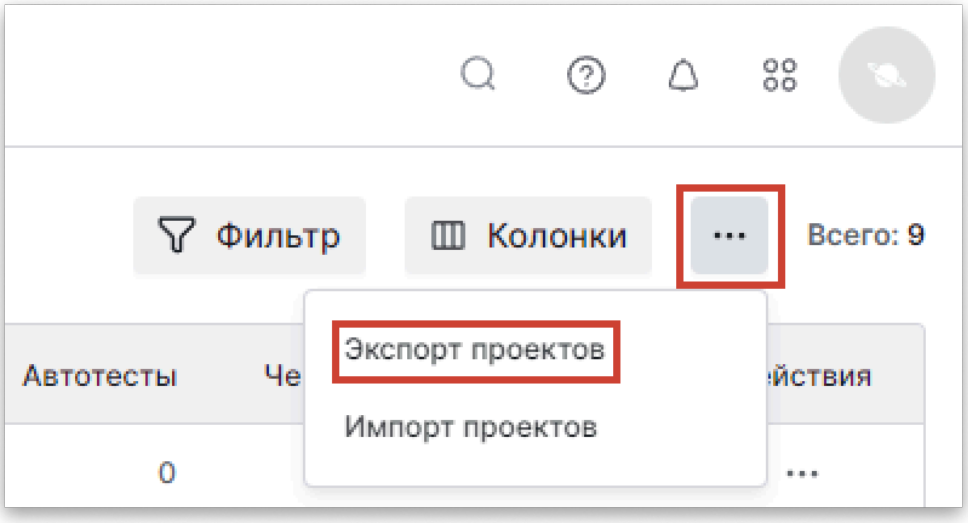
- Пользователи
- Системные настройки
- Интеграции с таск-трекерами
- Архивированные проекты
- Запросы и проектные фильтры
- Дашборды
- История изменений тест-кейсов

Экспорт проектов

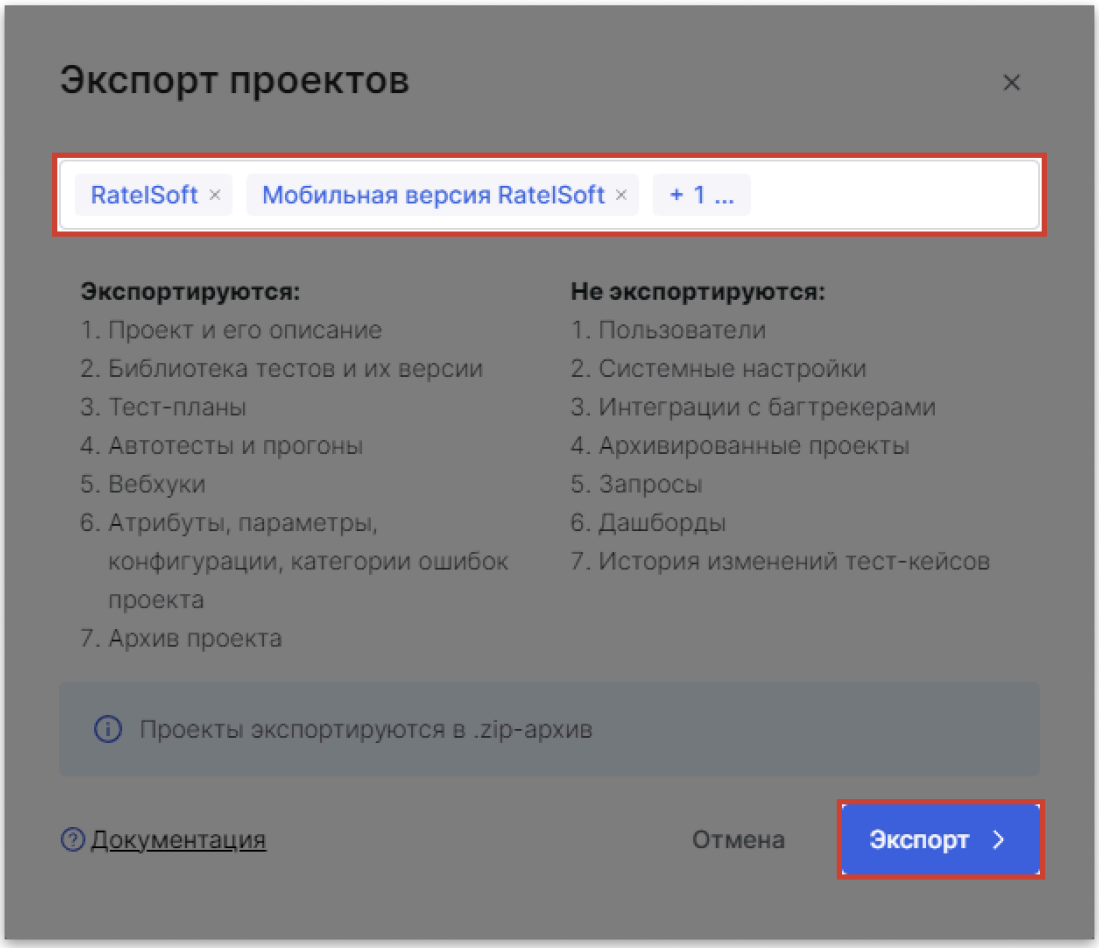
Вы можете экспортировать проекты из Test IT в формате `.zip`. Все экспортируемые проекты сохраняются в одном `.zip`-архиве. Массовый экспорт проектов доступен только при наличии действующей коммерческой лицензии.

Чтобы экспортировать проекты:

1. Откройте вкладку **Проекты**.
2. В правой части окна над списком проекта нажмите значок **Массовые действия**. В выпадающем меню выберите **Экспорт проектов**.



3. В открывшемся окне экспорта выберите проекты для импорта, используя выпадающее меню.
4. Нажмите **Экспорт**.



Отобразится сообщение об экспорте проекта.

5. **Опционально:** Отслеживайте процесс экспорта на странице Фоновых задач.
6. По окончании процесса скачайте `.zip`-архив с проектами на странице **Фоновых задач**.

Импорт проектов

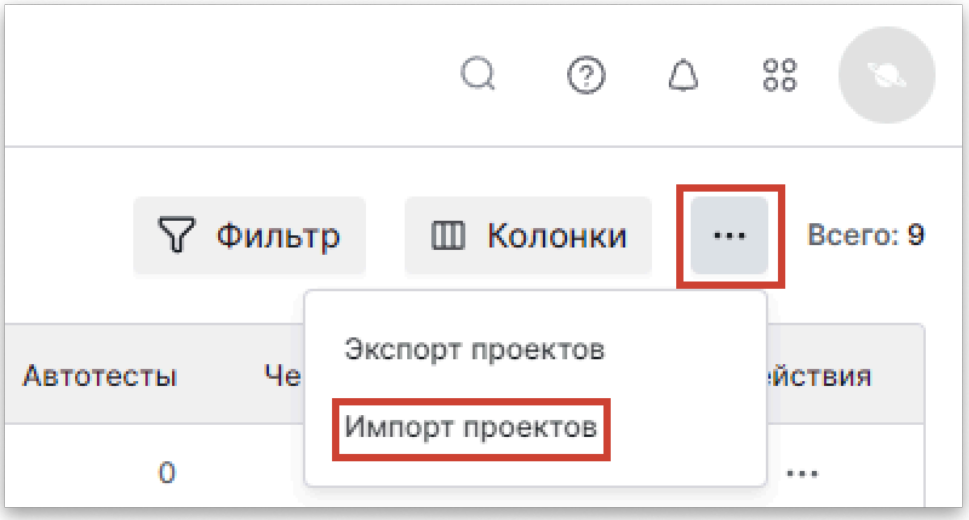
Вы можете импортировать проекты, архивированные в один `.zip`-файл. За один раз можно импортировать только один архив.

> **Внимание!**
>
> При повторном импорте проекты не будут заменены: в Test IT будут созданы новые идентичные проекты.

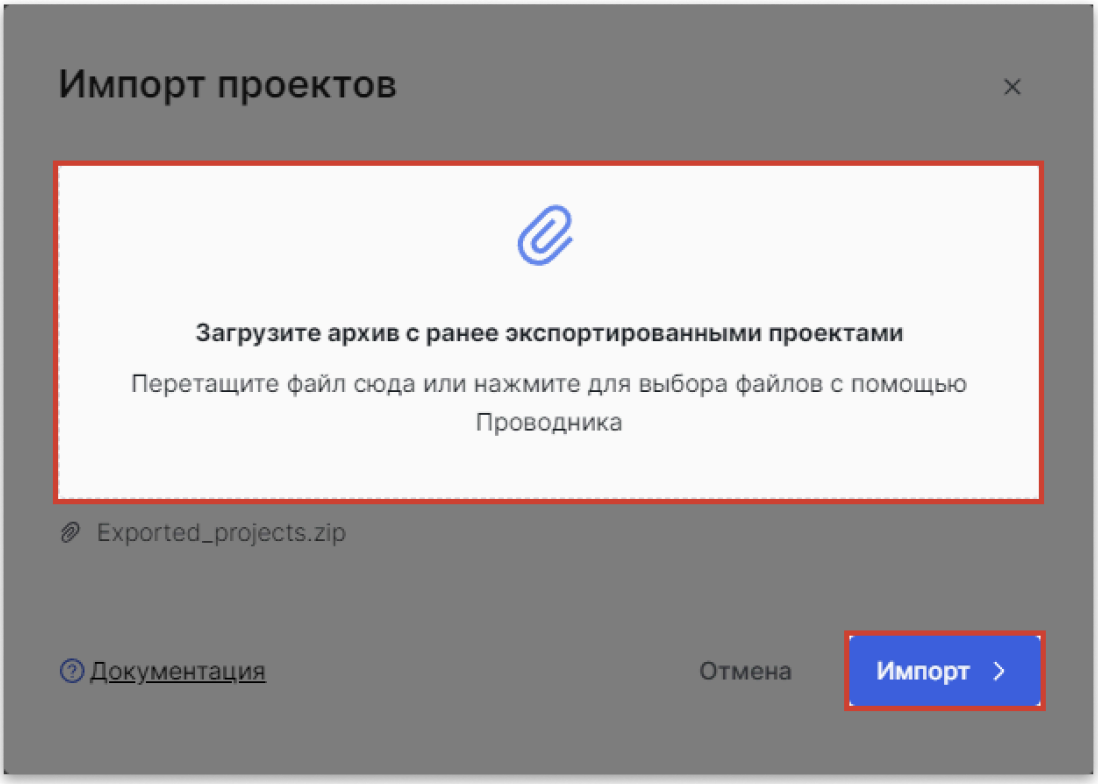
Чтобы импортировать проекты:

1. Откройте список проектов.

2. В правой части окна над списком проекта нажмите значок **Массовые действия**. В выпадающем меню выберите **Импорт проектов**.



3. В открывшемся окне импорта загрузите `.zip` -архив с проектами. Для этого перетащите архив с проектами в область для загрузки или воспользуйтесь **Проводником**.
4. **Опционально:** Чтобы заменить архив с импортируемыми проектами, загрузите новый архив. Старый архив будет автоматически удален.
5. Нажмите **Импорт**.



Отобразится сообщение об импорте проекта.

6. Закройте окно импорта проектов или дождитесь окончания процесса. По окончании импорта проекты будут доступны во вкладке **Проекты**.

Обновлено: 9/16/26, 11:33 AM

Подключение таск-трекера к проекту

Требуется настроенная интеграция

Для подключения таск-трекера к проекту предварительно требуется настроить интеграцию с одним из поддерживаемых таск-трекеров.

Test IT можно подключить к таск-трекеру и создавать задачи из интерфейса Test IT.

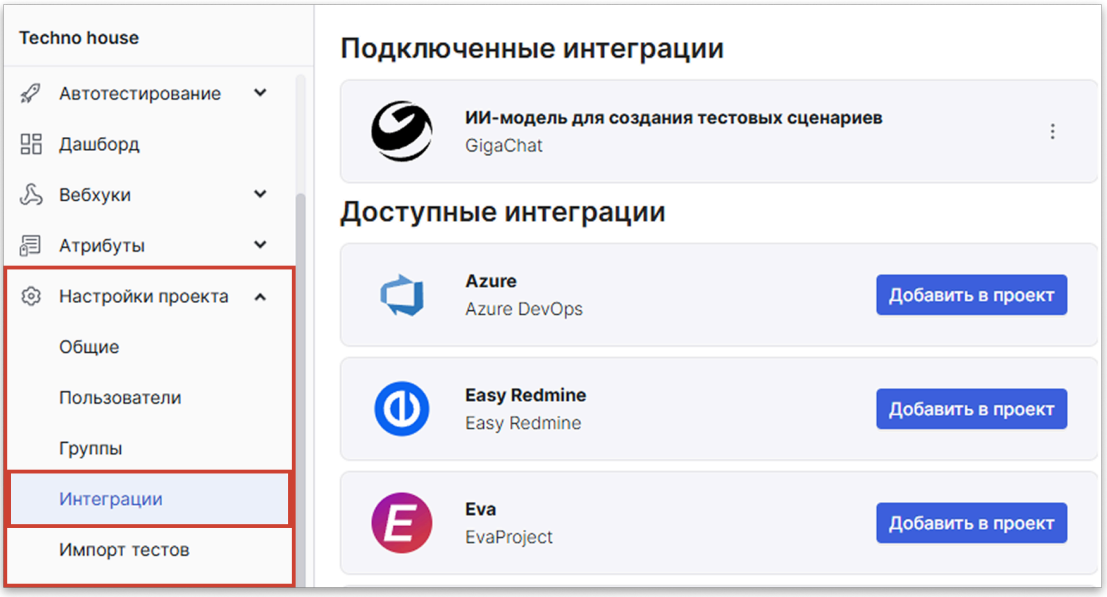
Список поддерживаемых таск-трекеров

- Azure DevOps
- ClickUp
- Easy Redmine
- EvaProject
- Jira:
  - Jira Server
  - Jira Cloud
- Kaiten
- Monday
- Redmine
- TeamStorm
- YouTrack
- Яндекс Трекер

Чтобы подключить таск-трекер к проекту Test IT:

1. Откройте проект.
2. Перейдите в раздел **Настройки проекта → Интеграции**.
   Откроется список доступных интеграций, созданных ранее в разделе

Администрирование.



3. В списке **Доступные интеграции** выберите созданную ранее интеграцию с таск-трекером и нажмите **Добавить в проект**.
4. Укажите проект в таск-трекере и типы дефектов, которые будут создаваться из интерфейса Test IT.
5. **Если вы подключаете проект к Jira**, в поле **Создание дубликатов** укажите проект Jira, который хотите использовать для дублирования тест-кейсов в Jira. Дублированные тест-кейсы смогут просматривать все, у кого есть доступ в Jira, но отсутствует доступ к Test IT. В Test IT ссылка на дубликат в Jira доступна на странице тест-поинта в секции справа.
6. Нажмите **Сохранить**.
   Интеграция появится в списке **Подключенные интеграции**.

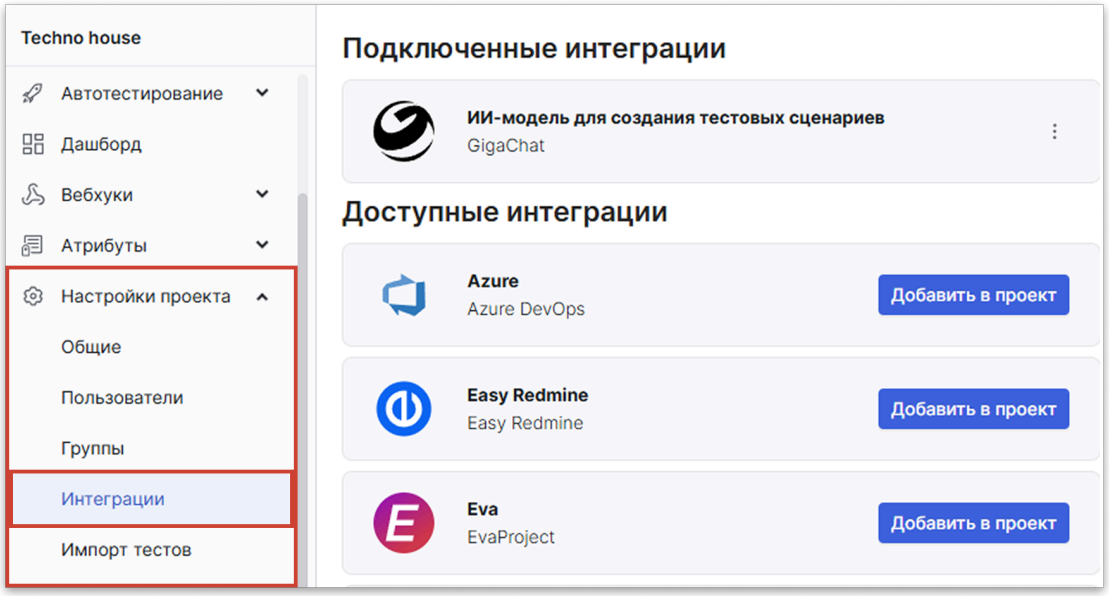
Обновлено: 6/18/26, 8:10 AM

Добавление ИИ-модели в проект

Вы можете подключать ИИ-модели к проектам Test IT и создавать тесты с помощью искусственного интеллекта по задачам из вашего таск-трекера. В 1 проект можно добавить до 3 ИИ-сервисов. Выбор сервиса осуществляется при генерации тестов.

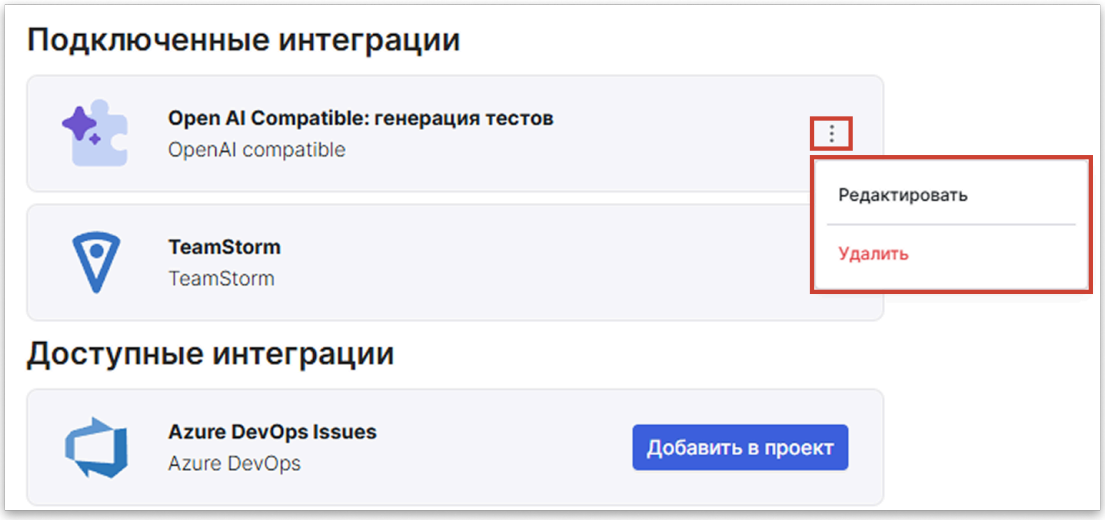
> **Требуется настроенная интеграция**
>
> Для подключения проекта требуется настроенная интеграция с ИИ-моделью. Для генерации тестов по задачам из таск-трекера необходима интеграция с таск-трекером.

1. Откройте проект.
2. В навигационном меню слева откройте **Настройки**, затем выберите **Интеграции**.
   Откроется список подключенных и доступных интеграций с ИИ-моделями и таск-трекерами.



3. В меню **Доступные интеграции** выберите созданную ранее интеграцию с ИИ-моделью.
4. Нажмите **Добавить в проект**.
   Откроется окно создания интеграций.
5. В выпадающем меню выберите модель ИИ.
6. Если в проект уже добавлена ИИ-интеграция, подтвердите замену в открывшемся окне. Можно добавить до 3 сервисов.

7. Выберите внешний сервис и языковую модель. В зависимости от выбранной модели стоимость ее использования может различаться.
8. Нажмите **Сохранить**.
   Новая интеграция появится в списке **Подключенные интеграции**.
9. **Опционально:** Чтобы изменить модель ИИ или удалить интеграцию, справа от названия интеграции нажмите значок **Действия**, затем в выпадающем меню выберите **Редактировать** или **Удалить**.



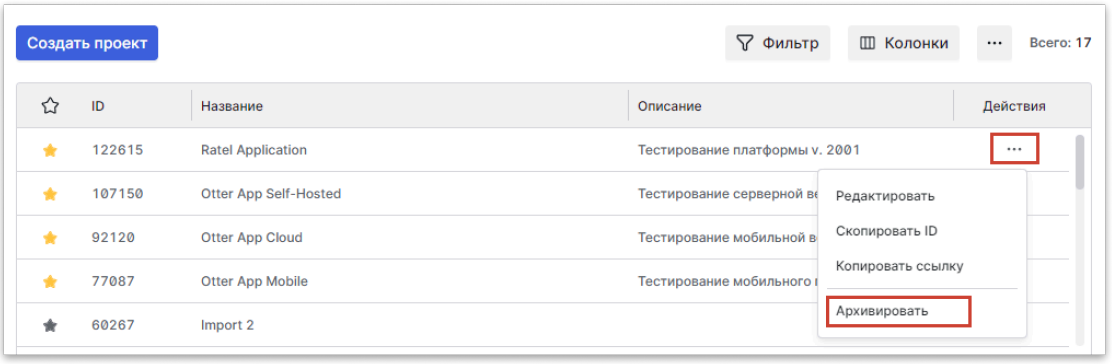
Обновлено: 4/15/26, 7:08 PM

Архивация, восстановление и удаление проектов

Вы можете архивировать проект, если он утратил актуальность. Архивные проекты недоступны для редактирования и использования. Вы можете восстановить проект из архива.

Архивация проекта

1. Перейдите на вкладку **Проекты**.
2. В таблице проектов откройте меню в колонке **Действия** и нажмите **Архивировать**.



3. Подтвердите действие в диалоговом окне.

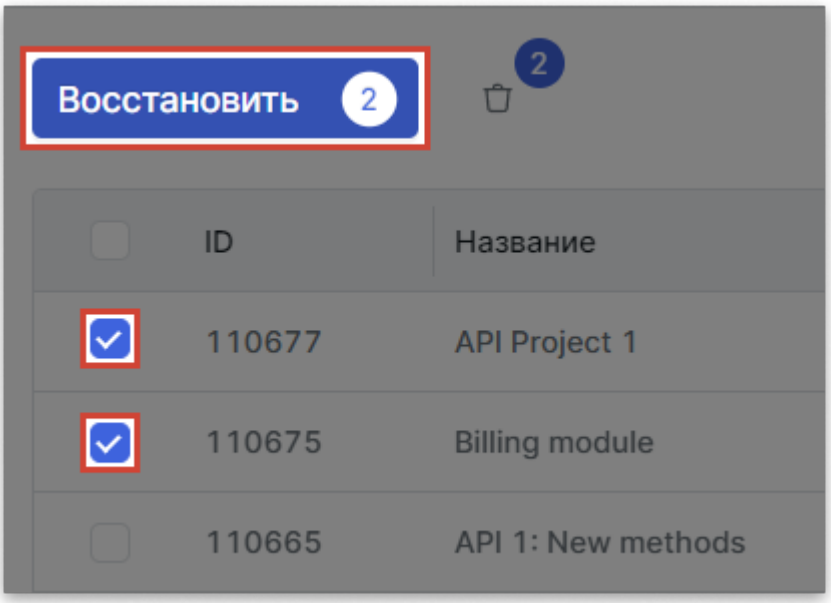
Восстановление и удаление проектов

Вы можете восстанавливать или удалять архивные проекты. Проекты, удаленные из архива, не могут быть восстановлены.

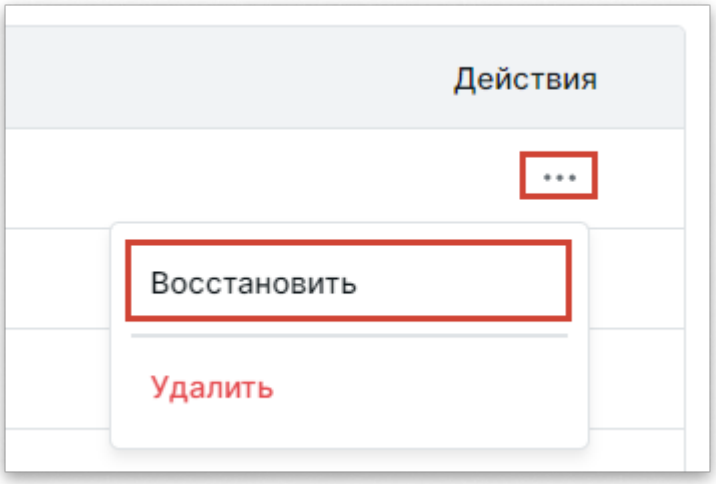
Чтобы восстановить проекты:

1. В верхней навигационной панели выберите **Архив**.
2. В списке архивных проектов выберите проекты, которые хотите восстановить, затем нажмите **Восстановить**. Чтобы выбрать все

проекты, отметьте флажком ячейку **ID**.



- **Опционально:** Чтобы восстановить отдельный проект, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.

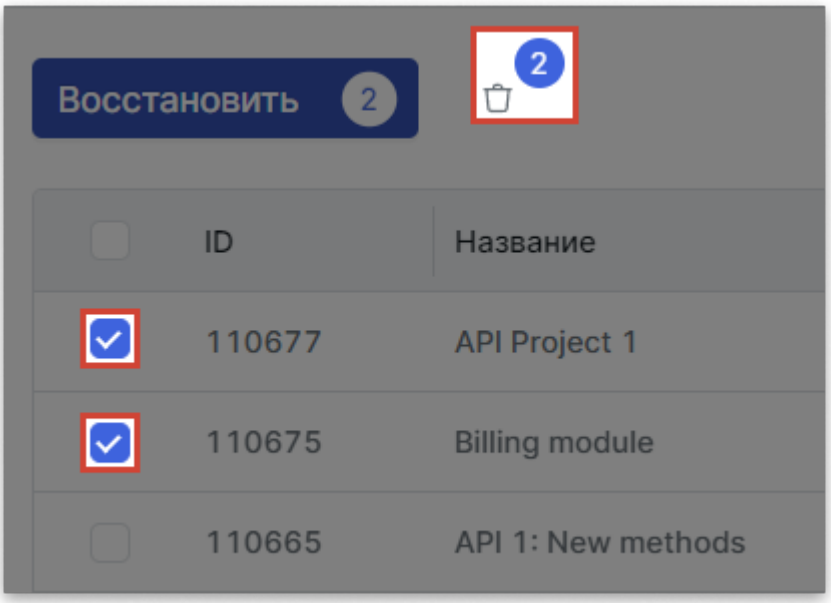


3. Подтвердите действие в диалоговом окне. Проект снова станет активным и появится в списке проектов.

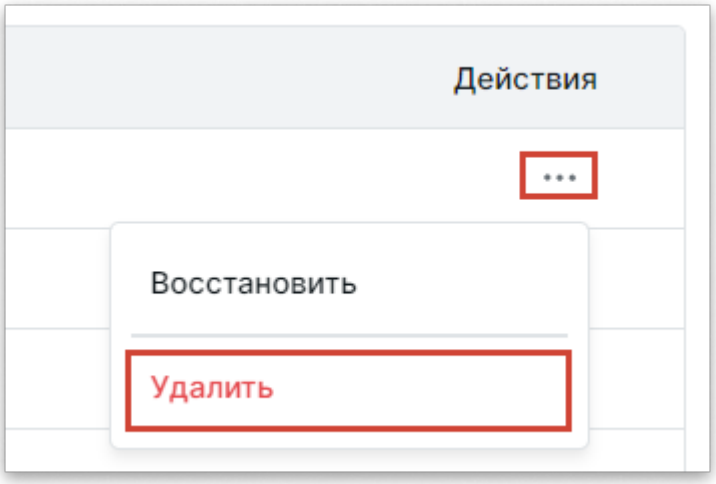
Чтобы удалить проекты:

1. В верхней навигационной панели выберите **Архив**.
2. В списке проектов выберите проекты, которые хотите удалить, затем нажмите значок **Удалить**. Чтобы выбрать все проекты, отметьте

флажком ячейку **ID**.



- **Опционально:** Чтобы удалить отдельный проект, откройте выпадающее меню в колонке **Действия** и выберите **Удалить**.



3. Подтвердите действие в диалоговом окне.

Обновлено: 4/15/26, 7:08 PM

Создание тестов

В Test IT можно создавать тест-кейсы, чек-листы и общие шаги. Это рабочие элементы, которые создаются, генерируются ИИ-моделями и импортируются в библиотеке тестов. Если у вас есть тесты в другой TMS-системе, вы можете автоматически перенести их в Test IT с помощью мигратора.

> **Используйте горячие клавиши**
>
> Для действий с рабочими элементами вы можете воспользоваться горячими клавишами.

Смотрите также: Обновление тест-кейсов на основе связанных автотестов

Типы рабочих элементов Test IT

В Test IT есть следующие виды рабочих элементов:

- Тест-кейс — ручной тест, включающий в себя предусловия и постусловия, шаги и их ожидаемые результаты. В тест-кейсе полностью описывается тестовый сценарий для той или иной функциональности.
- Чек-лист — список проверок, позволяющий добавить шаги теста, но не описывать предусловия и постусловия и ожидаемые результаты. Чек-лист можно использовать на раннем этапе разработки, пока к ней не написано полноценных тестов. Чек-лист можно преобразовать в тест-кейс.
- Общий шаг — шаг теста, который можно переиспользовать в нескольких тест-кейсах. Позволяет сэкономить время на описании тестовых сценариев с типичными шагами.

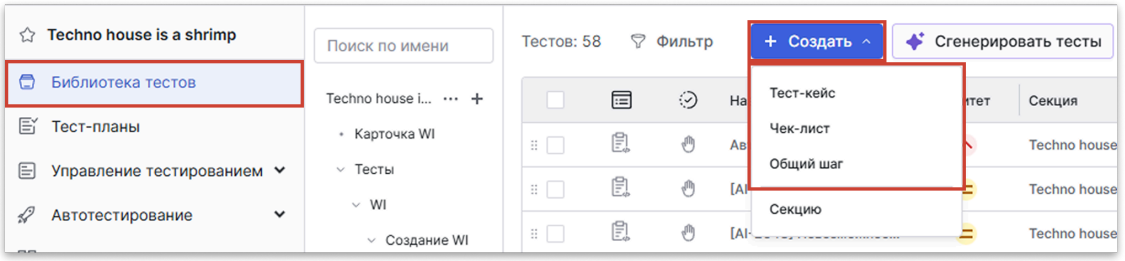
Больше информации о рабочих элементах вы найдете в сравнении типов рабочих элементов.

Создание тест-кейсов, чек-листов и общих шагов

- **Мы подготовили видео о создании тест-кейсов:**

Процесс создания тест-кейсов, чек-листов и общих шагов идентичен. Данная инструкция описывает его на примере создания тест-кейса.

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Нажмите **Создать**. В открывшемся меню выберите тип рабочего элемента (**Тест-кейс**, **Чек-лист** и **Общий шаг**).

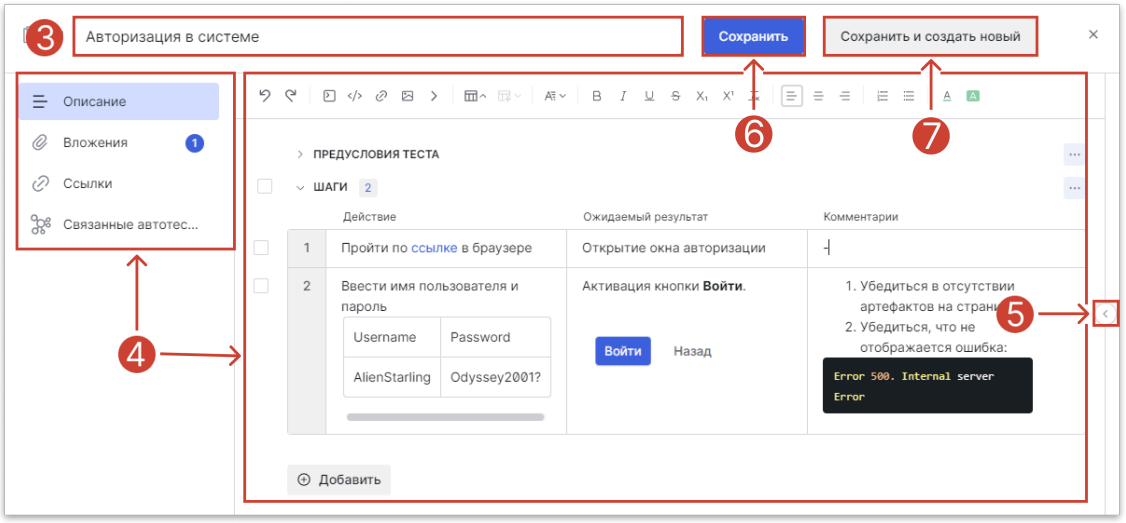


3. Введите название рабочего элемента.
4. Заполните данные рабочего элемента. Чтобы узнать больше о доступных полях для каждого типа рабочих элементов, смотрите Сравнение типов рабочих элементов.

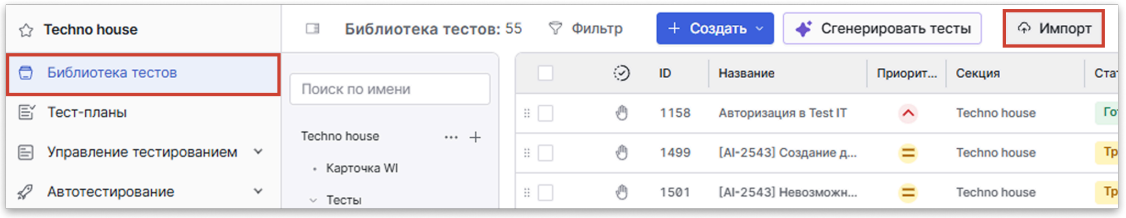
Используйте редактор:

- Вы можете редактировать текст, изменять шрифт, добавлять ссылки, изображения (через буфер обмена или методом drag & drop), блоки кода, и создавать таблицы с помощью редактора рабочих элементов. Вы также можете копировать таблицы из файлов Excel и Word и вставлять их в шаги, предусловия и постусловия рабочих элементов. Аналогичным образом можно копировать таблицы из шагов, предусловий и постусловий в Test IT и вставлять их в файлы Excel и Word.
- Вы можете воспользоваться редактором для изменения уже существующих рабочих элементов.

5. В правой части окна задайте время его прохождения, приоритет, статус готовности и значения атрибутов, добавленных в проект.
6. Нажмите **Сохранить**.
7. **Опционально:** Чтобы перейти к созданию нового теста, нажмите **Сохранить и создать новый**.



8. **Опционально:** Вы можете импортировать рабочие элементы в .xlsx, нажав значок импорта справа от кнопки **Сгенерировать тесты**.



Генерация тестов с помощью ИИ-моделей

Требуется настроенная интеграция в проекте

Для генерации тестов в проект должны быть добавлены интеграции с ИИ-моделью. Для генерации тестов по задачам из таск-трекера необходима интеграция с таск-трекером.

В Test IT вы можете создавать тест-кейсы с помощью искусственного интеллекта (ИИ). Тесты создаются по задачам в вашем таск-трекере. Из одной задачи создается несколько тестовых сценариев. ИИ-генерация платная, оплата производится провайдеру ИИ-модели.

Рекомендации по формулированию задач и запросов

Чтобы создание тестов с помощью ИИ-модели было продуктивным и безопасным, перед началом работы убедитесь, что:

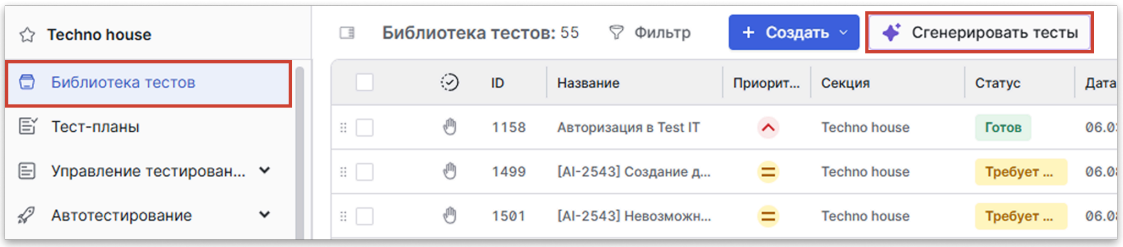
- Задачи, по которым создаются тест-кейсы, не содержат чувствительных данных, подлежащих NDA. Все данные передаются нейросети и обрабатываются ей.
- Задачи, по которым создаются тест-кейсы, имеют полное и развернутое описание и соответствуют спецификации. Изображения, вложенные файлы и комментарии к задачам в настоящее время не обрабатываются: нейросеть генерирует тест-кейсы по текстовому описанию в задаче. Специального формата для описания задач в таск-трекере нет.

Подробнее читайте в блоге Test IT .

Чтобы сгенерировать тест-кейсы:

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. **Опционально:** Откройте секцию, в которую хотите поместить сгенерированные тест-кейсы.

3. Нажмите **Сгенерировать тесты**.



4. Введите данные для генерации тест-кейсов. Воспользуйтесь одним из способов или их сочетанием:
   - В поле **Номер задачи** введите номер задачи из таск-трекера (например, *TS-2567*).
   - В поле **Пользовательский контекст** укажите контекст в качестве дополнения к используемой задаче или как самостоятельный запрос (номер задачи необязателен).
5. В поле **Сервис ИИ** выберите сервис, подключенный к проекту, для генерации тестов.
6. ₽ **Опционально:** Используя слайдеры в правой верхней части окна, укажите нужное количество тест-кейсов и температуру.

   > **Что такое температура ИИ**
   >
   > Температура — это параметр искусственного интеллекта, определяющий случайность генерируемого ответа. От температуры зависит предсказуемость или вариативность поведения ИИ-модели. В Test IT можно выбрать температуру от 0 до 1, где
   >
   > - 0 — результат, максимально приближенный к тексту задачи
   > - 1 — вариативный и менее предсказуемый результат

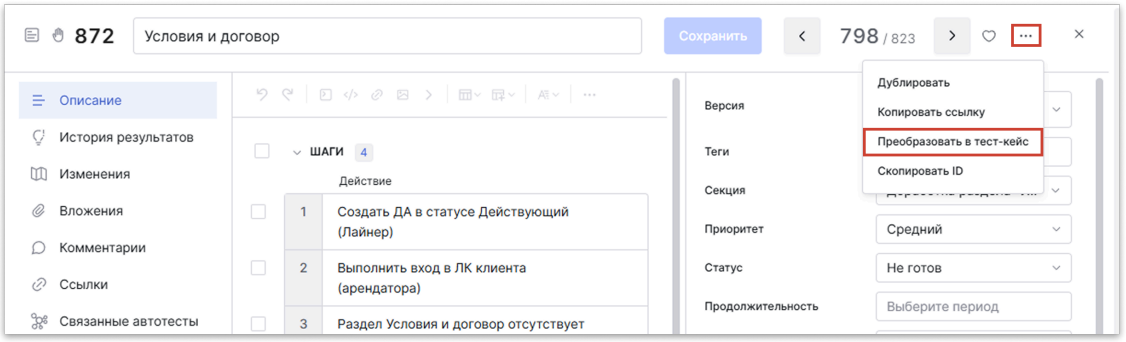
7. Нажмите **Сгенерировать** и дождитесь окончания процесса.
8. Просмотрите созданные тест-кейсы. Если тесты не соответствуют вашим требованиям, повторно нажмите **Сгенерировать**. По вашей задаче будут сгенерированы новые тесты, а предыдущие будут потеряны.
9. Если в проекте есть обязательные атрибуты, заполните их.
10. **Опционально:** Отредактируйте созданные тесты, нажав значок редактирования справа от названий тестов или их шагов.
11. **Опционально:** Выберите тесты для сохранения, используя чек-боксы справа от названий тестов.
12. Нажмите **Сохранить и добавить тесты**. Чтобы отказаться, закройте окно и подтвердите действие.
    После сохранения тест-кейсы будут добавлены в библиотеку. Вы сможете отредактировать тесты вручную.

Создание тест-кейсов из чек-листов

> **Учитывайте версию рабочего элемента**
>
> При возвращении к предыдущей версии рабочего элемента тест-кейс конвертируется обратно в чек-лист.

Если вам необходимо расширить описание чек-листа, вы можете конвертировать его в полноценный тест-кейс. Для этого:

1. Откройте проект.
2. В разделе **Библиотека тестов** откройте чек-лист.
3. Нажмите значок **Действия** в правом верхнем углу и выберите **Преобразовать в тест-кейс**.



   В поле описания появятся предусловия и постусловия теста, а также поля ожидаемых результатов, которых не было в чек-листе.
4. Заполните необходимые поля данными.
5. Нажмите **Сохранить**.

Обновлено: 8/12/26, 4:43 PM

Сравнение типов рабочих элементов

Test IT поддерживает три типа рабочих элементов — тест-кейсы, чек-листы и общие шаги. В таблице ниже приведено сравнение рабочих элементов по действиям, которые можно с ними выполнить.

Действие	Тест-кейс	Чек-лист	Общий шаг
Проставление ожидаемого результата	Да	Нет	Да
Отображение колонок **Тестовые данные** и **Комментарии**	Да	Нет	Да
Отслеживание истории результатов	Да	Да	Нет
Использование параметров	Да	Да	Да
Указание предусловий/постусловий	Да	Нет	Нет
Преобразование в тест-кейс	Нет	Да	Нет
Автоматизация рабочих элементов, просмотр и поиск связанных элементов	Да	Да	Нет
Просмотр и поиск связанных тест-кейсов	Нет	Нет	Да
Добавление общего шага	Да	Нет	Нет

Обновлено: 5/20/26, 12:33 PM

Импорт тестовой документации из XLS и XLSX

- **Для этого раздела мы подготовили видео:**

Вы можете импортировать тест-кейсы в систему Test IT, загружая файлы `.xls` и `.xlsx`, содержащие необходимые поля и их значения. Это позволяет сэкономить время на создание тест-кейсов. Например, если вам требуется создать большое количество тест-кейсов с одинаковыми полями. Чтобы скачать образцы файлов для импорта, вы можете экспортировать тест-кейсы из Test IT.

Максимальный размер загружаемых файлов:

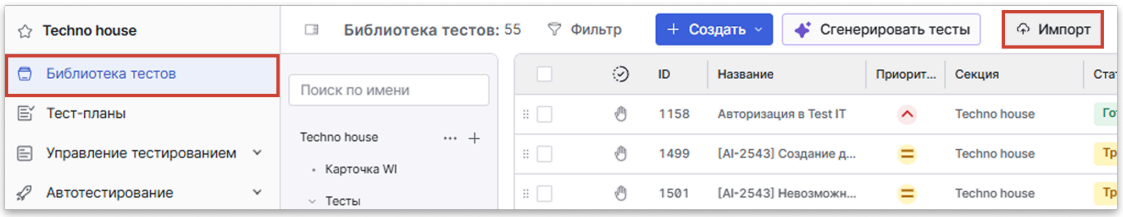
- В тарифах Базовый и Стандарт — 50 МБ
- В тарифе Про — 1 ГБ

> **Импорт пользовательских значений**
>
> Если импортируемый файл содержит пользовательские значения (колонки), которые требуется перенести в тест-кейсы, перед началом импорта создайте проектные атрибуты, соответствующие значениям, которые вы хотите перенести.

Чтобы импортировать тест-кейсы:

1. Откройте проект. По умолчанию откроется библиотека тестов.

2. Нажмите значок импорта рядом с кнопкой **Создать**.



3. **Опционально:** В навигационном меню слева откройте **Настройки проекта**, выберите **Импорт** и нажмите **Импортировать**.
4. Выберите способ импорта **XLSX**.
5. Перетащите Excel-файл в поле **Файл с данными для импорта** или загрузите его с помощью проводника, нажав **Обзор**. Нажмите **Далее**.
6. Если у вас есть конфигурационный файл `.json`, выберите опцию **Использовать конфигурационный файл**. Это позволит настроить соответствия полей автоматически. Если у вас нет конфигурационного файла, выберите опцию **Создать новую конфигурацию**. Нажмите **Далее**.
7. Укажите строку начала. Считывание наполнения Excel-файла начнется с указанной вами строки.
8. Настройте форматирование текста при импорте, выбрав одну из опций:
   - Сохранить исходное форматирование текста
   - Очистить форматирование и отображать HTML-теги
   - Очистить форматирование и отображать только текст

   Нажмите **Далее**.
9. Настройте соответствие полей импортируемого файла и тест-кейса в системе Test IT. В правой колонке требуется выбрать необходимое поле тест-кейса в Test IT или оставить **Не отображать** в случае избыточных данных. Если был загружен конфигурационный файл, поля будут настроены автоматически. Нажмите **Далее**.
10. Для атрибута **Варианты на выбор** необходимо настроить соответствие вариантов в импортируемом файле и в Test IT. Для этого укажите значение атрибута, которое соответствует значению столбца, затем нажмите **Далее**.
11. Выберите секцию в библиотеке тестов, в которую вы хотите импортировать тесты. Воспользуйтесь выпадающим списком в поле **Секция**. Нажмите **Далее**.
12. **Опционально:** Сохраните конфигурационный файл, чтобы сохранить настроенные вами соответствия полей. В дальнейшем вы сможете использовать его для импорта тестов.
13. Нажмите **Импортировать**.

> **Шаблон файла для импорта**
>
> Вы можете скачать шаблон .xlsx-файла для импорта тестов.
> Этот шаблон содержит большее количество атрибутов, чем обычно требуется. Для корректной работы рекомендуем создать свои атрибуты.

Обновлено: 8/12/26, 4:43 PM

Версионирование тестов

Тарифные ограничения

Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

- **Для этого раздела мы подготовили видео:**

В Test IT доступно переключение между версиями тест-кейсов, чек-листов и общих шагов (рабочих элементов). После сохранения изменений в тесте фиксируется его текущая версия. При внесении изменений в сохраненный тест автоматически создается новая версия. Вы можете переключаться между версиями теста и возвращаться к предыдущим версиям.

Изменения тест-поинтов, связанные с версионированием тест-кейсов

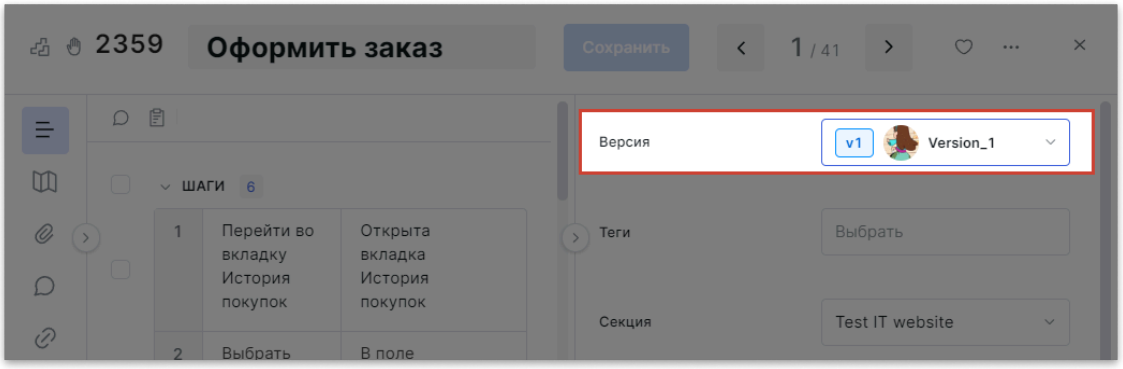
При изменении актуальной версии тест-кейса, изменяется и соответствующий тест-поинт в тест-плане кроме следующих случаев:

- **Тест-поинт пройден**. Для актуализации версии требуется сбросить результат тест-поинта.
- **В тест-плане используется тест-поинт с различными конфигурациями**. Чтобы актуализировать версию, нужно сбросить результаты данного тест-поинта для всех используемых конфигураций.
- **Тест-план заблокирован**. Для актуализации версии требуется разблокировать тест-план.

Смотрите также: Обновление тест-кейсов на основе связанных автотестов

Переключение между версиями

1. Откройте рабочий элемент, для которого хотите изменить версию.
2. На панели информации о рабочем элементе в поле **Версия** выберите необходимое значение.



Выбранная вами версия откроется в режиме просмотра.

Возврат к предыдущим версиям

1. На панели информации о рабочем элементе в поле **Версия** выберите версию, которую хотите сделать актуальной.
2. Сделайте данную версию актуальной. При этом актуализированная версия отображается как новая. Например, если вы, находясь в версии 2, вернулись к изменениям из версии 1, данные изменения применятся к версии 3. Последняя версия рабочего элемента по умолчанию является актуальной.

При создании новой версии тест-кейса копируются шаги, теги, атрибуты, вложения и ссылки.

Отслеживание изменений

Вы можете отслеживать изменения, внесенные в каждую из версий рабочего элемента от его создания до последней актуальной версии. Историю изменения теста от самого его создания до актуальной версии можно просмотреть в журнале. Чтобы открыть журнал изменений:

1. Откройте рабочий элемент.
2. Перейдите в раздел **Изменения** в навигационном меню слева.

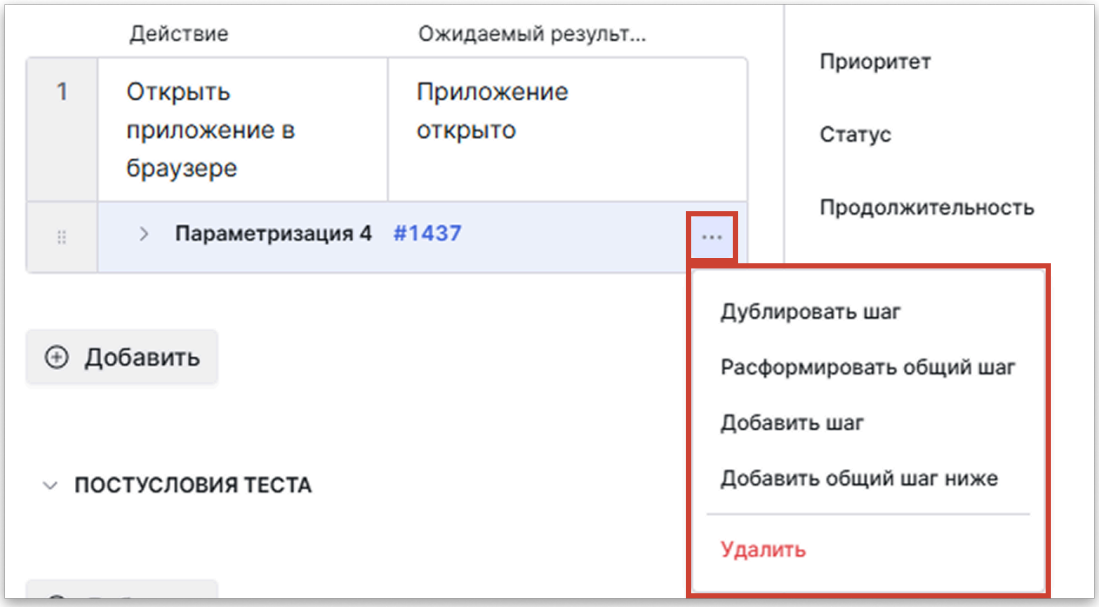
В журнале отображается, когда и кем было внесено изменение, а также значения до и после изменений. Удаления символов в текстовых полях выделяются красным цветом с перечеркиванием, а добавления символов — зеленым.

Обновлено: 8/18/26, 4:54 PM

Добавление общего шага в тест

Если вам необходимо добавить повторяющиеся действия в несколько тест-кейсов, вы можете воспользоваться общими шагами. Общие шаги являются одним из типов рабочих элементов системы Test IT и хранятся в библиотеке, как и тест-кейсы. Если у вас еще нет общих шагов, вы можете создавать их как рабочие элементы или преобразовать шаги теста в общий шаг. Для добавления общего шага в тест-кейс:

1. Откройте проект, перейдите в раздел **Библиотека тестов**.
2. Создайте новый тест-кейс или откройте существующий.
3. В пред-/постусловиях или шагах теста нажмите **Добавить**. Отобразится пустая строка для нового шага.
4. В меню **Действия** в правой части строки шага нажмите **Добавить общий шаг ниже**.



5. Выберите нужный общий шаг из списка или найдите его с помощью строки поиска.
6. Нажмите **Добавить**.

Общий шаг отображается на панели шагов тест-кейса. Вы можете:

- Развернуть общий шаг, чтобы ознакомиться с действиями, из которых он состоит. Чтобы развернуть общий шаг, нажмите на него.
- Расформировать общий шаг, чтобы все действия из общего шага отображались как шаги тест-кейса. Вы можете вносить в расформированные шаги изменения, в то время как в других тест-

кейсах данный общий шаг останется неизменным. Чтобы расформировать общий шаг, нажмите **Расформировать общий шаг** в выпадающем меню в правой части строки общего шага.

- Дублировать общий шаг. Нажмите **Дублировать общий шаг** в выпадающем меню в правой части строки общего шага.
- Удалить общий шаг. Нажмите **Удалить** в выпадающем меню в правой части строки общего шага.

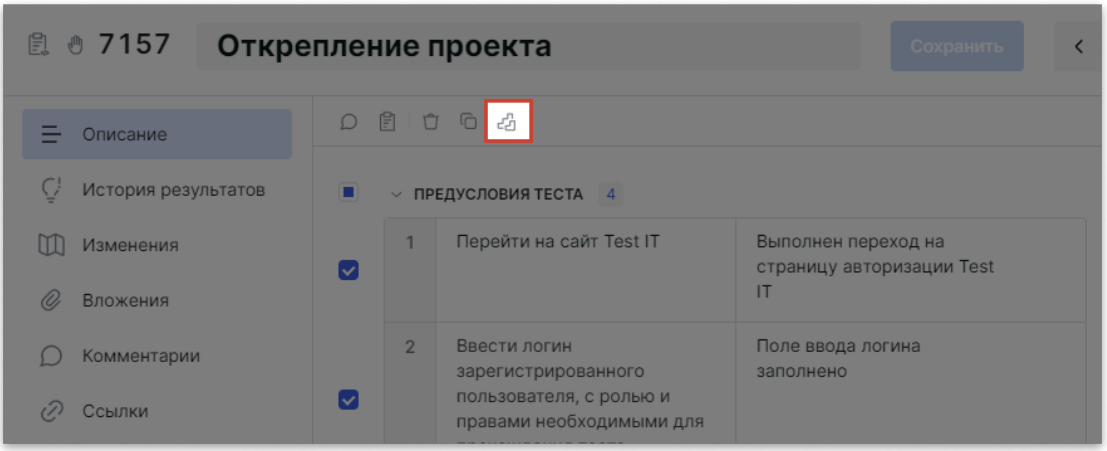
Обновлено: 6/25/26, 1:39 PM

Выделение шагов теста в общий шаг

Общий шаг — это шаг теста, который можно переиспользовать в нескольких тест-кейсах. Позволяет сэкономить время на описании тестовых сценариев с типичными шагами.

Помимо создания общего шага как рабочего элемента в библиотеке тестов вы можете создать общий шаг на основе уже написанных вами шагов тест-кейса. Для этого:

1. Откройте карточку тест-кейса, на основе шагов которого хотите создать общий шаг.
2. Выберите шаги, которые хотите объединить в общий шаг.
3. В появившемся меню действий нажмите **Создать общий шаг**.



4. В открывшемся окне введите название нового общего шага.
5. **Опционально:** Если вы не хотите добавлять создаваемый общий шаг в тест-кейс, шаги которого вы используете как основу для общего шага, поставьте флажок **Оставить исходные шаги после создания общего шага**. В таком случае общий шаг сохранится в библиотеку тестов, но не будет добавлен в исходный тест-кейс. Если не поставить данный флажок, созданный общий шаг заменит исходные шаги тест-кейса.
6. Нажмите **Создать общий шаг.**

Чтобы узнать больше о работе с общими шагами в описании тест-кейса, смотрите Добавление общего шага в тест-кейс.

Обновлено: 4/15/26, 7:08 PM

Добавление параметров в тесты

Вы можете добавлять параметры в ручные тесты. Если вы укажете в тест-кейсе параметр и несколько его значений, а затем добавите его в тестовый набор, в тест-плане получится несколько тест-поинтов по количеству указанных значений. Например, для тестов по авторизации в сервис можно создать параметры `login` и `password` с несколькими возможными значениями для каждого из них.

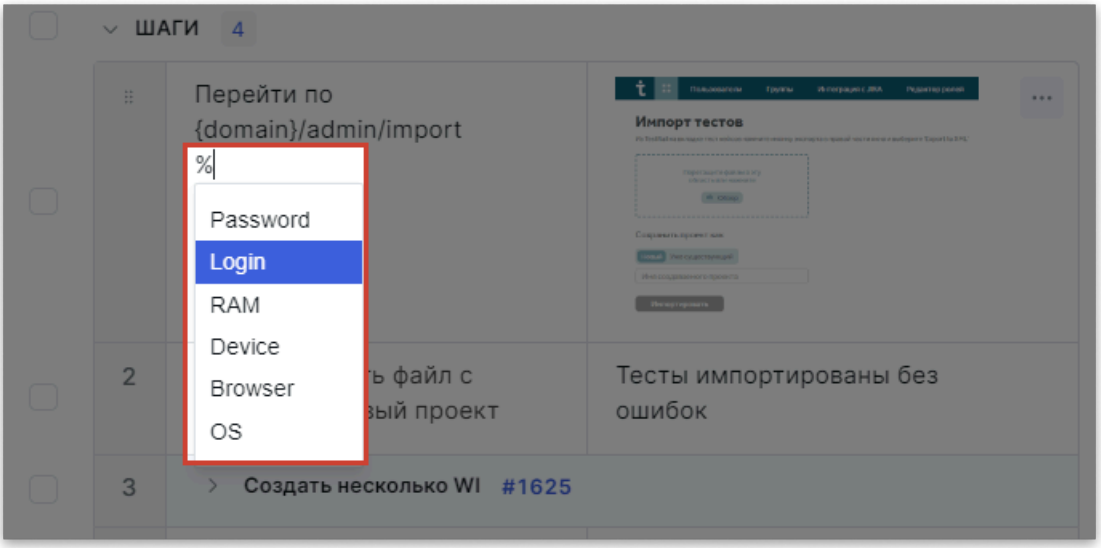
Параметры отображаются в нижней части окна рабочего элемента.

Особенности работы с параметрами в Test IT

- **Необходимо указать проекты**, в которых параметр может быть использован. Если вы создали параметр и не обнаружили его в проекте, требуется отредактировать параметр в настройках проектов, добавив нужный проект.

Чтобы добавить параметр ручного теста:

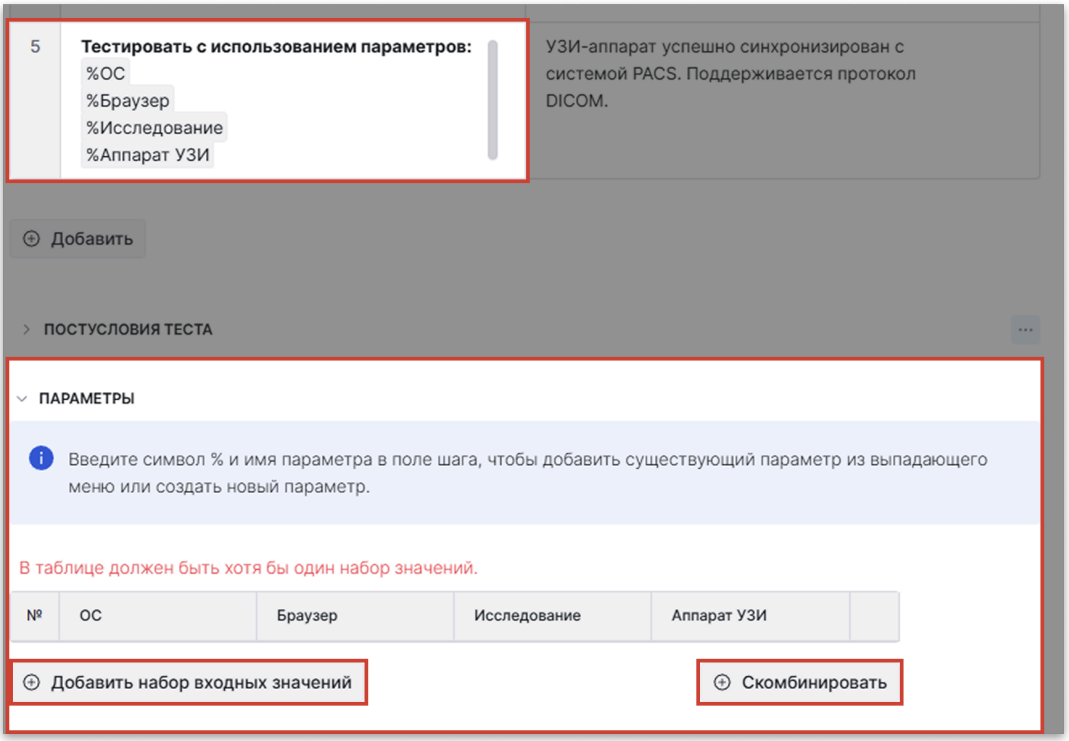
1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Создайте или откройте рабочий элемент, в который хотите добавить параметр.
3. Введите `%` в поле редактирования шага, чтобы указать параметр. Например, для теста по авторизации вы можете добавить параметры в шаги **Ввести логин** и **Ввести пароль**.



Вы можете:

- Выбрать уже существующий параметр из выпадающего списка.
- Создать новый параметр. Для этого продолжайте вводить название параметра после `%` и нажмите **Создать**. В открывшемся окне создания параметра укажите его возможные значения, затем нажмите **Сохранить**. Отредактируйте входные значения в разделе **Настройки проектов → Параметры**. Если вы удалите `%` из поля шага теста, параметр будет удален из теста, но сохранится в системе.

4. В секции **Параметры** в нижней части окна теста добавьте наборы значений параметров, с которыми тест будет пройден. Для этого выполните одно из действий:
   - Нажмите **Добавить набор входных значений** и добавьте нужные значения.
   - ₽ Нажмите **Скомбинировать**, затем в открывшемся окне выберите значения параметров и нажмите **Скомбинировать**.



5. Сохраните изменения в тесте.

Дополнительные ресурсы

- Создание тест-плана
- Выполнение тест-плана

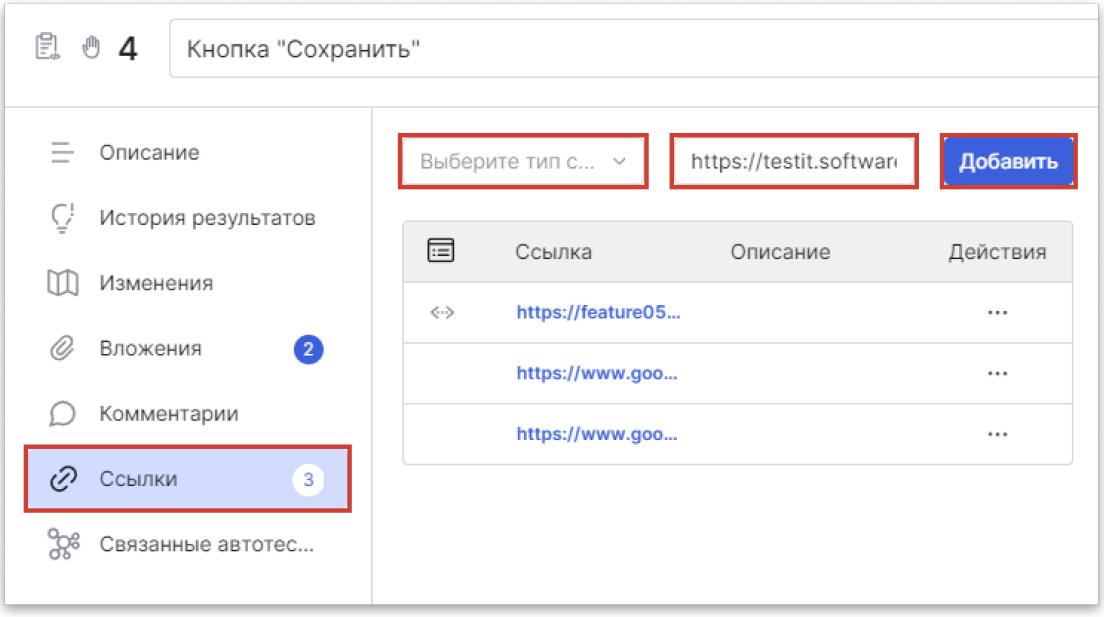
Обновлено: 8/17/26, 5:43 PM

Добавление ссылок в тесты

Вы можете добавлять в рабочие элементы ссылки на задачи, дефекты и связанные рабочие элементы, а также отслеживать статус связанных задач и дефектов. Если тест сгенерирован ИИ-моделью, ссылка на задачу-источник добавляется автоматически.

Добавление пользовательских ссылок

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте рабочий элемент, в который хотите добавить ссылку.
3. В навигационном меню слева перейдите в раздел **Ссылки**.



4. Введите тип (опционально) и URL ссылки, которую хотите добавить в рабочий элемент.
5. Нажмите **Добавить**. Ссылка отобразится в таблице.

Вы можете редактировать созданные ссылки и добавлять к ним описание. Чтобы редактировать ссылку:

1. Откройте меню в колонке **Действия**.
2. Нажмите **Редактировать**.
3. Вы можете добавить или изменить:
   - URL ссылки

    - Заголовок ссылки
    - Описание ссылки
    - Тип ссылки (**Связанный**, **Заблокирован**, **Дефект**, **Задача**, **Требование**, **Репозиторий**)
4. Нажмите **Сохранить**.

Отслеживание задач в Jira через ссылку

Вы можете добавлять ссылки на задачи в Jira. Если вы настроили интеграцию с Jira и подключили ее к вашему проекту Test IT, вы можете навести курсор на добавленную вами ссылку и отследить во всплывающем окне следующую информацию:

- Тип задачи
- Статус задачи
- Приоритет
- Исполнителя

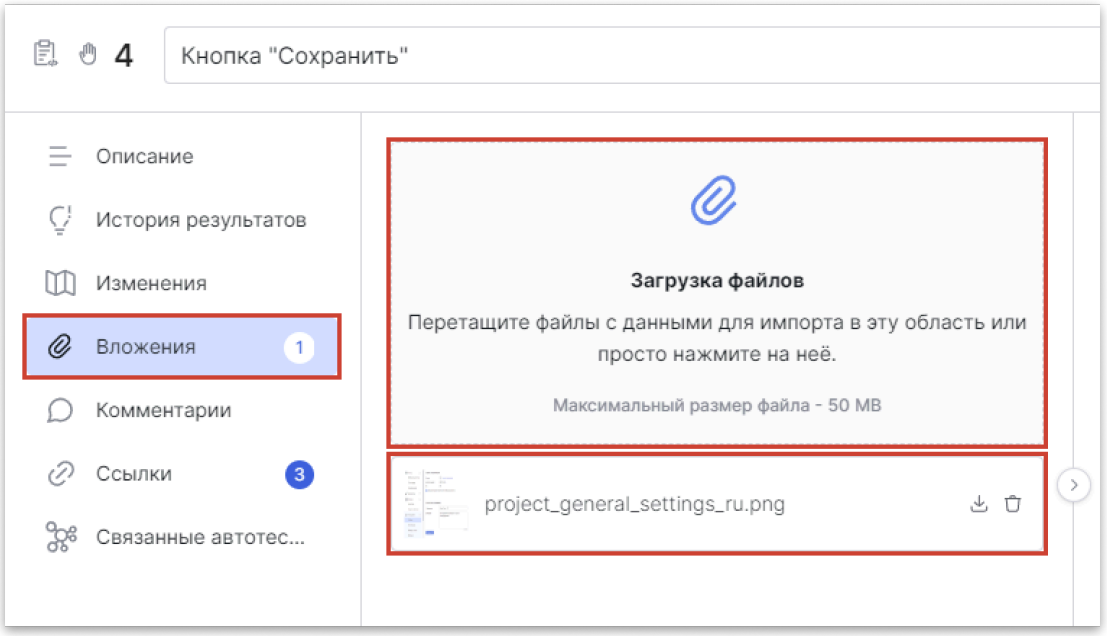
Обновлено: 4/15/26, 7:08 PM

Работа с вложениями тестов

Вы можете добавлять вложения к рабочим элементам или к конкретным шагам тестов. Вы можете перетащить изображение в поле для вложений, вставить его из буфера обмена или добавить с помощью проводника ОС.

Добавление вложений в тест

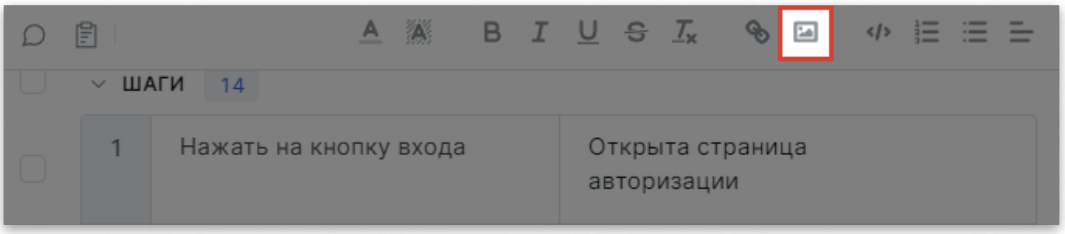
1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте рабочий элемент, в который хотите добавить вложение.
3. В окне редактирования рабочего элемента перейдите в раздел **Вложения**.



4. Перетащите вложение в поле загрузки или нажмите на поле загрузки, чтобы загрузить вложение с помощью проводника. Прикрепленные вложения отображаются под полем загрузки.
5. **Опционально:** Чтобы скачать вложение, нажмите значок загрузки на вложении.
6. **Опционально:** Чтобы удалить вложение, нажмите значок корзины на вложении, в открывшемся окне подтвердите действие.

Добавление вложений в шаги теста

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте рабочий элемент, в шаг которого хотите добавить вложение. По умолчанию откроется раздел, содержащий поля предусловий и постусловий, шагов теста и ожидаемых результатов.
3. Чтобы добавить вложение для конкретного шага:
   - Перетащите изображение в поле шага.
   - Вставьте вложение из буфера обмена в нужный вам шаг.
   - Используйте значок картинки на панели форматирования, чтобы добавить в шаг изображение, используя проводник.

   Добавленные к шагу вложения будут отображаться в разделе **Вложения**. Вы можете скачать или удалить файлы, прикрепленные к рабочему элементу.
4. **Опционально:** Чтобы удалить вложение, нажмите значок корзины на вложении, в открывшемся окне подтвердите действие.

Обновлено: 4/15/26, 7:08 PM

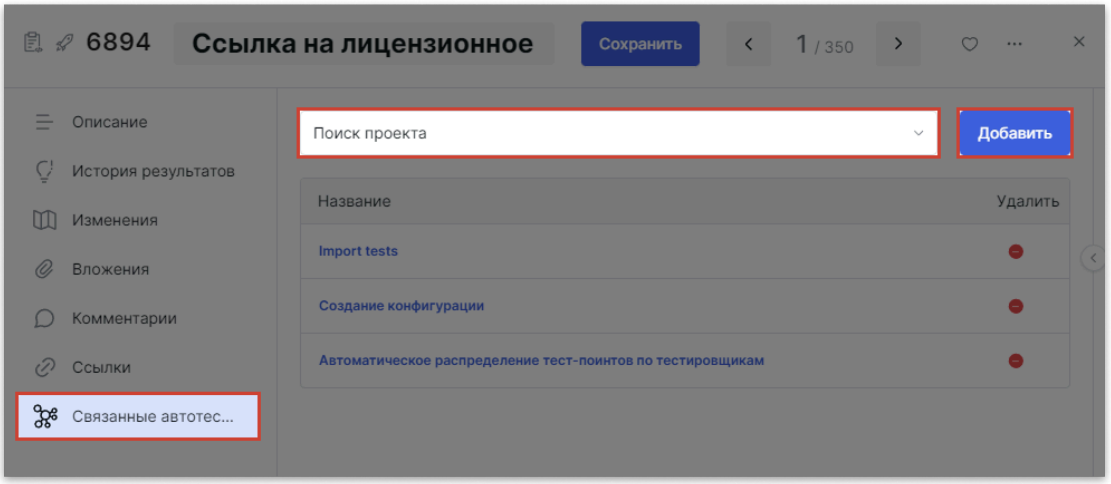
Автоматизация ручных тестов

Вы можете привязать автотест к тест-кейсу или чек-листу в библиотеке, чтобы обеспечить прозрачность покрытия тестами и связать тесты одной функциональности, хранящиеся в разных источниках (тест-кейсы хранятся в библиотеке тестов, а автотесты во внешнем репозитории)(например, IDE-системе). Тест-кейс с привязанным автотестом считается автоматизированным во время выполнения тест-плана.

Если тест-кейс связан с автотестом 1 к 1, его можно обновлять, заменяя его содержимое метаданными связанного автотеста.

Чтобы привязать автотест к тест-кейсу:

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте тест-кейс.
3. Перейдите в раздел **Связанные автотесты**.
4. В выпадающем меню выберите автотест.



5. Нажмите **Добавить**.
6. Нажмите **Сохранить**.

После связывания тест-кейса с автотестом система считает тест-кейс автоматизированным. При выполнении тест-плана автоматизированные тест-кейсы позволяют запускать автотесты прямо из тест-плана. В библиотеке тестов такой тест-кейс помечается специальным значком.

Смотрите также: Обновление тест-кейсов на основе связанных автотестов

Обновлено: 4/22/26, 2:57 PM

Экспорт тестов

Test IT поддерживает экспорт тестов. Вы можете экспортировать тесты в `.xlsx` или конвертировать рабочие элементы из библиотеки тестов в версию для печати.

Экспорт тестов в XLSX

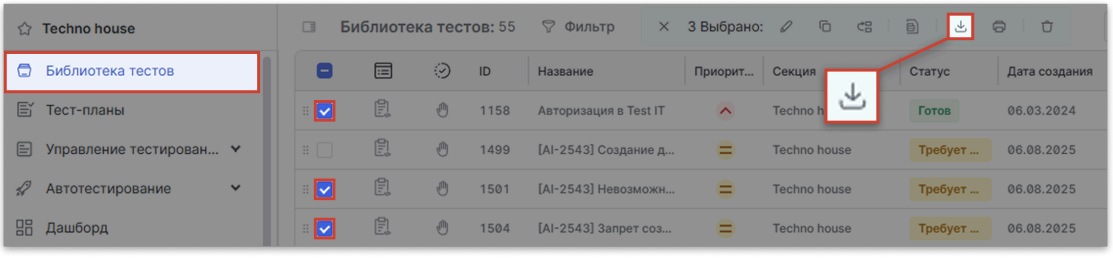
Вы можете экспортировать созданную вами в Test IT тестовую документацию в `.xlsx`.

Test IT позволяет экспортировать:

- ID рабочих элементов
- Шаги тестов с ожидаемыми результатами
- Входные параметры
- Все поля описания теста с пользовательскими атрибутами
- Директория секции
- Ссылки на тесты из библиотеки тестов

Чтобы экспортировать тесты:

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Выберите рабочие элементы, которые хотите экспортировать.
3. Нажмите на значок экспорта.



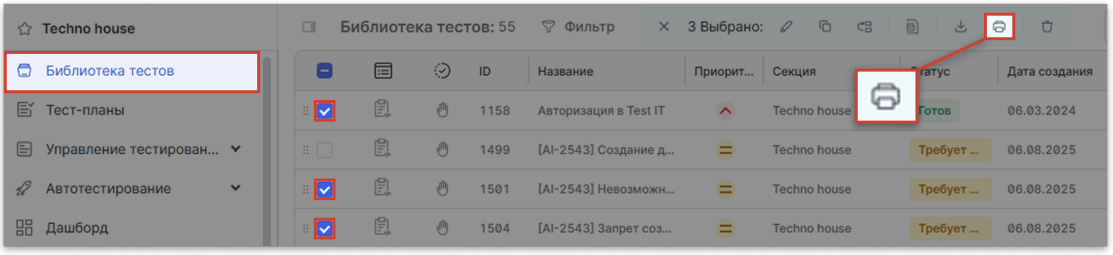
Файл с экспортированными тестами загрузится на ваше устройство. Чтобы узнать больше о том, как импортировать тестовую документацию из Excel-файлов в Test IT, смотрите Импорт тестовой документации из XLS и XLSX.

Вы можете проверить статус экспорта в разделе Фоновые задачи.

Версия тестовой документации для печати

Вы можете конвертировать рабочие элементы из библиотеки тестов в документ для печати (поддерживается формат PDF).

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Выберите рабочие элементы, которые хотите распечатать.
3. Нажмите на значок печати.



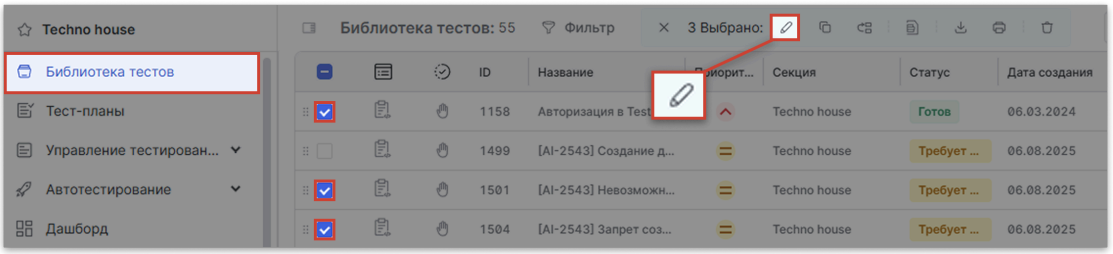
   В новой вкладке откроется документ с рабочими элементами, конвертированными в версию для печати.

Обновлено: 4/15/26, 7:08 PM

Массовое изменение тестов

Вы можете вносить изменения в несколько рабочих элементов одновременно.

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Выберите тесты, в которые хотите внести изменения.
3. Нажмите на значок массового изменения.



   Откроется окно массового изменения рабочих элементов.
4. Вы можете вносить в рабочие элементы следующие изменения:
   - Вкладка **Настройка тестов** позволяет изменить статус, приоритет и время прохождения выбранных тестов.
   - Вкладка **Пользовательские атрибуты** позволяет изменить значения атрибутов, добавленных в выбранные рабочие элементы.
   - Вкладки **Теги**, **Вложения** и **Ссылки** позволяют выбрать действие, которое вы хотите выполнить — **Добавить**, **Заменить**, **Удалить** и **Очистить**. После выбора действия вы можете выбрать теги, вложения или ссылки, с которыми хотите выполнить выбранное действие.
5. Нажмите **Сохранить**.

После сохранения изменения применяются ко всем выбранным рабочим элементам.

Смотрите также: Обновление тест-кейсов на основе связанных автотестов

Обновлено: 4/22/26, 2:57 PM

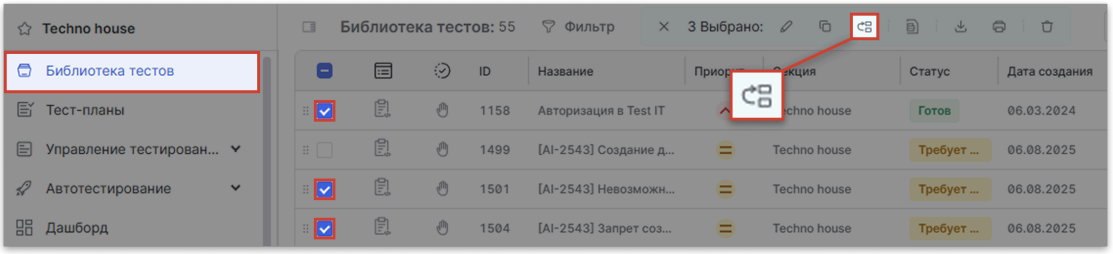
Перемещение и копирование тестов в библиотеке

Вы можете перемещать тест-кейсы, чек-листы и общие шаги между секциями библиотеки внутри одного проекта, а также копировать их в другую секцию или в другой проект.

Перемещение тестов

Массовое перемещение

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Выберите рабочие элементы, которые хотите переместить.
3. Нажмите на значок перемещения.



4. Выберите секцию, в которую хотите переместить выбранные элементы.
5. Нажмите **Переместить**.

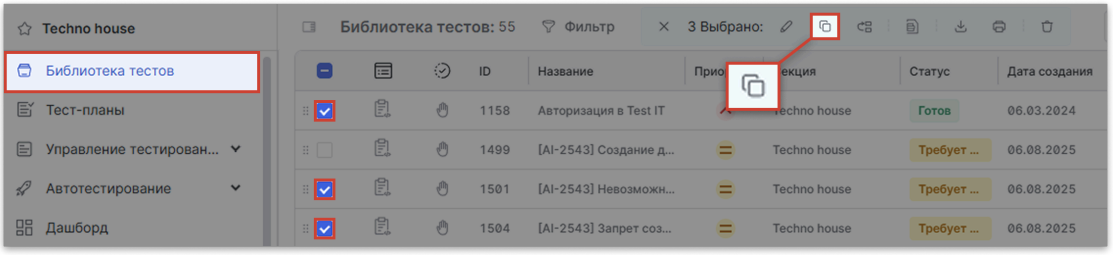
Перемещение одного теста

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте рабочий элемент, который хотите переместить.
3. В поле **Секция** укажите секцию, в которую хотите переместить элемент.
4. Нажмите **Сохранить**.

Копирование тестов

> **Изменение данных при копировании**
>
> При копировании рабочих элементов пользовательские атрибуты теряются, а общие шаги разворачиваются в теле теста.

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Выберите рабочие элементы, которые хотите скопировать.
3. Нажмите на значок копирования.



4. Выберите способ копирования:
   - **Скопировать рабочие элементы с секциями** — позволяет скопировать рабочие элементы вместе с секцией, в которой они находятся.
   - **Скопировать только рабочие элементы** — позволяет скопировать рабочие элементы в другую секцию в текущем проекте или в секцию библиотеки тестов другого проекта.
5. Нажмите **Далее**.
6. Выберите проект и секцию библиотеки тестов, в которые хотите скопировать выбранные рабочие элементы.
7. Нажмите **Копировать**.

Обновлено: 4/15/26, 7:08 PM

Архивация, восстановление и удаление тестов

Вы можете архивировать рабочие элементы из библиотеки тестов. Архивированные рабочие элементы становятся неактивными и не могут быть изменены: из них невозможно создавать тест-кейсы, общие шаги, чек-листы, а также наполнять их содержимым.

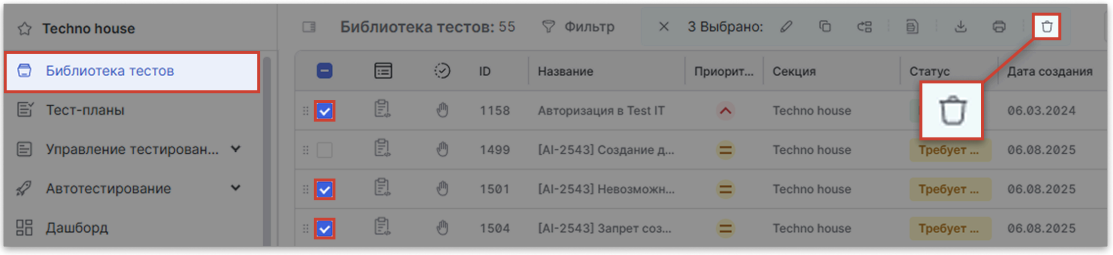
Вы можете восстановить рабочие элементы из архива в течение периода, заданного администратором системы в системных настройках. По окончании этого периода архив автоматически очищается.

Также можно безвозвратно удалить рабочие элементы. После удаления или очистки архива восстановить тесты невозможно.

Архивация тестов

Чтобы архивировать рабочий элемент:

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Выберите рабочие элементы, которые хотите архивировать.
3. Нажмите на значок архивации.



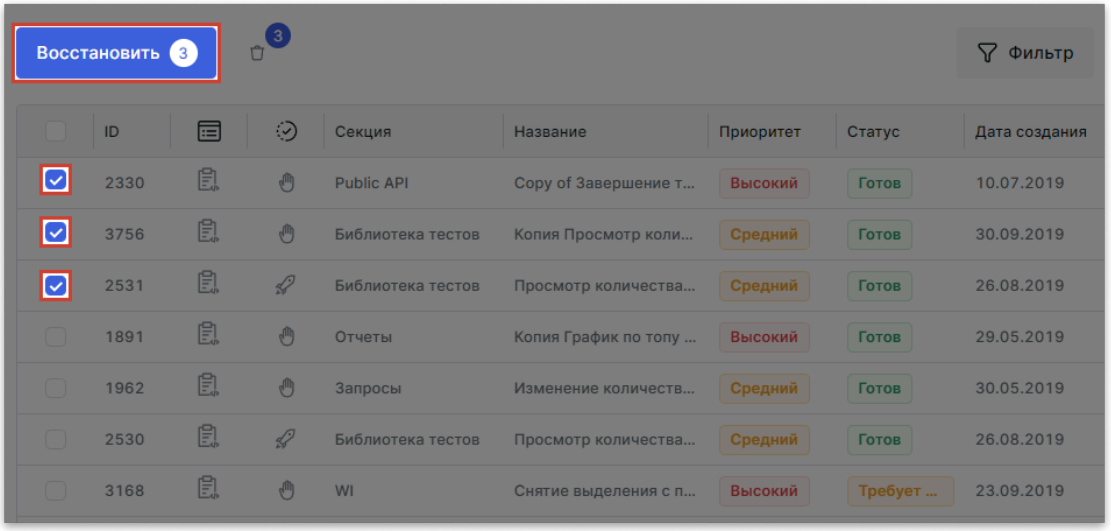
4. Подтвердите действие в диалоговом окне.
   Выбранные тесты перемещены в архив.

Восстановление тестов

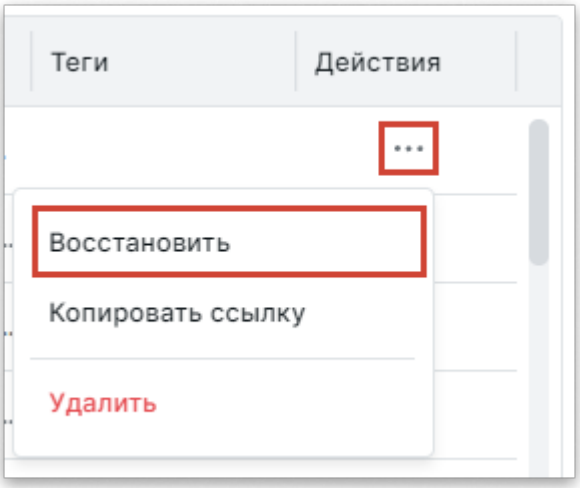
Рабочие элементы хранятся в архиве в течение периода, установленного в системных настройках. В течение этого периода рабочие элементы возможно восстановить.

Чтобы восстановить рабочие элементы:

1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Библиотека тестов**.
3. В списке ручных тестов выберите тесты, которые хотите восстановить, затем нажмите **Восстановить**.



4. **Опционально:** Чтобы восстановить отдельные выбранные рабочие элементы, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.



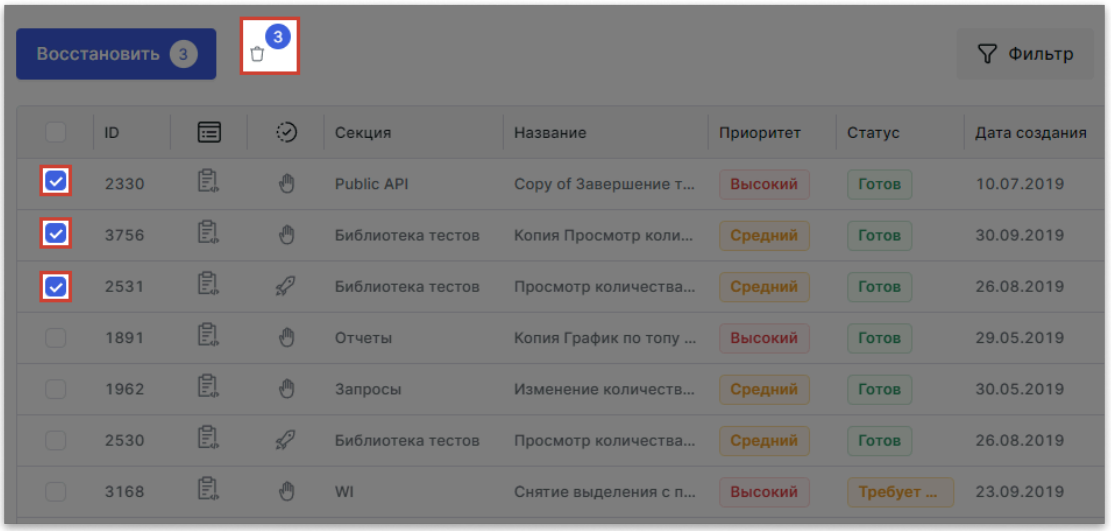
5. Подтвердите действие в диалоговом окне.
   Восстановленные рабочие элементы снова будут отображаться в разделе **Библиотека тестов**.

Удаление тестов

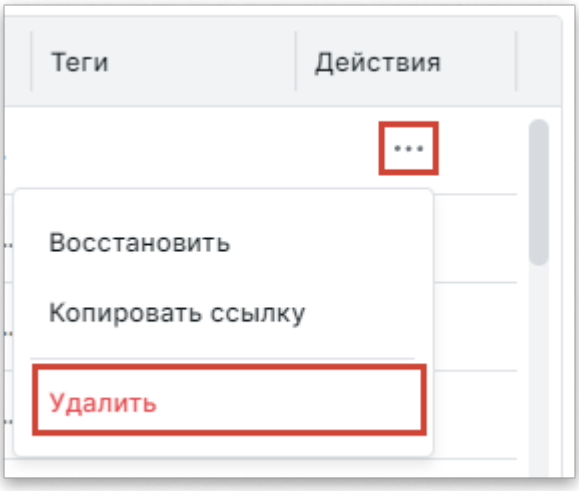
Вы можете безвозвратно удалить рабочие элементы из архива.

Чтобы удалить рабочие элементы:

1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Библиотека тестов**.
3. В списке ручных тестов выберите тесты, которые хотите удалить, затем нажмите **Удалить**.



4. **Опционально:** Чтобы удалить отдельные выбранные рабочие элементы, откройте выпадающее меню в колонке **Действия** и выберите **Удалить**.



5. Подтвердите действие в диалоговом окне.

Обновлено: 4/15/26, 7:08 PM

Работа с секциями

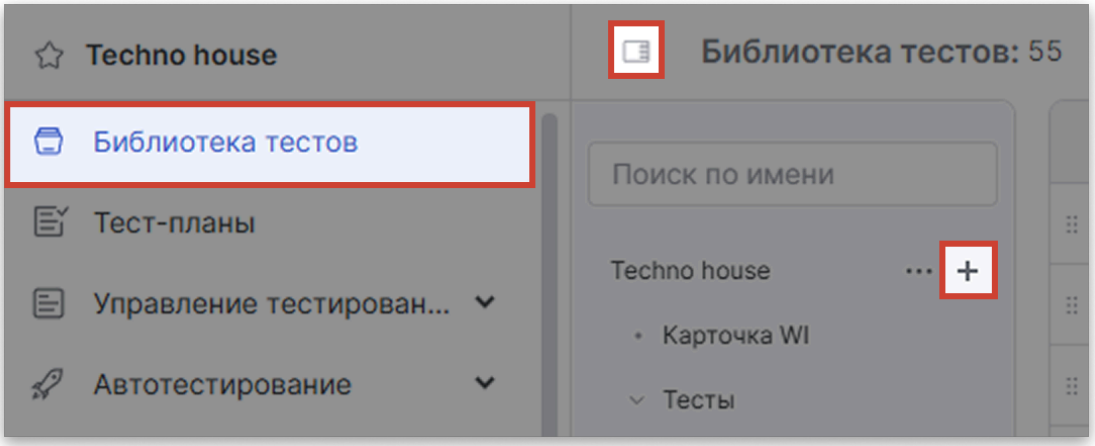
- **Для этого раздела мы подготовили видео**: Как организовать тестовую модель с Test IT .

Секция — это группа тестов (рабочих элементов) в проекте, создаваемая для удобства навигации.

> **Добавляйте пред- и постусловия секций.**
>
> В Test IT можно создать секцию и указывать пред- и постусловия для входящих в нее тестов.
>
> - Предусловия применяются к тестам, вложенным в данную секцию, к дочерним секциям они не применяются и указываются отдельно.
> - Когда предусловия и постусловия секции заданы, они будут применяться ко всем тестам в этой секции.
> - Предусловия и постусловия секции невозможно удалить или редактировать из рабочего элемента.
> - В предусловиям и постусловиям применяются общие шаги.

Создание секций

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. В списке секций нажмите **+**.



3. Введите название новой секции.
4. **Опционально:** Добавьте пред- и постусловия секции. При необходимости добавьте в пред-/постусловие общий шаг. Это можно

сделать и после создания секции.
5. Нажмите **Сохранить**.
6. **Опционально:** Чтобы скрыть или показать дерево секций, слева от счетчика тестов нажмите **Скрыть секции** / **Показать секции**.

Созданная секция становится вложенной в ту секцию, из которой она была создана. Test IT поддерживает неограниченное количество уровней вложенности секций.

Редактирование, удаление и другие действия с секциями

При удалении секций возможна потеря данных!

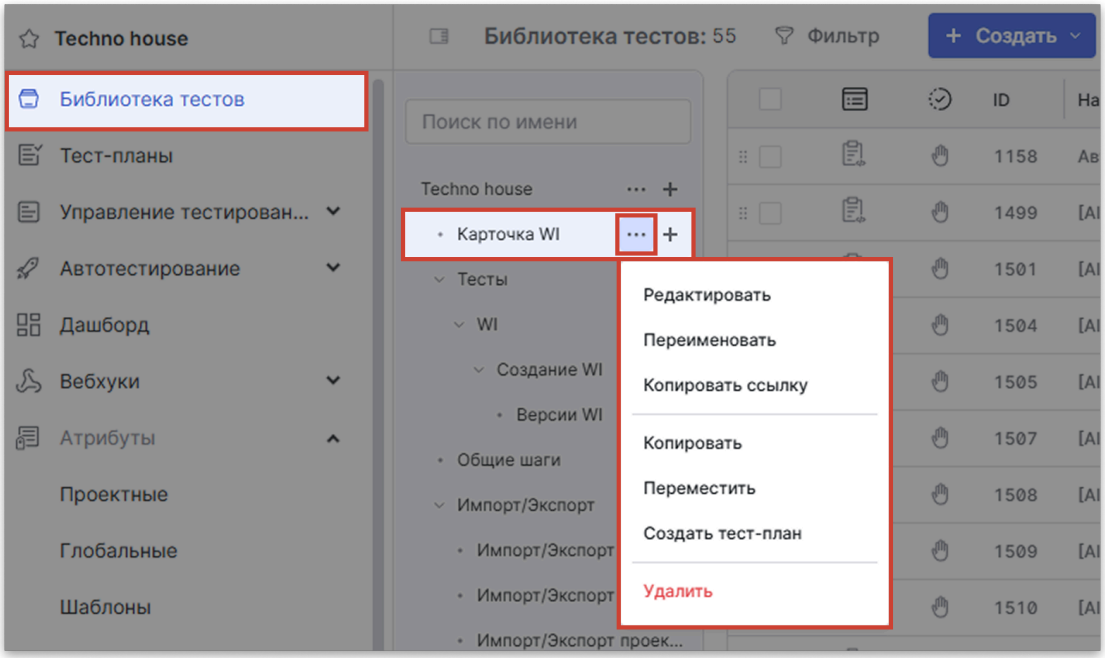
- При удалении секции тест-кейсы (рабочие элементы), входящие в нее, перемещаются в архив.
- Удаление секции приведет к потере тест-поинтов, созданных из тест-кейсов, входящих в секцию, если соблюдены оба следующих условия:
  1. Набор тестов был создан с сохранением структуры библиотеки тестов
  2. Тест-поинтам не проставлены результаты либо тест-план не заблокирован

Вы можете редактировать, перемещать, удалять, копировать секцию, копировать ссылку на нее, создавать из нее тест-планы и др.

Для совершения действий с секциями:

1. Откройте библиотеку тестов и найдите нужную секцию.

2. Справа от названия секции нажмите меню **Действия**.



3. Выполните нужное действие. При необходимости сохраните изменения или подтвердите действие.

Обновлено: 4/15/26, 7:08 PM

Работа с параметрами и конфигурациями в Test IT

Параметры и конфигурации в Test IT

Параметр — это характеристика окружения, на котором проводится тестирование ПО (например, операционная система определенной версии, браузер, и т. д.).

Test IT Базовый и Стандарт

В Test IT есть несколько типов параметров:

- Параметры ручных тестов — если вы укажете в тест-кейсе параметр и несколько его значений, а затем добавите его в набор, в тест-плане получится несколько тест-поинтов по количеству указанных значений.
- Параметры конфигураций — для создания конфигураций, использования в тест-планах и прогонах автотестов
- Параметры автотестов — такие параметры нельзя создавать/изменять на UI, они создаются в Test IT при загрузке автотестов и их результатов. Такие параметры нельзя создать из интерфейса Test IT, но можно удалять.

Test IT Про 5.8 и ранее

Параметры используются:

- В конфигурациях для описания тестового окружения
- В тестах без создания конфигураций. Если вы укажете в тест-кейсе параметр и несколько его значений, а затем добавите его в набор, в тест-плане получится несколько тест-поинтов по количеству указанных значений.

Конфигурация — это набор параметров и их значений. Конфигурации позволяют привязать наборы тестов к определенной связке параметров окружения, на которых они запущены.

Как запускать тесты на разных конфигурациях

Конфигурации указываются для тестовых наборов, а не для отдельных тестов. Чтобы запускать тесты на разных конфигурациях, вы можете:

- Добавить несколько конфигураций к одному набору
- Создать несколько тестовых наборов с разными конфигурациями

Чтобы создать конфигурацию, вам нужно:

- Создать параметры конфигурации, которые будут лежать в основе конфигураций
- Создать конфигурацию, используя эти параметры

Настройте уведомления об изменениях конфигураций

С Test IT вы можете получать в мессенджер уведомления о создании, редактировании или архивировании конфигурации. Для этого нужно настроить вебхук, указав необходимые переменные.

Создание параметров

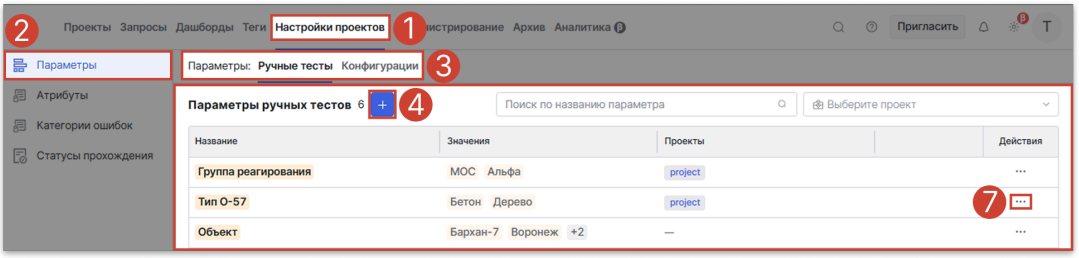
Вы можете задавать параметры, которые требуется проверить. Например, если ваш тестируемый продукт поддерживает несколько операционных систем, вы можете создать параметр `ОС`. Созданные параметры могут быть использованы в различных проектах и ручных тестах.

Test IT Облако; Test IT Про 6.0 и позднее

Чтобы создать параметр ручного теста или конфигурации:

1. Войдите в систему, используя аккаунт с ролью **Администратор**, затем в верхней навигационной панели выберите **Настройки проектов**.
2. В навигационном меню слева выберите **Параметры**. Отобразится список параметров.

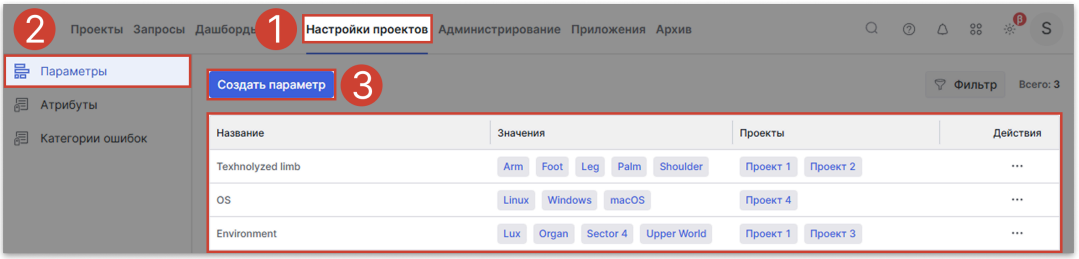
3. В правом верхнем углу выберите тип параметра, который хотите создать, открыв вкладку **Ручные тесты** или **Конфигурации**. После создания изменить тип параметра нельзя.
4. Нажмите **Создать параметр**.
5. Введите данные параметра:
   - **Название**
   - **Значения** параметра. Например, для параметра `ОС` можно создать значения `Windows 10`, `Ubuntu 21.04` и `macOS`.
   - Проекты, в которых параметр будет использоваться. Если проект не указан, параметр не может быть в нем использован. Проекты можно добавить при редактировании параметра.
6. Нажмите **Сохранить**.
7. **Опционально:** Чтобы отредактировать параметр, нажмите значок **Действия** справа от его названия, выберите **Редактировать**, затем внесите необходимые изменения и сохраните их.



Test IT Про 5.8 и ранее

Чтобы создать параметр:

1. Войдите в систему, используя аккаунт с ролью **Администратор**, затем в верхней навигационной панели выберите **Настройки проектов**.
2. В навигационном меню слева выберите **Параметры**. Отобразится список параметров.
3. Нажмите **Создать параметр**.



4. Введите данные параметра:
   - Название

- Проекты, в которых параметр будет использоваться. Если проект не указан, параметр не может быть в нем использован. Проекты можно добавить при редактировании параметра.
- Значения параметра. Например, для параметра `ОС` можно создать значения `Windows 10`, `Ubuntu 21.04` и `macOS`.

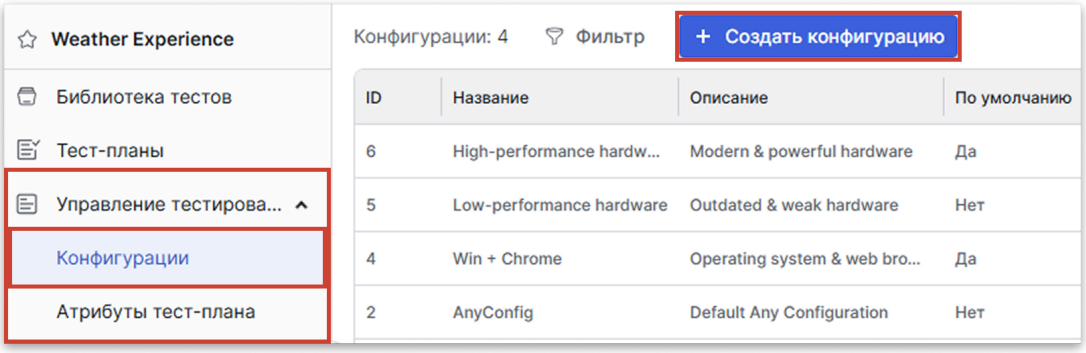
5. Нажмите **Сохранить**.
6. **Опционально:** Чтобы отредактировать параметр, нажмите значок **Действия** справа от его названия, выберите **Редактировать**, затем внесите необходимые изменения и сохраните их.

Информация об использовании параметров доступна в статье Добавление параметров в тесты.

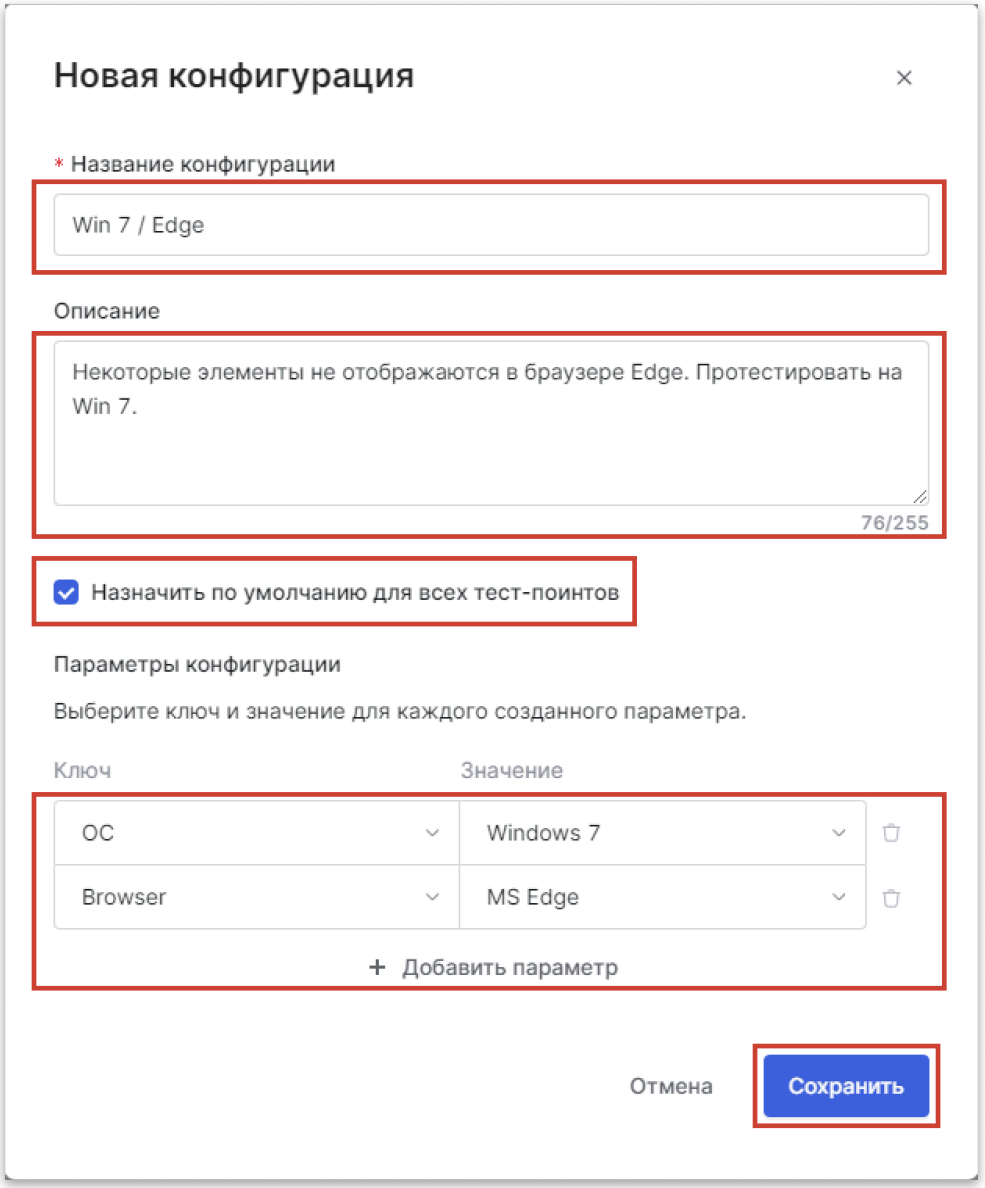
Работа с конфигурациями

Создание конфигураций

1. Откройте проект.
2. Перейдите в раздел **Управление тестированием → Конфигурации**.
3. Нажмите **Создать конфигурацию**.



Откроется окно создания конфигурации.



4. Введите название конфигурации. При необходимости добавьте описание. Вы можете включать новую конфигурацию во все создаваемые вами наборы тестов по умолчанию. Для этого активируйте опцию **Назначить по умолчанию для всех тест-поинтов**.
5. Добавьте в параметры конфигурации и присвойте им значения:
   а. В выпадающем меню **Ключ** выберите нужный параметр. Например, параметр `ОС`.
   б. В выпадающем меню **Значение** выберите значение параметра. Например, `Windows 10`.
   в. Нажмите **Добавить**.
   г. **Опционально:** Добавьте другие параметры конфигурации. Например, если помимо параметра `ОС` вы создали параметр `Browser` со значениями `Chrome`, `Firefox` и `Safari`, вы также можете добавить его в конфигурацию и присвоить ему одно из значений. Конфигурация будет состоять из сочетания параметров операционной системы и

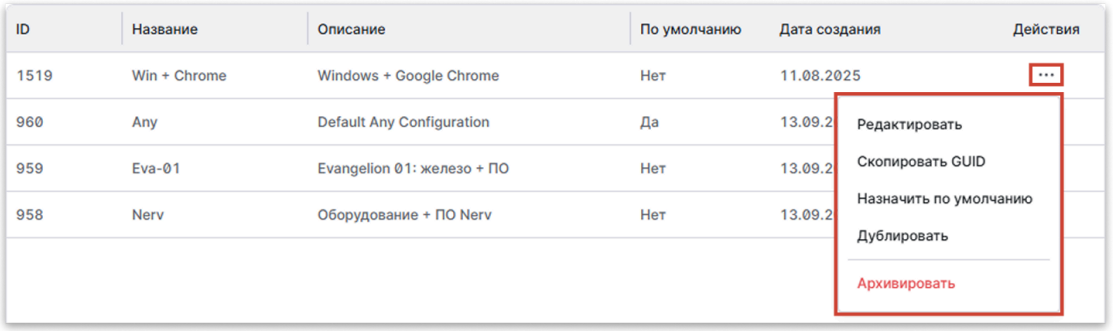
браузера. Все добавленные параметры и их значения отображаются в общем списке.

д. **Опционально:** Чтобы удалить параметр, нажмите значок **Корзина** слева от строки параметра.

6. Нажмите **Сохранить**.
   Созданная конфигурация отображается в таблице в разделе **Конфигурации**.

Действия с конфигурациями

- Для совершения действия воспользуйтесь меню в колонке **Действия**.



Вы можете:

- Редактировать настройки конфигурации
- Скопировать GUID конфигурации
- Назначить по умолчанию или удалить из списка по умолчанию
- Дублировать
- Архивировать конфигурацию. Архивные конфигурации нельзя изменять, добавлять в тестовые наборы и тест-планы. Архивные конфигурации и созданные на их основе тест-поинты скрываются из тест-планов.

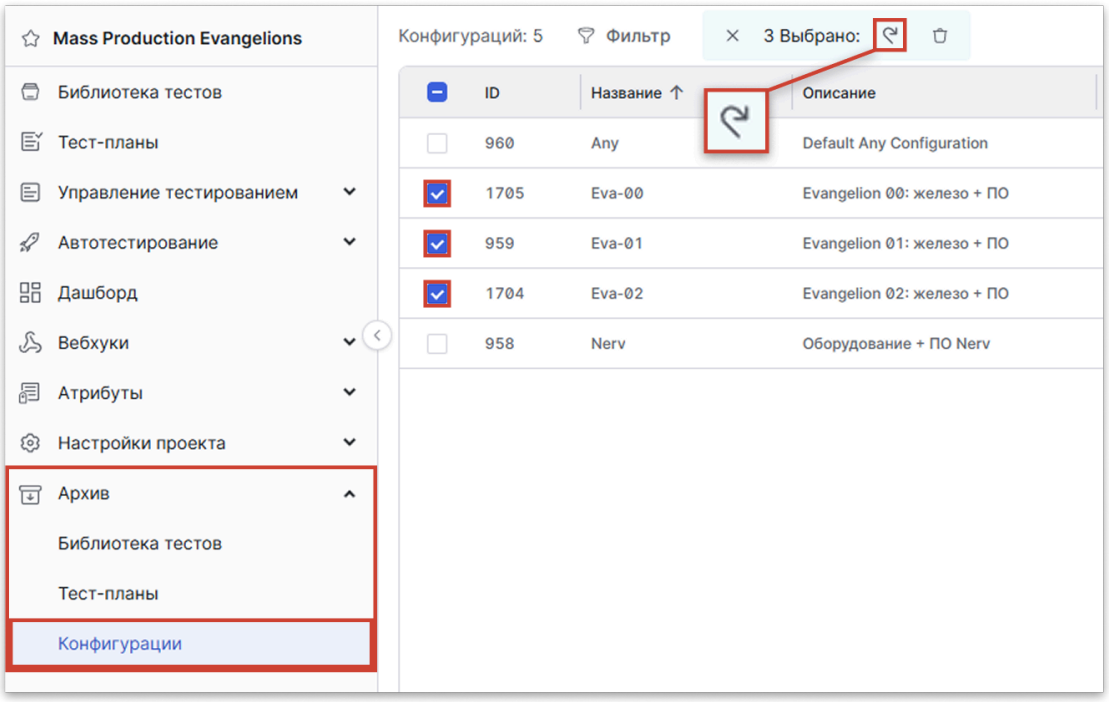
Восстановление и удаление конфигураций

Вы можете восстанавливать или удалять архивные конфигурации. Конфигурации, удаленные из архива, не могут быть восстановлены. Если в списке одна конфигурация, архивировать ее нельзя.

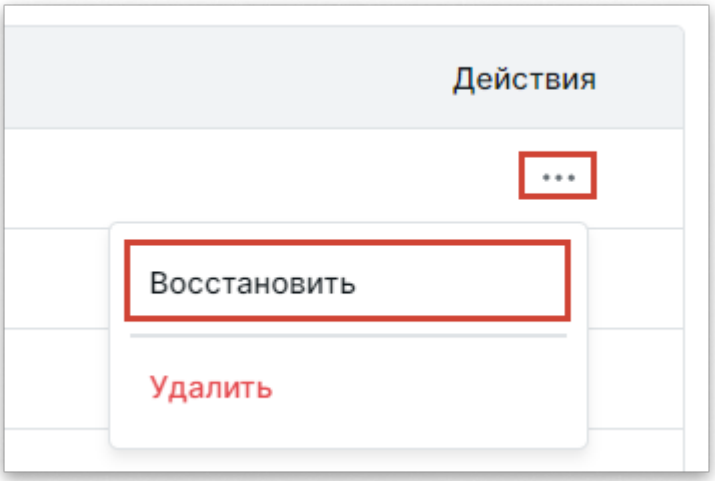
Чтобы восстановить конфигурации:

1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Конфигурации**.
3. В списке конфигураций выберите конфигурации, которые хотите восстановить, затем нажмите **Восстановить**. Чтобы выбрать все

конфигурации, отметьте флажком ячейку **ID**.



4. **Опционально:** Чтобы восстановить отдельную конфигурацию, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.

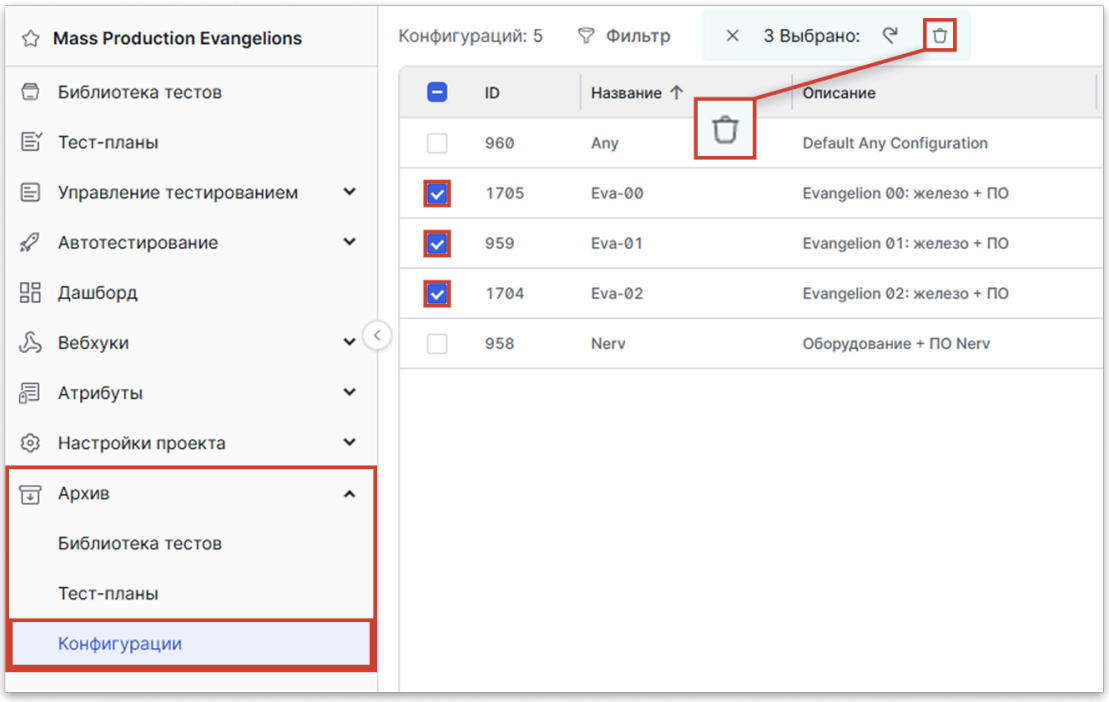


5. Подтвердите действие в диалоговом окне.
   Конфигурация снова станет активной и появится в списке конфигураций.

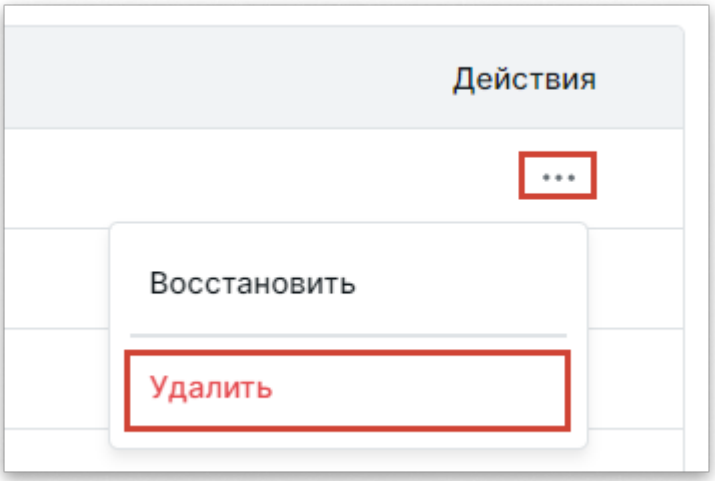
Чтобы удалить конфигурации:

1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Конфигурации**.
3. В списке конфигураций выберите конфигурации, которые хотите удалить, затем нажмите значок **Удалить**. Чтобы выбрать все

конфигурации, отметьте флажком ячейку **ID**.



4. **Опционально:** Чтобы удалить отдельную конфигурацию, откройте выпадающее меню в колонке **Действия** и выберите **Удалить**.



5. Подтвердите действие в диалоговом окне.

Обновлено: 10/29/26, 1:42 PM

Создание тест-плана

Назначьте конфигурацию

Перед созданием тест-плана убедитесь, что в проекте назначена конфигурация по умолчанию. Вы также можете назначить нужные вам конфигурации.

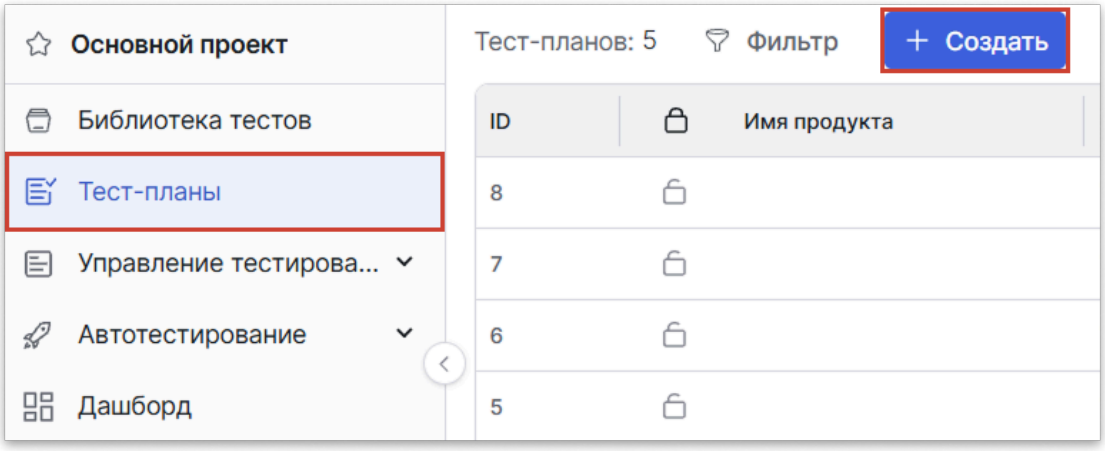
Вы можете создать тест-план:

- Из раздела тест-планов
- Из библиотеки тестов

Создание тест-плана из раздела тест-планов

Чтобы создать тест-план:

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Нажмите **Создать**.



4. Введите информацию о тест-плане: название, тестируемый продукт и версию, даты начала и окончания, описание и теги. В поле **Время прохождения** выберите режим работы таймера:
   - **Отключено** — таймер не активирован.
   - **Ручное** — таймер запускается и отключается пользователем нажатием на кнопку таймера в тест-поинте.

- **Автоматическое** — таймер автоматически запускается при открытии тест-поинта и отключается при его закрытии или переходе к следующему тест-поинту.

5. Нажмите **Сохранить**.

Тест-план отобразится в таблице в разделе **Тест-планы**.

Создание тест-плана из библиотеки тестов

В библиотеке тестов вы можете создать тест-план:

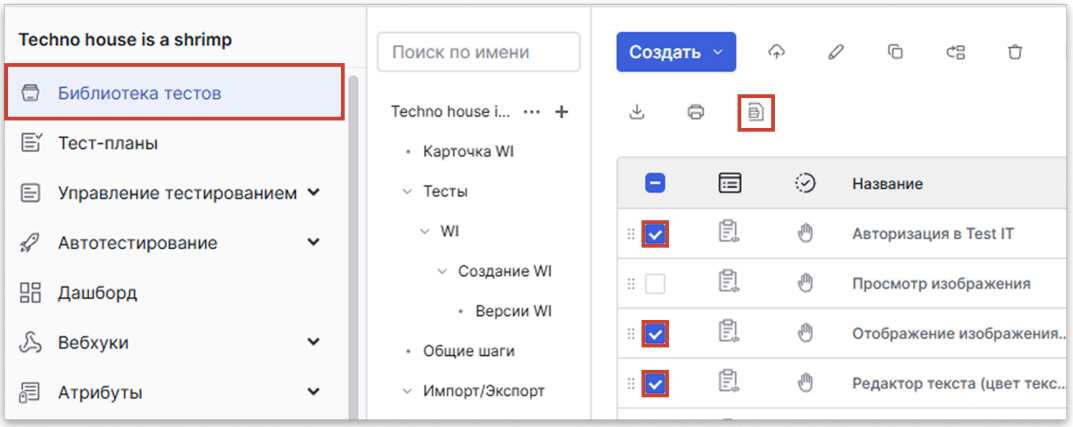
- Из списка тестов с созданием пользовательского набора тестов
- Из отдельной секции библиотеки с созданием набора, сохраняющего структуру секции
  При создании тест-плана наборы тестов создаются автоматически.

Чтобы создать тест-план:

1. Откройте проект. Откроется библиотека тестов.
2. Выберите тесты, которые хотите добавить в тест-план — из списка тестов или из секции библиотеки:

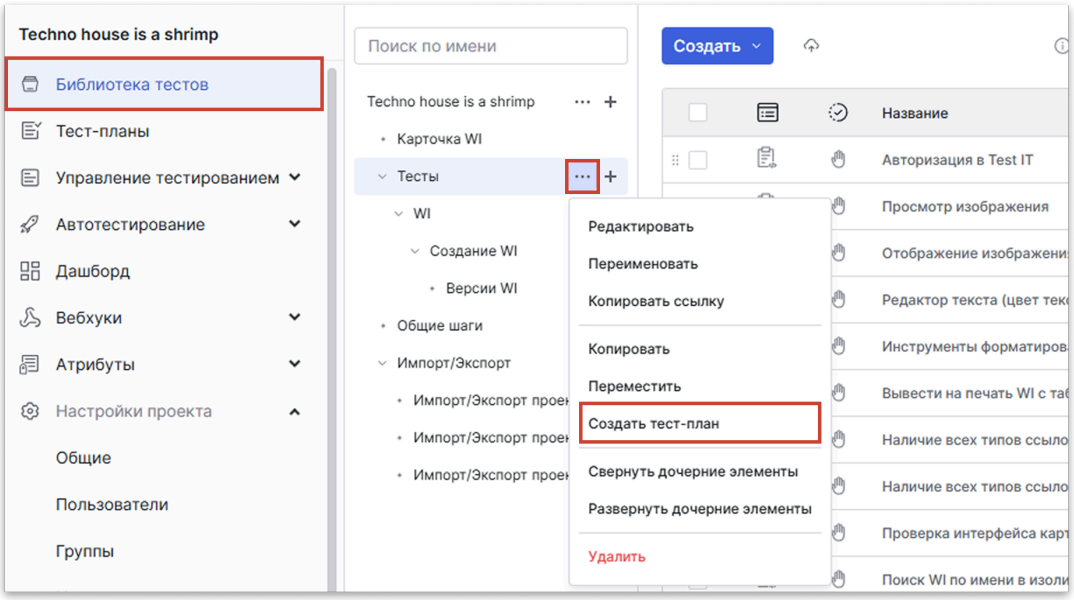
Из списка тестов

- В списке библиотеки (таблице) выберите тесты и нажмите значок создания тест-плана . При необходимости воспользуйтесь Фильтром.



Из секции

- В списке секций наведите курсор на название секции и нажмите значок **Действия** ··· справа. В открывшемся меню выберите **Создать тест-план**.



4. **Опционально:** В открывшемся окне укажите данные о вашем тест-плане: название (измените название, заданное по умолчанию), тестируемый продукт и его версию, даты начала и окончания, описание тест-плана, теги, конфигурации для создания тестового набора.
5. Нажмите **Создать**.
   Созданный тест-план откроется в разделе **Тест-планы**.

Действия с тест-планами и просмотр тест-планов

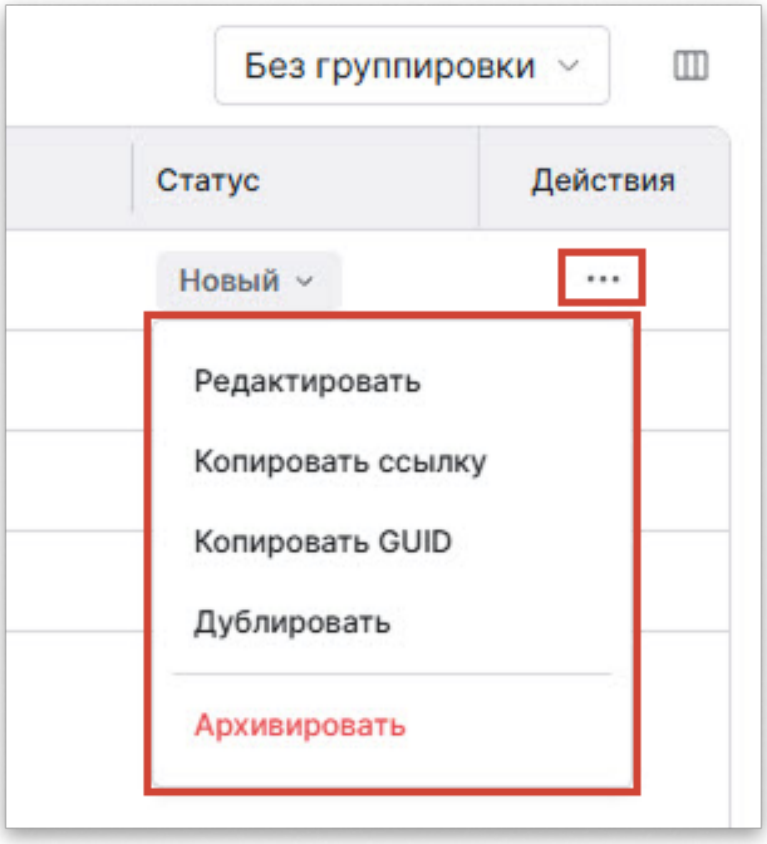
В разделе тест-планов вы можете:

- Редактировать название или описание тест-плана
- Копировать ссылку на тест-план
- Копировать GUID тест-плана
- Дублировать тест-план
- Архивировать тест-план

Для этого:

1. Откройте проект и перейдите в раздел **Тест-планы**.

2. Справа он названия тест-плана откройте меню **Действия**.



3. Выберите и выполните нужное действие.

- Чтобы настроить отображаемые в таблице колонки, нажмите **Колонки** над таблицей .
- Для поиска и сортировки тест-планов нажмите **Фильтр** над таблицей . Настроенные фильтры в разделе **Тест-планы** автоматически сохраняются при переходе между вкладками тест-плана.

Обновлено: 4/15/26, 7:08 PM

Изменение тест-плана

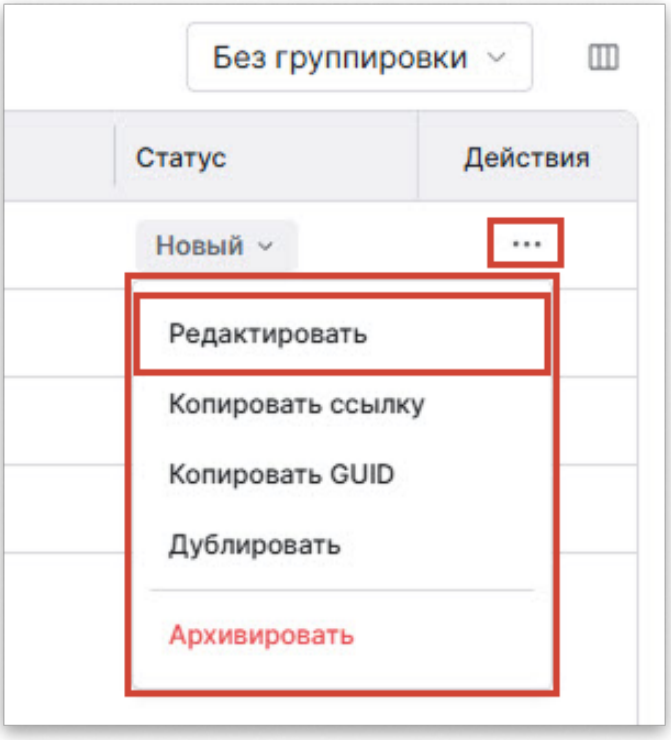
Настройте уведомления об изменениях тест-плана

С Test IT вы можете получать в мессенджер уведомления об изменении тест-плана и его статуса. Для этого нужно настроить вебхук, указав переменные для изменения тест-плана или его статуса.

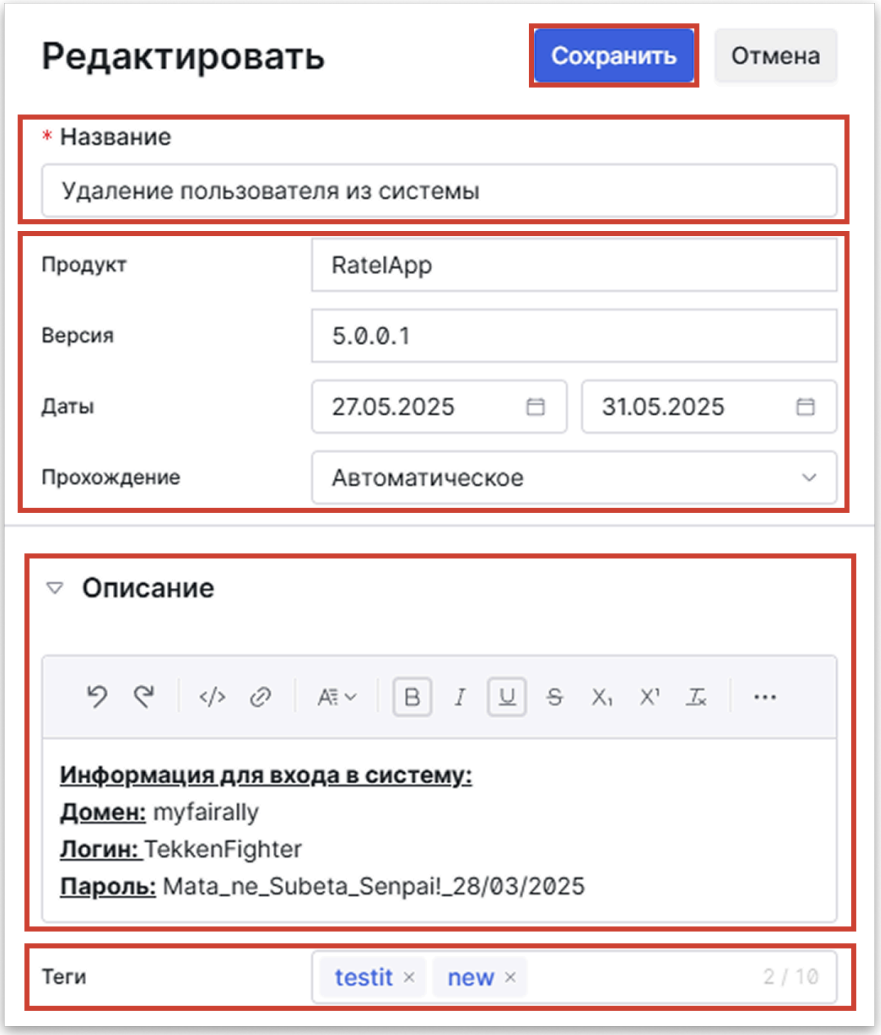
Вы можете изменять тест-план, включая его название, описание, атрибуты, время прохождения и теги.

Чтобы изменить тест-план:

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. В списке тест-планов нажмите **Действия** справа от названия тест-плана и выберите **Редактировать**.



   В правой части экрана откроется окно редактирования.
4. Опционально: откройте тест-план и нажмите значок **Действия** между значками блокировки и скрытия правой боковой панели.
5. Внесите необходимые изменения.
6. Нажмите **Сохранить**.



Автоматическое обновление статуса тест-плана

Статус тест-плана обновляется автоматически при исполнении входящих в него в него тест-поинтов или архивации тест-плана.

Тест-план переходит в статус *В процессе* в следующих случаях:

- Указан первый результат исполнения тест-поинта, входящего в тест-план — для тест-планов в статусе *Новый*.
- Указан результат исполнения тест-поинта, результат по которому не был указан или был сброшен — для тест-планов в статусе *Завершен* или *Приостановлен*.

Тест-план переходит в статус *Завершен* в следующих случаях:

- Указан результат по последнему выполненному тест-поинту, входящему в тест-план (тест-план не содержит тест-поинтов без указанных результатов) — для тест-планов в статусе *В процессе* или *Приостановлен*.
- Тест-план отправлен в архив.

Обновлено: 4/15/26, 7:08 PM

Блокировка тест-плана и запрос на его разблокировку

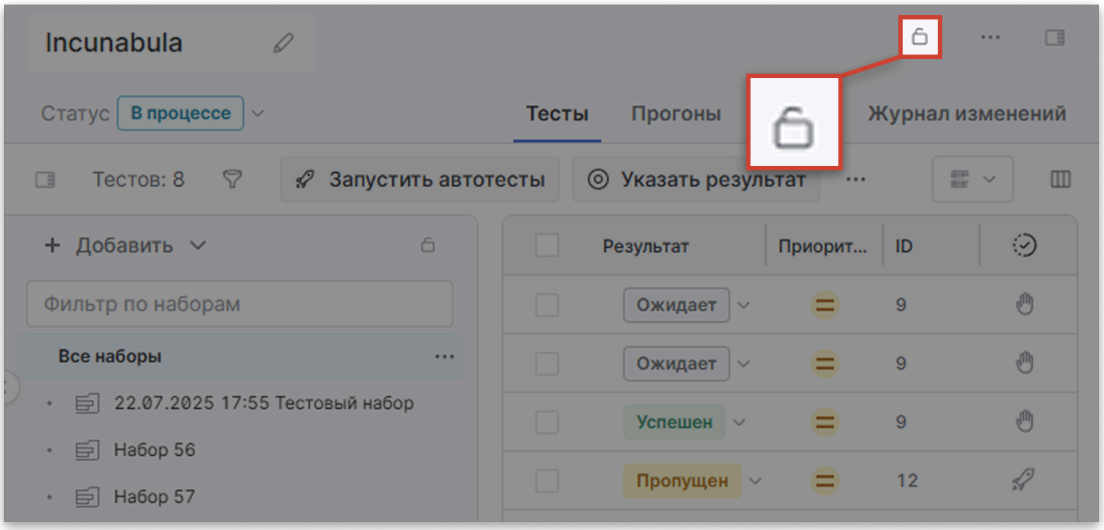
> **Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Test IT позволяет блокировать тест-планы, чтобы фиксировать версии тестов и не допускать изменений в тестовой документации. В заблокированном тест-плане невозможно создавать новые тестовые наборы, а также редактировать уже созданные. При блокировке тест-плана все пользователи, имеющие к нему доступ, могут просматривать его, делиться им с помощью ссылки, а также выполнять его.

Разблокировать тест-план может только пользователь с проектной ролью, предусматривающей права на редактирование модуля тест-планов и выше. Пользователи с проектными ролями без доступа к редактированию тест-планов могут отправить запрос на разблокировку тест-плана.

Чтобы заблокировать тест-план или отправить запрос на его разблокировку:

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. Нажмите значок **Замок** справа от названия тест-плана.



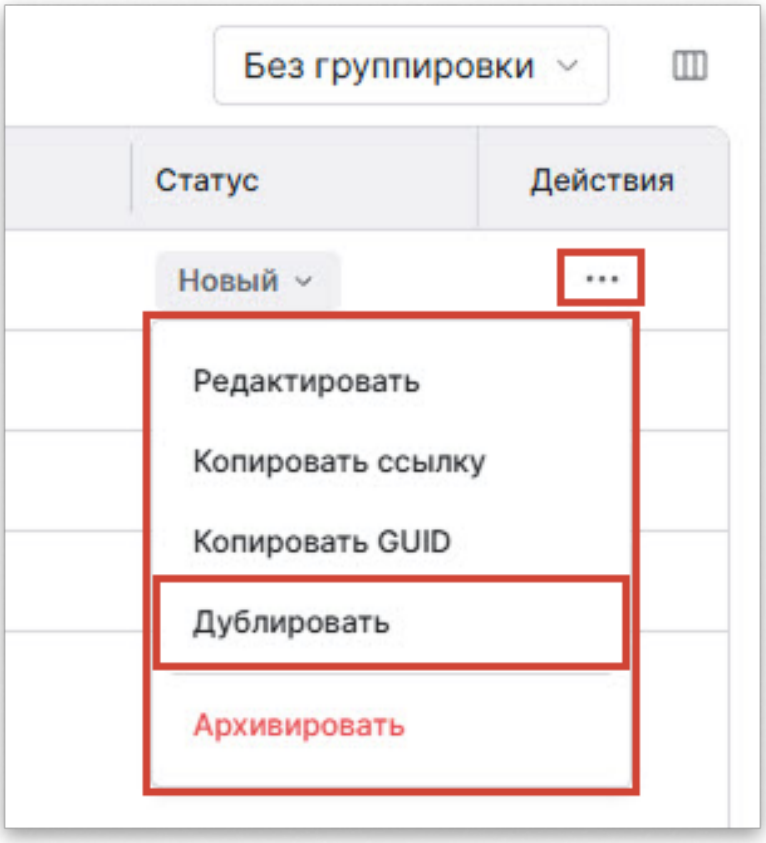
После отправления запроса пользователю, заблокировавшему тест-план, отправляется уведомление с запросом на разблокировку.

Обновлено: 8/18/26, 4:54 PM

Дублирование тест-плана

Если вам нужно создать новый тест-план, идентичный уже существующему, вы можете дублировать существующий тест-план.

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. В таблице тест-планов откройте меню в колонке **Действия** и нажмите **Дублировать**.



4. Подтвердите действие в диалоговом окне.

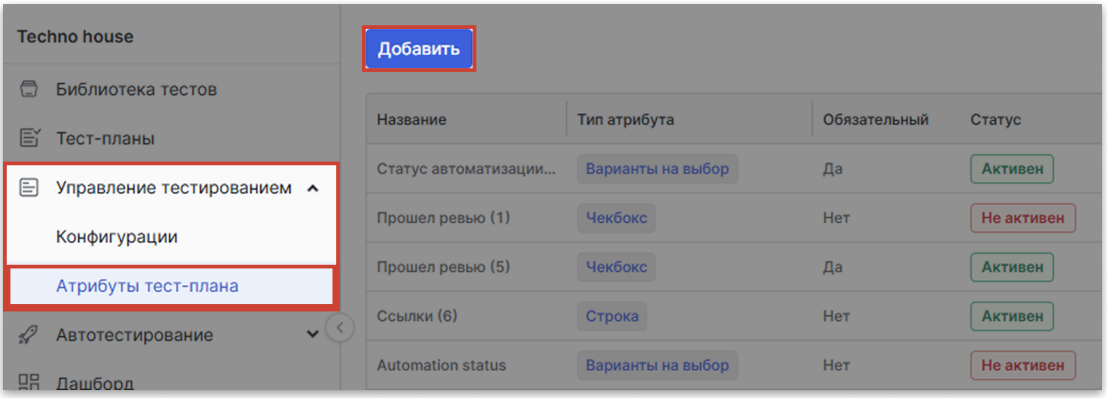
Дубликат тест-плана с добавлением (1) в названии появится в таблице тест-планов. Статус дублированного тест-плана будет установлен как **Новый**, а поля **Дата начала** и **Дата окончания** будут пустыми.

Обновлено: 4/15/26, 7:08 PM

Добавление атрибутов тест-плана

Вы можете добавлять к тест-планам проекта глобальные атрибуты.

1. Откройте проект.
2. Перейдите в раздел **Управление тестированием → Атрибуты тест-плана**.
3. Нажмите **Добавить**.



4. Выберите глобальный атрибут из списка. При необходимости воспользуйтесь поиском.
5. Нажмите **Добавить**.
6. **Опционально**: Чтобы отредактировать или удалить атрибут, используйте меню **Действия** справа от строки атрибута.

Глобальный атрибут появится в списке. При добавлении данный атрибут появляется во всех тест-планах проекта.

Смотрите также:

- Настройка атрибутов
- Настройка атрибутов в проекте

Обновлено: 4/22/26, 2:57 PM

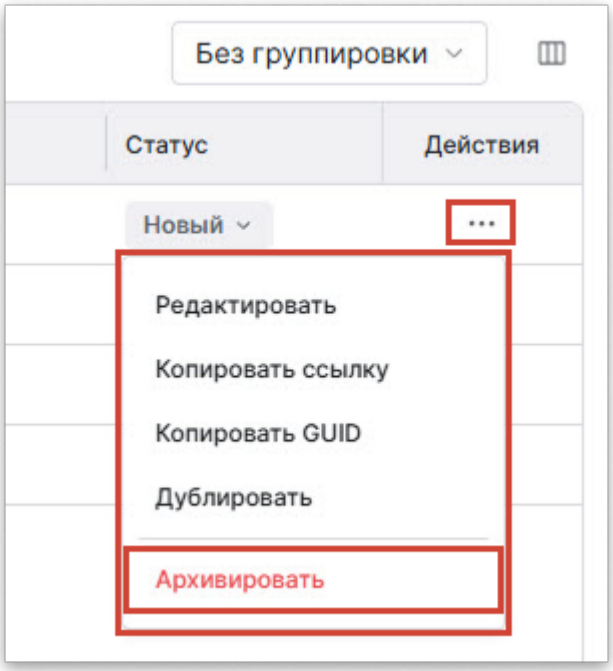
Архивация, восстановление и удаление тест-плана

Вы можете архивировать тест-план, чтобы пользователи не имели к нему доступа. Архивные тест-планы нельзя редактировать, проходить, блокировать, дублировать и экспортировать в XLSX. Вы можете восстановить тест-план из архива.

Архивация

Чтобы добавить тест-план в архив:

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. В таблице тест-планов откройте меню в колонке **Действия** и нажмите **Архивировать**.



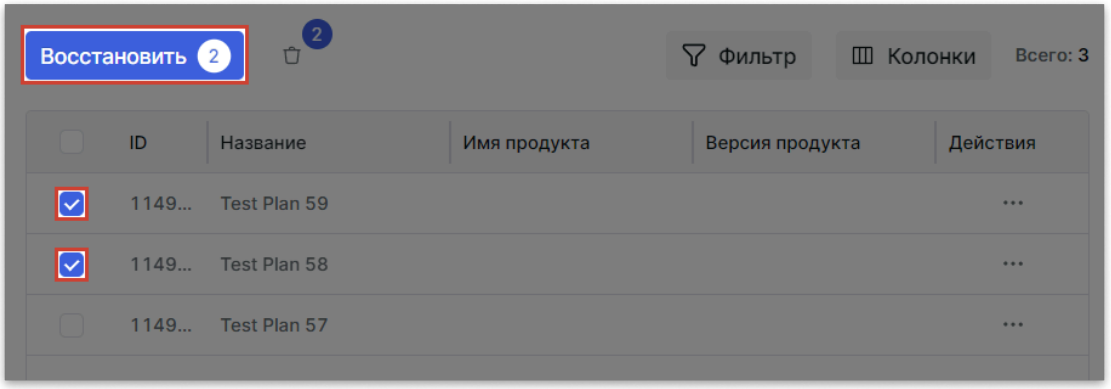
4. Подтвердите действие в диалоговом окне.

Восстановление и удаление тест-планов

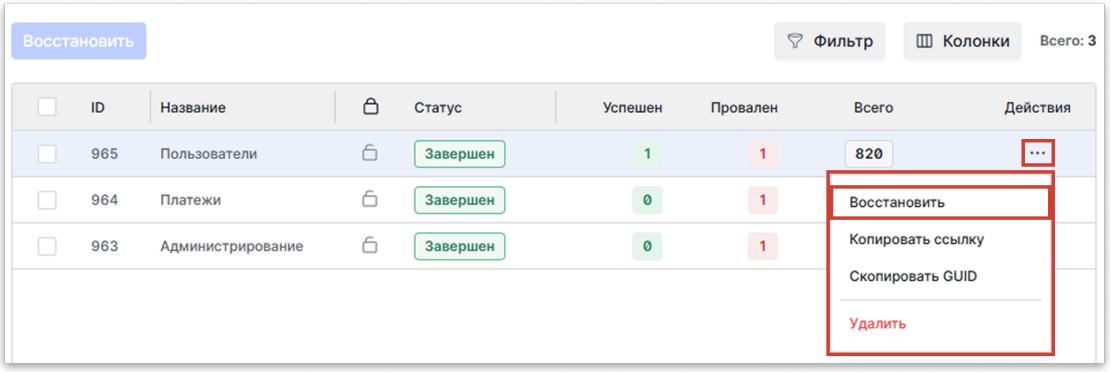
Вы можете восстанавливать или удалять архивные тест-планы. Тест-планы, удаленные из архива, не могут быть восстановлены.

Чтобы восстановить тест-планы:

1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Тест-планы**.
3. В списке тест-планов выберите тест-планы, которые хотите восстановить, затем нажмите **Восстановить**.



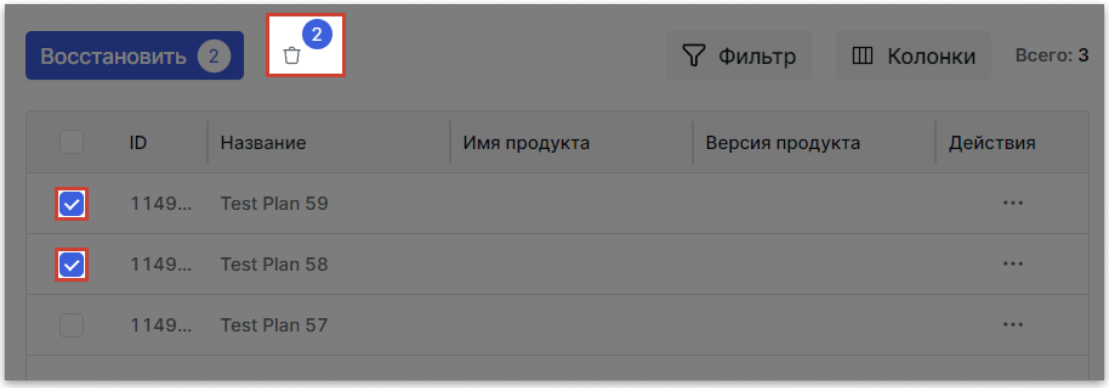
4. **Опционально:** Чтобы восстановить отдельный выбранный тест-план, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.



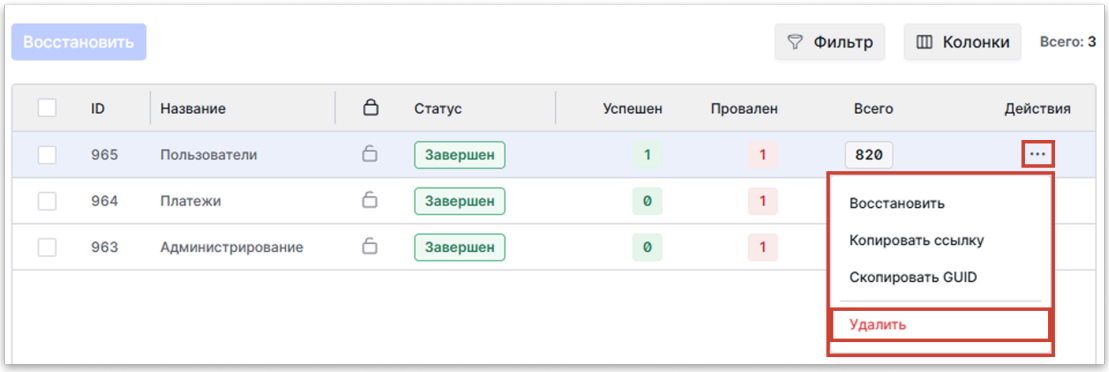
5. Подтвердите действие в диалоговом окне.
   Тест-планы будут восстановлены и появятся в разделе Тест-планов.

Чтобы удалить тест-планы:

1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Тест-планы**.
3. В списке тест-планов выберите тест-планы, которые хотите удалить, затем нажмите значок **Удалить**.



4. **Опционально:** Чтобы удалить выбранный тест-план, откройте выпадающее меню в колонке **Действия** и выберите **Удалить**.



5. Подтвердите действие в диалоговом окне.

Обновлено: 4/15/26, 7:08 PM

Работа с тестовыми наборами

Тестовые наборы в Test IT

После создания тест-плана вам необходимо создать тестовый набор и наполнить его тестами, которые будут выполнены в ходе данного тест-плана. Вы также можете добавлять в тестовый набор конфигурации, на которых планируете выполнять тесты. Test IT позволяет создавать пользовательский тестовый набор, набор из секций библиотеки и динамический набор.

- **Пользовательский набор** содержит тесты, добавленные пользователем.
- **Набор из секции библиотеки** формируется из выбранной секции библиотеки тестов, имеет идентичное название и включает в себя все вложенные секции.
- **Динамический набор** формируется на основании критериев, указанных в фильтре. Новые тесты, соответствующие заданным критериям, будут добавляться в тест-план автоматически. Вы можете создавать, редактировать, и обновлять динамические тестовые наборы. Для создания тестовых наборов, используйте меню **Все наборы** в разделе **Тест-планы**.

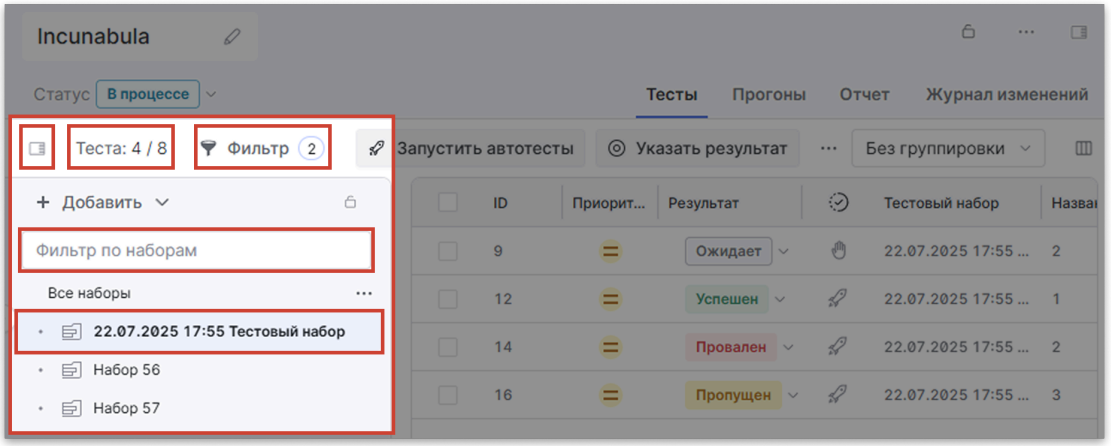
> **Назначьте тестовому набору конфигурацию**
>
> Для корректной работы с тест-планами необходимо назначить тестовому набору хотя бы одну конфигурацию. Если вы не назначили конфигурацию, будет использована конфигурация "По умолчанию" ("Any") — убедитесь, что она не удалена и не отключена по умолчанию.

Просмотр тестовых наборов в тест-плане

На странице тест-плана вы можете просматривать все тестовые наборы, входящие в тест-план, отображать и скрывать их атрибуты, сортировать и находить наборы с помощью фильтра.

Чтобы просмотреть тестовые наборы:

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. В верхней части списка тестовых наборов нажмите **Все наборы**. Тестовые наборы отобразятся в правой части экрана в виде таблицы.
5. Чтобы выбрать отдельный набор, нажмите на него в списке. Количество тестов в наборе будет отображено над списком.
6. Чтобы найти набор, начните вводить его название в поле Фильтр по наборам.
7. **Опционально:** Чтобы скрыть список наборов, нажмите значок свертывания наборов слева от количества тестов.



Вы можете настраивать отображение тестовых наборов:

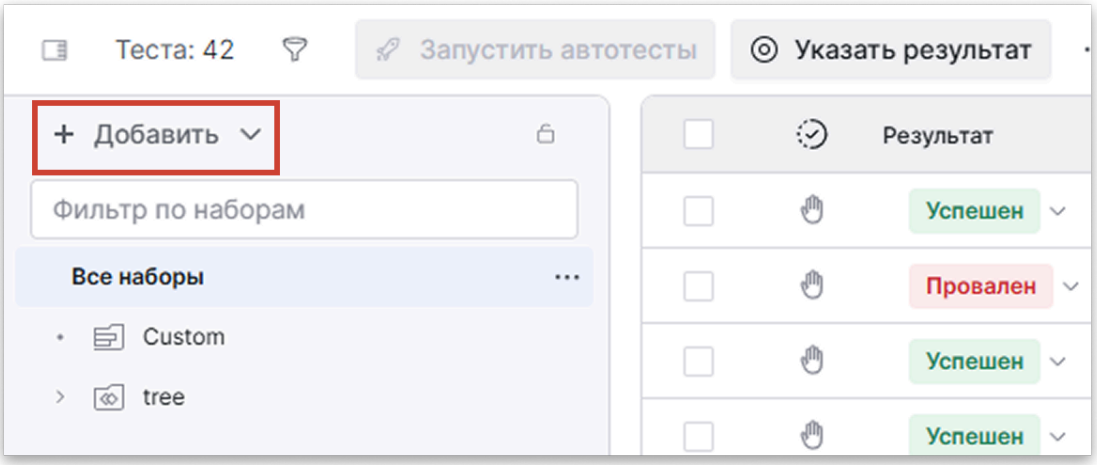
- Чтобы развернуть или свернуть вложенные наборы (дочерние элементы), нажмите **Действия** рядом с выбранным набором, затем выберите нужную опцию.
- Чтобы отсортировать тестовые наборы по значениям, нажмите на заголовок соответствующей колонки в таблице.
- Чтобы выбрать отображаемые колонки, нажмите **Колонки**, затем в выпадающем меню выберите необходимые колонки.
- Чтобы найти тестовые наборы по атрибутам, нажмите **Фильтр**.

Создание тестовых наборов

** Тарифные ограничения: динамические наборы**

В тарифе **Базовый** динамические тестовые наборы недоступны. Если вы создали динамический набор в тарифе **Стандарт**, а затем перешли на тариф **Базовый**, автообновление будет недоступно.
Подробнее: о тарифах Test IT .

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. Над списком тестовых наборов нажмите **Добавить** и выберите тип тестового набора.



5. Введите название тестового набора.
6. Выберите тесты, которые будут добавлены в набор:
   - **Для пользовательских наборов:** Выберите тесты из библиотеки, которые хотите добавить. При необходимости воспользуйтесь фильтром.
   - **Для наборов из секции:** Выберите секцию, из которой хотите добавить тесты в набор. При необходимости воспользуйтесь фильтром.
   - **Для динамических наборов:** Укажите критерии выбора тест-поинтов из фильтра (например *Статус автоматизации*, *Приоритет* и т. д.). Чтобы сохранить структуру библиотеки тестов и настроить

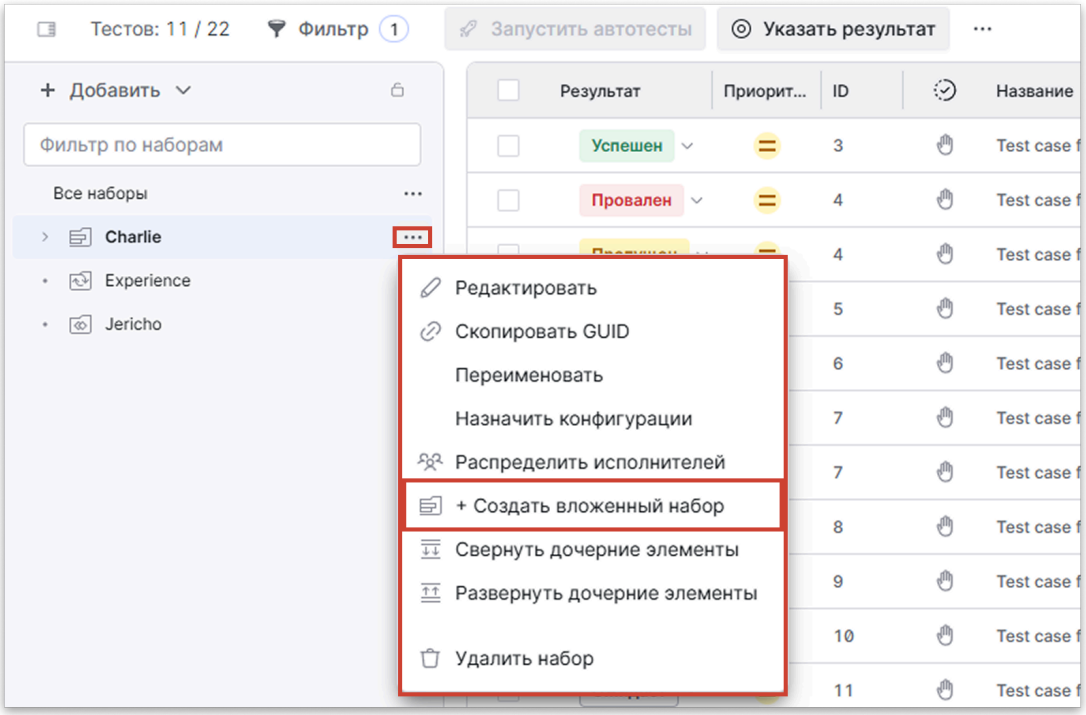
автоматическое обновление набора, выберите Сохранить структуру и Обновлять автоматически в верхней части окна.

7. Нажмите **Сохранить**.
   Тестовый набор появится в списке.

Создание вложенного тестового набора

Внутри пользовательских тестовых наборов вы можете создавать вложенные тестовые наборы. Количество уровней вложенности не ограничено. Чтобы создать тестовый набор:

1. Откройте меню **Действий** справа от названия тестового набора, в который хотите добавить вложенный набор. Выберите **Создать вложенный набор**.



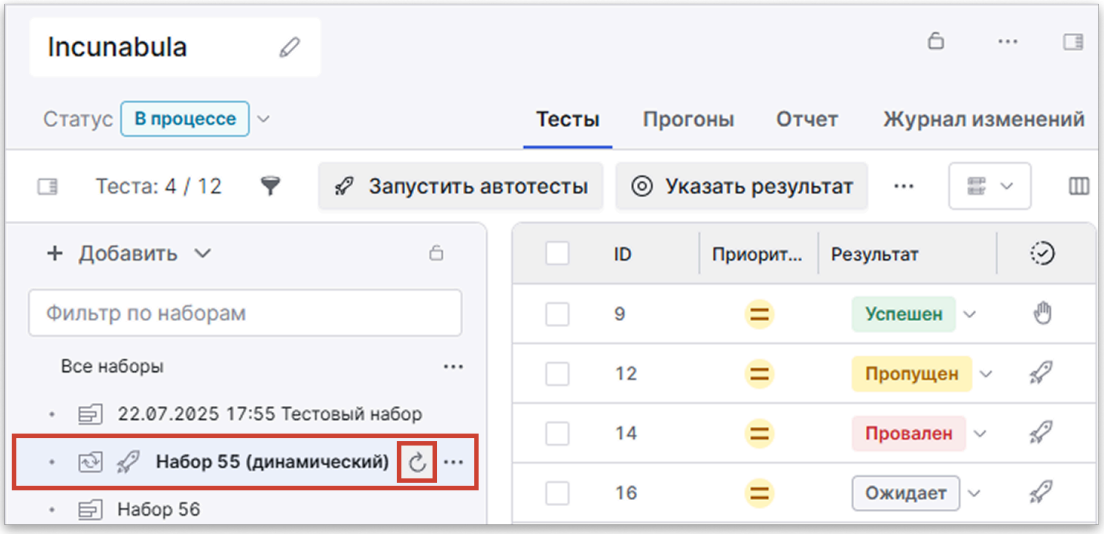
2. Укажите название вложенного тестового набора и выберите тесты, которые хотите в него включить.
3. Нажмите **Сохранить**.
   Вложенный тестовый набор появится в панели тестовых наборов.

Действия с динамическим тестовым набором

Вы можете обновлять, редактировать, сворачивать, разворачивать и удалять динамический набор, а также настраивать его автоматическое обновление.

Чтобы обновить сохраненный динамический набор:

- Наведите курсор на динамический набор в меню **Все наборы** и нажмите **Обновить**.



  Если динамические набор содержит тесты в статусе *В процессе*, обновление не поддерживается.

Чтобы отредактировать, переименовать, свернуть и развернуть дочерние элементы или удалить набор:

- Наведите курсор на динамический набор в меню **Все наборы**, нажмите **Действия**, затем выберите нужное действие.

Чтобы настроить автообновление динамического набора:

1. Наведите курсор на динамический набор в меню **Все наборы**.
2. В выпадающем меню **Действия** выберите **Редактировать**.
3. Выберите **Обновлять автоматически** в верхней части окна.
4. Нажмите **Применить**.

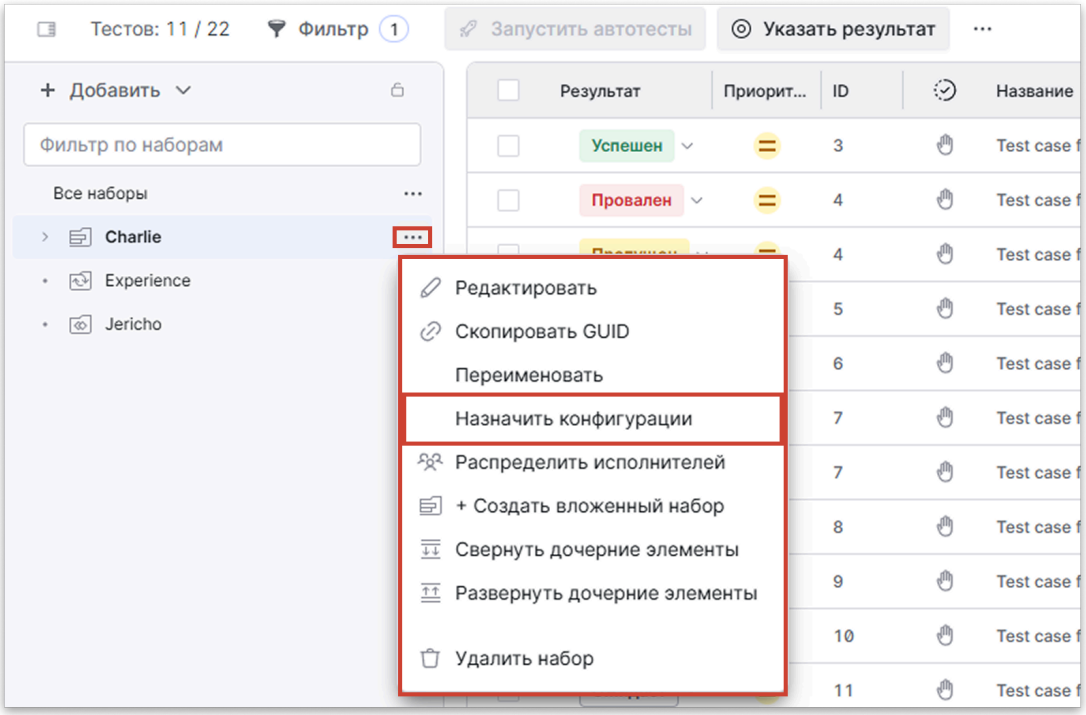
Добавление тестов в тестовый набор

Вы можете добавлять тесты в созданный тестовый набор. Для этого:

1. Откройте меню **Действий** справа от названия тестового набора, в который хотите добавить тесты и выберите **Редактировать**.
2. Выберите тесты, которые хотите добавить в тестовый набор. В редакторе тестового набора тесты отображаются в том же формате, что и в библиотеке тестов, а также разбиты по секциям. Нажмите **Фильтр** над таблицей, чтобы отфильтровать тесты по необходимым вам критериям.
3. Нажмите **Сохранить**.
   Выбранные тесты отобразятся в тестовом наборе.

Добавление конфигураций в тестовый набор

1. В списке тестовых наборов наведите курсор на тестовый набор с добавленными тестами.
2. Справа он названия набора откройте меню **Действия** и выберите **Назначить конфигурации**.



3. Введите названия конфигураций, которые хотите добавить, и выберите их из выпадающего списка. Вы можете выбрать одну из созданных ранее конфигураций или создать новую. Для этого нажмите **Создать конфигурацию**.
   Чтобы узнать больше о создании конфигураций на основе параметров, смотрите Работа с параметрами и конфигурациями.
4. Нажмите **Сохранить**.

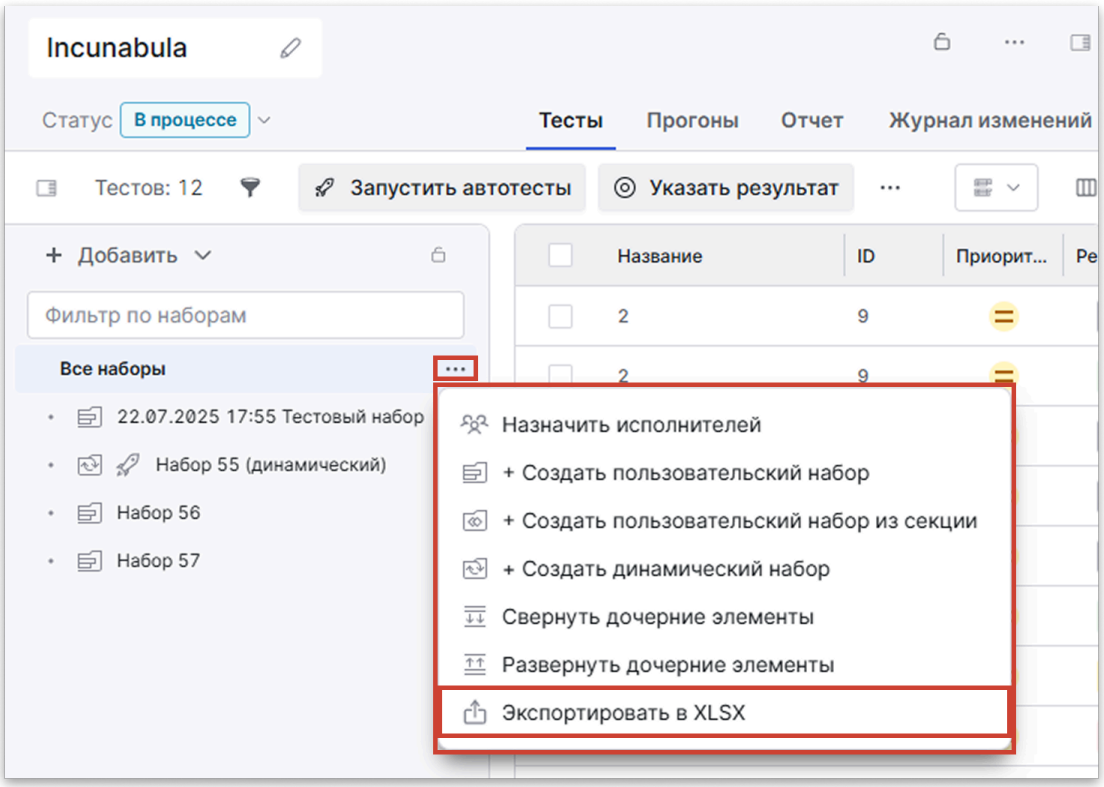
Добавленные конфигурации отображаются в таблице тестов отдельной колонкой. Для каждой конфигурации можно назначить исполнителя.

Экспорт тестовых наборов

> **₽ Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Вы можете загружать тестовые наборы и входящие в них тест-кейсы в формате `.xlsx`. При экспорте происходит загрузка всех наборов, входящих в тест-план. Экспорт динамических тестовых наборов временно недоступен.

1. Откройте проект.
2. В навигационном меню слева нажмите **Тест-планы**.
3. Откройте тест-план.
4. **Опционально:** Выберите тесты для экспорта, применив фильтр, расположенный над списком тестовых наборов.
5. Слева от списка **Все наборы** нажмите **Действия**, затем из выпадающего меню выберите **Экспортировать в XLSX**.



Откроется окно **Экспорт в XLSX**.

6. В окне **Экспорт в XLSX** выберите конфигурации из выпадающего меню, затем нажмите **Сохранить**.
   Файл с тест-кейсами в формате .xlsx будет добавлен в очередь экспорта. Вы можете отслеживать статус экспорта в разделе Фоновые задачи.

Обновлено: 8/18/26, 4:54 PM

Назначение исполнителей тест-плана

После того, как вы сформировали тестовый набор для тест-плана, вы можете перейти к назначению исполнителей тест-поинтов.

Делитесь тест-поинтами с командой

Вы можете скопировать в адресной строке ссылку на отдельный тест-поинт и отправить вашим коллегам в чат или по электронной почте.
Если вы скопировали ссылку на тест-поинт, при переходе по ссылке откроется карточка тест-поинта в соответствующей вкладке раздела **Тест-планы**.

Фильтр поможет найти неназначенные тесты

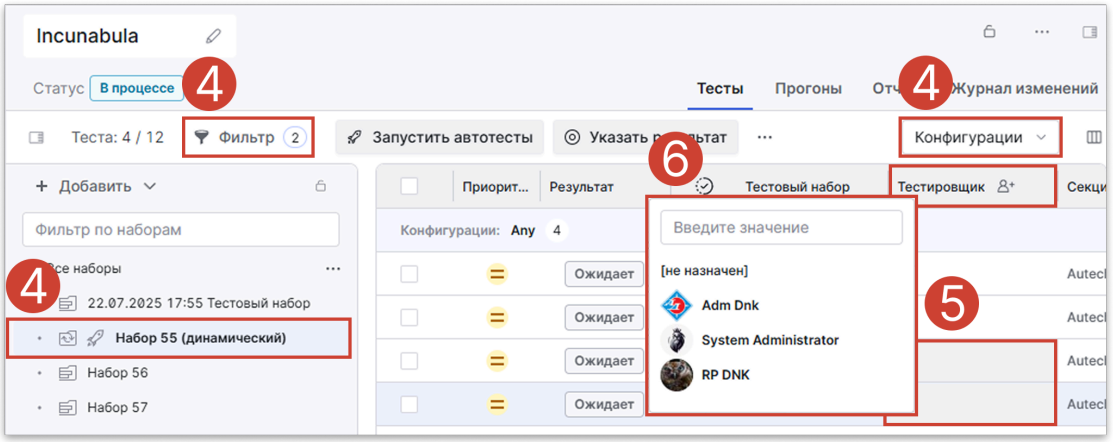
С помощью Фильтра вы легко найдете тесты, на которые не назначен инженер. Для этого в поле **Тестировщик** выберите **Не назначено**.

Назначение исполнителей вручную

Вы можете назначить исполнителя на один или несколько тест-поинтов, входящих в тестовый набор, или на все наборы.

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. **Опционально:** Отсортируйте тесты так, чтобы назначать тестировщика было удобнее. Для этого:
   - Выберите нужный тестовый набор.
   - Воспользуйтесь фильтром.
   - С помощью меню группировки упорядочьте тесты по нужному критерию. Например, вы можете сгруппировать тесты по конфигурациям и на каждую назначить отдельного исполнителя.
   - Отсортируйте тесты по атрибутам в соответствующих колонках.

5. В колонке **Тестировщик** выберите тест-поинты, на которые хотите назначить исполнителя:
   - Чтобы назначить исполнителя на один тест-поинт, в строке теста нажмите на ячейку колонки **Тестировщик**.
   - Чтобы назначить исполнителя на несколько тест-поинтов, в колонке **Тестировщик** выделите несколько тест-поинтов, удерживая левую кнопку мыши зажатой.
     Откроется окно назначения исполнителя. Если для выбранных тестов уже есть тестировщик, он будет переназначен.
6. Выберите исполнителя выделенных тест-поинтов из списка. Если у тест-поинта уже есть исполнитель, и вы хотите отменить назначение (очистить ячейку), выберите **[не назначен]**.



Автоматическое распределение тест-поинтов

> **Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

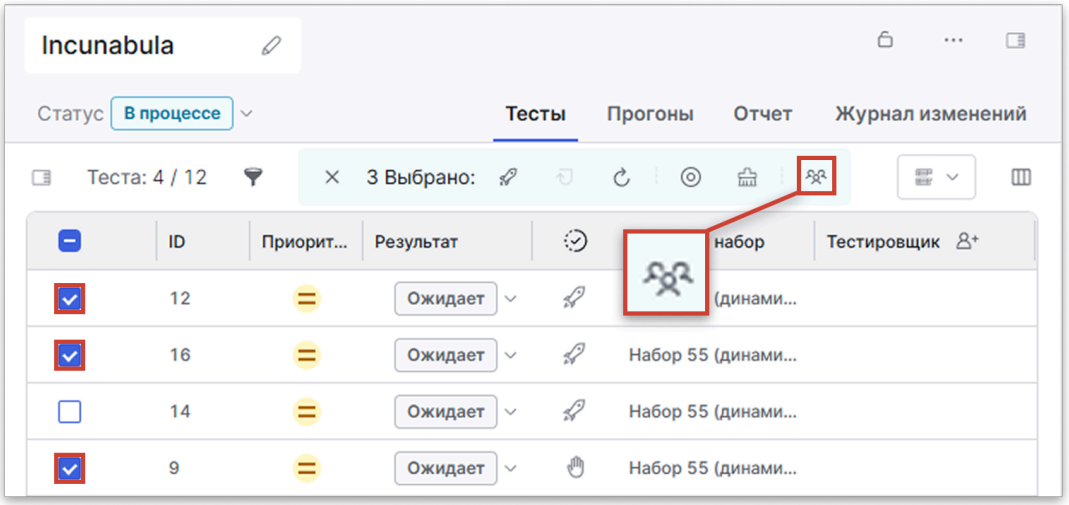
При автоматическом назначении исполнителей тесты распределяются в равной пропорции между выбранными пользователями с учетом количества тест-поинтов и времени их выполнения, указанного в тест-кейсах.

Пример из практики

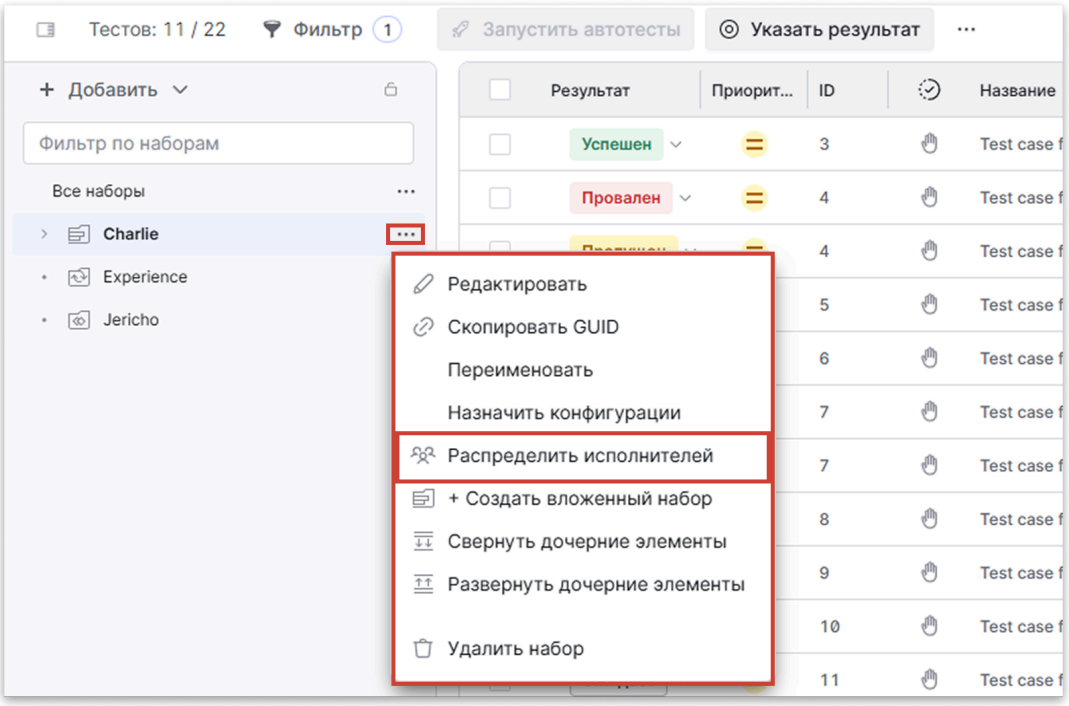
Например, в тест-плане 10 тестов по 1 минуте каждый и 1 тест на 10 минут. Суммарно 11 тестов и 11 минут на выполнение. На тестировщика А будет назначено 10 тестов по 1 минуте, на тестировщика Б — 1 тест на 10 минут. Если тестировщиков 5, на одного из них будет назначен тест на 10 минут, на остальных четверых распределены 10 тестов по 1 минуте в равных пропорциях, если это возможно математически.

Чтобы автоматически распределить тест-поинты:

1. В проекте откройте тест-план.
   Откроется список тест-поинтов.
2. Выберите тесты, на которые хотите назначить исполнителей одним из способов:
   - Чтобы распределить исполнителей на выбранные тест-поинты, в первой колонке списка выберите тест-поинты, отметив их флажками, затем нажмите значок назначения исполнителей.



   - Чтобы распределить исполнителей на тестовый набор, в списке тестовых наборов откройте меню **Действия** справа от названия набора и выберите **Распределить исполнителей**.



Откроется окно назначения исполнителей.

3. В поле **Пользователи** выберите пользователей, между которыми хотите распределить тест-поинты.
4. Нажмите **Сохранить**.

Обновлено: 8/18/26, 4:54 PM

Выполнение тест-плана

Используйте горячие клавиши

Для действий с тест-поинтами вы можете воспользоваться горячими клавишами.

Выбор назначенных тест-поинтов

После формирования тестового набора и назначения исполнителей тест-плана вы можете перейти к выполнению тест-плана.

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. Отфильтруйте тест-поинты, назначенные на вас.

В таблице отобразятся тест-поинты, назначенные на вас.

Делитесь планами и тестами с командой!

- Вы можете скопировать URL тест-плана, чтобы поделиться с коллегами. Параметры **Фильтра**, заданные вами, сохранятся.
- Вы можете скопировать URL тест-поинта. При переходе по ссылке откроется карточка тест-поинта в соответствующей вкладке раздела **Тест-планы**.

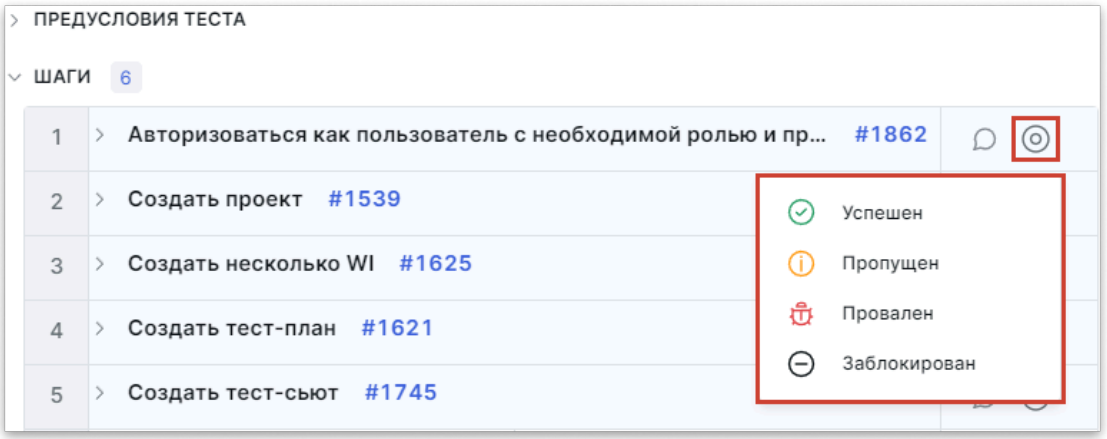
Запуск автотестов

Вы можете найти информацию по запуску автотестов из тест-планов в соответствующем разделе.

Отметка результатов шагов теста

Вы можете указывать результаты для отдельных шагов теста и сохранять тесты в статусе *В процессе*.

1. Откройте тест-поинт.
2. Запустите таймер прохождения теста, если он настроен на ручной режим при создании тест-плана.
3. По прохождении шага, укажите результат: *Успешен*, *Пропущен*, *Провален* или *Заблокирован*.



4. Вы можете оставить комментарий к результату шага.
5. Нажмите **Сохранить**.

После сохранения тест-поинта таймер останавливается и открывается следующий тест-поинт из тестового набора. Сохраненный тест-поинт отображается в таблице в статусе **В процессе**. Вы можете открыть тест-поинт и продолжить его прохождение.

> **Запустите таймер вручную**
>
> При продолжении прохождения теста необходимо запустить таймер вручную.

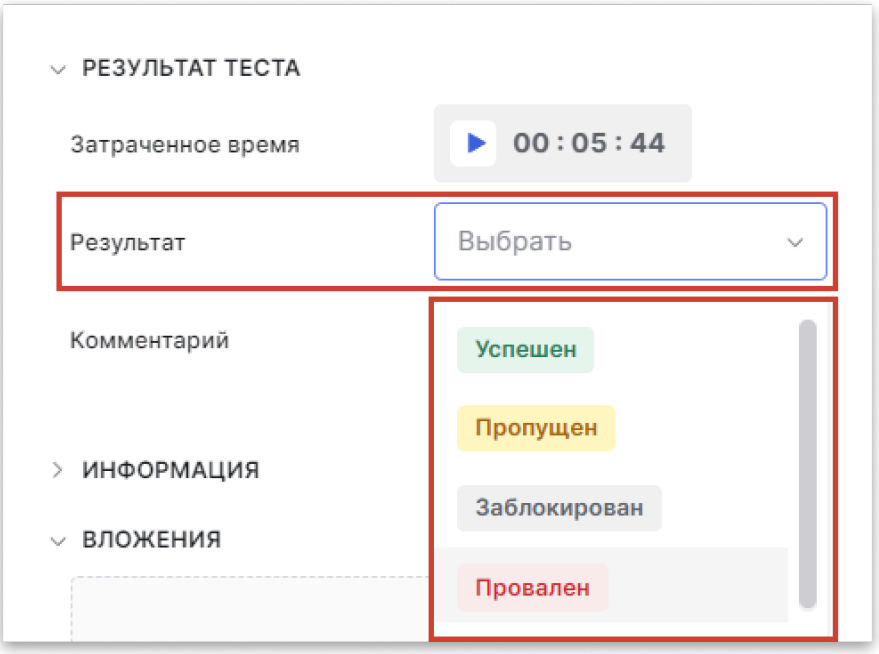
Отметка результата одного теста

Вы можете указать результат одного теста:

- Со страницы теста — указывается результат, есть возможность учитывать время тестирования, добавлять комментарии и прикреплять файлы
- Со страницы тест-плана — указывается только результат (без возможности учитывать время, комментировать и прикреплять файлы) После указания результата теста вы будете назначены на него в качестве исполнителя.

Отметка результата со страницы теста

1. Откройте тест-поинт.
2. Запустите таймер прохождения теста, если он настроен на ручной режим при создании тест-плана.
3. По прохождении теста в поле **Результат** выберите: *Успешен*, *Пропущен*, *Провален*, *Заблокирован*.



4. Вы можете оставить комментарий к результату теста в поле **Комментарий**. Если хотите добавить вложения, вы можете перетащить их в область загрузки в разделе **Вложения** или загрузить их с помощью Проводника.
5. Нажмите **Сохранить**.

Отметка результата из тест-плана

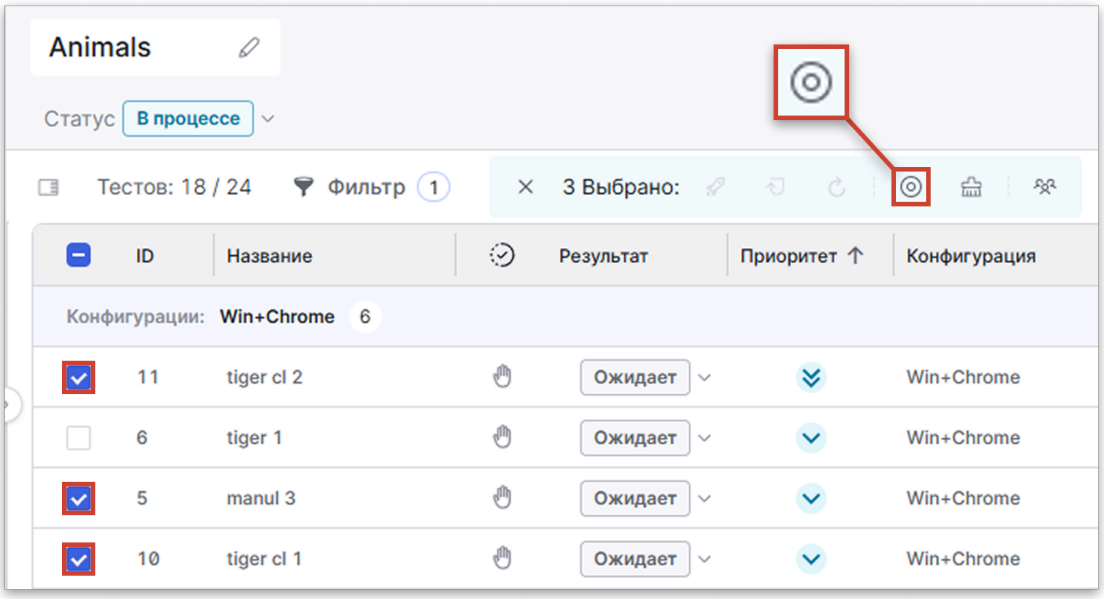
1. В проекте откройте тест-план.
2. В колонке **Результат** выберите *Успешен*, *Пропущен*, *Провален* или *Заблокирован*.

Test IT сохранит результат прохождения теста и обновит его статус в тест-плане в зависимости от присвоенного результата. Вы можете сбросить результат теста, пройти его еще раз и присвоить ему новый результат. Test IT сохранит все присвоенные этому тесту результаты.

Массовая отметка результатов тестов

Вы можете отметить результаты нескольких тест-поинтов одновременно. После указания результатов тестов вы будете назначены на него в качестве исполнителя.

1. В проекте откройте тест-план.
2. Выберите тест-поинты, для которых хотите указать результат.
3. Нажмите значок массового указания результата.



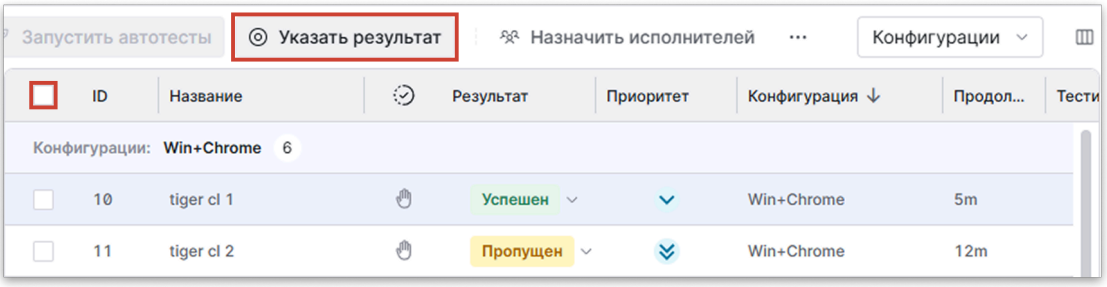
4. Во вкладке **Статус** укажите результат, который хотите указать для выбранных тестов: *Успешен*, *Пропущен*, *Провален* или *Заблокирован*.
5. При необходимости вы можете добавить в результат ссылки, комментарии и вложения в соответствующих вкладках.
6. Нажмите **Сохранить**.

Указание результатов для всех тестов

Чтобы указать результат для всех тестов в выбранном наборе или для всех наборов:

1. В проекте откройте тест-план. При необходимости откройте набор, для которого хотите указать результат.
2. Убедитесь, что ни один из тестов в таблице не выбран.

3. Нажмите **Указать результат**.



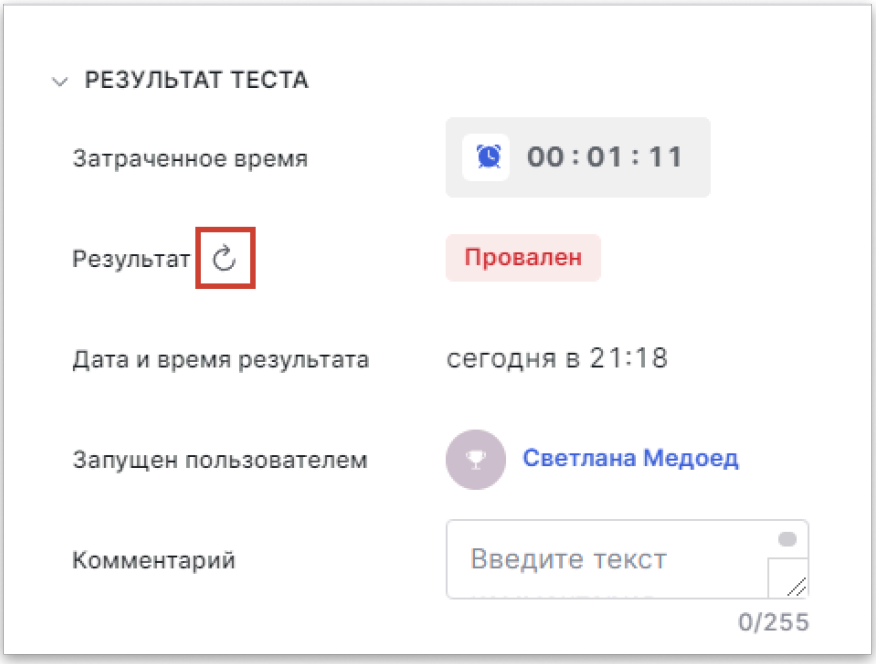
4. Во вкладке **Статус** укажите результат, который хотите указать для выбранных тестов: *Успешен*, *Пропущен*, *Провален* или *Заблокирован*.
5. При необходимости вы можете добавить в результат ссылки, комментарии и вложения в соответствующих вкладках.
6. Нажмите **Сохранить**. :::

Test IT сохранит результат прохождения отмеченных флажками тестов и обновит их статус в тест-плане в зависимости от присвоенного результата.

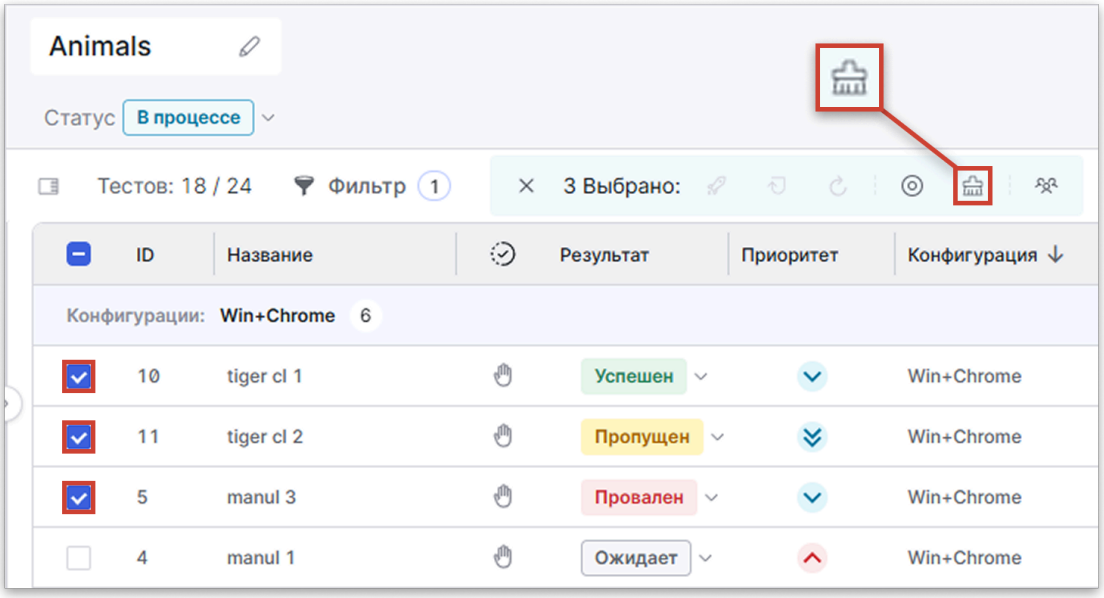
Повторное прохождение тестов

Вы можете проходить тест неограниченное количество раз. Для этого необходимо сбросить текущий результат теста и пройти его заново. Вы можете сбрасывать результат конкретного теста или нескольких тестов одновременно.
После указания результатов тестов вы будете назначены на него в качестве исполнителя.

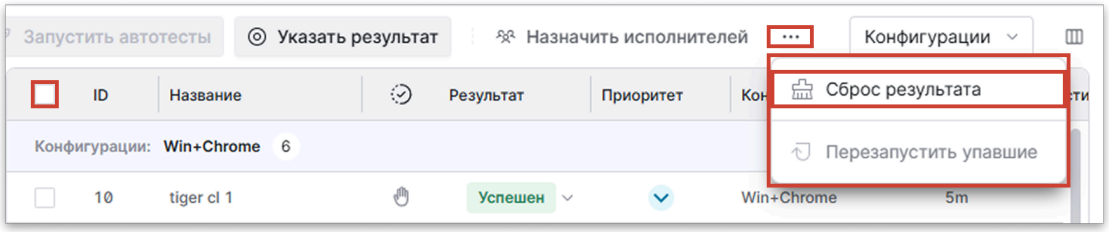
- Для сброса результата конкретного теста откройте его и нажмите на значок сброса.



- Для массового сброса результатов выделите тесты флажками, затем нажмите значок массового сброса результата, напоминающий **щётку**.



- Для сброса результатов всех тестов очистите выбор тест-поинтов в первой колонке слева, затем откройте меню действий справа от кнопки **Назначить исполнителей** и выберите **Сбросить результат**.

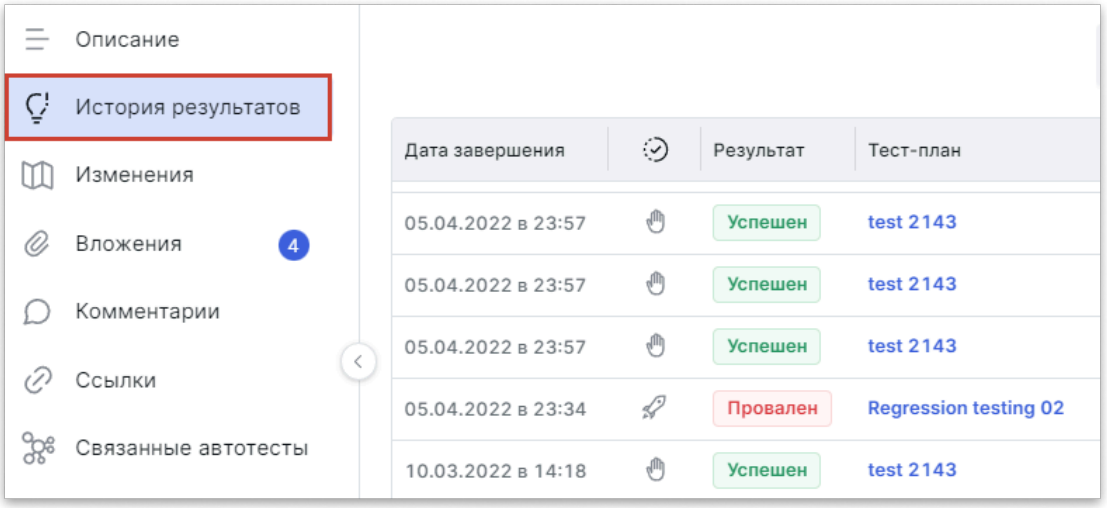


После сброса результатов тестов вы можете пройти их повторно. Test IT сохраняет все результаты, указанные для тест-поинта.

Отслеживание истории результатов

Test IT сохраняет все результаты прохождения тест-поинтов. Чтобы посмотреть, какие результаты были указаны для тест-поинта в рамках выполнения тест-плана:

1. Выйдите из тест-плана.
2. Перейдите в раздел **Библиотека тестов**.
3. Откройте тест-кейс, для которого хотите отследить результаты.
4. Перейдите в раздел **История результатов**.



В открывшемся окне отображается таблица результатов теста. Воспользуйтесь фильтром, чтобы отсортировать содержимое таблицы. Вы можете открывать результаты тестов, чтобы посмотреть результаты шагов, комментарии, ссылки и вложения, прикрепленные к результату.

Обновлено: 4/15/26, 7:08 PM

Создание дефектов в таск-трекерах

> **Требуется настроенная интеграция и подключение проекта**
>
> Чтобы создавать тест-кейсы по задачам в вашем таск-трекере, нужно настроить и протестировать интеграцию с таск-трекером и добавить ее в проект.

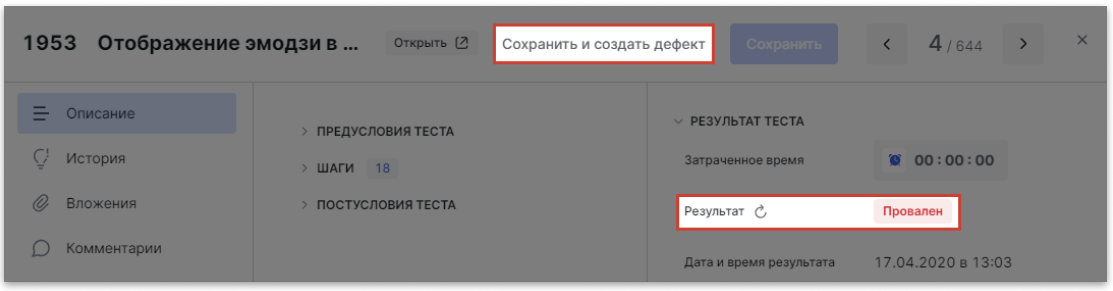
Если при прохождении тест-плана вы обнаружили дефект, можно создать задачу в таск-трекере из интерфейса Test IT. При создании дефекта задача в таск-трекере связывается с тестом, в котором обнаружен дефект, и с результатом прогона теста, на котором был обнаружен дефект. Таким образом, тестировщики могут быстро создавать задачи для команды разработки, которой не требуется доступ к Test IT.

Из тест-плана можно создавать дефекты для ручных и автоматизированных тестов.

Создание дефектов для ручных тестов

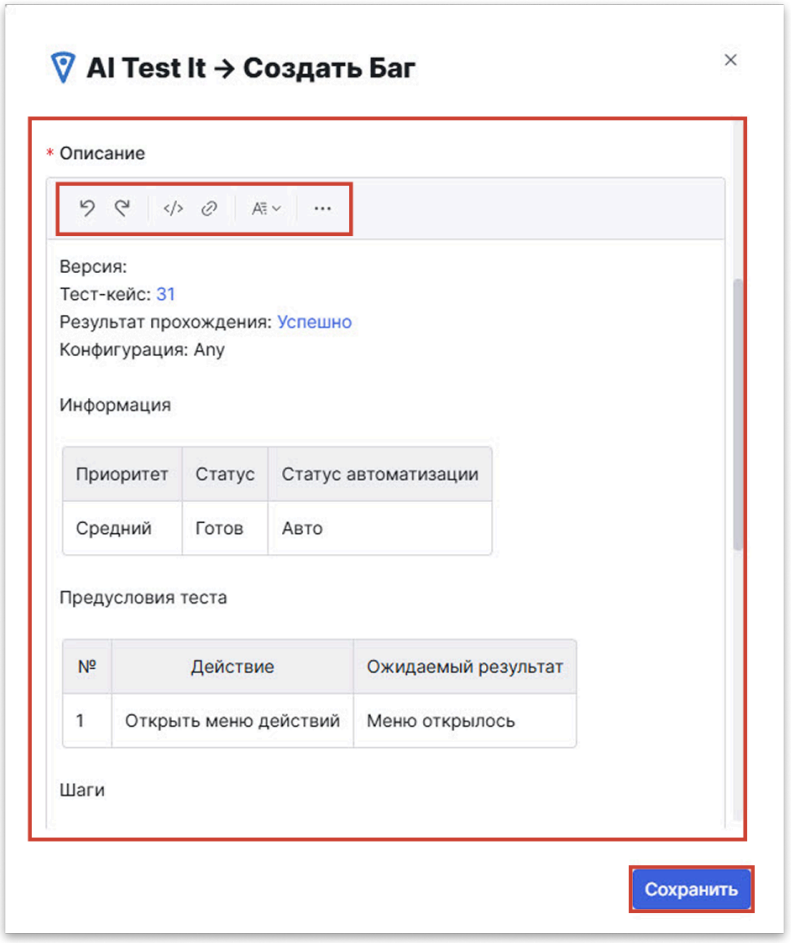
Чтобы создать дефект из Test IT:

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. Откройте тест с результатом *Пропущен*, *Заблокирован* или *Провален*, для которого хотите создать дефект.
5. Нажмите **Сохранить и создать дефект**.



Если вы настроили интеграцию с несколькими проектами в таск-трекере, выберите проект, в котором хотите создать дефект.

6. В открывшемся окне таск-трекера заполните недостающие данные о дефекте (при необходимости).
7. **Опционально:** Отредактируйте описание дефекта, используя панель инструментов в секции **Описание**.
8. Нажмите **Сохранить**.



После создания дефекта в таск-трекере появится ссылка на тест и на результат его прохождения, во время которого нашли дефект. Вы также можете отследить, на какой конфигурации был обнаружен дефект.

Чтобы отследить ссылку на созданный дефект:

- Перейдите в раздел **Ссылки** в окне результата теста.

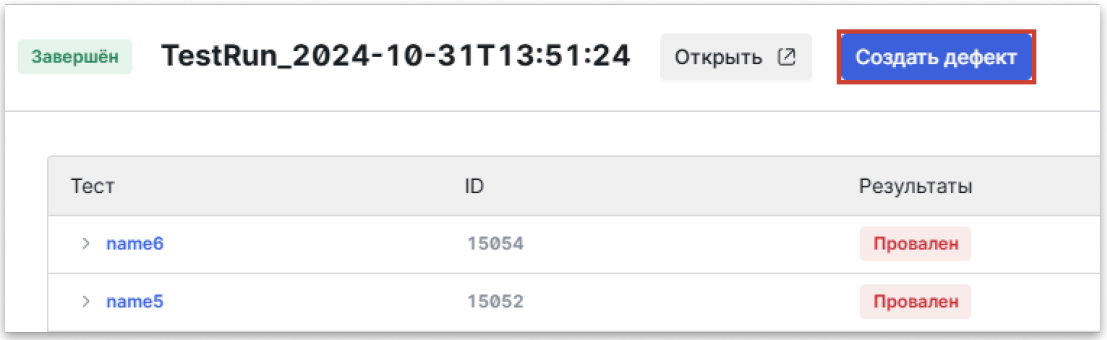
  Вы можете отслеживать статус дефекта, его приоритет и исполнителя прямо в Test IT, наведя на него курсор.

Создание дефектов для автотестов из тест-плана

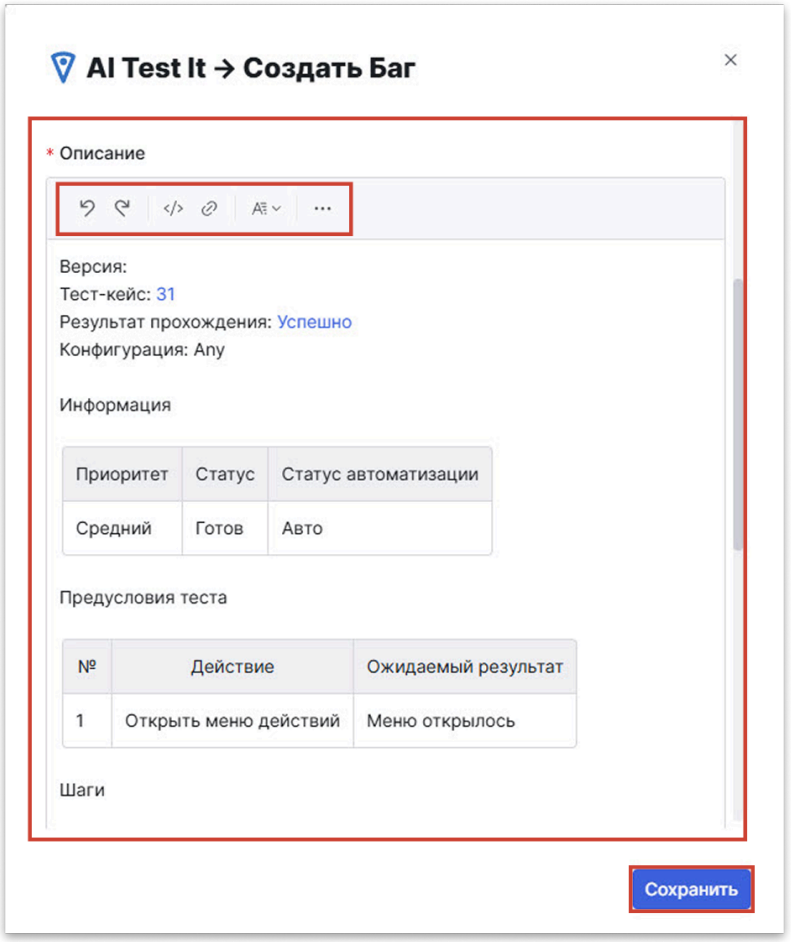
Из раздела тест-планов можно создавать дефекты для автотестов со статусом *Провален*. Для автотестов с любым статусом можно создать дефект из раздела Прогоны автотестов.

Чтобы создать дефект:

1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. На странице тест-плана перейдите во вкладку **Прогоны**.
5. Откройте прогон, содержащий проваленные автотесты.
6. Нажмите **Создать дефект**.



   Если вы настроили интеграцию с несколькими проектами в таск-трекере, выберите проект, в котором хотите создать дефект.
7. В открывшемся окне таск-трекера заполните недостающие данные о дефекте (при необходимости).
8. **Опционально:** Отредактируйте описание дефекта, используя панель инструментов в секции **Описание**.
9. Нажмите **Сохранить**.



После создания дефекта в таск-трекере появится ссылка на тест и на результат его прохождения, во время которого нашли дефект. Вы также можете отследить, на какой конфигурации был обнаружен дефект.

Чтобы отследить ссылку на созданный дефект:

- Перейдите в раздел **Ссылки** в окне результата теста.

  Вы можете отслеживать статус дефекта, его приоритет и исполнителя прямо в Test IT, наведя на него курсор.

Обновлено: 9/8/26, 6:41 PM

Отчет по тест-плану

> **₽ Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Вы можете отслеживать текущее состояние тест-плана с помощью отчета. Чтобы открыть отчет:

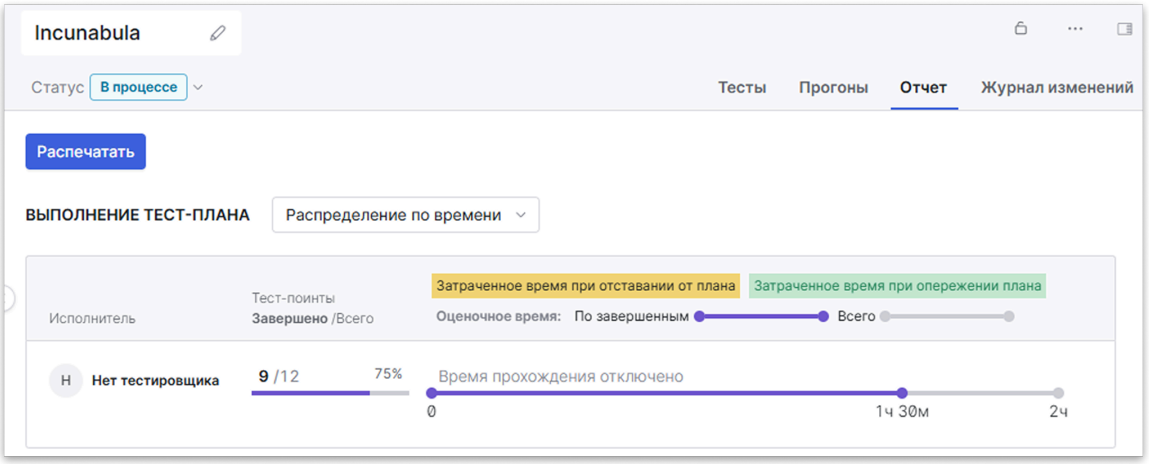
1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Перейдите на вкладку **Отчет**.

В отчете представлена аналитическая информация по тест-плану.
Вы можете:

- Отслеживать загрузку тестировщиков в секции **Выполнение тест-плана**. Если один из тестировщиков перегружен, вы можете перераспределить нагрузку
- Анализировать прогресс выполнения тест-плана с помощью Диаграммы сгорания задач
- Экспортировать отчет по результатам тестов в формате `XLSX`
- Экспортировать отчет по тест-плану в формате `PDF`

Выполнение тест-плана

В отчете секция **Выполнение тест-плана** показывает распределение тест-поинтов по специалистам (тестировщикам). Также вы можете отслеживать количество дефектов, заведенных в процессе выполнения тест-плана (включая основную информацию о дефекте, статус и приоритет дефекта в Jira) и отслеживать результаты тестов, даже если их запускали несколько раз.



- Чтобы отобразить время, затраченное на выполнение тест-поинтов каждым тестировщиком, а также количество завершенных тест-поинтов из их общего числа, в выпадающем меню выберите **Распределение по времени**. Чтобы просмотреть запланированное время, затраченное время и отставание от графика, наведите курсор на шкалу **Оценочное время**.
- Чтобы отобразить статус тест-поинтов, назначенных на каждого тестировщика, выберите **Распределение по тест-поинтам**. Чтобы просмотреть количество тест-поинтов и их статус, наведите курсор на шкалу тест-поинтов справа от имени тестировщика.

Диаграмма сгорания задач

Диаграмма сгорания задач помогает анализировать прогресс выполнения тест-плана и оценивать, насколько прохождение всех тест-поинтов, входящих в тест-план, соответствует установленным срокам.

Диаграмма находится во вкладке **Отчет** раздела **Тест-планы**. Она отображается, когда тест-план переходит из статуса *Новый* в статус *В процессе*.

На диаграмме представлены следующие данные:

- Период исполнения тест-плана — ось X
- Количество тест-поинтов или оценочная продолжительность их выполнения (в зависимости от выбранного режима отображения) — ось Y

На основании данных тест-плана формируются два графика:

- **Идеальный план** (идеальный график сгорания задач) — сценарий, при котором тест-план выполняется в соответствии с установленными сроками (все тест-поинты будут пройдены в соответствии с датой окончания тест-плана). Идеальный план формируется на основании тест-поинтов, включенных в тест-план на момент его начала и планируемой даты окончания.
- **Оставшиеся тест-поинты** (график фактического сгорания задач) — количество невыполненных тест-поинтов, входящих в тест-план. График изменяется при выполнении, добавлении, удалении тест-поинтов и очистке их результатов. Если таких изменений не происходит, с течением времени, график фактического сгорания задач проходит параллельно оси X. Данные графика обновляются каждые 10 минут.

Сопоставляя эти графики, вы можете оценить, выполняется ли тест-план в соответствии с установленными сроками.

> **Учитывайте статус тест-плана**
>
> - Если на момент перехода тест-плана в из статуса *Новый* в статус *В процессе* тест-план не содержит тест-поинтов, идеальный план не отобразится.
> - После перехода тест-плана в статус *Завершен* изменений в диаграмме не происходит.

Вы можете настраивать отображение диаграммы сгорания задач:

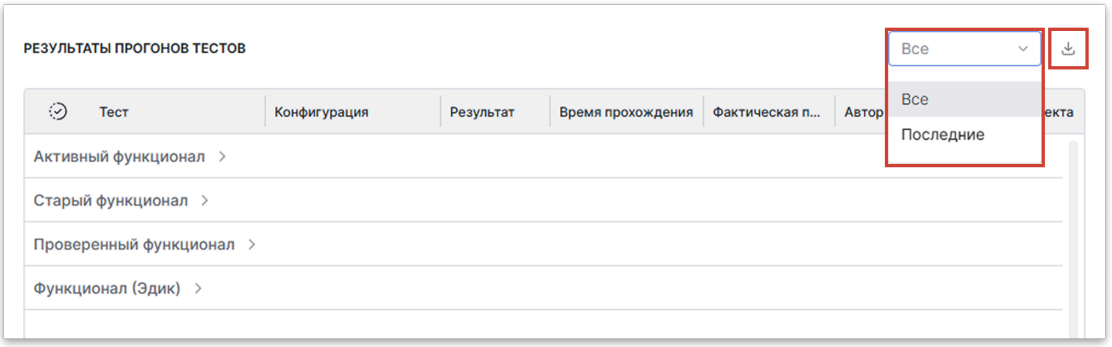
- Чтобы переключать отображение тест-поинтов или время их выполнения на оси Y, выберите **Время** или **Тест-поинты** из выпадающего меню **График сгорания задач по тест-плану**.
- Чтобы увеличить или уменьшить детализацию оси X, используйте шкалу в нижней части диаграммы или колесико мыши.

Экспорт отчета по результатам тестов в XLSX

Вы можете экспортировать данные по результатам тестов в `.xlsx` файл. Файлы с отчетами доступны для загрузки в Фоновых задачах.

1. Откройте тест-план и выберите вкладку **Отчет**, затем перейдите к секции **Результаты прогонов тестов**.

2. **Опционально:** В меню, расположенном слева от значка экспорта, выберите, какие результаты вы хотите загрузить:
   - **Все** — результаты всех прогонов тестов, если тесты запускались более одного раза
   - **Последние** — результаты последнего запуска тестов
3. Нажмите значок экспорта.



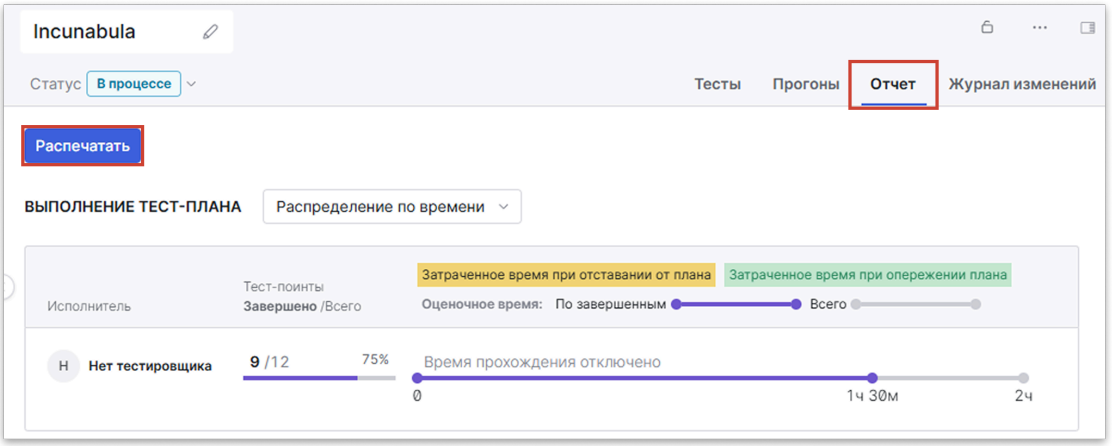
Откроется окно экспорта отчета.

4. **Опционально:** Выберите опцию **Включить шаги** чтобы включить в отчет шаги тестов. В окне экспорта вы также можете выбрать, какие результаты вы хотите экспортировать: все или последние.
5. Нажмите **Экспортировать**. Отобразится сообщение о начале экспорта с кнопкой **Перейти к фоновым задачам**.
6. Скачайте отчет из модуля Фоновые задачи, когда экспорт будет завершен.

Экспорт отчета в PDF

Отчет по тест-плану можно распечатать или сохранить в формате `PDF`.

1. Нажмите **Распечатать**.



2. Выберите данные, которые хотите включить в версию для печати, и нажмите **Распечатать**. Версия для печати откроется в новой вкладке.

3. Укажите данные, которые будут включены в отчет.
   Данные, которые вы указали, сохранятся в вашем браузере.
4. Нажмите **Распечатать**.

Обновлено: 8/18/26, 4:54 PM

Работа с автоматизированными тестами

Интегрируйтесь с системами автоматизированного тестирования

Test IT предоставляет широкие возможности для интеграции с системами автоматизированного тестирования. Подробности читайте в соответствующем разделе.

Быстрый старт автотестирования

Test IT поможет подготовить стартовую сборку для вашего языка программирования, фреймворка и проекта. Укажите нужные данные и получите архив со сборкой.

Страница сборки доступна по ссылке , а также в меню **Информация** в верхнем правом углу окна, в разделах **Библиотека автотестов** и **Прогоны** (если они пока не содержат прогонов и автотестов).

Настройте уведомления об изменениях автотестов

С Test IT вы можете получать в мессенджер уведомления о событиях, происходящих с автотестами. Для этого нужно настроить вебхук, указав необходимые переменные.

Из интерфейса Test IT вы можете управлять автотестами, расположенными в вашем репозитории: запускать автотесты из системы, отслеживать их прогоны (тест-раны), историю результатов и многое другое. Вы можете формировать аналитику по результатам прогонов автотестов, а также создавать категории ошибок, чтобы автоматически определять причины проваленных результатов автотестов.

Данный раздел описывает следующие сценарии работы с автоматизированными тестами:

- Добавление автотестов в Test IT из внешнего репозитория и действия с ними
- Работа с нестабильными автотестами

- Связывание автотестов с тест-кейсами
- Обновление тест-кейсов на основе связанных автотестов
- Запуск автотестов из UI
- Запуск автотестов с помощью клиентских библиотек
- Импорт результатов из Allure Adapters
- Отслеживание запусков автотестов
- Создание дефектов в таск-трекерах для автотестов
- Анализ результатов автотестов
- Таймлайн
- Управление прогонами автотестов

Обновлено: 7/21/26, 5:46 PM

Добавление автотестов в Test IT из внешнего репозитория и действия с ними

- **Для этого раздела мы подготовили видео:**

Настройте уведомления о добавлении, изменении и удалении автотестов

С Test IT вы можете получать в мессенджер уведомления о добавлении, редактировании и удалении автотеста. Для этого нужно настроить вебхук, указав необходимые переменные.

Чтобы управлять автотестами, расположенными в вашем репозитории, из интерфейса Test IT, нужно добавить автотесты в систему, создав их карточки. После этого вы сможете запускать автотесты из UI Test IT, отслеживать результаты их прогонов (тест-ранов), присваивать автотестам статус нестабильности, привязывать автотесты к рабочим элементам и совершать другие действия.

Вы можете добавить автотесты в систему Test IT:

- С помощью API
- С помощью клиентских библиотек

Добавление автотеста через API

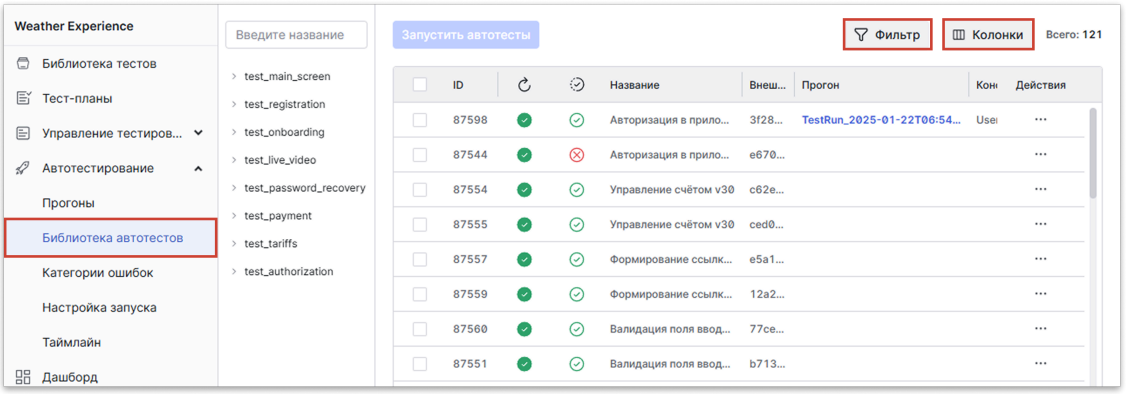
1. Авторизуйтесь в Swagger.
2. В секции **AutoTests** раскройте метод `POST /api/v2/autoTests` .
3. Нажмите **Try it out**. Тело запроса откроется для редактирования.
4. В запросе укажите: external ID, ссылку на автотест, ID проекта, предусловия тестов, постусловия тестов и шаги.
5. Нажмите **Execute**. В случае успешного выполнения запроса, API вернет код 201.

После создания карточки автотеста через API автотест отображается в системе Test IT.

Просмотр автотестов

Чтобы найти карточку автотеста:

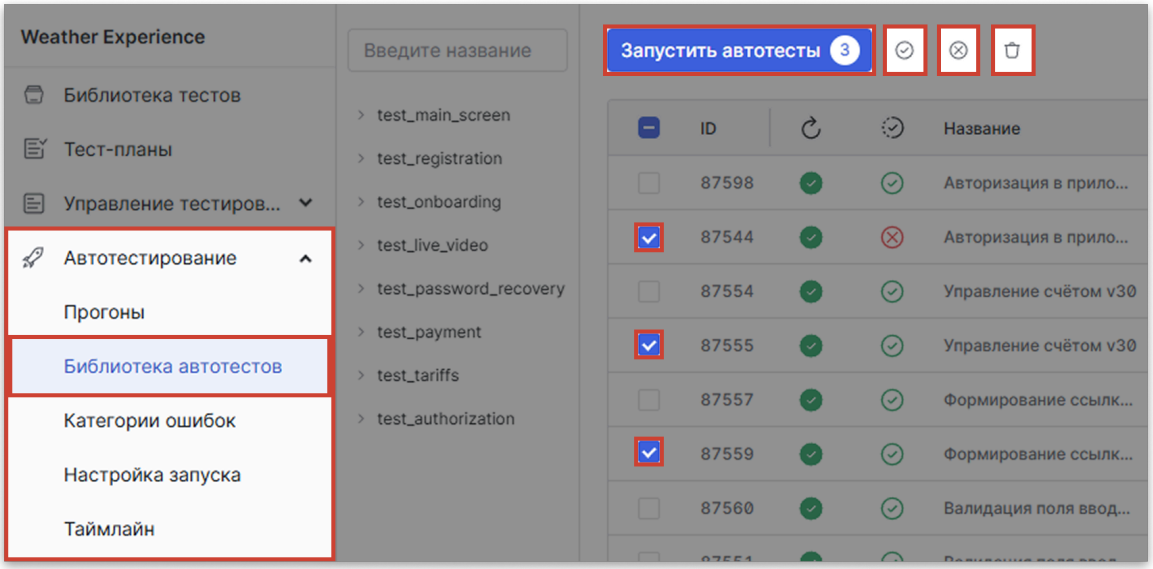
1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Библиотека автотестов**. В данном разделе отображается список автотестов, добавленных в ваш проект. В нем отображены названия автотестов, автор, ссылки на репозиторий с автотестами, внешние ID, последние прогоны (тест-раны) и их результаты.



3. **Опционально:** Для точного поиска используйте Фильтр или выберите нужные **Колонки**.

Массовые действия с автотестами

Для применения массовых действий к автотестам необходимо выбрать несколько из них. Кнопки массовых действий расположены над списком автотестов.



Вы можете:

- Запустить автотесты. Они будут сформированы и запущены в рамках автоматически созданного прогона.
- Отметить автотест как стабильный, используя значок ⊘ или нестабильный, используя значок ⊗.
- Удалить автотесты, используя значок удаления 🗑. Удаляются только карточки автотестов (во внешнем репозитории автотесты сохраняются). Удаление происходит безвозвратно.

Статус нестабильности

Вы можете присвоить автотесту статус нестабильности — flaky.

Статус — это информационная метка, по которой можно строить аналитику.

Статус нестабильности присваивается вручную (через интерфейс или API). Система помогает выявить нестабильные автотесты с помощью расчета процента стабильности автотеста. Процент стабильности можно посмотреть в описании автотеста в поле **Стабильность прохождения**.

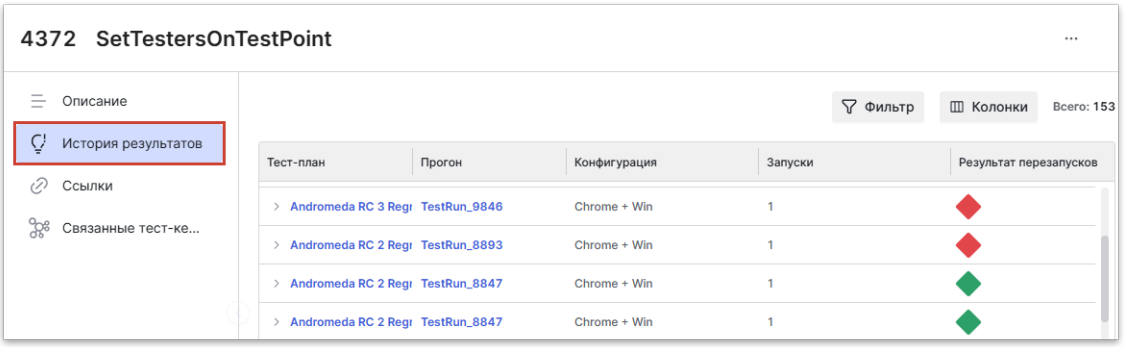
Подробнее: Работа с нестабильными автотестами.

История результатов автотестов

Вы можете посмотреть историю результатов автотестов: тест-план, прогон, конфигурацию, количество и результаты перезапусков и другие данные. Любой исторический результат можно детализировать для просмотра дополнительной информации.

Чтобы просмотреть историю прохождения автотеста:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Библиотека автотестов**.
   Откроется окно автотеста.
3. Откройте автотест.
4. В меню слева откройте вкладку **История результатов**.
   Откроется список результатов автотеста.



5. **Опционально**: Чтобы посмотреть детализированную информацию исторического результата, нажмите на строку результата в списке.

Обновлено: 8/12/26, 4:43 PM

Работа с нестабильными автотестами

Вы можете помечать автотесты в системе как нестабильные, чтобы анализировать стабильность ваших автотестов. Отметить автотест как нестабильный (flaky) можно:

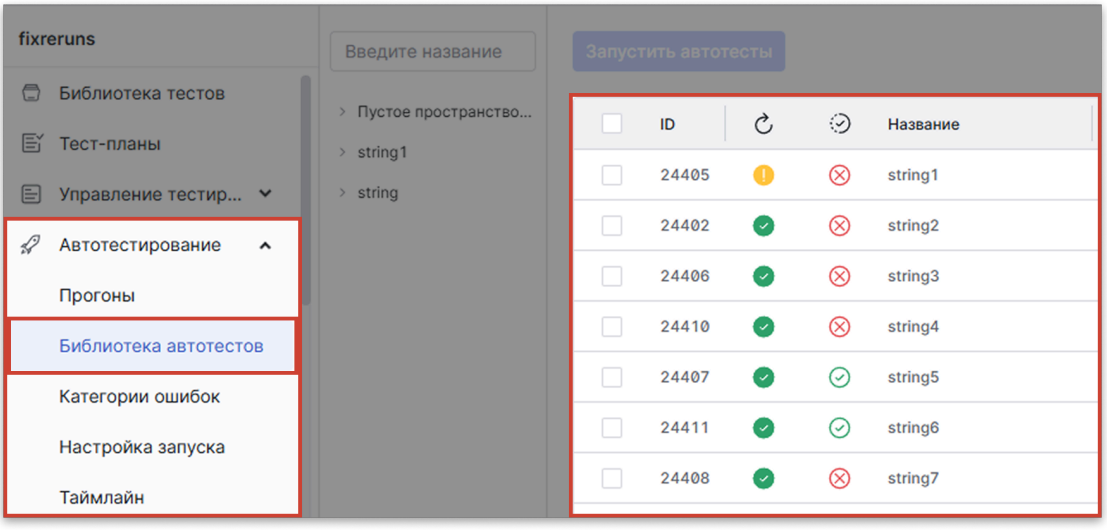
- В пользовательском интерфейсе Test IT
- Автоматически (необходимо произвести настройки)
- С помощью API

Идентификация нестабильных автотестов

Test IT позволяет идентифицировать нестабильные тесты с помощью статистики запусков. По умолчанию система отслеживает историю запусков и формирует статистику по последним 100 запускам автотеста. Частота прохождений с разными результатами влияет на процент стабильности автотеста. Стабильным считается автотест, который за последние 100 запусков дает один и тот же результат.

Чтобы идентифицировать нестабильный автотест:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Библиотека автотестов**.



3. Откройте карточку автотеста.

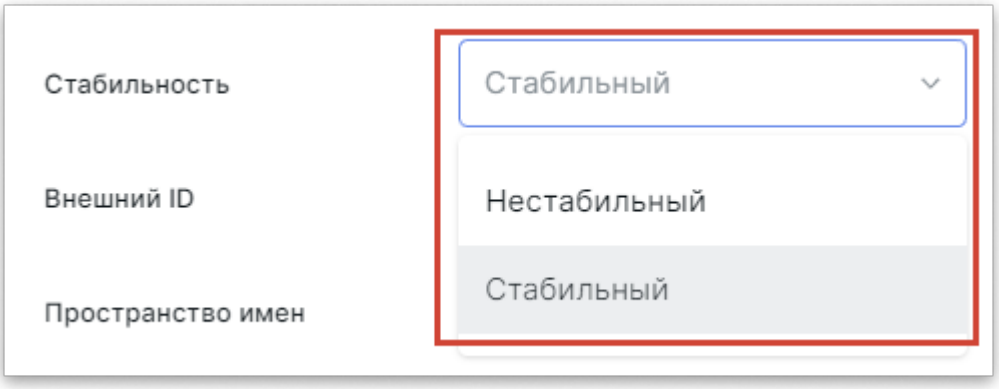
4. Проверьте поле **Стабильность прохождения** в описании автотеста. Если стабильность теста менее 100%, вы можете пометить тест как нестабильный.

Вы можете проанализировать ошибки нестабильных автотестов и создать категории ошибок, чтобы система находила их автоматически.

Проставление отметки о нестабильности через UI

Проставление отметки о нестабильности для одного автотеста

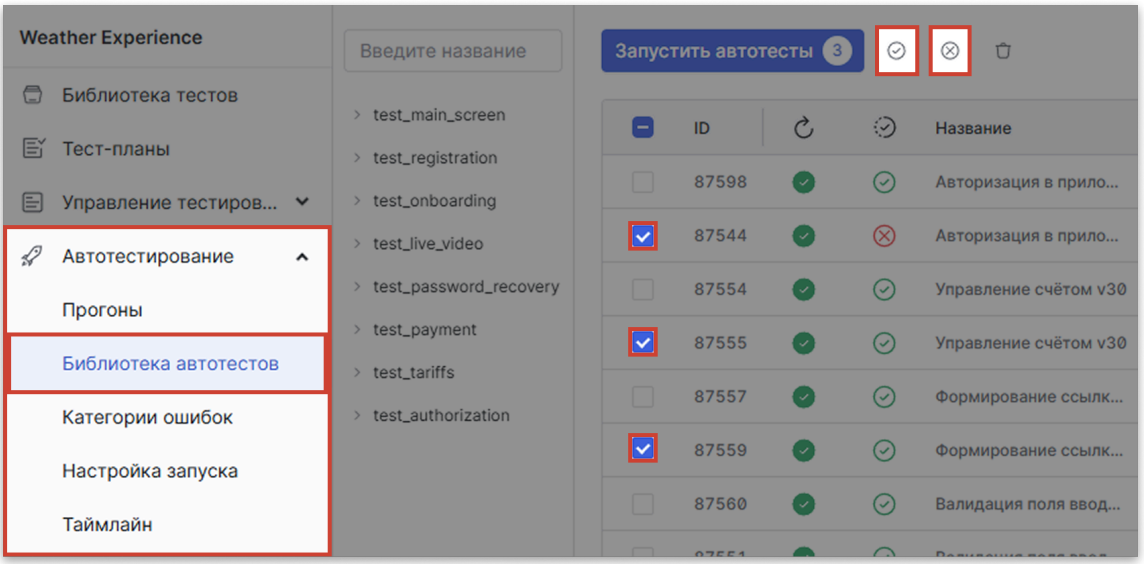
1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Библиотека автотестов**.
3. Откройте карточку автотеста.
4. В описании автотеста найдите поле **Стабильность** и выберите статус в выпадающем меню.



Проставление отметки о нестабильности для нескольких автотестов

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Библиотека автотестов**.
3. Отметьте автотесты, которым хотите поставить отметку о нестабильности.
4. Над списком автотестов нажмите **Отметить автотест как нестабильный**, для этого нажмите значок ⊗.

- **Опционально**: Отметьте тест как стабильный, для этого нажмите значок ⊘.



Смотрите также: Массовые действия с автотестами.

Автоматическая отметка нестабильных автотестов

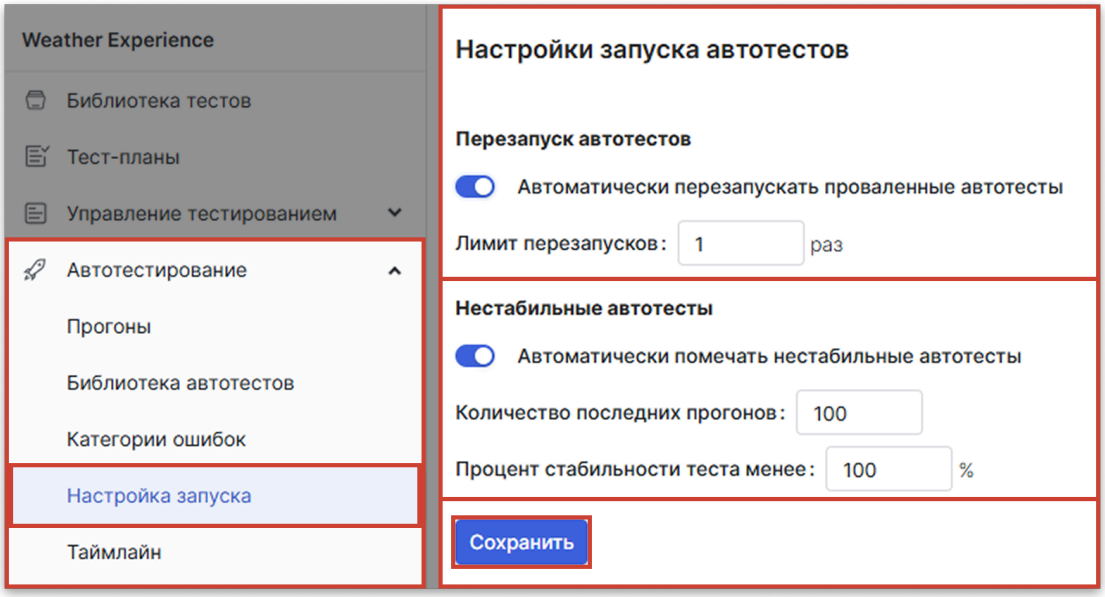
> **₽ Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Test IT может автоматически отмечать тесты как нестабильные. Для этого нужно включить соответствующую опцию в настройках проекта (по умолчанию отключена).

> **Как рассчитывается стабильность автотестов**
>
> Стабильность рассчитывается как отношение количества изменений результатов к их общему количеству за последние 100 прогонов. В подсчете стабильности участвуют результаты *Успешен*, *Провален*, *Пропущен* и *Заблокирован*. Подсчет начинается со второго результата, первый результат не учитывается.
> Стабильность теста не означает его успешность. Например, последние 100 прогонов Тест 1 прошел с результатом *Успешен*, а Тест 2 — с результатом *Провален*. Стабильность обоих тестов составит 100%.

Чтобы включить и настроить автоматическую отметку нестабильных автотестов:

1. Откройте проект.
2. В навигационном меню слева откройте раздел **Автотестирование → Настройка запуска**.



3. Включите опцию **Автоматически помечать нестабильные автотесты**.
4. Укажите количество последних прогонов и процент стабильности автотестов. На основании этих данных система будет определять нестабильные автотесты.
5. Нажмите **Сохранить**.

Проставление отметки о нестабильности через API

Чтобы отметить тесты как нестабильные с помощью API:

- При создании автотеста с помощью метода `POST /api/v2/flaky/bulk` укажите значение параметра `isFlaky` как `true`.

Автотест, созданный в системе, будет автоматически отмечен как нестабильный.

Обновлено: 8/18/26, 4:54 PM

Связывание автотестов с ручными тестами

Настройте уведомления об изменении связи автотестов с рабочими элементами

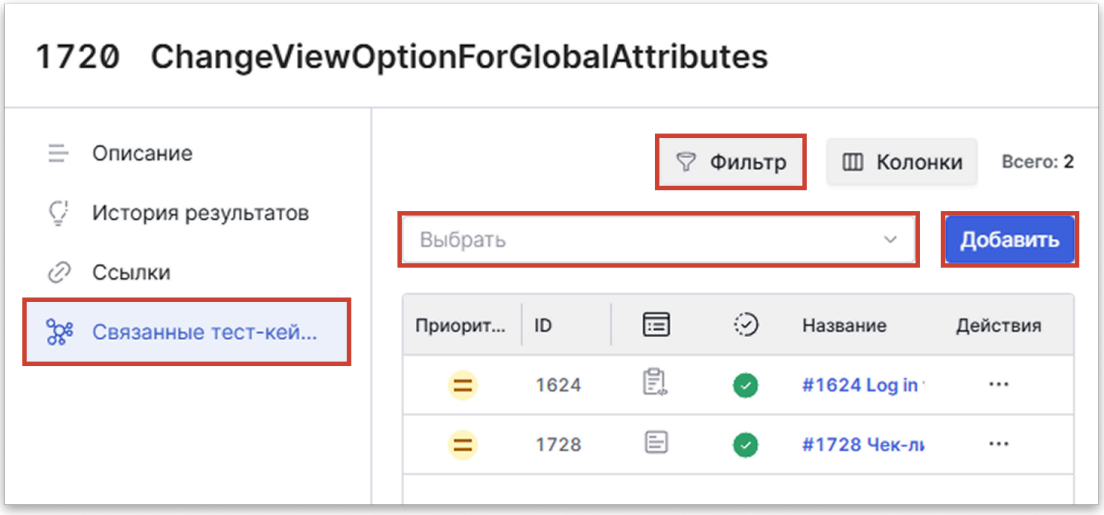
С Test IT вы можете получать в мессенджер уведомления о создании или удалении связи автотеста с рабочим элементом. Для этого нужно настроить вебхук, указав необходимые переменные.

Вы можете связывать автотесты с тест-кейсами или чек-листами, чтобы сделать их автоматизированными и запускать автотесты из тест-планов.

Если тест-кейс связан с автотестом 1 к 1, его можно обновлять, заменяя его содержимое метаданными связанного автотеста.

Чтобы связать автотест с ручным тестом:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Библиотека автотестов**.
3. Откройте карточку автотеста.
4. Перейдите в раздел **Связанные тест-кейсы**.



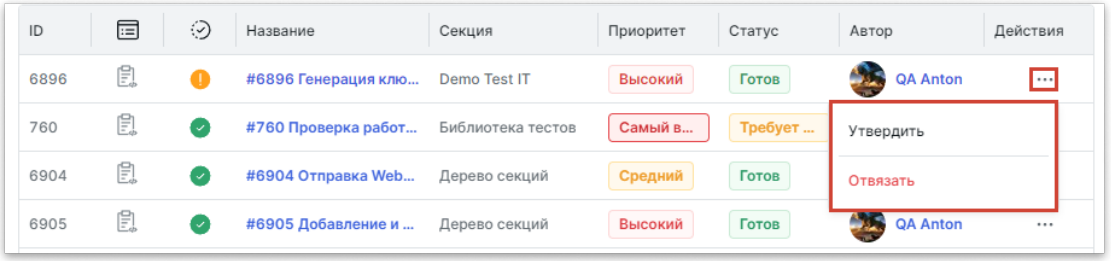
5. Введите глобальный ID (в колонке **ID** в библиотеке тестов) или название рабочего элемента (теста), к которому хотите привязать автотест. При необходимости воспользуйтесь Фильтром.
6. Нажмите **Добавить**.

Тест-кейс появится в таблице связанных элементов. Вы можете отслеживать его ID, тип, название, секцию библиотеки тестов, в которой он находится, приоритет, статус готовности и автора в соответствующих колонках.

В колонке **Изменения** отображается статус изменений, внесенных в рабочий элемент после связки с автотестом. Если в шаги рабочего элемента внесены изменения, вы можете подтвердить их актуальность. Например, если в тест-кейс внесены новые параметры, вам нужно убедиться, что эти параметры описаны в автотесте, и подтвердить актуальность изменений в связанном тест-кейсе. Индикатор неподтвержденных изменений носит информативный характер и не влияет на возможность запуска автотеста. Чтобы подтвердить актуальность изменений, откройте меню в колонке **Действия** и нажмите **Утвердить**.

Чтобы отвязать рабочий элемент от автотеста:

- Откройте меню в колонке **Действия** и нажмите **Отвязать**.



  Вы также можете добавлять связанные автотесты из рабочих элементов. Для этого смотрите раздел Автоматизация рабочих элементов.

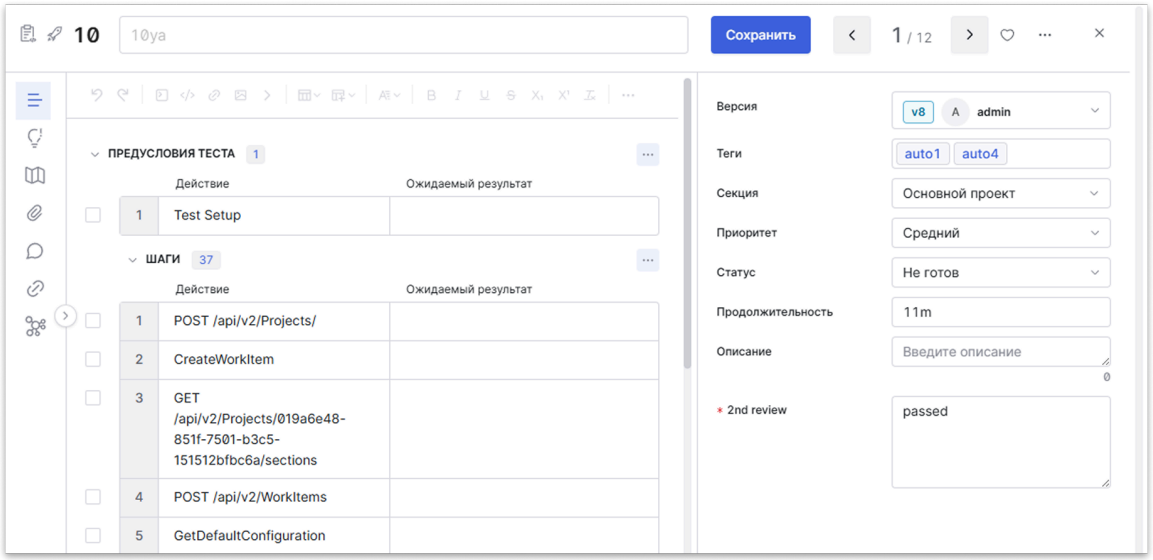
Обновлено: 4/15/26, 7:08 PM

Обновление тест-кейсов на основе связанных автотестов

Принцип обновления тест-кейсов на основе связанных автотестов

> **Настройте уведомления об изменении рабочего элемента**
>
> С Test IT вы можете получать в мессенджер уведомления о создании или удалении связи автотеста с рабочим элементом. Для этого нужно настроить вебхук, указав необходимые переменные .

Вы можете обновлять содержимое тест-кейсов на основе метаданных связанных автотестов. После прогона автотеста его метаданные переносятся в связанный тест-кейс и заменяют собой его содержимое:



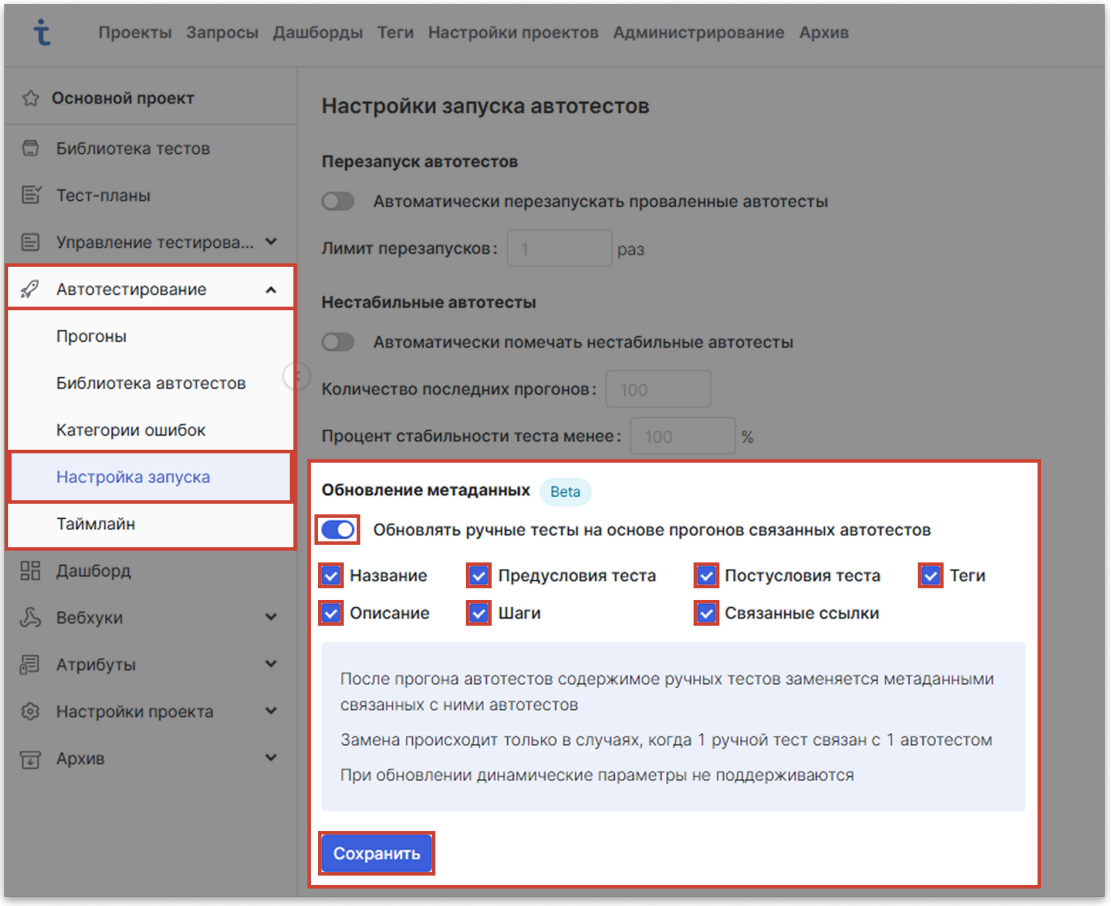
Особенности обновления тест-кейсов:

1. Тест-кейс должен быть связан с автотестом 1 к 1. Если к 1 тест-кейсу привязано несколько автотестов или к 1 автотесту привязано несколько тест-кейсов, функциональность работать не будет.
2. Содержимое тест-кейса обновляется после прогона автотеста.
3. После прогона автотеста прежнее содержимое привязанного тест-кейса, включая параметры и их значения, заменяется метаданными автотеста.

4. Если опция обновления тест-кейсов активна, редактирование полей тест-кейса, включенных при настройке, недоступно. Для редактирования нужно отключить опцию обновления.
5. Чтобы в названии тест-кейса (ручного теста) указывалось значение параметра, в адаптере (в коде с описанием тест-методов) укажите `{param_name}` . Чтобы в названии тест-кейса (ручного теста) указывалось название параметра, в адаптере укажите `param_name` .

Настройка обновления тест-кейсов на основе связанных автотестов

1. Откройте проект.
2. Перейдите в раздел **Автотестирование** → **Настройки запуска**.
3. В секции **Обновление метаданных** включите опцию **Обновлять ручные тесты на основе прогонов связанных автотестов**.
4. Выберите атрибуты, которые хотите заменять на метаданные автотеста: название теста, его предусловия, постусловия, теги и т.д.
5. Нажмите **Сохранить**.



Тест-кейсы будут обновлены после прогона связанных с ними автотестов.

Обновлено: 8/12/26, 4:43 PM

Запуск автотестов из UI

Настройте уведомления о запуске автотестов

С Test IT вы можете получать в мессенджер уведомления о запуске автотестов из UI. Для этого нужно настроить вебхук, указав необходимые переменные.

В Test IT автотесты можно запускать со страницы автотестов, из тест-плана и из прогона автотестов (повторно). Для проваленных автотестов можно настроить автоматический перезапуск.

Запуск автотестов

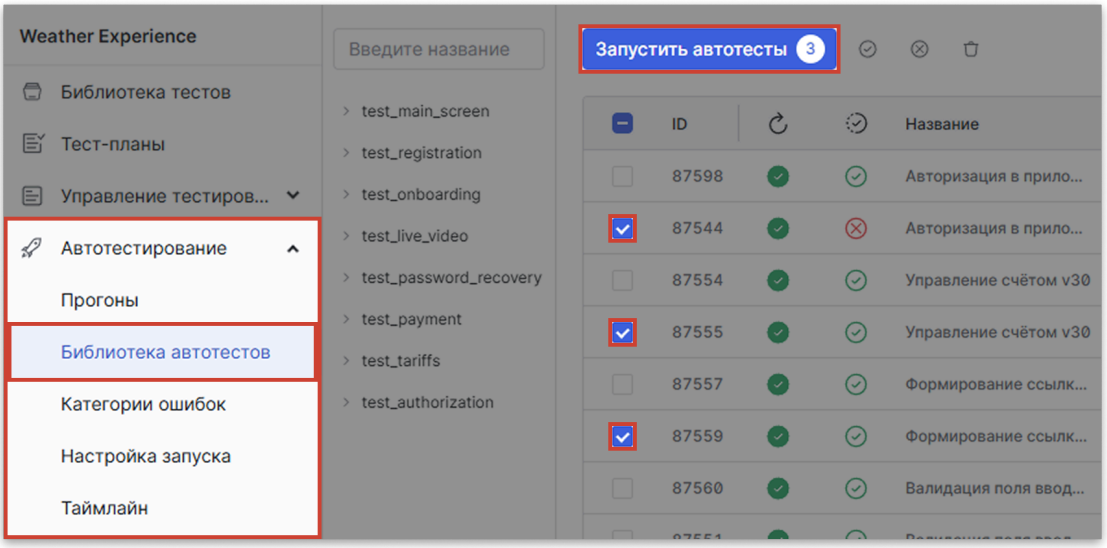
Тарифные ограничения

Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Автотесты, добавленные в Test IT, можно запускать из пользовательского интерфейса (UI). Данный тип запуска может быть осуществлен автономно от тест-планов.

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Библиотека автотестов**.
3. Выберите автотесты, которые хотите запустить.

4. Нажмите **Запустить автотесты**.



5. Укажите данные для запуска автотестов:
   - Конфигурации, для которых хотите запустить автотесты
   - Дополнительные параметры
   - Вебхуки, которые вы хотите использовать при запуске автотестов. Для запуска доступны только вебхуки в статусе *Запущен* для события “Запуск автотестов”. Если нужного вебхука нет, создайте его в разделе **Вебхуки**.
   - Теги
6. Нажмите **Создать прогон и запустить автотесты**.

После запуска вам придет уведомление о нем, позволяющее перейти к прогону, где вы можете проанализировать причины падения автотестов.

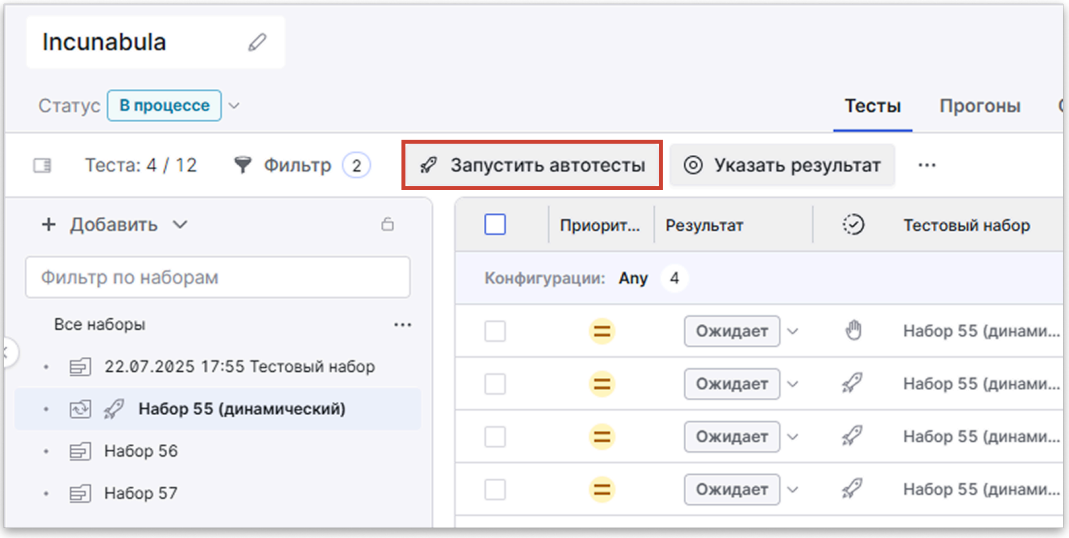
Запуск автотестов из тест-плана

После того, как вы добавили автотесты в систему Test IT и привязали их к тест-кейсам, вы можете запускать автотесты прямо из системы управления тестированием. Чтобы запустить автотесты:

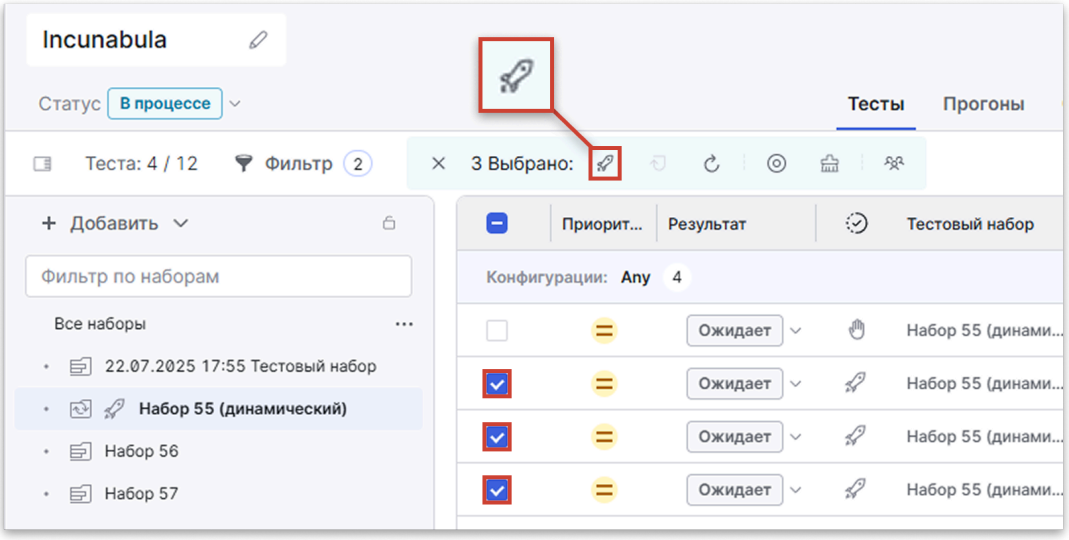
1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. Выберите автотесты одним из способов:
   - Нажмите **Фильтр** над таблицей тест-поинтов и в поле **Статус автоматизации** выберите **Автоматизированный**. В списке отобразятся только автоматизированные тест-поинты.

- Отсортируйте автоматизированные тест-поинты с помощью колонки **Статус автоматизации**.

5. Выберите один из способов запуска автотестов:
   - Чтобы запустить все автотесты, нажмите **Запустить автотесты**.



   - Чтобы запустить один или несколько автотестов, выберите их и нажмите значок запуска автотестов.



     Автотесты, не находящиеся *В процессе*, будут запущены.

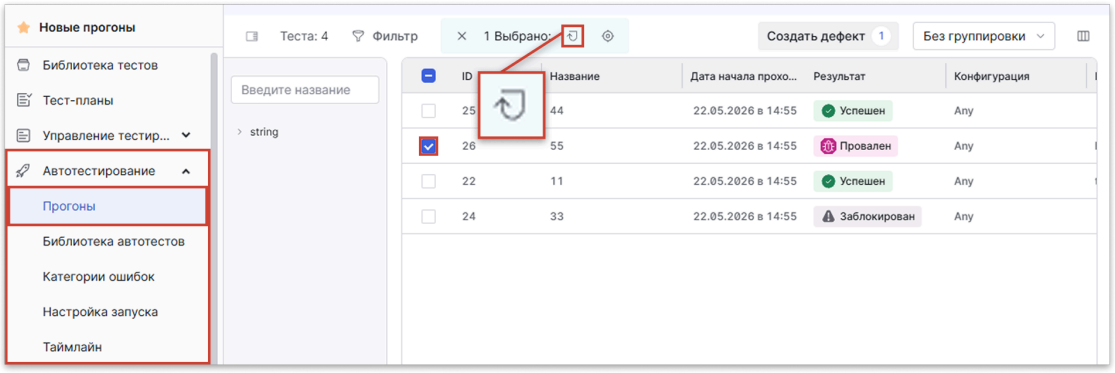
6. **Опционально:** Выберите вебхуки для запуска автотестов.
7. Нажмите **Создать прогон и запустить автотесты**.
8. **Опционально:** Чтобы просмотреть прогоны автотестов из тест-плана, откройте вкладку **Прогоны**.

В случае ошибок автотестов, проанализируйте причину их результатов: инфраструктуру, автотесты или продукт.

Перезапуск автотестов из прогона

Вы можете перезапускать автотесты из прогона. Тесты в статусе *В процессе* не могут быть перезапущены до их окончания.

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Прогоны**.
3. Откройте прогон.
4. Выберите автотесты и нажмите **Перезапустить**.



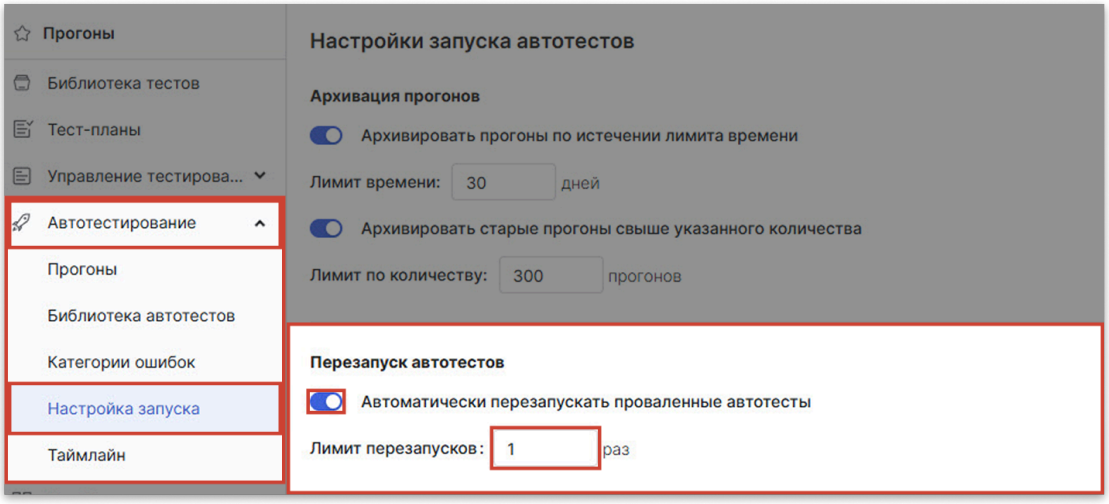
5. **Опционально:** Выберите вебхуки для запуска автотестов.
6. Нажмите **Перезапустить**. Отобразится сообщение о запуске автотестов.

Автоматический перезапуск автотестов с ошибкой

> **Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Если автотесты возвращают ошибку, вы можете настроить автоматический перезапуск.

1. Откройте проект.
2. В навигационном меню слева откройте раздел **Автотестирование → Настройка запуска**.



3. Включите опцию **Перезапускать автотесты**.
4. Установите ограничение на количество перезапусков (лимит перезапуска).
5. Нажмите **Сохранить**.

Обновлено: 8/18/26, 4:54 PM

Запуск автотестов с помощью клиентских библиотек

Вы можете автоматизировать создание карточек автотестов в Test IT с помощью адаптеров для тестовых фреймворков. Вы можете использовать следующие клиентские библиотеки:

- **Адаптеры**:
  - C# (фреймворк Dotnet)
  - Golang
  - Java
  - JavaScript
  - Python
- **API-клиенты**:
  - C# (фреймворк Dotnet)
  - Golang
  - Java
  - JavaScript
  - Python

Принцип работы клиентских библиотек

1. Адаптер срабатывает при запуске тестового фреймворка.
2. Автотесты в коде обогащаются метаданными при помощи аннотаций/декораторов/атрибутов (в зависимости от возможностей языка программирования).
3. При запуске тестов собираются данные о запуске тестов, которые предоставляются тестовым фреймворком, а также дополнительные метаданные, которыми были обогащены тесты.
4. Все собранные метаданные об автотестах и их запуске сохраняются, собираются и конвертируются в формат пригодный для отправки через API-клиент.
5. После сбора и преобразования данных они отправляются в Test IT через API-клиент.

После запуска вы можете найти карточки автотестов в вашем проекте в разделе **Автотестирование**. Вы можете отслеживать результаты запущенных автотестов из раздела **Автотестирование → Прогоны** или

Тест-планы → Прогоны. Чтобы впоследствии запускать данные автотесты через UI, привяжите их карточки к тест-кейсам.

Обновлено: 4/15/26, 7:08 PM

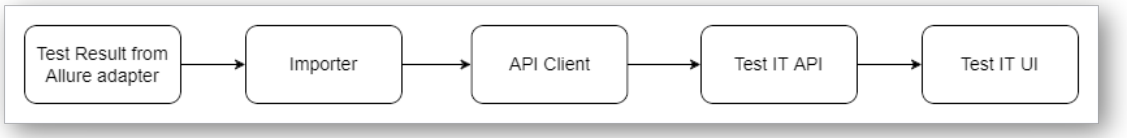
Импорт результатов из Allure Adapters

Test IT поддерживает импорт результатов автотестов, полученных из адаптеров Allure, с помощью клиентских библиотек. Используйте утилиты для импорта, чтобы конвертировать полученные из адаптеров Allure `.json` файлы с результатами в формат, поддерживаемый Test IT. Вы можете найти данную утилиту на нашем Github:

- Test IT Importers

Принцип работы

1. Импортер собирает результаты автотестов, полученные из адаптеров Allure в формате `.json`.
2. Импортер конвертирует считанные данные в формат, поддерживаемый системой Test IT.
3. Импортер создает новый прогон (тест-ран), чтобы передать в него результаты тестов, или передает данные в существующий прогон, указанный пользователем.
4. Импортер передает конвертированные данные результатов тестов в Test IT с помощью API-клиента. Результаты отображаются в системе. Вы можете ознакомиться с форматом отображения результатов тестов в соответствующем разделе.



Обновлено: 4/15/26, 7:08 PM

Отслеживание запусков автотестов

Настройте уведомления о запусках автотестов

С Test IT вы можете получать в мессенджер уведомления о запуске и прохождении автотестов, а также об остановке прогона. Для этого нужно настроить вебхук, указав необходимые переменные.

Test IT позволяет отслеживать запуски автотестов из UI, через API, а также с помощью клиентских библиотек. Чтобы открыть список прогонов (тест-ранов) и отследить их статус или посмотреть их историю:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Прогоны**.

Внимание

Результаты автотестов, проставленные через UI, не попадают в таблицу прогонов.

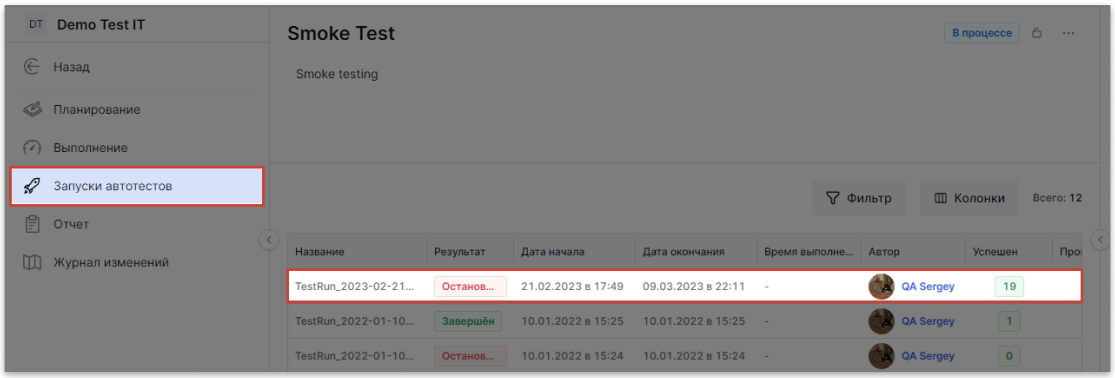
Прогоны отображаются в таблице. В ней вы можете отслеживать их названия, статус, дату начала, инициатора запуска, количество запущенных автотестов, результаты автотестов и категории ошибок. При необходимости вы можете отфильтровать прогоны.

Отслеживание запусков автотестов из тест-плана

Чтобы отслеживать результаты автотестов, запущенных во время прогона (тест-рана):

1. Откройте проект.
2. Перейдите в раздел **Тест-планы** и откройте тест-план.
3. На странице тест-плана перейдите во вкладку **Прогоны**.

4. Откройте прогон.



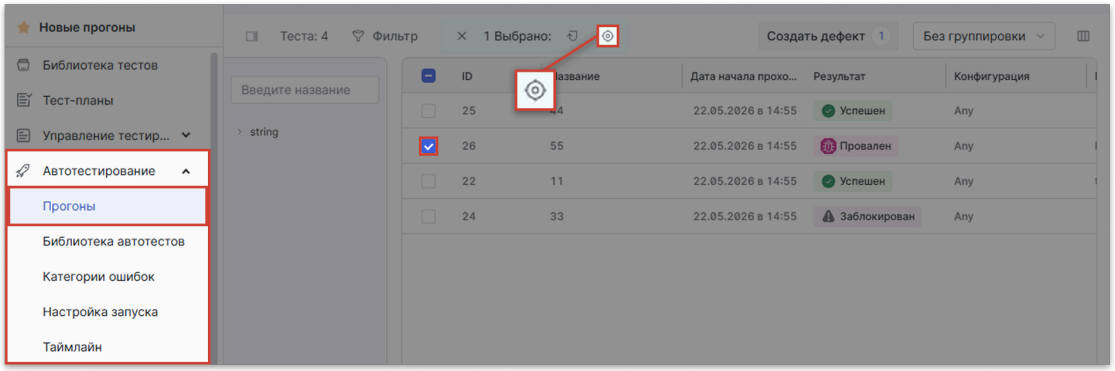
   В окне прогона открывается таблица с данными о запущенных автотестах. Воспользуйтесь фильтром, чтобы отсортировать содержимое таблицы. Чтобы открыть результат конкретного теста, нажмите на строку данного теста в таблице.

Вы можете изменить присваиваемое по умолчанию название прогона. Для этого воспользуйтесь методом `POST /api/v2/testRuns` в нашем API.

Отметка причины результата

Вы можете формировать аналитику по автотестам, проставляя причины результата автотестам в прогоне (тест-ране). Чтобы проставить причину результата:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Прогоны**.
3. В открывшемся списке откройте прогон.
4. Поставьте флажки около автотеста или нескольких автотестов, для которых хотите указать причину результата.
5. Нажмите значок **Указать результат** над таблицей.



6. В открывшейся панели выберите причину падения (**Продукт**, **Автотест** и/или **Инфраструктура**). При необходимости добавьте комментарий,

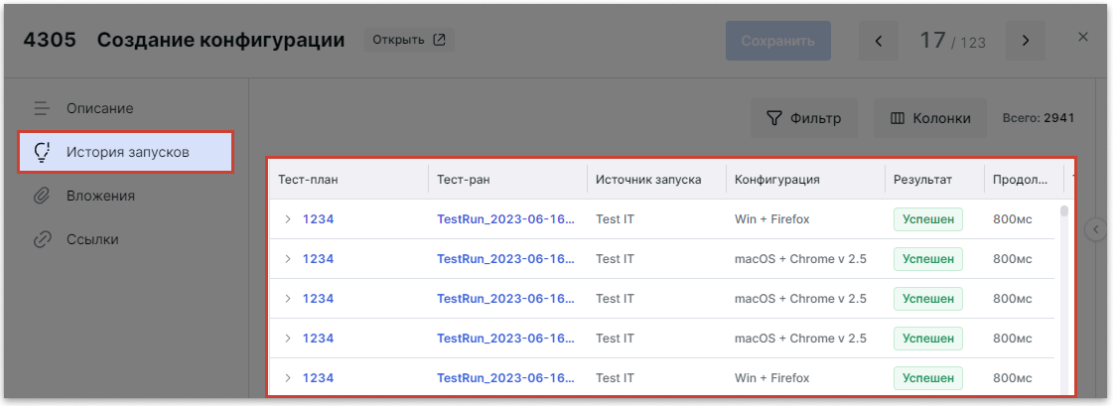
вложение или ссылку.
7. Нажмите **Сохранить**.

Чтобы узнать больше о причинах падения автотестов и категориях ошибок, смотрите Анализ ошибок автотестов.

Просмотр истории запусков автотеста

Детальная информация о результатах прохождения конкретных тестов хранится в окне результата. Для просмотра детальной информации:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Прогоны**.
3. Откройте прогон.
4. Откройте результат из таблицы.
5. Перейдите в раздел **История запусков**.



6. Разверните результат.

В окне истории запусков теста вы можете:

- Отслеживать результат прохождения теста, время его завершения и конфигурацию, для которой был запущен автотест.
- Отслеживать результаты предусловий и постусловий автотеста, а также каждого из шагов.
- Отслеживать строки кода, на которых произошла ошибка автотестов. Вы можете сохранить их как регулярное выражение, чтобы автоматически проставлять причину падения для автотестов, содержащих данные выражения.
- Проставить причину падения автотеста.
- Отслеживать историю запусков данного автотеста в журнале запусков.

Обновлено: 6/2/26, 1:50 PM

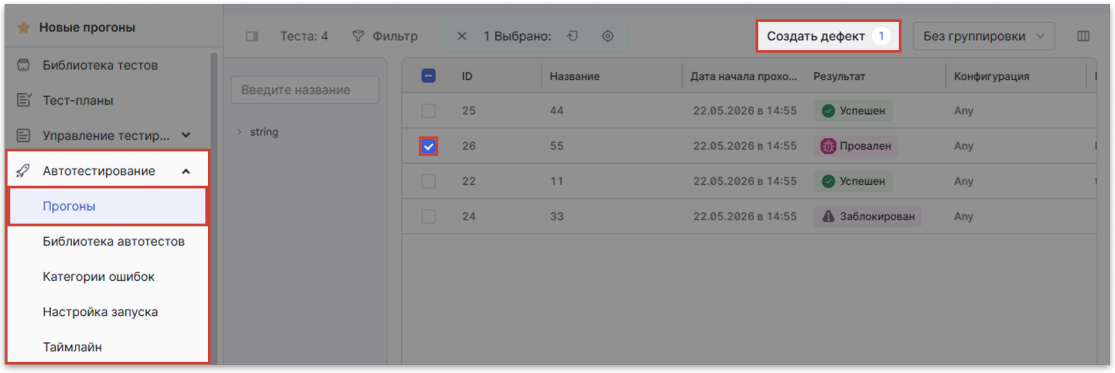
Создание дефектов в таск-трекерах из прогонов автотестов

> **Требуется настроенная интеграция и подключение проекта!**
>
> Чтобы создавать тест-кейсы по задачам в вашем таск-трекере, нужно настроить и протестировать интеграцию с таск-трекером и добавить ее в проект.

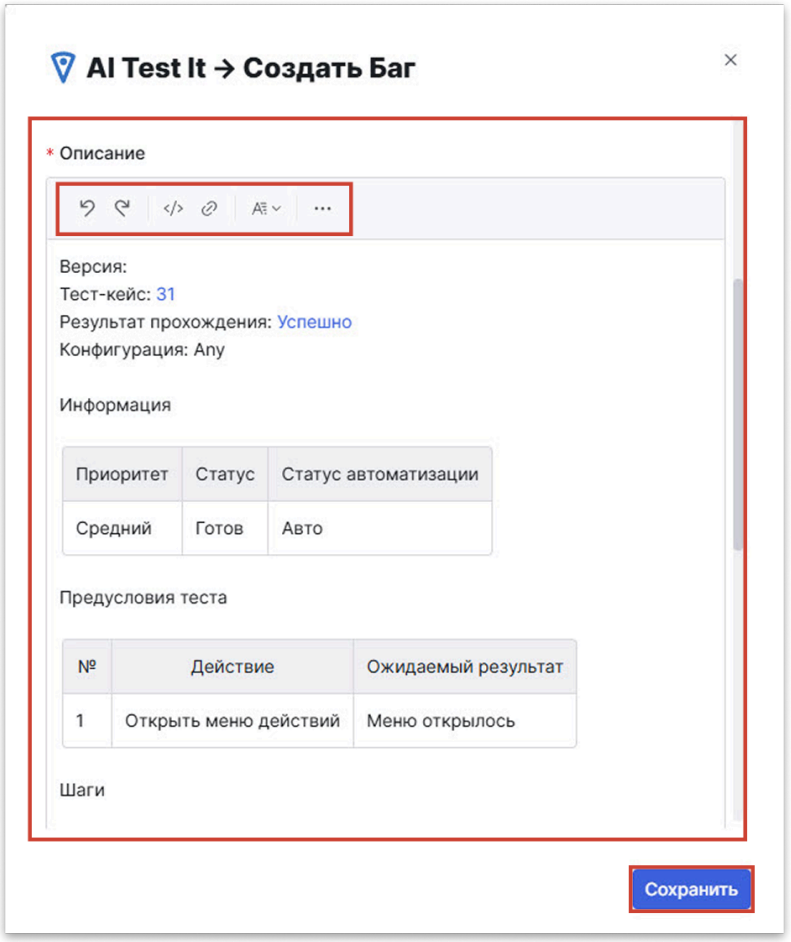
В Test IT можно создавать дефекты для проваленных автотестов.

Чтобы создать дефект:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Прогоны**.
3. Откройте прогон.
4. В прогоне выберите автотесты с результатом *Провален*, для которых нужно создать дефект. При необходимости воспользуйтесь Фильтром.
5. Нажмите **Создать дефект**.



   Если вы настроили интеграцию с несколькими проектами в таск-трекере, выберите проект, в котором хотите создать дефект.
6. В открывшемся окне таск-трекера заполните недостающие данные о дефекте (при необходимости).
7. **Опционально:** Отредактируйте описание дефекта, используя панель инструментов в секции **Описание**.
8. Нажмите **Сохранить**.



После создания дефекта в таск-трекере появится ссылка на тест и на результат его прохождения, во время которого нашли дефект. Вы также можете отследить, на какой конфигурации был обнаружен дефект.

Чтобы отследить ссылку на созданный дефект:

- Перейдите в раздел **Ссылки** в окне результата теста.

  Вы можете отслеживать статус дефекта, его приоритет и исполнителя прямо в Test IT, наведя на него курсор.

Обновлено: 9/8/26, 6:41 PM

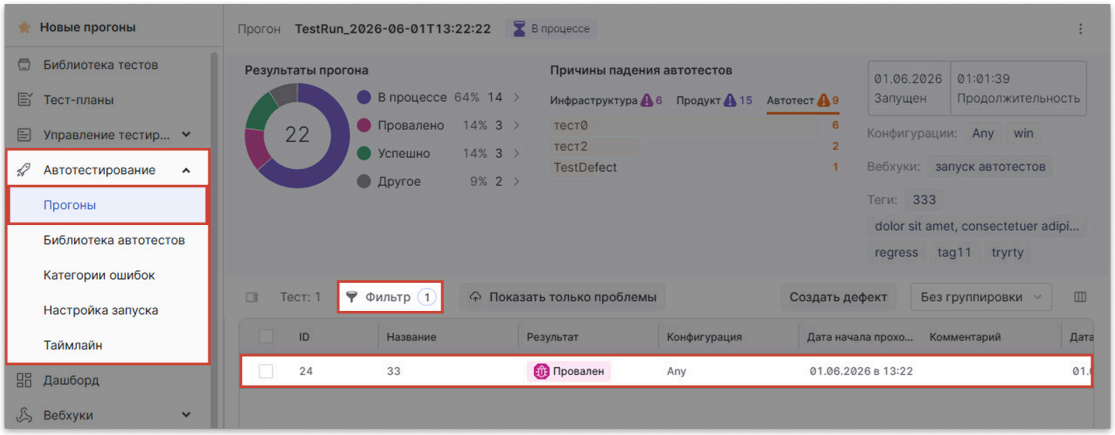
Анализ результатов автотестов

После запуска автотестов вы можете проанализировать их прогон (тест-ран). Вы можете:

- Отметить в системе автотесты без дефектов
- Задать причину падения автотестов, выставив одну из категорий ошибок: продукт, автотест или инфраструктура
- Задать регулярное выражение, при обнаружении которого система Test IT будет автоматически определять причину падения теста

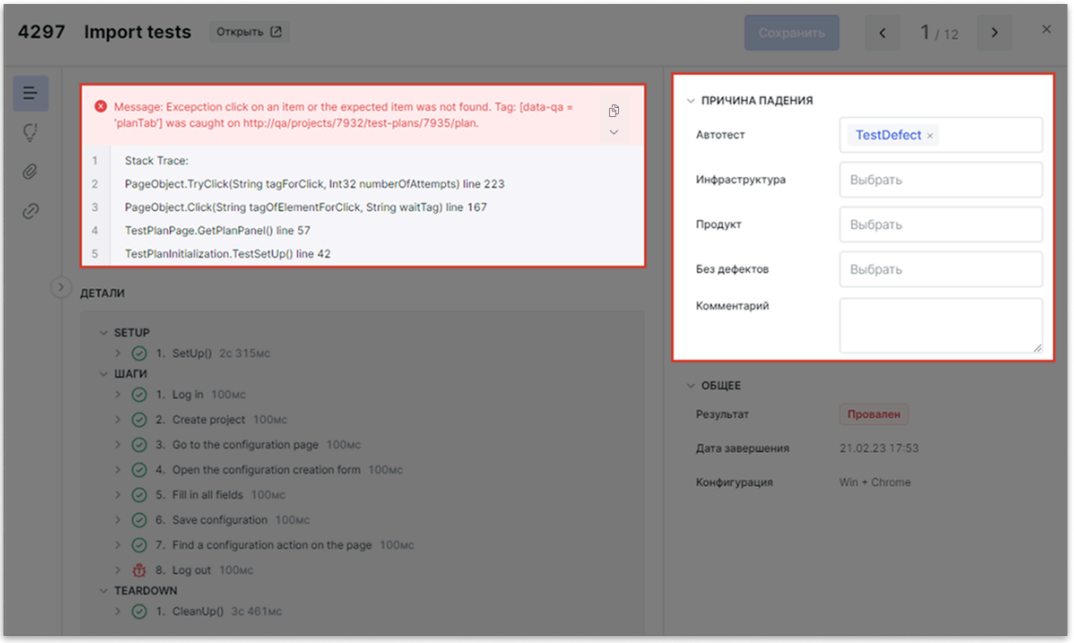
Определение причины падения автотестов

1. Откройте проект.
2. Перейдите в раздел **Прогоны**.
3. Откройте прогон.
4. Отфильтруйте проваленные автотесты. Для этого нажмите **Фильтр** над таблицей результатов тестов и в поле **Статус** выберите **Провален**.



5. Откройте проваленный автотест. В верхней части описания результата вы увидите панель **Stack Trace**, отображающую строки кода, на которых тест упал.
6. Когда вы определите, в чем именно причина падения автотеста на данных строках кода, укажите причину падения данного автотеста в поле **Причина падения**. Вам доступны следующие варианты:
   - **Продукт**
   - **Инфраструктура**

- **Автотест**



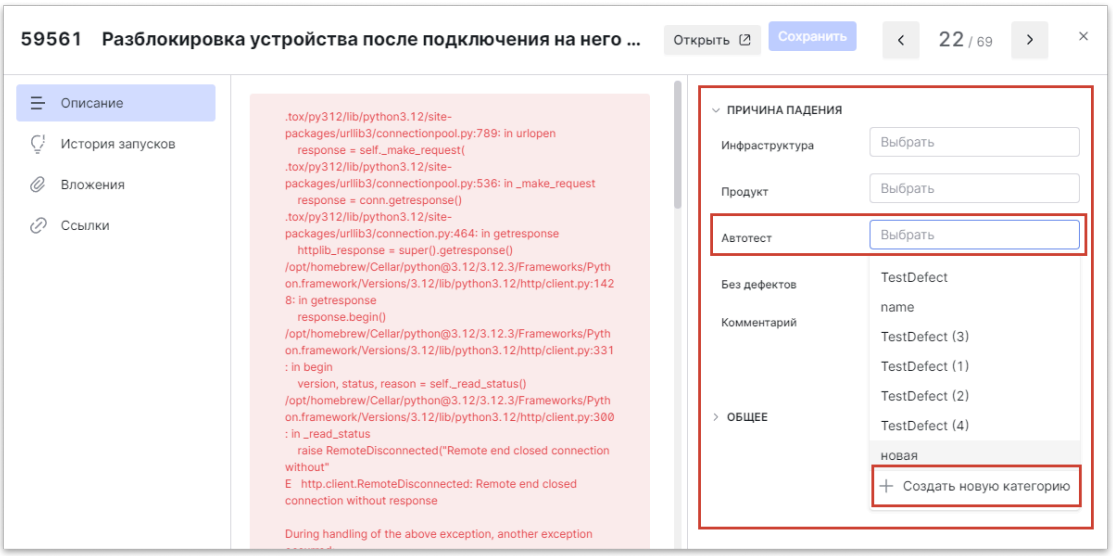
7. Нажмите **Сохранить**.

Создание категорий ошибок

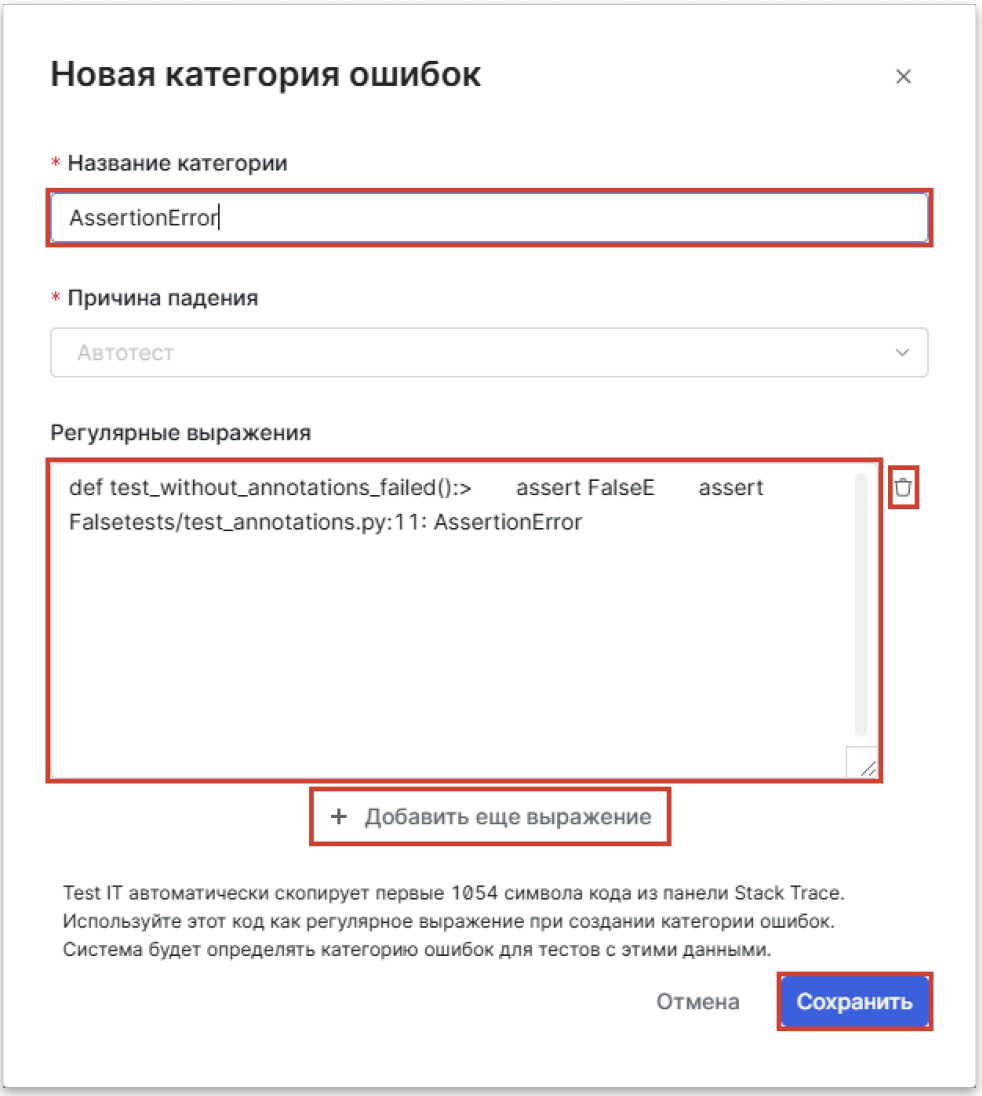
Вы можете обучать систему Test IT распознавать типичные ошибки в автотестах с помощью регулярных выражений и категорий ошибок, ассоциирующихся с ними. У проваленных автотестов регулярное выражение формируется из панели **Stack Trace**. При сохраненном регулярном выражении система Test IT автоматически проставляет автотестам, содержащим это выражение, заданную категорию ошибки. Категории ошибок можно создавать в прогоне или в разделе Настройки проектов.

Чтобы создать категорию ошибок в прогоне:

1. Откройте проект.
2. Откройте раздел **Автотестирование → Прогоны**.
3. Откройте прогон.
4. В прогоне откройте автотест.
5. В окне автотеста в правой панели разверните секцию **Причина падения**.
6. Нажмите левой кнопкой мыши в поле **Автотест**. Откроется выпадающее меню категорий ошибок.
7. В выпадающем меню нажмите **Создать новую категорию**. Откроется окно создания категории ошибок.



8. Заполните данные новой категории ошибок:



- Введите название.
- Заполните поле **Регулярные выражения**. Для автотестов в статусе *Провален* система автоматически скопирует первые 1054 символа кода из панели **Stack Trace**.

    - **Опционально**: Отредактируйте регулярное выражение.
    - **Опционально**: Добавьте в категорию ошибок другие регулярные выражения.
    - **Опционально**: Чтобы удалить регулярное выражение, нажмите значок **Корзина** справа от него.
9. Нажмите **Сохранить**. Новая категория ошибок сохранится в разделе **Автотестирование → Категории ошибок**.
10. В окне автотеста нажмите **Сохранить**, чтобы сохранить новую категорию ошибок в данном автотесте.

Категории ошибок отображаются в Настройках проектов и в разделе проекта Автотестирование → Категории ошибок.

Управление категориями ошибок

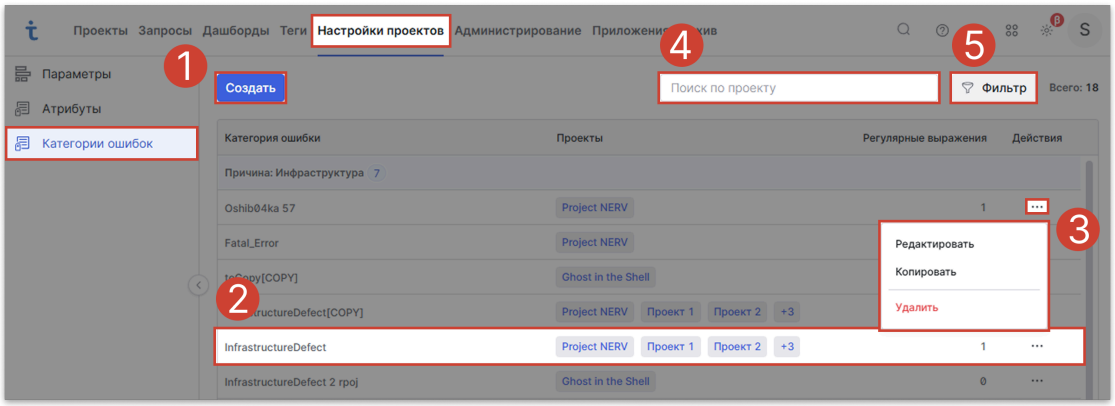
Для категорий ошибок доступны:

- Глобальное управление — на уровне всей системы Test IT в разделе **Настройки проектов**. Здесь можно назначать категории ошибок на различные проекты.
- Управление в проекте — на уровне отдельного проекта в разделе **Автотестирование**

Глобальное управление категориями ошибок

Вы можете создавать категории ошибок, указывать, в каких проектах Test IT они будут использоваться или вносить в них изменения. Чтобы открыть раздел **Категории ошибок**:

- В верхней навигационной панели Test IT нажмите **Настройки проектов**, затем в меню слева выберите **Категории ошибок**.



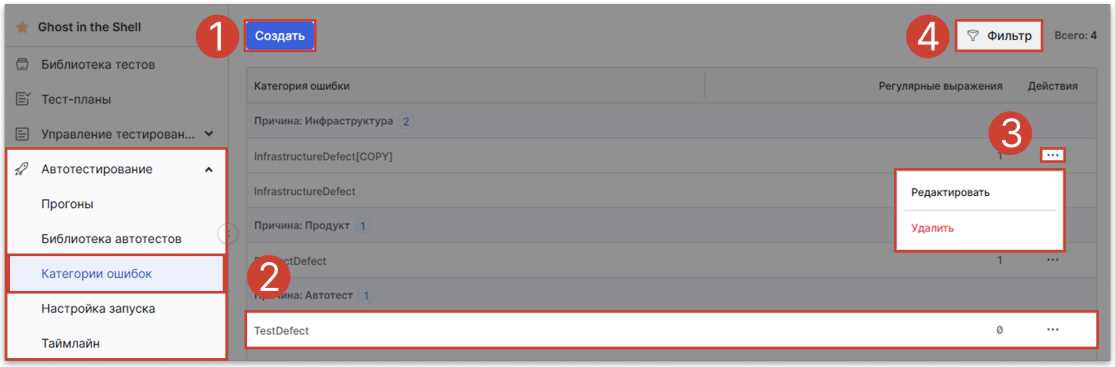
В разделе **Категории ошибок** вам доступны действия:

- Чтобы создать категорию ошибок, нажмите **Создать** (1). Укажите данные о категории, включая проекты, в которых эта категория ошибок будет использоваться.
- Чтобы изменить категорию ошибок, нажмите на нее в списке (2) или нажмите **Действия** и выберите **Редактировать** (3). Здесь вы можете выбрать проекты, в которых категория ошибок будет использоваться.
- Чтобы скопировать или удалить категорию ошибок, выберите **Действия** (3) и выберите нужную опцию.
- Чтобы найти категории ошибок, используемые в конкретных проектах, используйте поле **Поиск по проекту** (4).
- Для расширенного поиска воспользуйтесь Фильтром (5).

Управление категориями ошибок в проекте

Чтобы открыть раздел управления категориями ошибок в проекте,

- Откройте проект, затем в левом навигационном меню выберите **Автотестирование → Категории ошибок**.



В разделе **Категории ошибок** вам доступны действия:

- Чтобы создать категорию ошибок, нажмите **Создать** (1).
- Чтобы изменить категорию ошибок, нажмите на нее в списке (2) или нажмите **Действия** и выберите **Редактировать** (3). Если категория

ошибок используется в других проектах, редактирование недоступно.
- Чтобы удалить категорию ошибок из проекта, выберите **Действия** (3) и выберите нужную опцию. Категории ошибок, удаленные из проекта, доступны в разделе Настройки проектов.
- Для поиска воспользуйтесь Фильтром (4).

Обновлено: 5/19/26, 1:43 PM

Таймлайн

Вы можете посмотреть все запуски автотестов в вашем проекте в разрезе времени. Для этого:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Таймлайн**.

Вы можете использовать слайдеры, чтобы сфокусироваться на более конкретном периоде времени, а также отфильтровать прогоны (тест-раны) по времени их создания, нажав **Фильтр** над графиком. При наведении курсора на прогон на графике отображается детальная информация о прогоне.

Обновлено: 4/15/26, 7:08 PM

Управление прогонами автотестов

- Редактирование прогонов
- Действия с прогонами
- Архивация и восстановление прогонов
- Автоархивация прогонов
- Сравнение прогонов

Чтобы узнать о том, как создавать и перезапускать прогоны, обратитесь к соответствующим инструкциям.

Редактирование прогонов

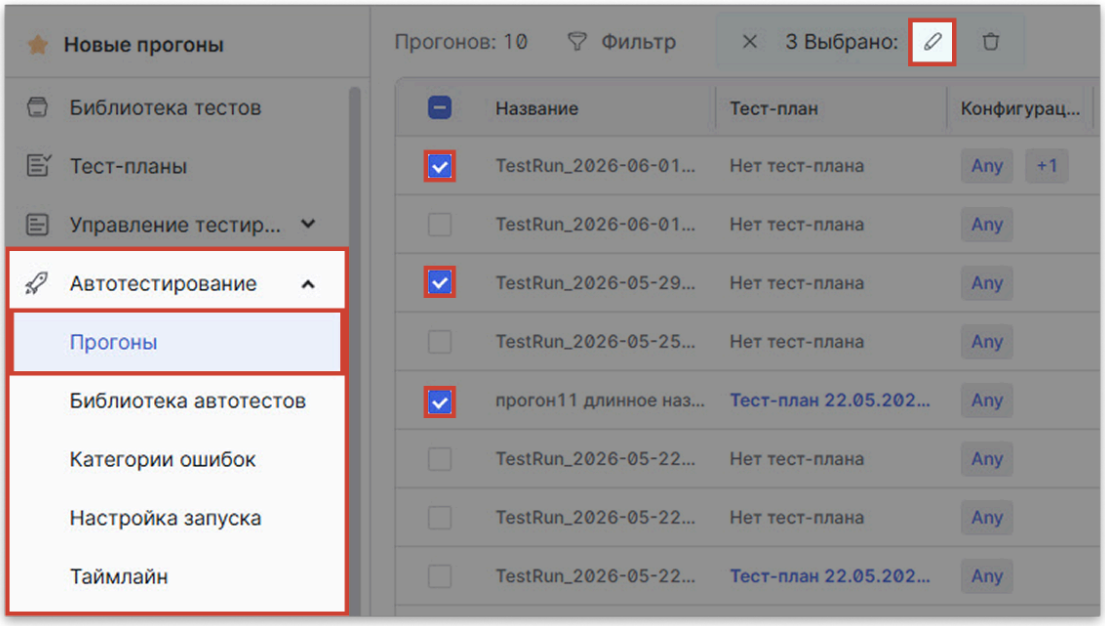
Вы можете редактировать прогоны автотестов (тест-раны), изменяя их атрибуты. Вам доступны следующие опции:

- **Редактирование одного прогона** — позволяет изменить название, описание, источник запуска, вложенные файлы и ссылки.
- **Массовое редактирование** — позволяет изменять описание, источник запуска, вложенные файлы и ссылки.

Чтобы отредактировать прогон:

1. Откройте проект.
2. Перейдите в раздел **Автотестирование → Прогоны**.
3. Выберите нужные прогоны. Вы можете выбрать один или более прогонов. Чтобы выбрать все прогоны, поставьте флажок в ячейке **Название**.

4. Нажмите значок **Редактировать**.



   - **Опционально**: Чтобы отредактировать отдельный прогон, нажмите значок **Действия** в правой части списка прогонов и выберите **Редактировать**.
5. В открывшемся окне внесите требуемые изменения.
6. Нажмите **Сохранить**.

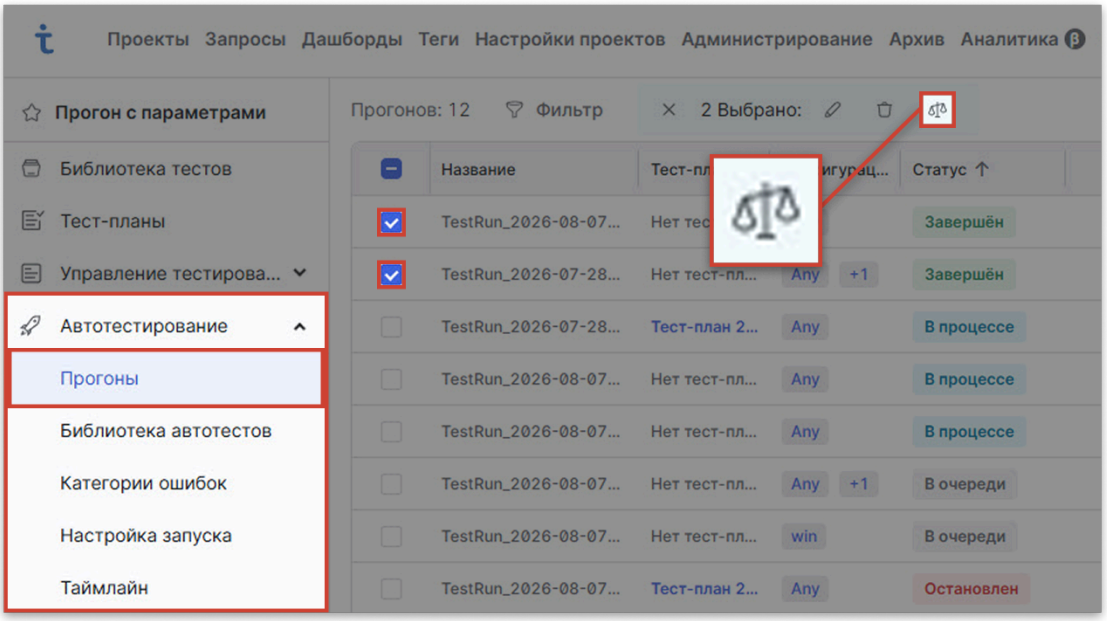
Сравнение прогонов

> **Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Вы можете сравнивать прогоны автотестов в статусе *Завершен* или *Остановлен*. Для сравнения необходимо выбрать 2 прогона.

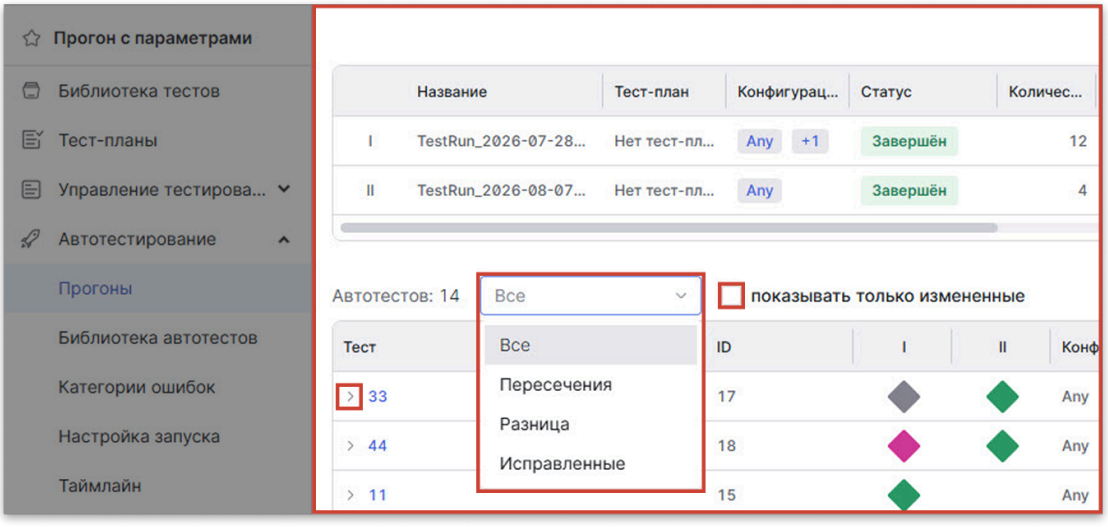
Чтобы удалить прогоны:

1. Откройте проект
2. Перейдите в раздел **Автотестирование → Прогоны**.
3. Выберите 2 прогона, которые хотите сравнить, затем нажмите значок сравнения прогонов **Весы**.



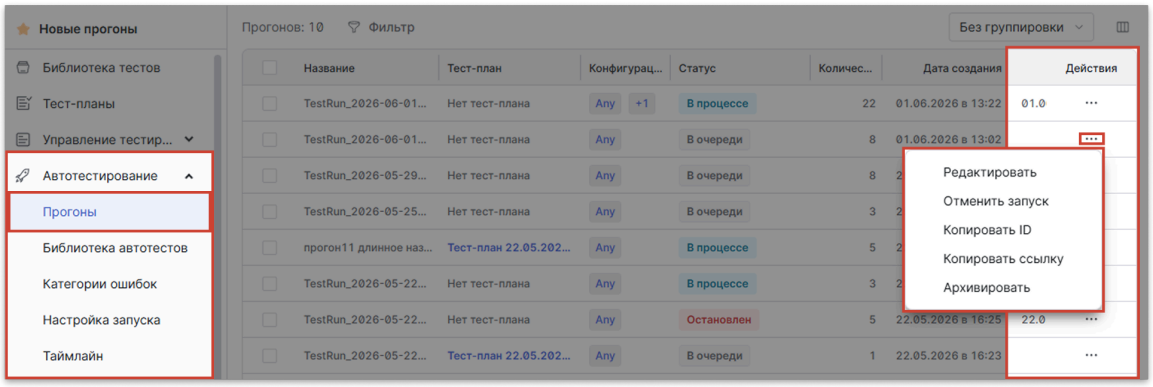
Откроется страница сравнения прогонов.

4. **Опционально:** Выберите **показать только измененные**, чтобы отобразить только измененные автотесты.
5. **Опционально:** Выберите автотесты для сравнения с помощью меню над списком тестов. Доступные опции:
   - Все
   - Общие — общие результаты обоих прогонов: один тест, параметры и конфигурация
   - Разница — тесты, которые были только в одном из прогонов
   - Исправленные — тесты, результаты которых в одном прогоне положительные (*Успешен*), а в другом отрицательные (*Провален*, *Пропущен* или *Заблокирован*)
   - Упавшие — тесты с результатом *Провален*
6. **Опционально:** В списке автотестов разверните их детали, нажав **>** в колонке **Тест**.



Действия с прогонами

Меню **Действия** расположено в правой части списка прогонов.



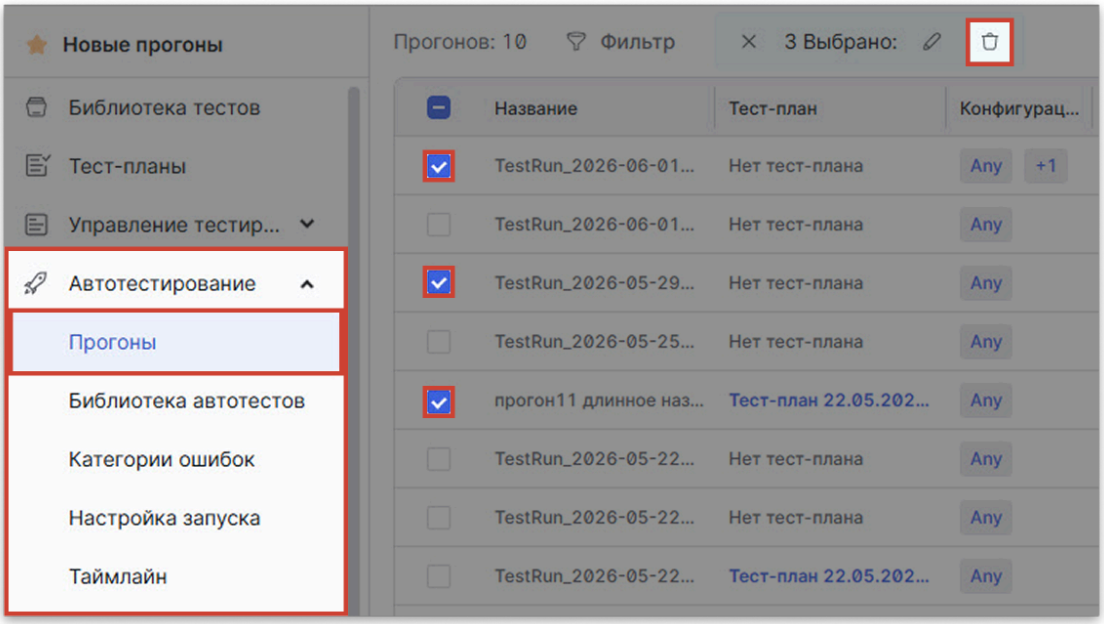
С помощью меню действий вы можете:

- **Отредактировать отдельный прогон**
- **Отменить запуск**
- **Скопировать ID прогона**
- **Скопировать ссылку на прогон**
- **Архивировать прогон**
  Чтобы открыть меню действий:
- Нажмите значок **Действия** справа в списке прогонов.

Архивация и восстановление прогонов

Чтобы удалить прогоны:

1. Откройте проект
2. Перейдите в раздел **Автотестирование → Прогоны**.
3. Отметьте прогоны, которые хотите удалить, затем нажмите значок **Корзина**.

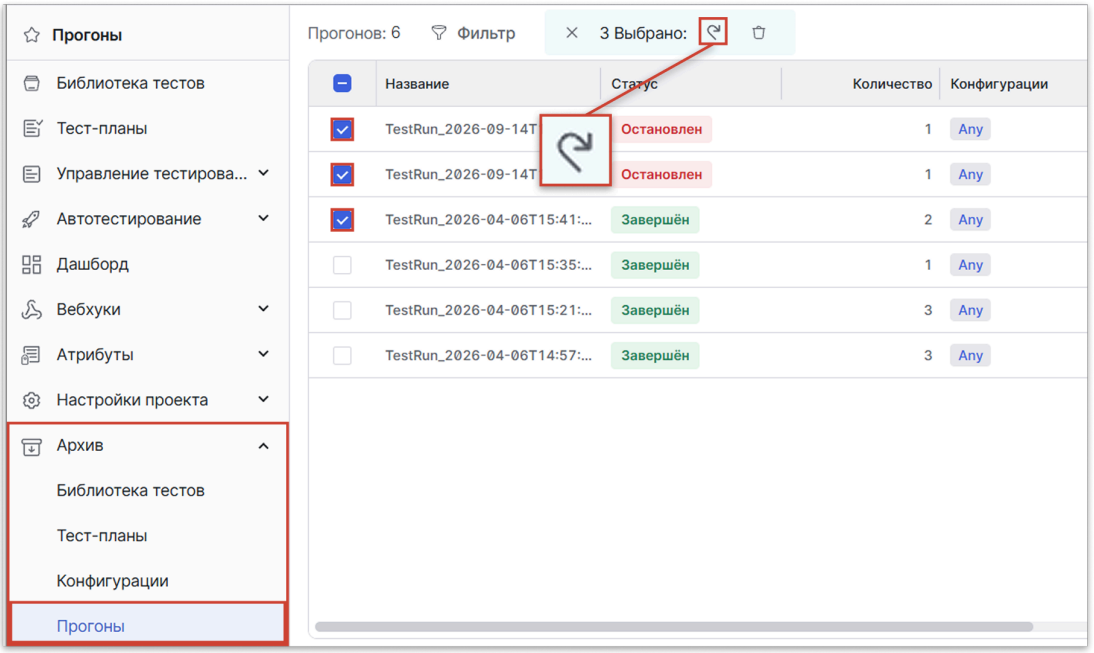


4. В открывшемся окне подтвердите действие.

Восстановление и удаление прогонов

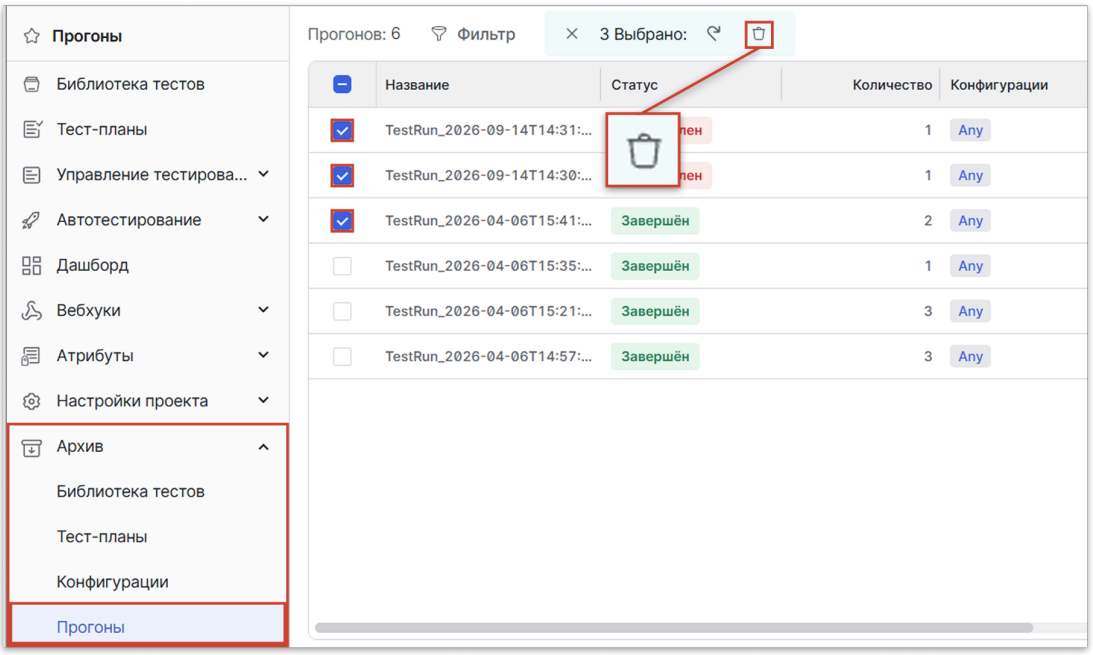
Чтобы восстановить прогоны:

1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Прогоны**. В списке выберите прогоны, которые хотите восстановить, затем нажмите значок **Восстановить**.



3. **Опционально:** Чтобы восстановить отдельный прогон, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.

4. **Опционально**: Чтобы **удалить прогоны**, выберите их в списке и нажмите значок **Корзина**, затем подтвердите действие в диалоговом окне.



   - Чтобы удалить отдельный прогон, откройте меню **Действия** ••• справа в списке прогонов, выберите Удалить, затем подтвердите действие в диалоговом окне.

> **API-методы для восстановления прогонов**
>
> Для восстановления удаленных прогонов через API используются методы:
>
> - Для восстановления одного прогона: `/api/v2/testRuns/{id}/restore`
> - Для массового восстановления прогонов: `/api/v2/testRuns/{id}/restore`

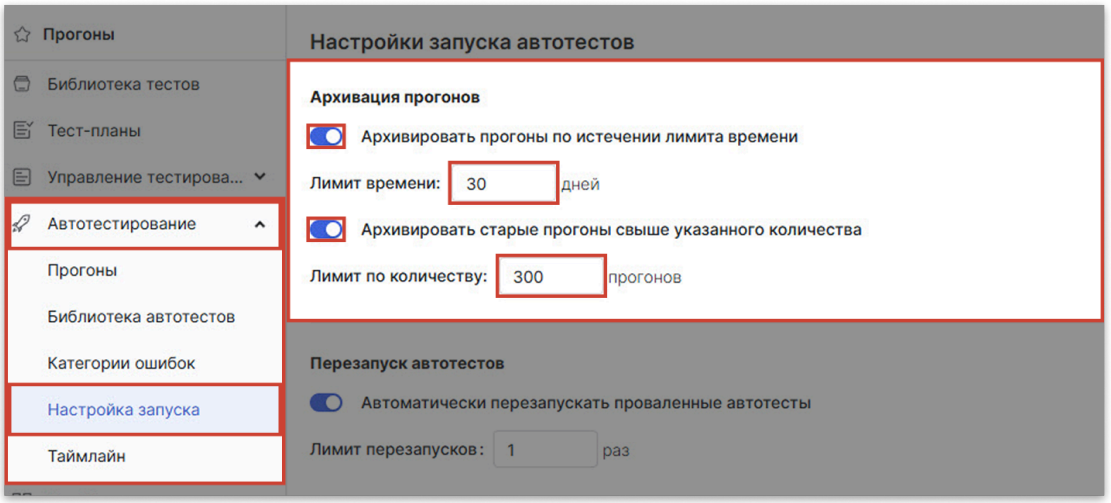
Автоархивация прогонов

> **Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Вы можете настроить автоматическую архивацию прогонов по истечении указанного количества дней или по превышении установленного лимита.

1. Откройте проект.

2. Перейдите в раздел **Автотестирование → Настройки запуска**.
3. Выберите нужные опции и установите лимиты на количество дней и/или прогонов



4. Нажмите **Сохранить**.

Обновлено: 10/29/26, 1:42 PM

Test IT CLI

> **Используйте адаптер Test IT**
>
> Помимо Test IT CLI, при работе с многими тестовыми фреймворками вы можете использовать адаптер Test IT, дающий больше возможностей.

Test IT CLI — это утилита командной строки, которая помогает взаимодействовать с Test IT. Она позволяет легко и быстро собирать и загружать результаты ваших автоматизированных тестов, работать с любыми тестовыми фреймворками и интегрировать их.

Test IT CLI распространяется как пакет Python и легко устанавливается на любые платформы (Linux, Windows и macOS) и окружения (отдельные сервера, контейнеры Docker, CI/CD системы и т.д.).

Test IT CLI доступна для скачивания на Python Package Index .

Основные преимущества

Основные преимущества Test IT CLI:

- Простая и быстрая загрузка результатов автоматизированных тестов
- Простая интеграция с CI/CD системами, такими как GitHub, GitLab, Jenkins и др.
- Работа с любыми тестовыми фреймворками, которые поддерживают формат JUnit XML
- Поддержка параллельного запуска тестов
- Простая установка на любые платформы и окружения

Установка

Для начала работы с Test IT CLI:

1. Установите пакет `testit-cli` из официального репозитория Python. После установки вы сможете использовать утилиту `testit` из командной строки. Для установки используйте команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Если Python отсутствует в вашей системе или окружении, установите его согласно официальной документации (большинство систем и окружений уже содержат установленный Python).

Загрузка первых результатов выполнения тестов

После установки Test IT CLI вы можете загрузить результаты выполнения автоматизированных тестов в систему управления тестированием Test IT.

Для загрузки результатов:

1. Запустите автоматизированные тесты с опцией экспортирования отчета в формате JUnit XML. Данный отчет является стандартным способом обмена результатами между утилитами тестирования, а также CI/CD системами. Вы можете ознакомиться с примерами экспорта JUnit XML отчета для тестовых фреймворков.

2. После получения JUnit XML отчета используйте Test IT CLI со следующими параметрами:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "TestNG test run" \
  --results results/report.xml
```

   - <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT

- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 6 result file with a total of 14 tests
Created new testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
Sending test results to Test IT ...
Uploading: 100%|██████████| 14/14 [00:06<00:00, 2.09it/s]
Successfully sent test results
Completed testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
```

Параметры запуска Test IT CLI

- Для получения полной справки о доступных параметрах используйте команду `testit results import --help` :

```sh
$ testit results import --help
usage: testit [options]

This tool is the command line wrapper of Test IT
allowing you to upload the test results in real time to
Test IT

optional arguments:
  -h, --help            show this help message and exit
  -u https://demo.testit.software, --url
https://demo.testit.software
                        Set url address of the Test IT
instance
  -t T2lKd2pLZGI4WHRhaVZUejNl, --token
T2lKd2pLZGI4WHRhaVZUejNl
                        Set API token
  -pi 5236eb3f-7c05-46f9-a609-dc0278896464, --project-id
5236eb3f-7c05-46f9-a609-dc0278896464
                        Set project id
  -ci 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb, --
configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb
                        Set configuration id
  -ti 3802f329-190c-4617-8bb0-2c3696abeb8f, --testrun-id
3802f329-190c-4617-8bb0-2c3696abeb8f
                        Set test run id
  -tn TestRun01, --testrun-name TestRun01
                        Set test run name
  -s ., --separator .
                        Separate the classname value in
the results into namespace and classname
  -ns NameSpace01, --namespace NameSpace01
                        Set namespace
  -cn ClassName01, --classname ClassName01
                        Set classname
  -r DIR, --results DIR
                        Set directory with results file
  -d, --debug
                        Set debug logs
```

```
  -a TEXT, --attachments TEXT
                          Path to attachments for test run
(multiple)
```

Описание параметров

В Test IT CLI используются следующие параметры:

- `--url`, `-u`: задает адрес вашей системы Test IT (например https://tms.testit.software/)
- `--token`, `-t`: задает приватный токен, сгенерированный в вашем профиле Test IT. Можно указывать как параметр или переменная окружения `TMS_TOKEN`
- `--project-id`, `-pi`: задает id проекта, в который будут загружены тесты
- `--configuration-id`, `-ci`: задает id конфигурации, на которой выполнялись тесты
- `--testrun-id`, `-ti`: задает id прогона (тест-рана), в который будут загружены результаты или который необходимо будет отметить как завершенный
- `--testrun-name`, `-tn`: задает название прогона
- `--separator`, `-s`: разделяет classname на namespace и classname по указанной строке
- `--namespace`, `-ns`: задает название namespace по умолчанию
- `--classname`, `-cn`: задает название classname по умолчанию
- `--results`, `-r`: задает путь до директории или файла с результатами выполнения тестов
- `--debug`, `-d`: включает расширенный режим логирования
- `--attachments`, `-a`: задает путь к вложениям для прогона (может быть указан несколько раз)

Режимы работы

Test IT CLI поддерживает следующие режимы работы:

- Стандартный запуск
- Параллельный запуск

Стандартный запуск

Режим стандартного запуска позволяет загрузить результаты выполнения тестов с помощью одной команды. Данный режим используется по умолчанию.



Процесс состоит из следующих этапов:

1. Сбор информации из JUnit XML отчета о пройденных тестах.
2. Если не указан `testrun_id` , создается новый прогон
3. В созданный прогон загружаются все результаты.
4. После загрузки прогон помечается как завершенный.

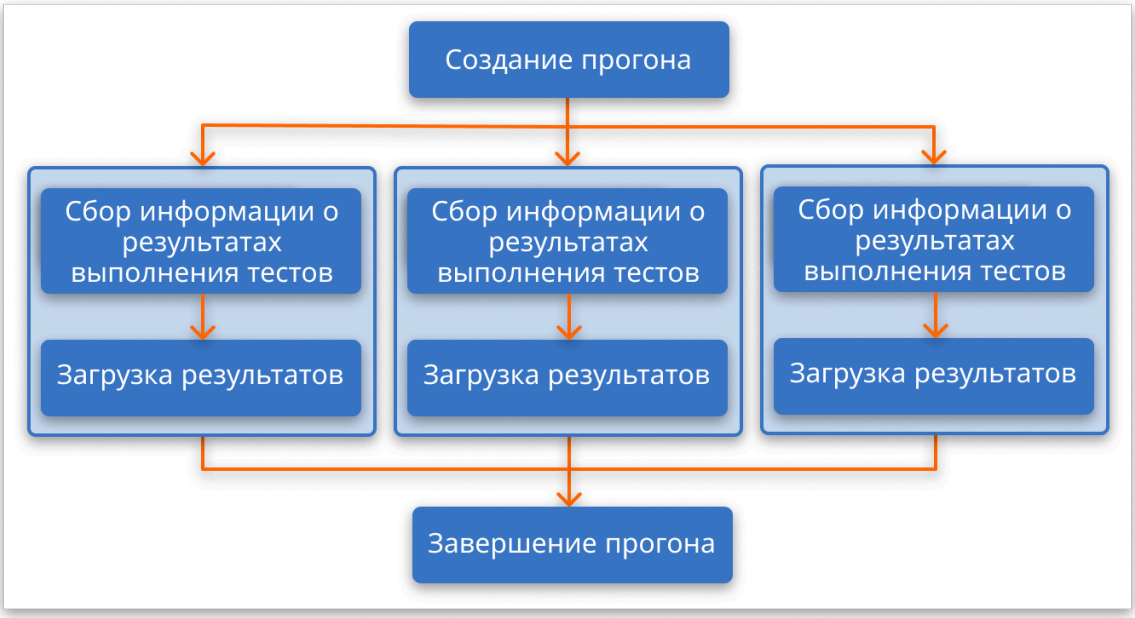
▼ **Пример выполнения команды**

```sh
# загрузка результатов в новый прогон
$ testit results import \
  --url https://tms.testit.software \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-name "New test run" \
  --results results/report.xml

# загрузка результатов в существующий прогон
$ testit results import \
  --url https://tms.testit.software \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-id da02746b-7734-4220-a7f3-126d9f291317 \
  --results results/report.xml
```

Параллельный запуск

Режим параллельного запуска позволяет загружать результаты тестов при параллельном запуске тестов.



Процесс параллельного запуска включает в себя:

1. ▼ **Создание нового прогона**

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit testrun create \
  --url https://tms.testit.software \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --testrun-name "New test run" \
  --output tmp/output.txt
```

2. ▼ **Загрузка результатов из разных потоков**

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results upload \
  --url https://tms.testit.software \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-id $(cat tmp/output.txt) \
  --results results/report.xml
```

3. ▼ **Завершение прогона**

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit testrun complete \
  --url https://tms.testit.software \
  --testrun-id $(cat tmp/output.txt) \
```

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Интеграции с CI/CD-системами
- Примеры автоматизации с другими тестовыми фреймворками
- Плагин Test IT Менеджмент для систем IDE
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM

Работа с системами CI/CD

> **Используйте адаптер Test IT**
>
> Помимо Test IT CLI, при работе с многими тестовыми фреймворками вы можете использовать адаптер Test IT, который дает больше возможностей.

Test IT поддерживает интеграцию с множеством популярных систем CI/CD. Вы можете использовать Test IT CLI для загрузки результатов выполнения тестов.

В Test IT доступны интеграции со следующими системами:

- Azure DevOps
- Bamboo
- Bitbucket
- CircleCI
- GitFlic
- GitHub Actions
- GitLab CI/CD
- Jenkins
- TeamCity

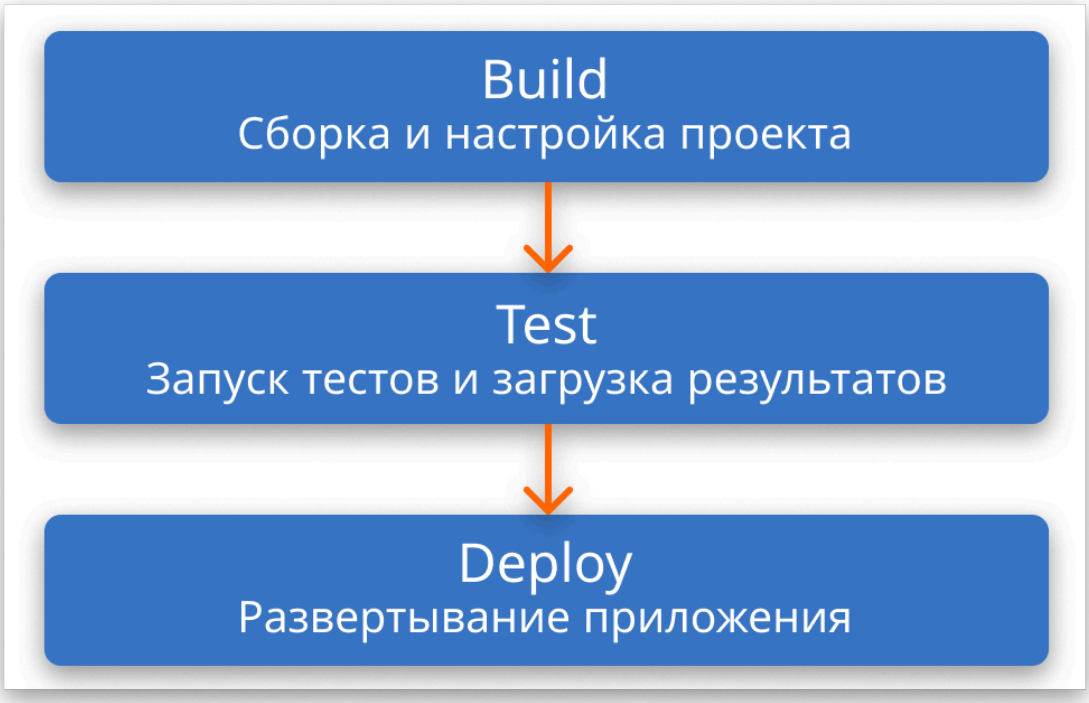
Запуск тестов из CI/CD систем

Независимо от выбранной вами CI/CD системы, в большинстве случаев необходимо выполнить несколько простых шагов:

1. В пайплайн запуска тестов добавить шаг с установкой Test IT CLI. Установка производится с помощью простой команды и поддерживает все операционные системы и окружения.
2. Настроить генерацию JUnit XML отчета для вашего тестового фреймворка.
3. Использовать утилиту Test IT CLI для загрузки результатов тестов.

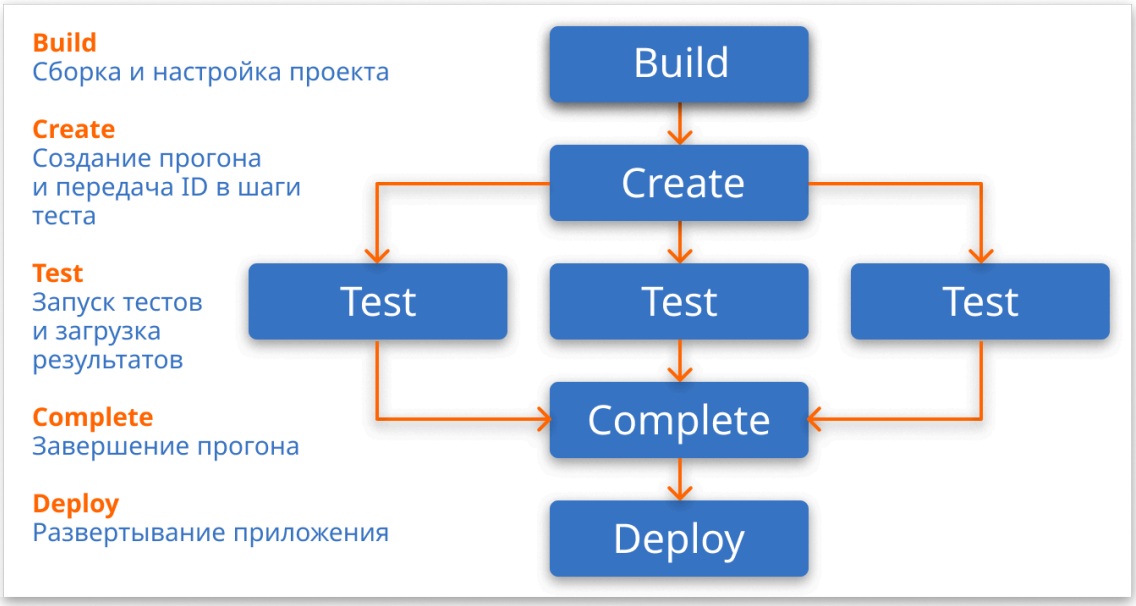
Стандартный запуск

При стандартном запуске, обычно вам необходимо выполнить сборку и настройку вашего проекта перед запуском тестов. После этого вы можете использовать утилиту Test IT CLI для загрузки результатов выполнения тестов. Утилита автоматически обработает результаты выполнения тестов, создаст новый прогон (тест-ран) и загрузит все результаты в Test IT.



Параллельный запуск

Test IT также поддерживает параллельный запуск автоматизированных тестов. Для отправки результатов автотестов из параллельных потоков вам необходимо создать прогон, передать его ID в следующие потоки, залить результаты автотестов и пометить прогон как завершенный.



Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 4/22/26, 2:57 PM



Azure DevOps

Test IT поддерживает интеграцию с Azure DevOps. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в Azure DevOps. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `azure-pipelines.yml`, который отвечает за работу пайплайнов в Azure DevOps (необходимо указать URL вашего ресурса):

```yml
# azure-pipelines.yml
trigger:
- main

pool:
  vmImage: ubuntu-latest
strategy:
  matrix:
    Python37:
      python.version: '3.7'

steps:
# Install project dependencies
- script: |
    python -m pip install --upgrade pip
    pip install -r requirements.txt
  displayName: 'Install dependencies'

# Execute test and generate report
- script: |
    pytest --junitxml=results/test-results.xml
src/tests.py
  displayName: 'Run tests'

# Install Test IT CLI and submit test results to TMS
- script: |
    pip install testit-cli
    testit results import \
      --url <YOUR_INSTANCE_URL> \
      --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
      --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
      --testrun-name "Azure DevOps test run" \
      --results results/test-results.xml
  displayName: 'Submit results'
  condition: always()
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в Azure DevOps. При этом добавляются дополнительные независимые шаги для создания, завершения прогона (тест-рана) и запуска тестов.

Ниже представлен пример файла `azure-pipelines.yml`, который отвечает за работу пайплайнов в Azure DevOps (необходимо указать URL вашего ресурса и токен переменной):

```yml
# azure-pipelines.yml
trigger:
- main

pool:
    vmImage: ubuntu-latest

jobs:
- job: Create
  steps:
  # Install Test IT CLI, create testrun and save testrun
id to variable
  - script: |
      pip install testit-cli
      testit testrun create \
        --url <YOUR_INSTANCE_URL> \
        --project-id 5236eb3f-7c05-46f9-a609-dc0278896464
\
        --token <TOKEN_VARIABLE> \
        --testrun-name "Azure DevOps test run" \
        --output output.txt
      echo "##vso[task.setvariable
variable=testRunId;isoutput=true]$(cat output.txt)"
    name: 'create'

- job: ParallelTesting
  dependsOn: Create
  variables:
    testRunId: $[
dependencies.Create.outputs['create.testRunId'] ]
  strategy:
    parallel: 2
  steps:
  # Install project dependencies
  - script: |
      python -m pip install --upgrade pip
      pip install -r requirements.txt
    displayName: 'Install dependencies'

  # Execute test and generate report
  - script: |
      pytest --junitxml=results/test-results.xml
```

```
src/tests.py
    displayName: 'Run tests'

  # Install Test IT CLI and submit test results to TMS
  - script: |
      pip install testit-cli
      testit results upload \
        --url <YOUR_INSTANCE_URL> \
        --testrun-id $(testRunId) \
        --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
        --results results/test-results.xml
    displayName: 'Submit results'
    condition: always()

- job: Complete
  condition: always()
  dependsOn: [ParallelTesting, Create]
  variables:
    testRunId: $[
dependencies.Create.outputs['create.testRunId'] ]
  steps:
  # Install Test IT CLI and complete testrun
  - script: |
      pip install testit-cli
      testit testrun complete \
        --url <YOUR_INSTANCE_URL> \
        --testrun-id $(testRunId)
    displayName: 'Complete'
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте секреты Azure DevOps для хранения чувствительной информации. Подробнее с секретами вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM



Bamboo

Test IT поддерживает интеграцию с Bamboo. Для настройки используется всего несколько строк кода, поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в Bamboo. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `bamboo.yaml`, который отвечает за работу пайплайнов в Bamboo (необходимо указать URL вашего ресурса):

```yml
# bamboo-specs/bamboo.yaml
plan:
  project-key: <PROJECT_KEY>
  key: BI
  name: Bamboo Integration

stages:
  - Test:
    - Tests

Tests:
    tasks:
        - script:
            # Install project dependencies
            - pip install -r requirements.txt

            # Execute test and generate report
            - pytest --junitxml=results/test-results.xml
src/tests.py
    final-tasks:
        - script:
            # Install Test IT CLI
            - pip install testit-cli

            # Submit test results to TMS
            - testit
              --url <YOUR_INSTANCE_URL> \
              --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464 \
              --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
              --testrun-name "Bamboo test run" \
              --results results/test-results.xml
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в Bamboo. При этом добавляются дополнительные независимые шаги для создания, завершения прогона (тест-рана) и запуска тестов.

Ниже представлен пример файла `bamboo.yaml`, который отвечает за работу пайплайнов в Bamboo (необходимо указать URL вашего ресурса и токен переменной):

```yml
# bamboo-specs/bamboo.yaml
version: 2
plan:
  project-key: <PROJECT_KEY>
  key: BI
  name: Bamboo Integration

stages:
  - Create:
    - Create testrun
  - Test:
      jobs:
        - Tests 1
        - Tests 2
  - Complete:
    - Complete testrun

Create testrun:
    tasks:
        - script:
            # Install Test IT CLI
            - pip install testit-cli

            # Create testrun
            - testit testrun create \
              --url <YOUR_INSTANCE_URL> \
              --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464 \
              --token <TOKEN_VARIABLE> \
              --testrun-name "Bamboo test run" \
              --output output.txt
    artifacts:
        -
           name: Reports
           location: .
           pattern: 'output.txt'
           shared: true

Tests 1:
    tasks:
        - script:
            # Install project dependencies
```

```
            - pip install -r requirements.txt

            # Execute test and generate report
            - pytest --junitxml=results/test-results.xml
src/tests.py
    final-tasks:
        - script:
            # Install Test IT CLI
            - pip install testit-cli

            # Submit test results to TMS
            - testit results upload \
              --url <YOUR_INSTANCE_URL> \
              --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
              --testrun-id "$(cat output.txt)" \
              --results results/test-results.xml

Tests 2:
    tasks:
        - script:
            # Install project dependencies
            - pip install -r requirements.txt

            # Execute test and generate report
            - pytest --junitxml=results/test-results.xml
src/tests.py
    final-tasks:
        - script:
            # Install Test IT CLI
            - pip install testit-cli

            # Submit test results to TMS
            - testit results upload \
              --url <YOUR_INSTANCE_URL> \
              --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
              --testrun-id "$(cat output.txt)" \
              --results results/test-results.xml

Complete testrun:
    tasks:
        - script:
```

```
            # Install Test IT CLI
            - pip install testit-cli

            # Complete testrun
            - testit testrun complete \
              --url <YOUR_INSTANCE_URL> \
              --testrun-id "$(cat output.txt)"
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте переменные Bamboo для хранения чувствительной информации. Подробнее с секретами вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM



Bitbucket

Test IT поддерживает интеграцию с Bitbucket. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в Bitbucket. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `bitbucket-pipelines`, который отвечает за работу пайплайнов в Bitbucket (необходимо указать URL вашего ресурса):

```yml
# bitbucket-pipelines.yml
image: python:3.8

pipelines:
  default:
    - step:
        name: Run tests and generate report
        script:
          # Install project dependencies
          - pip install -r requirements.txt

          # Execute test and generate report
          - pytest --junitxml=results/test-results.xml
src/tests.py
        after-script:
          # Install Test IT CLI
          - pip install testit-cli

          # Submit test results to TMS
          - testit
            --url <YOUR_INSTANCE_URL>
            --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464
            --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb
            --testrun-name "Bitbucket test run"
            --results results/test-results.xml
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в Bitbucket. При этом добавляются дополнительные независимые шаги для создания, завершения прогона (тест-рана) и запуска тестов.

Ниже представлен пример файла `bitbucket-pipelines.yml`, который отвечает за работу пайплайнов в Bitbucket (необходимо указать URL вашего ресурса и токен переменной):

```yml
# bitbucket-pipelines.yml
image: python:3.8

pipelines:
  default:
    - step:
        name: Create testrun
        script:
          # Install Test IT CLI
          - pip install testit-cli

          # Create testrun
          - testit testrun create \
            --url <YOUR_INSTANCE_URL>
            --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464
            --token <TOKEN_VARIABLE> \
            --testrun-name "Bitbucket test run"
            --output output.txt

          # Export TMS_TEST_RUN_ID env variable
          - echo "export TMS_TEST_RUN_ID=$(cat
output.txt)" >> tms_env.sh
        artifacts:
          - tms_env.sh
    - parallel:
      - step:
          name: Test group 1
          script:
            # Install project dependencies
            - pip install -r requirements.txt

            # Execute test and generate report
            - pytest --junitxml=results/test-results.xml
src/tests.py
          after-script:
            # Use the artifact from the previous step
            - cat tms_env.sh
            - source tms_env.sh

            # Install Test IT CLI
            - pip install testit-cli
```

```
43
44            # Submit test results to TMS
45            - testit results upload \
46              --url <YOUR_INSTANCE_URL>
47              --configuration-id 15dbb164-c1aa-4cbf-830c-
48  8c01ae14f4fb
49              --testrun-name "Bitbucket test run"
50              --results results/test-results.xml
51        - step:
52            name: Test group 2
53            script:
54              # Install project dependencies
55              - pip install -r requirements.txt
56
57              # Execute test and generate report
58              - pytest --junitxml=results/test-results.xml
59  src/tests.py
60            after-script:
61              # Use the artifact from the previous step
62              - cat tms_env.sh
63              - source tms_env.sh
64
65              # Install Test IT CLI
66              - pip install testit-cli
67
68              # Submit test results to TMS
69              - testit results upload \
70                --url <YOUR_INSTANCE_URL>
71                --configuration-id 15dbb164-c1aa-4cbf-830c-
72  8c01ae14f4fb
73                --testrun-name "Bitbucket test run"
74                --results results/test-results.xml
75      - step:
76          name: Complete
77          script:
78            # Use the artifact from the previous step
79            - cat tms_env.sh
80            - source tms_env.sh
81
            # Install Test IT CLI
            - pip install testit-cli
```

```
# Complete testrun
- testit testrun complete \
  --url <YOUR_INSTANCE_URL>
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте переменные и секреты Bitbucket для хранения чувствительной информации. Подробнее с секретами вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM



CircleCI

Test IT поддерживает интеграцию с CircleCI. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в CircleCI. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `.circleci/config.yml`, который отвечает за работу пайплайнов в CircleCI (необходимо указать URL вашего ресурса):

```yml
# .circleci/config.yml
version: 2.1

jobs:
  test:
    docker:
      - image: python:3.10-bullseye

    steps:
      - checkout

      - attach_workspace:
          at: ~/

     # Install project dependencies
      - run:
          command:  |
            python -m pip install --upgrade pip
            pip install -r requirements.txt

      # Execute test and generate report
      - run:
          command: |
            pytest --junitxml=results/test-results.xml
src/tests.py

      # Install Test IT CLI and submit test results to TMS
      - run:
          when: always
          command: |
            pip install testit-cli
            testit results import \
              --url <YOUR_INSTANCE_URL> \
              --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464 \
              --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
              --testrun-name "CircleCI test run" \
              --results results/test-results.xml

      - store_artifacts:
          path: results
```

```
            destination: test_results
        - store_test_results:
            path: results

workflows:
  version: 2
  full:
    jobs:
      - test
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в CircleCI. При этом добавляются дополнительные независимые шаги для создания, завершения прогона и запуска тестов.

Ниже представлен пример файла `.circleci/config.yml`, который отвечает за работу пайплайнов в CircleCI (необходимо указать URL вашего ресурса и токен переменной):

```yml
# .circleci/config.yml
version: 2.1

jobs:
  create:
    docker:
      - image: python:3.10-bullseye

    steps:
      - attach_workspace:
          at: ~/

      # Install Test IT CLI and create testrun
      - run:
          command: |
            pip install testit-cli
            testit testrun create \
              --url <YOUR_INSTANCE_URL> \
              --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464 \
              --token <TOKEN_VARIABLE> \
              --testrun-name "CircleCI test run" \
              --output output.txt
      - persist_to_workspace:
          root: ~/
          paths:
            - project

  test:
    docker:
      - image: python:3.10-bullseye

    parallelism: 4

    steps:
      - checkout

      - attach_workspace:
          at: ~/

      # Install project dependencies
      - run:
```

```
          command:  |
            python -m pip install --upgrade pip
            pip install -r requirements.txt

      # Execute test and generate report
      - run:
          command: |
            pytest --junitxml=results/test-results.xml
src/tests.py

      # Install Test IT CLI and submit test results to TMS
      - run:
          when: always
          command: |
            pip install testit-cli
            testit results upload \
              --url <YOUR_INSTANCE_URL> \
              --testrun-id "$(cat output.txt)" \
              --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
              --results results/test-results.xml

      - store_artifacts:
          path: results
          destination: test_results
      - store_test_results:
          path: results

  complete:
    docker:
      - image: python:3.10-bullseye

    steps:
      - attach_workspace:
          at: ~/

      # Install Test IT CLI and complete testrun
      - run:
          when: always
          command: |
            pip install testit-cli
            testit testrun complete \
              --url <YOUR_INSTANCE_URL> \
```

```
              --testrun-id "$(cat output.txt)"
workflows:
  version: 2
  full:
    jobs:
      - create
      - test:
          requires:
            - create
      - complete:
          requires:
            - test
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте секреты CircleCI для хранения чувствительной информации. Подробнее с секретами вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM



GitFlic

Test IT поддерживает интеграцию с GitFlic. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Смотрите также: Настройка интеграции с GitFlic с помощью вебхуков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitFlic
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

> **Используйте переменную окружения**
>
> Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN` . О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в GitFlic. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `.gitflic-ci.yml`, который отвечает за работу пайплайнов в GitFlic (необходимо указать URL вашего ресурса):

```yml
# .gitflic-ci.yml
default:
  image: python:3.10-bullseye

stages:
  - test

test:
  stage: test
  script:
    # Install project dependencies
    - pip install -r requirements.txt

    # Execute test and generate report
    - pytest --junitxml=results/test-results.xml
src/tests.py
  after_script:
    # Install Test IT CLI
    - pip install testit-cli

    # Submit test results to TMS
    - testit results import \
        --url <YOUR_INSTANCE_URL> \
        --project-id 5236eb3f-7c05-46f9-a609-dc0278896464
\
        --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
        --testrun-name "GitFlic test run" \
        --results results/test-results.xml
  artifacts:
    when: always
    reports:
      junit:
        - results/test-results.xml
```

Больше информации о запуске конвейеров (пайплайнов) вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в GitFlic. При этом добавляются дополнительные независимые шаги для создания, завершения прогона (тест-рана) и запуска тестов.

Ниже представлен пример файла `.gitflic-ci.yml`, который отвечает за работу пайплайнов в GitFlic (необходимо указать URL вашего ресурса и токен переменной):

```yml
# .gitflic-ci.yml
default:
  image: python:3.10-bullseye

stages:
  - create
  - test
  - complete

create:
  stage: create
  script:
    # Install Test IT CLI
    - pip install testit-cli

    # Create testrun
    - testit testrun create \
      --url <YOUR_INSTANCE_URL> \
      --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
      --token <TOKEN_VARIABLE> \
      --testrun-name "GitFlic test run" \
      --output output.txt

    # Export TMS_TEST_RUN_ID env variable
    - echo "TMS_TEST_RUN_ID=$(cat output.txt)" > tms.env
  artifacts:
    reports:
      dotenv: tms.env

test:
  parallel: 4
  stage: test
  script:
    # Install project dependencies
    - pip install -r requirements.txt

    # Execute test and generate report
    - pytest --junitxml=results/test-results.xml
src/tests.py
  after_script:
    # Install Test IT CLI
    - pip install testit-cli
```

```
43
44      # Submit test results to TMS
45      - testit results upload \
46        --url <YOUR_INSTANCE_URL> \
47        --configuration-id 15dbb164-c1aa-4cbf-830c-
48  8c01ae14f4fb \
49        --results results/test-results.xml
50    artifacts:
51      when: always
52      reports:
53        junit:
54          - results/test-results.xml
55    dependencies:
56      - create
57
58  complete:
59    stage: complete
60    script:
61      # Install Test IT CLI
62      - pip install testit-cli
63
64      # Complete testrun
65      - testit testrun complete \
66        --url <YOUR_INSTANCE_URL>
67    dependencies:
68      - test
69      - create
    when: always
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте секреты GifFlic для работы с чувствительной информацией. Подробнее — в официальной документации GitFlic .

Обновлено: 4/22/26, 2:57 PM



GitHub Actions

Test IT поддерживает интеграцию с GitHub Actions. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в GitHub Actions. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `.python-app.yml`, который отвечает за работу пайплайнов в GitHub Actions (необходимо указать URL вашего ресурса):

```yml
# .python-app.yml
name: Python application

on:
  push:
    branches: [ "main" ]

permissions:
  contents: read

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3
    - name: Set up Python 3.10
      uses: actions/setup-python@v3
      with:
        python-version: "3.10"

    # Install project dependencies
    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt

    # Execute test and generate report
    - name: Test with pytest
      run: |
        pytest --junitxml=results/test-results.xml
src/tests.py

    # Install Test IT CLI and submit test results to TMS
    - name: Submit results
      if: always()
      run: |
        pip install testit-cli
        testit results import \
          --url <YOUR_INSTANCE_URL> \
          --project-id 5236eb3f-7c05-46f9-a609-
```

```
dc0278896464 \
          --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
          --testrun-name "GitHub test run" \
          --results results/test-results.xml
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в GitHub Actions. При этом добавляются дополнительные независимые шаги для создания, завершения прогона и запуска тестов.

Ниже представлен пример файла `.python-app.yml`, который отвечает за работу пайплайнов в GitHub Actions (необходимо указать URL вашего ресурса и токен переменной):

```yml
# .python-app.yml
name: Python application

on:
  push:
    branches: [ "main" ]

permissions:
  contents: read

jobs:
  create:
    runs-on: ubuntu-latest
    outputs:
      test-run-id: ${{ steps.create-testrun.outputs.test-run-id }}
    steps:
    # Install Test IT CLI, create testrun and save testrun id to variable
    - name: Create testrun
      id: create-testrun
      run: |
        pip install testit-cli
        testit testrun create \
          --url <YOUR_INSTANCE_URL> \
          --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
          --token <TOKEN_VARIABLE> \
          --testrun-name "GitHub test run" \
          --output output.txt
        ID=$(cat output.txt)
        echo "test-run-id=$ID" >> $GITHUB_OUTPUT
  test:
    needs: create
    runs-on: ubuntu-latest

    strategy:
      fail-fast: false
      matrix:
        ci_index: [0, 1, 2, 3]
        ci_total: [4]

```

```yaml
    steps:
    - uses: actions/checkout@v3
    - name: Set up Python 3.10
      uses: actions/setup-python@v3
      with:
        python-version: "3.10"

    # Install project dependencies
    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt

    # Execute test and generate report
    - name: Test with pytest
      run: |
        pytest --junitxml=results/test-results.xml
src/tests.py

    # Install Test IT CLI and submit test results to TMS
    - name: Submit results
      if: always()
      run: |
        pip install testit-cli
        testit results upload \
          --url <YOUR_INSTANCE_URL> \
          --testrun-id $(testRunId) \
          --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
          --results results/test-results.xml
    env:
          CI_TOTAL: ${{ matrix.ci_total }}
          CI_INDEX: ${{ matrix.ci_index }}

  complete:
    needs: [create, test]
    if: always()
    runs-on: ubuntu-latest
    steps:
    # Install Test IT CLI and complete testrun
    - name: Complete testrun
      if: always()
      run: |
```

```
pip install testit-cli
testit testrun complete \
  --url <YOUR_INSTANCE_URL> \
  --testrun-id $(testRunId)
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте секреты GitHub для хранения чувствительной информации. Подробнее с секретами вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM



GitLab

Test IT поддерживает интеграцию с GitLab. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в GitLab. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `.gitlab-ci.yml`, который отвечает за работу пайплайнов в GitLab (необходимо указать URL вашего ресурса):

```yml
# .gitlab-ci.yml
default:
  image: python:3.10-bullseye

stages:
  - test

test:
  stage: test
  script:
    # Install project dependencies
    - pip install -r requirements.txt

    # Execute test and generate report
    - pytest --junitxml=results/test-results.xml
src/tests.py
  after_script:
    # Install Test IT CLI
    - pip install testit-cli

    # Submit test results to TMS
    - testit results import \
        --url <YOUR_INSTANCE_URL> \
        --project-id 5236eb3f-7c05-46f9-a609-dc0278896464
\
        --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
        --testrun-name "GitLab test run" \
        --results results/test-results.xml
  artifacts:
    when: always
    reports:
      junit:
        - results/test-results.xml
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в GitLab. При этом добавляются дополнительные независимые шаги для создания, завершения прогона (тест-рана) и запуска тестов.

Ниже представлен пример файла `.gitlab-ci.yml`, который отвечает за работу пайплайнов в GitLab (необходимо указать URL вашего ресурса и токен переменной):

```yml
# .gitlab-ci.yml
default:
  image: python:3.10-bullseye

stages:
  - create
  - test
  - complete

create:
  stage: create
  script:
    # Install Test IT CLI
    - pip install testit-cli

    # Create testrun
    - testit testrun create \
      --url <YOUR_INSTANCE_URL> \
      --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
      --token <TOKEN_VARIABLE> \
      --testrun-name "GitLab test run" \
      --output output.txt

    # Export TMS_TEST_RUN_ID env variable
    - echo "TMS_TEST_RUN_ID=$(cat output.txt)" > tms.env
  artifacts:
    reports:
      dotenv: tms.env

test:
  parallel: 4
  stage: test
  script:
    # Install project dependencies
    - pip install -r requirements.txt

    # Execute test and generate report
    - pytest --junitxml=results/test-results.xml
src/tests.py
  after_script:
    # Install Test IT CLI
    - pip install testit-cli
```

```
43
44      # Submit test results to TMS
45      - testit results upload \
46        --url <YOUR_INSTANCE_URL> \
47        --configuration-id 15dbb164-c1aa-4cbf-830c-
48  8c01ae14f4fb \
49        --results results/test-results.xml
50    artifacts:
51      when: always
52      reports:
53        junit:
54          - results/test-results.xml
55    dependencies:
56      - create
57
58  complete:
59    stage: complete
60    script:
61      # Install Test IT CLI
62      - pip install testit-cli
63
64      # Complete testrun
65      - testit testrun complete \
66        --url <YOUR_INSTANCE_URL>
67    dependencies:
68      - test
69      - create
    when: always
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте проектные переменные GitLab для хранения чувствительной информации. Подробнее с переменными вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM



Jenkins

Test IT поддерживает интеграцию с Jenkins. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в Azure Jenkins. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `Jenkinsfile`, который отвечает за работу пайплайнов в Jenkins (необходимо указать URL вашего ресурса):

```groovy
# Jenkinsfile
pipeline {
    agent {
        docker { image 'python:3.10-bullseye' }
    }

    stages {
        stage('Test') {
            steps {
                // Install project dependencies
                sh '''
                  python -m pip install --upgrade pip
                  pip install -r requirements.txt
                '''

                // Execute test and generate report
                sh 'pytest --junitxml=results/test-
results.xml src/tests.py'

                // Install Test IT CLI and submit test
results to TMS
                sh '''
                  pip install testit-cli
                  testit results import \
                    --url <YOUR_INSTANCE_URL> \
                    --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464 \
                    --configuration-id 15dbb164-c1aa-4cbf-
830c-8c01ae14f4fb \
                    --testrun-name "Jenkins test run" \
                    --results results/test-results.xml
                '''
            }
        }
    }
    post {
        always {
            junit 'results/*.xml'
        }
    }
}
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Параллельный запуск

Test IT поддерживает параллельный запуск автоматизированных тестов в Jenkins. При этом добавляются дополнительные независимые шаги для создания, завершения прогона (тест-рана) и запуска тестов.

Ниже представлен пример файла `Jenkinsfile`, который отвечает за работу пайплайнов в Jenkins (необходимо указать URL вашего ресурса и токен переменной):

```groovy
# Jenkinsfile
pipeline {
    agent {
        docker { image 'python:3.10-bullseye' }
    }

    stages {
        stage('Create') {
            steps {
                // Install Test IT CLI and create testrun
                sh '''
                  pip install testit-cli
                  testit testrun create \
                    --url <YOUR_INSTANCE_URL> \
                    --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464 \
                    --token <TOKEN_VARIABLE> \
                    --testrun-name "Jenkins test run" \
                    --output output.txt
                '''

                // Install project dependencies
                sh '''
                  python -m pip install --upgrade pip
                  pip install -r requirements.txt
                '''
            }
        }

        stage('Test') {
            steps {
                parallel(
                    thread_1: {
                        // Execute test and generate
report
                        sh 'pytest --
junitxml=results/test-results-1.xml src/tests.py'

                        // Submit test results to TMS
                        sh '''
                          testit results upload \
                            --url <YOUR_INSTANCE_URL> \
```

```
                            --testrun-id "$(cat
output.txt)" \
                            --configuration-id 15dbb164-
c1aa-4cbf-830c-8c01ae14f4fb \
                            --results results/test-
results-1.xml
                        '''

                    },
                    thread_2: {
                        // Execute test and generate
report
                        sh 'pytest --
junitxml=results/test-results-2.xml src/tests.py'

                        // Submit test results to TMS
                        sh '''
                          testit results upload \
                            --url <YOUR_INSTANCE_URL> \
                            --testrun-id "$(cat
output.txt)" \
                            --configuration-id 15dbb164-
c1aa-4cbf-830c-8c01ae14f4fb \
                            --results results/test-
results-2.xml
                        '''
                    }
                    ...
                    thread_n: {
                        // Execute test and generate
report
                        sh 'pytest --
junitxml=results/test-results-n.xml src/tests.py'

                        // Submit test results to TMS
                        sh '''
                          testit results upload \
                            --url <YOUR_INSTANCE_URL> \
                            --testrun-id "$(cat
output.txt)" \
                            --configuration-id 15dbb164-
c1aa-4cbf-830c-8c01ae14f4fb \
                            --results results/test-
```

```
results-n.xml
                        '''
                    }
                )
            }
        }
    }
    post {
        always {
            junit 'results/*.xml'
            // Complete testrun
            sh '''
              testit testrun complete \
                --url <YOUR_INSTANCE_URL> \
                --testrun-id "$(cat output.txt)"
            '''
        }
    }
}
```

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте секреты Jenkins для хранения чувствительной информации. Подробнее с секретами вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM



TeamCity

Test IT поддерживает интеграцию с TeamCity. Для настройки используется всего несколько строк кода и поддерживаются любые тестовые фреймворки и платформы.

Вы можете интегрировать как простые проекты (только с запуском тестов), так и сложные проекты (несколько отдельных проектов), которые запускаются в несколько независимых потоков.

Независимо от ваших требований процесс интеграции будет одинаков:

1. Запуск автоматизированных тестов в пайплайнах GitLab
2. **Для Test IT CLI:** Генерация JUnit XML отчета с результатами автоматизированных тестов
3. Загрузка результатов с помощью Test IT CLI или адаптера Test IT для тестовых фреймворков

Используйте переменную окружения

Чтобы утилита Test IT CLI работала корректно, передавайте приватный токен через переменную окружения `TMS_TOKEN`. О безопасной передаче токена читайте в разделе Безопасное хранение данных.

Стандартный запуск

Данная инструкция описывает запуск простого проекта в Azure TeamCity. Она актуальна для любых тестовых фреймворков и платформ. В качестве примера описан запуск с помощью тестового фреймворка pytest для Python.

Для быстрой и простой интеграции с Test IT вы можете использовать утилиту `Test IT CLI`, которая позволяет загрузить все результаты автотестов с помощью одной команды.

Ниже представлен пример файла `settings.kts`, который отвечает за работу пайплайнов в TeamCity (необходимо указать URL вашего ресурса):

```kt
import jetbrains.buildServer.configs.kotlin.*
import
jetbrains.buildServer.configs.kotlin.buildFeatures.parallelTe
import
jetbrains.buildServer.configs.kotlin.buildFeatures.perfmon
import jetbrains.buildServer.configs.kotlin.buildSteps.script
import jetbrains.buildServer.configs.kotlin.triggers.vcs


version = "2022.10"

project {

    buildType(Build)
}

object Build : BuildType({
    name = "Build"

    vcs {
        root(DslContext.settingsRoot)
    }

    steps {
        script {
            name = "Install dependencies"
            scriptContent = """
                python -m pip install --upgrade pip
                pip install -r requirements.txt
            """.trimIndent()
        }

        script {
            name = "Run tests"
            scriptContent = """ pytest --junitxml=results/tes
results.xml src/tests.py """
        }
        script {
            executionMode = BuildStep.ExecutionMode.ALWAYS
            name = "Submit results"
            scriptContent =
                """
```

```
                    pip install testit-cli
                    testit --url <YOUR_INSTANCE_URL> \
                        --project-id 5236eb3f-7c05-46f9-a609-
dc0278896464 \
                        --configuration-id 15dbb164-c1aa-4cbf
830c-8c01ae14f4fb \
                        --testrun-name "Teamcity test run" \
                        --results results/test-results.xml
                """.trimIndent()
        }
    }

    triggers {
        vcs {
        }
    }

    features {
        perfmon {
        }
    }
})
```

Простым добавлением необходимых шагов вы можете запускать множество необходимых вам тестов и загружать их в Test IT. Больше информации вы можете найти в официальной документации .

Безопасное хранение данных

Для работы утилиты Test IT CLI вам необходимо использовать приватный токен для авторизации. Хранение данного токена и другой чувствительной информации в коде строго не рекомендуется, так как доступ к этой информации будет иметь любой у кого есть доступ к коду.

Используйте переменные TeamCity для хранения чувствительной информации. Подробнее с переменными вы можете ознакомится в официальной документации .

Обновлено: 4/21/26, 7:08 PM

Шаблон интеграции JMeter, Test IT и GitLab CI/CD

Содержание

- Назначение и функциональность
- Инструменты, применяемые в шаблоне
- Структура проекта
- Вынесенная конфигурация JMeter
- Конфигурация
- Процесс работы шаблона
- Подготовка инфраструктуры GitLab и JMeter
- Настройка Test IT
- Условия успешного использования шаблона
- Документация

Назначение и функциональность

Шаблон интеграции JMeter, Test IT и GitLab CI/CD (далее — "Шаблон") представляет готовое решение для интеграции тестов производительности Apache JMeter с Test IT и GitLab CI/CD. Он предназначен для автоматизации тестирования производительности. При работе с шаблоном выполняются операции:

1. Запуск тестов с помощью GitLab CI/CD
2. Запуск тест-планов JMeter в режиме без графического интерфейса
3. Обработка результатов тестов и создание отчетов
4. Загрузка результатов тестов, логов и HTML-отчетов в Test IT

Шаблон поддерживает два основных метода запуска тестов:

- **Ручной/запланированный запуск из GitLab CI/CD** — осуществляется непосредственно из интерфейса GitLab или по расписанию
- **Запуск через вебхуки из Test IT** — производится из интерфейса Test IT, что позволяет динамически передавать параметры

Инструменты, применяемые в шаблоне

- **GitLab CI/CD** — система CI/CD, оркестрирующая весь процесс, от настройки окружения до выполнения тестов и составления отчетов
- **Apache JMeter** — инструмент для тестирования производительности, используемый для выполнения тест-планов `.jmx`
- **Test IT** — система управления тестированием, в которой хранятся и анализируются результаты тестов
- **Test IT CLI** — утилита командной строки для загрузки результатов в Test IT
- **JMeter JUnit Reporter Plugin** — плагин для JMeter, который генерирует результаты тестов в формате JUnit XML, совместимом с Test IT

Структура проекта

- Для использования шаблона создайте проект со следующей структурой:

```text
/ваш-проект
|-- .gitlab-ci.yml
|-- common.properties
|-- /jmeter_tests
|   |-- Login_Test.jmx
|   `-- Checkout_Process.jmx
|-- /jmeter_plugins
|   `-- jmeter-junit-reporter-3.1.jar
|-- /test_data
|   `-- users.csv
`-- /jmeter_reports (создается автоматически)
```

  В проект входят файлы и каталоги:

  - `.gitlab-ci.yml` — основной файл конфигурации CI/CD
  - `common.properties` — основной файл свойств JMeter для параметризации тестов

- `jmeter_tests/` — каталог, содержащий все тест-планы JMeter (файлы `.jmx`)
- `jmeter_plugins/` — каталог, содержащий необходимые плагины JMeter, такие как JUnit Reporter
- `test_data/` — каталог, содержащий любые внешние файлы данных, необходимых для ваших тестов (например, наборы данных CSV)
- `jmeter_reports/` — каталог, содержащий артефакты тестов (логи, файлы JTL, отчеты JUnit и HTML). Создается автоматически конвейером CI/CD.

Вынесенная конфигурация JMeter

Преимущества вынесенной конфигурации

В шаблоне используется подход, при котором конфигурационные файлы и плагины JMeter выносятся в отдельную папку `JmeterRep`. Это обеспечивает следующие преимущества:

- **Контроль версий**. Все конфигурационные файлы и плагины находятся под контролем версий Git, что позволяет отслеживать изменения и возвращаться к предыдущим версиям.
- **Централизованное управление**. Все настройки JMeter собраны в одном месте, что упрощает администрирование и поддержку.
- **Переносимость**. Конфигурация легко переносится между различными окружениями и серверами.
- **Изоляция**. Изменения в конфигурации не затрагивают стандартную установку JMeter.

Структура вынесенной конфигурации

Вынесенная конфигурация имеет следующую структуру:

```text
/JmeterRep
|-- /config                    # Конфигурационные файлы JMeter
|   |-- jmeter.properties       # Основной файл конфигурации JMeter
|   |-- common.properties       # Пользовательские свойства для тестов
|   |-- log4j2.xml              # Конфигурация логирования
|   |-- user.properties         # Дополнительные пользовательские настройки
|   `-- system.properties       # Системные свойства JMeter
|-- /extPlugins                 # Внешние плагины JMeter
|   |-- jmeter-plugins-*.jar    # JAR-файлы плагинов
|   `-- ApacheJMeter_*.jar      # Компоненты JMeter
`-- /JMeter-TestIT-GitLab-Template  # Шаблон интеграции
```

Стандартное расположение файлов JMeter

Конфигурационные файлы JMeter как правило располагаются в директориях:

- `$JMETER_HOME/bin/jmeter.properties`
- `$JMETER_HOME/bin/user.properties`
- `$JMETER_HOME/bin/system.properties`
- `$JMETER_HOME/bin/log4j2.xml`

Плагины JMeter как правило располагаются в директориях:

- `$JMETER_HOME/lib/ext/` — для `JAR` -файлов плагинов
- `$JMETER_HOME/lib/` — для основных библиотек

Настройка вынесенной конфигурации

Для использования вынесенной конфигурации:

1. При запуске JMeter укажите путь к конфигурационным файлам:

```sh
jmeter -p /path/to/JmeterRep/config/jmeter.properties
```

2. Укажите путь к дополнительным плагинам через системное свойство:

```sh
-Djmeter.ext.dirs=/path/to/JmeterRep/extPlugins
```

3. **Опционально:** Вместо указания пути к дополнительным плагинам настройте переменные окружения:

```sh
export
JMETER_PROPERTIES=/path/to/JmeterRep/config/jmeter.properti
export JMETER_EXT_DIRS=/path/to/JmeterRep/extPlugins
```

Использование виртуальных ссылок на Windows

В операционной системе Windows для связи вынесенных конфигурационных файлов и плагинов с основной установкой JMeter используются виртуальные ссылки (symbolic links).

- Для папки с плагинами используется ссылка:

```cmd
mklink /D "%JMETER_HOME%\lib\ext" "
<YOUR_PROJECT_PATH>\extPlugins"
```

- Для конфигурационных файлов используются ссылки:

```cmd
mklink "%JMETER_HOME%\bin\jmeter.properties" "
<YOUR_PROJECT_PATH>\config\jmeter.properties"
mklink "%JMETER_HOME%\bin\common.properties" "
<YOUR_PROJECT_PATH>\config\common.properties"
mklink "%JMETER_HOME%\bin\log4j2.xml" "
<YOUR_PROJECT_PATH>\config\log4j2.xml"
```

Где:

  - `%JMETER_HOME%` — путь к установке JMeter (например, `C:\apache-jmeter-5.6.3`)
  - `<YOUR_PROJECT_PATH>` — путь к вашему проекту с репозиторием

Преимущества использования виртуальных ссылок:

- JMeter находит файлы и папки в стандартных местах
- Фактически файлы остаются в репозитории под контролем версий
- Не требуется изменение параметров запуска JMeter
- Упрощается развертывание и обновление конфигурации

Важно: Для создания виртуальных ссылок на Windows требуются права администратора.

Такой подход позволяет более гибко работать с конфигурационными файлами и параметризировать JMeter, обеспечивая при этом полный контроль версий над всеми настройками и плагинами.

Конфигурация

Файл .gitlab-ci.yml

В файле `.gitlab-ci.yml` содержатся ключевые области для настройки автоматизации:

- Глобальные переменные:

```yml
variables:
  HOST_VM_PERF: "your.performance.server.com"  # Пример: IP-адрес или домен сервера для performance тестирования
  JMETER_application: "cloud"   # По умолчанию используется cloud версия
```

- Переменные для интеграции с Test IT (в шаблоне `.common_job`):

```yml
variables:
  TESTIT_UPLOAD_URL: your_test_it_url  # Пример: замените на ваш адрес Test IT
  TESTIT_PROJECT_ID: 00000000-0000-0000-0000-000000000000  # Пример ID проекта
  TESTIT_CONFIGURATION_ID: 11111111-1111-1111-1111-111111111111  # Пример ID конфигурации
```

- Стадии пайплайна (конвейера):

```yml
stages:
  - prepare      # Подготовительная стадия - загрузка данных, настройка окружения
  - perf_tests   # Основные performance тесты (полные сценарии нагрузочного тестирования)
  - service      # Сервисные тесты (тестирование отдельных API endpoints)
  - custom       # Кастомные тесты (специальные сценарии для конкретных компонентов)
```

- Примеры определения задач (jobs):

  1. Основной performance тест:

```yml
perf-tms-cloud-job:
  extends:
    - .before
    - .common_job
  stage: perf_tests
  variables:
    TEST_PLAN_FILE: "jmx/perfTests/tms.jmx"
  rules:
    - if: '$CI_PIPELINE_SOURCE == "trigger" && $JMETER_testName == "tms" && $JMETER_application == "cloud"'
      when: on_success
    - when: manual
      allow_failure: true
      variables:
        RUN_TYPE: "tst"
```

  2. Сервисный тест:

```yml
perf-auth-cloud-job:
  extends:
    - .before
    - .common_job
  stage: service
  variables:
    TEST_PLAN_FILE: "jmx/service/auth.jmx"
  rules:
    - if: '$CI_PIPELINE_SOURCE == "trigger" &&
$JMETER_testName == "auth" && $JMETER_application ==
"cloud"'
      when: on_success
    - when: manual
      allow_failure: true
```

3. Enterprise-версия теста:

```yml
perf-tms-ent-job:
  extends:
    - .before
    - .common_job
  stage: perf_tests
  variables:
    TEST_PLAN_FILE: "jmx/perfTests/tms.jmx"
    RUN_TYPE: "tst"
    JMETER_application: "ent"  # Переопределяем для
enterprise
```

- Принципы именования:
  - Переменная `TEST_PLAN_FILE` указывает путь к `.jmx` -файлу относительно корня проекта.
  - Имя задачи (job) может быть любым, но должно быть уникальным.
  - Для автоматического запуска из Test IT используется переменная `JMETER_testName` , которая должна соответствовать значению в конфигурации Test IT.

Переменные окружения

- Обязательные переменные CI/CD (настраиваются в **GitLab Settings** > **CI/CD** > **Variables**):

Переменная	**Описание**	**Пример значения**	**Тип**
`TESTIT_PRIVATE_TOKEN`	Приватный токен для доступа к Test IT API	`your-secret-token`	Masked
`SSH_PRIVATE_KEY`	SSH ключ для доступа к performance серверу	`-----BEGIN OPENSSH PRIVATE KEY-----...`	Masked

- Переменные для настройки окружения:

Переменная	**Описание**	**Значение по умолча**
`HOST_VM_PERF`	IP-адрес performance сервера	`your.performance.se`
`JMETER_application`	Тип приложения (cloud/ent)	`cloud`
`TESTIT_UPLOAD_URL`	URL Test IT	`your_test_it_url`
`TESTIT_PROJECT_ID`	ID проекта в Test IT	`00000000-0000-0000-000000000000`

Переменная	Описание	Значение по умолча
`TESTIT_CONFIGURATION_ID`	ID конфигурации в Test IT	`11111111-1111-1111-111111111111`
`PUBLIC_REPORTS_URL`	Базовый URL для HTML отчетов	`your_reports_server`
`JMETER_HOME`	Путь к JMeter на runner	`/opt/jmeter`

- Переменные для запуска из Test IT с помощью вебхуков:

Переменная	Описание	Пример значения
`JMETER_testName`	Имя теста для запуска	`tms` , `auth` , `registration`
`JMETER_application`	Тип приложения	`cloud` , `ent`
`TESTIT_TESTRUN_ID`	ID тест-рана в Test IT	`12345678-1234-1234-1234-123456789012`
`RUN_TYPE`	Тип запуска	`TESTIT_RUN` , `tst`

Файлы JMeter common.properties и jmeter.properties

Для конфигурации Jmeter шаблон использует два ключевых файла: `jmeter.properties` и `common.properties` . Такая связка позволяет гибко управлять настройками.

- `jmeter.properties` — стандартный файл конфигурации JMeter. Его основная задача — указать JMeter на использование дополнительного файла свойств. Добавляется следующая настройка:

```properties
user.properties = ./config/common.properties
```

JMeter загрузит все свойства, определенные в `common.properties` при запуске. Таким образом, не нужно каждый раз указывать путь к `common.properties` через флаг `-q` в командной строке, что упрощает команду запуска в CI/CD.

- `common.properties` — основной файл для параметризации тестов, определяющий все параметры, специфичные для проекта: количество пользователей, целевые хосты, настройки окружения и другие конфигурационные данные. Использование отдельного файла для этих целей делает конфигурацию чистой, переносимой и легко управляемой.
- Работа Jmeter с файлами:
  1. При запуске JMeter читает `jmeter.properties`.
  2. Обнаружив директиву `user.properties=./config/common.properties`, он загружает все свойства из `common.properties`.
  3. **Важно:** Свойства из `common.properties` при загрузке JMeter становятся **параметрами JMeter**, а не переменными. В JMX-сценариях доступ к этим параметрам осуществляется через функцию `${__P(имя_параметра, значение_по_умолчанию)}`. Например, `${__P(threads, 1)}`. Не путайте параметры с обычными переменными JMeter.
- Пример структурированного файла `common.properties`:

```properties
#
================================================================
#                            ОСНОВНЫЕ ПАРАМЕТРЫ ТЕСТА
#
================================================================
# Количество виртуальных пользователей (потоков)
threads=10
# Время разгона (в секундах)
rampup=5
# Продолжительность теста (в секундах)
duration=300

#
================================================================
#                            ПАРАМЕТРЫ ОКРУЖЕНИЯ
#
================================================================
# Имя хоста или IP-адрес целевого сервера
target.host=your.test.server.com
# Протокол (http или https)
protocol=https
```

Тест-планы JMeter

Тест-планы JMeter создаются в формате `.jmx` .Чтобы сделать ваши тест-планы совместимыми с этим фреймворком:

1. Используйте функцию `${__P(property_name, default_value)}` для чтения значений из файла `common.properties` . - Пример: в группе потоков установите `Number of Threads (users)` в `${__P(threads, 1)}` .
2. Используйте плагин **JUnit Reporter:** 1. Добавьте слушатель `jp@gc - JUnit Reporter` в свой тест-план. 2. Настройте его для сохранения отчета по пути, который может найти скрипт CI. Шаблон `.gitlab-ci.yml` ожидает его по адресу `$REPORT_DIR/junit.xml` , что соответствует `jmeter_reports/Your_Job_Name/junit.xml` .

Процесс работы шаблона

Работа шаблона включает в себя этапы:

1. Обработка переменных
2. Выполнение тестов
3. Интеграция с Test IT

1. Обработка переменных

1. **Определение переменных GitLab CI** происходит в `.gitlab-ci.yml` или в настройках проекта GitLab. Они управляют процессом работы CI/CD.
2. **Запуск тестов из Test IT с помощью вебхука, содержащего переменные Test IT**. Когда тест запускается из Test IT, он может отправлять полезную нагрузку JSON с переменными (`TESTIT_VARIABLES`). Скрипт `.before` в файле CI включает раздел-заполнитель, показывающий, как можно разобрать этот JSON (с помощью инструмента `jq`) и динамически обновить файл `common.properties` перед запуском теста.
3. **Определение свойства JMeter**. JMeter загружает все переменные из `common.properties` во время выполнения. Таким образом, параметры, такие как количество потоков, хосты и продолжительность, передаются в тест-план.

2. Выполнение тестов

1. **Запуск конвейера GitLab CI** (например, вручную, по расписанию или через вебхук Test IT).
2. **Начало выполнения задания**, запуск скрипта `.before` , который подготавливает окружение.
3. **Выполнение скрипта** `.common_job` : a. Скрипт определяет план тестирования для запуска на основе имени задания (`$CI_JOB_NAME`). b. Скрипт запускает JMeter с указанным планом тестирования и файлом свойств. c. JMeter выполняет тест и с помощью плагина JUnit Reporter создает файл `junit.xml` . d. JMeter также создает подробный HTML-отчет.

4. **Вставка ссылки в общедоступный HTML-отчет**. Ссылка помещается в файл `junit.xml` с помощью скрипта. Это делает отчет доступным непосредственно из тестового запуска Test IT.
5. **Вызов `testit` CLI для загрузки результатов `junit.xml` в** Test IT.

Пример выполнения JMeter-теста

В Test IT мы запускаем тесты с использованием следующих параметров:

```text
      # -n : запуск в неинтерактивном режиме (без GUI)
      # -t : путь к файлу тестового плана (.jmx файл)
      # -l : файл для сохранения результатов в формате JTL
      # -e : генерация HTML отчета после выполнения
      # -o : директория для HTML отчета
      # -q : файл с дополнительными свойствами (наша
конфигурация)

      # Задержка только для ночных прогонов (nightly run)
      if [ "$FINAL_RUN_TYPE" = "nr" ]; then
        echo "Nightly run detected - sleeping for 900
seconds to avoid peak hours"
        sleep 900
      else
        echo "Non-nightly run - starting immediately
(FINAL_RUN_TYPE: $FINAL_RUN_TYPE)"
      fi

      jmeter -n -t ./$TEST_PLAN_FILE -l
./reports/results.jtl -e -o ./reports/report -q
./config/common.properties
```

3. Интеграция с Test IT

- **Создание тестового запуска**. Скрипт определяет, создавать ли новый тестовый запуск в Test IT или прикреплять результаты к существующему (если конвейер был запущен с помощью вебхука Test IT, который передает `$TESTIT_TESTRUN_ID`).
- **Загрузка результатов**. Команда `testit results import` выполняет основную работу. Она использует формат JUnit, связывает результаты с

правильным проектом и конфигурацией и прикрепляет весь каталог HTML-отчетов для детального анализа.
- **Организация пространства имен**. Флаг `--namespace` помогает организовать результаты в Test IT, предотвращая коллизии имен тестов, если в одном проекте тестируется несколько приложений.

Подготовка инфраструктуры GitLab и JMeter

Подготовка образа Docker

Важно: Для работы пайплайна необходимо предварительно собрать и разместить Docker образ в реестре контейнеров.

1. Подготовьте сборку Docker-образа:

```sh
# Перейдите в директорию с Dockerfile
cd JMeter-TestIT-GitLab-Template

# Соберите образ
docker build -t your-registry/jmeter-testit:latest .
```

2. Загрузите Docker-образ в реестр:

```sh
# Загрузите образ в ваш Docker реестр
docker push your-registry/jmeter-testit:latest
```

3. Обновите файл .gitlab-ci.yml:

```yml
# Укажите ваш образ в секции image
.common_джоб:
  image: your-registry/jmeter-testit:latest
```

Готовый Dockerfile с зависимостями находится в файле `Dockerfile` и содержит:

- OpenJDK 11 (Alpine Linux)
- Apache JMeter 5.6.3
- Test IT CLI для интеграции

- Необходимые системные утилиты (rsync, ssh, python3)
- Настроенное логирование для CI/CD

Dockerfile с зависимостями (развернуть / свернуть)

```docker
FROM adoptopenjdk/openjdk11:jdk-11.0.26_4-alpine
ARG JMETER_VERSION=5.6.3
ENV JMETER_HOME=/opt/apache-jmeter-${JMETER_VERSION}
ENV PATH=$JMETER_HOME/bin:$PATH
ENV TZ=Europe/Moscow
RUN apk update && apk add --no-cache curl tar bash
openssh rsync tzdata python3 py3-pip \
    && pip install testit-cli \
    && cp /usr/share/zoneinfo/$TZ /etc/localtime \
    && echo $TZ > /etc/timezone \
    && mkdir -p /tmp/dependencies \
    && curl -L
https://archive.apache.org/dist/jmeter/binaries/apache-
jmeter-${JMETER_VERSION}.tgz -o
/tmp/dependencies/apache-jmeter-${JMETER_VERSION}.tgz \
    && tar -xzf /tmp/dependencies/apache-
jmeter-${JMETER_VERSION}.tgz -C /opt/ \
    && rm -rf /tmp/dependencies
# Модифицируем конфигурацию log4j2 для вывода логов в
консоль (stdout)
RUN sed -i '/<Appenders>/a\
   <Console name="Console" target="SYSTEM_OUT">\
     <PatternLayout pattern="%d{yyyy-MM-dd
HH:mm:ss,SSS} %-5p [%c] - %m%n"/>\
   </Console>' $JMETER_HOME/bin/log4j2.xml \
   && sed -i '/<Root level="/a\
      <AppenderRef ref="Console"/>'
$JMETER_HOME/bin/log4j2.xml
```

Быстрый старт

1. Клонируйте репозиторий. Начните с этого репозитория-шаблона.
2. Настройте `.gitlab-ci.yml` :

- Заполните `TESTIT_URL`, `TESTIT_PROJECT_ID` и `TESTIT_CONFIGURATION_ID`.
- Установите `TESTIT_PRIVATE_TOKEN` как маскированную переменную CI/CD в GitLab.
- Обновите `HOST_VM_PERF` и другие переменные, специфичные для окружения.

3. Создайте файл `common.properties` и добавьте в него параметры вашего теста.
4. Добавьте тест-планы JMeter: поместите ваши файлы `.jmx` в каталог `jmeter_tests/`.
   - Убедитесь, что они параметризованы с помощью функций `${__P()}`.
   - Добавьте слушатель JUnit Reporter в каждый тест-план.
5. Добавьте плагины JMeter: поместите `jmeter-junit-reporter-*.jar` и любые другие необходимые плагины в каталог `jmeter_plugins/`.
6. Создайте задания GitLab CI/CD: для каждого файла `.jmx` создайте соответствующее задание в `.gitlab-ci.yml`, которое расширяет `.common_job`. Помните о соглашении об именах!
7. Запустите конвейер: отправьте свои изменения в GitLab и запустите конвейер.

Пошаговая настройка на основе существующего проекта

Если у вас уже есть настроенный GitLab CI для JMeter-тестов, выполните следующие шаги для интеграции с шаблоном:

Шаг 1: Анализ существующей структуры

1. Проверьте структуру ваших задач (jobs):
   - Определите, какие стадии используются (`perf_tests`, `service`, `custom`)
   - Найдите переменные `TEST_PLAN_FILE` в ваших задачах
   - Проверьте правила запуска (`rules`) для каждой задачи
2. Сопоставьте переменные:

```yml
# Ваши текущие переменные → Переменные шаблона
HOST_VM_PERF → HOST_VM_PERF (без изменений)
JMETER_application → JMETER_application (без изменений)
```

Шаг 2: Адаптация задач (jobs)

1. Замените extends в ваших задачах:

```yml
# Было:
your-job:
  extends: .your_template

# Стало:
your-job:
  extends:
    - .before
    - .common_job
```

2. Обновите переменные задач (jobs):

```yml
# Добавьте переменную TEST_PLAN_FILE если ее нет:
variables:
  TEST_PLAN_FILE: "jmx/perfTests/your-test.jmx"
```

Шаг 3: Настройка интеграции с Test IT

1. Добавьте переменные Test IT в .common_job:

```yml
variables:
  TESTIT_UPLOAD_URL: your_test_it_url  # Пример адреса Test IT
  TESTIT_PROJECT_ID: your-project-id  # Пример ID проекта (получить можно через Chrome DevTools)
  TESTIT_CONFIGURATION_ID: your-configuration-id  # Пример ID конфигурации (получить можно через Chrome DevTools)
```

2. Настройте правила запуска вебхуков:

```yml
rules:
  - if: '$CI_PIPELINE_SOURCE == "trigger" &&
$JMETER_testName == "your-test-name"'
    when: on_success
  - when: manual
    allow_failure: true
```

Шаг 4: Проверка совместимости

1. Убедитесь, что ваши `.jmx` -файлы соответствуют требованиям:
   - Используют переменные из `common.properties`
   - Генерируют JUnit XML отчеты
   - Настроены для работы с переменными окружения

2. Проверьте структуру папок:

```text
├── jmx/
|   ├── perfTests/     # Основные performance-тесты
|   ├── service/       # Сервисные тесты
|   └── custom/        # Кастомные тесты
├── testData/          # Тестовые данные
└── plugins/           # JMeter-плагины
```

Настройка Test IT

Настройка параметров проекта

- В разделе Параметры проекта Test IT создайте следующие параметры:

#	Параметр	Назначение	Значения
1	runType	Тип запуска тестов — используется для определения типа тестирования	api, regression, smoke, load

2	testName	Название теста (задачи (job)) — определяет название задачи в GitLab CI, которое должно совпадать с именем .jmx-файла. Например, для файла Login_Test.jmx параметр должен быть Login_Test.	Login_Test, Checkout_Process, Search_Functionality
3	application	Название приложения — используется для группировки результатов.	perf, cloud, api
4	host	Хост для тестирования — определяет целевой сервер для нагрузочного тестирования.	URL тестируемого приложения
5	loopCount	Количество итераций — определяет количество повторений теста	Числовые (1, 625, 1500, 5000)
6	userNum	Количество пользователей — определяет количество виртуальных пользователей	Числовые (1, 10, 15, 20, 22, 25)

Настройка конфигураций

- В разделе Конфигурации создайте конфигурации для различных сценариев:

1. **Базовые конфигурации**:
    - Создайте конфигурации (например, "Load Test — Cloud Standard")
    - Укажите описание конфигурации
    - Установите флаг **По умолчанию** для основной конфигурации
2. **Привязка параметров**:
    - Для каждой конфигурации установите специфичные значения параметров
    - Например:

- `application: cloud`
- `host: perf`
- `loopCount: 625`
- `runType: lst`
- `testName: Login_Test`
- `userNum: 4`

3. **Управление конфигурациями**:
   - Используйте различные конфигурации для разных типов тестирования
   - Настройте параметры в соответствии с требованиями каждого сценария
   - Регулярно обновляйте конфигурации при изменении требований к тестированию

Настройка вебхука в Test IT

- Для автоматического запуска тестов из Test IT необходимо настроить вебхук для запуска автотестов, который будет отправлять запросы в GitLab CI/CD.

Параметры вебхука в Test IT

1. **Общие настройки:**
   - Название: `Запуск автотестов`
   - Состояние: `Запущен`
2. **URL и метод:**
   - URL: `https://your-gitlab-instance.com/api/v4/projects/{PROJECT_ID}/trigger/pipeline`
   - Тип запроса: `POST`
   - `{PROJECT_ID}` — ID вашего проекта в GitLab (пример значения по умолчанию)
3. **Заголовки HTTP:**
   - Параметр: `Content-Type`
   - Значение: `application/json`

4. **Тело запроса HTTP:**

   - Заменить системные параметры: ✅
   - Экранировать параметры: ✅
   - Пользовательский контекст:

```json
{
"ref": "main",
"token": "YOUR_GITLAB_TOKEN",
"variables": {
   "TESTIT_PROJECT_ID": "$PROJECT_ID",
   "TESTIT_CONFIGURATION_ID": "$CONFIGURATION_IDS",
   "TESTIT_TEST_RUN_ID": "$TEST_RUN_ID",
   "JMETER_loopCount":
"$CONFIGURATIONS_PARAMETERS[loopCount]",
   "JMETER_usrNum":
"$CONFIGURATIONS_PARAMETERS[usrNum]",
   "JMETER_host": "$CONFIGURATIONS_PARAMETERS[host]",
   "JMETER_application":
"$CONFIGURATIONS_PARAMETERS[application]",
   "JMETER_testName":
"$CONFIGURATIONS_PARAMETERS[testName]",
   "runType": "$CONFIGURATIONS_PARAMETERS[runType]"
 }
}
```

Примечание: Все указанные токены и ID являются примерами значений по умолчанию. Замените их на реальные значения из вашего окружения.

Описание параметров:

- `ref` — ветка репозитория для запуска тестов
- `token` — токен GitLab для управления проектом
- `TESTIT_*` — системные переменные Test IT, формируются автоматически
- `JMETER_*` — параметры JMeter, настраиваются в конфигурациях Test IT

Все ID проекта и конфигурации, приведенные в теле запроса — это примеры значений по умолчанию. Получить реальные данные ID можно через просмотр данных в интерфейсе Test IT или Chrome DevTools при

работе с Test IT.

После настройки вебхуков и параметров, запуск любого автотеста в Test IT будет автоматически передавать данные в GitLab CI/CD, где будет выполняться соответствующий тест JMeter. Результаты тестирования будут автоматически переданы обратно в Test IT.

Условия успешного использования шаблона

Шаблон предоставляет полнофункциональную интеграцию между JMeter, GitLab CI/CD и Test IT. Он автоматизирует выполнение тестов производительности, генерацию отчетов и загрузку результатов в систему управления тестированием.

Для успешного использования:

1. Настройте переменные окружения в GitLab
2. Адаптируйте тест-планы под ваши требования
3. Настройте Test IT CLI для вашего экземпляра
4. Настройте вебхук в Test IT для автоматического запуска тестов
5. Протестируйте интеграцию на тестовом окружении

При возникновении вопросов обращайтесь к документации JMeter, GitLab CI/CD и Test IT, а также к указанным выше ссылкам на документацию компонентов.

Документация

Все возможности шаблона и его параметры описаны в документации:

- **JMeter JUnit Reporter Plugin:** https://github.com/tilln/jmeter-junit-reporter
- **Test IT CLI:** /user-guide/integrations/cli.md
- **Документация Test IT:** docs.testit.software

Обновлено: 8/13/26, 6:19 PM

Работа с тестовыми фреймворками

Test IT позволяет интегрироваться со множеством тестовых фреймворков для загрузки результатов автоматизированных тестов. Вы можете использовать любые способы запуска автоматизированных тестов, такие как CI/CD системы, сервера для сборки, локальные запуски.

Доступные способы интеграции:

- Test IT CLI
- Адаптер (список поддерживаемых фреймворков смотрите в статье)
- API

Ниже приведен список поддерживаемых тестовых фреймворков:

Поддерживаемые тестовые фреймворки

- Appium
- Behave
- BrowserStack
- Catch2
- CodeceptJS
- Cucumber
- Cypress
- Emcee
- Flutter
- Golang
- GoogleTest
- Jasmine
- JBehave
- Jest
- JUnit
- Kotest
- Marathon Cloud
- Minitest
- Mocha
- MSTest
- Nose2
- NUnit
- PHPUnit
- Playwright
- Postman
- pytest
- pytest-BDD
- Robot Framework
- RSpec
- Sauce Labs
- Selenide
- Selenium
- SoapUI
- SpecFlow
- Telerik Test Studio
- TestCafe
- TestComplete

- Testim
- TestNG
- Unittest
- WebdriverIO
- XCTest
- xUnit.net

Обновлено: 8/12/26, 4:43 PM

Адаптер Test IT

Следите за обновлениями адаптера

Подпишитесь на обновления адаптера в GitHub, чтобы получать уведомления о новых версиях.

- **Для этого раздела мы подготовили видео:**

- Что такое адаптер Test IT
- Основные преимущества
- Передаваемые метаданные
- Подключение
- Подписка на обновления версий адаптера и совместимость версий
  - Подписка на обновления адаптера
  - Совместимость версий адаптера и системы Test IT
- Дополнительные ресурсы

Что такое адаптер Test IT

Адаптер — это утилита для переноса автотестов, их результатов и метаданных из тестовых фреймворков в Test IT, устанавливаемая в проект с автотестами. Для каждого фреймворка адаптер разрабатывается индивидуально.

Список поддерживаемых фреймворков

- CodeceptJS
- Cucumber
- Cypress
- Flutter
- Golang
- JBehave
- Jest
- JUnit
- Kotest
- Mocha
- MSTest
- NUnit
- Playwright
- pytest
- Robot Framework
- Selenide
- SpecFlow
- TestCafe
- TestNG
- XCTest
- xUnit.net

Возможно влияние адаптера на автотесты

Адаптер устанавливается в проект с автотестами как отдельный пакет ПО и может оказывать влияние на код автотестов.

Основные преимущества

Адаптер предоставляет широкие возможности сбора информации об автотестах и обладает гибкостью в интеграции. В отличие от CLI, поддерживающего формат JUnit XML и передающего основные параметры автотестов, адаптер позволяет:

- Выводить дополнительные метаданные (внешний идентификатор, теги, ссылки, шаги тестов, вложения и т.д.)
- Использовать фильтры при запуске автотестов
- Применять различные способы конфигурирования (конфигурационный файл, переменная окружения, параметры командной строки)

Также адаптер поддерживает фильтрацию через утилиту Test IT CLI.

Передаваемые метаданные

Адаптер позволяет передавать следующие метаданные:

- `workItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `displayName` — имя автотеста, используемое в Test IT
- `externalId` (не путать с `globalId` или `id` автотеста) — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `description` — описание автотеста, указанное в карточке автотеста
- `labels` — лейблы, указанные в карточке автотеста
- `tags` — теги, указанные в карточке автотеста
- `links` — ссылки, указанные в карточке автотеста
- `adapter.addLinks` — ссылки на результаты автотестов
- `nameSpace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `className` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `adapter.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `adapter.addMessage` — сообщения о результатах автотестов
- `parameter` — параметр, указанный в результате автотеста
- `step` — шаг автотеста

Список передаваемых метаданных может несущественно различаться в зависимости от используемого тестового фреймворка.

Подключение

Инструкции по подключению адаптеров для различных фреймворков и языков программирования доступны на GitHub Test IT .

Мы непрерывно работаем над созданием новых адаптеров для различных тестовых фреймворков. Вы можете предложить интеграцию на официальном сайте Test IT .

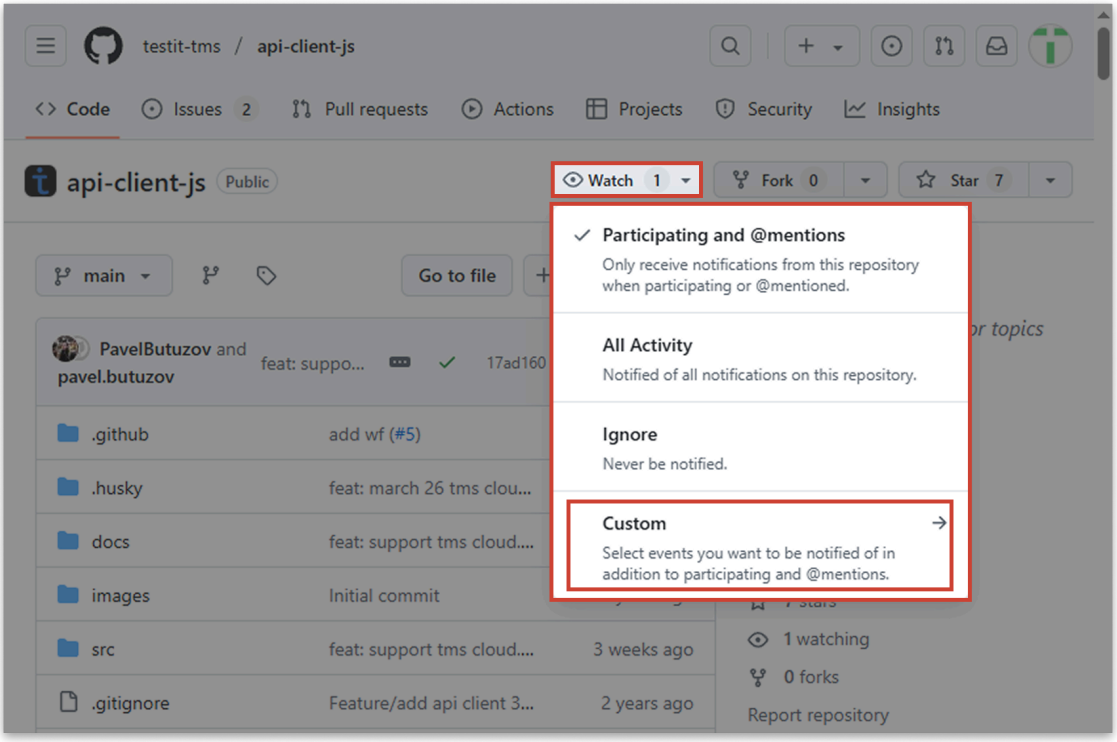
Подписка на обновления версий адаптера и совместимость версий

Чтобы быть в курсе об обновлениях адаптера, вы можете подписаться на его обновления на странице GitHub Test IT . Подписка оформляется для каждого адаптера по отдельности. Также на GitHub представлена информация о совместимости версий адаптера с версиями Test IT.

Подписка на обновления адаптера

Чтобы подписаться на обновления адаптера Test IT:

1. Авторизуйтесь в GitHub.
2. Перейдите в репозиторий нужного адаптера, например https://github.com/testit-tms/adapters-python .
3. В панели в правой верхней части окна нажмите **Watch**, затем в выпадающем меню выберите **Custom**.



4. В открывшемся окне выберите **Releases**, затем нажмите **Apply**.
   Вы будете получать уведомления об обновлении версии выбранного адаптера в GitHub.

Совместимость версий адаптера и системы Test IT

Проверить совместимость версий адаптера и Test IT можно в соответствующих разделах GitHub Test IT:

- .NET
- Flutter
- Go
- Java
- JavaScript
- Kotlin
- Python
- Swift

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для систем IDE
- Взаимодействие с REST API

Обновлено: 8/12/26, 4:43 PM



Appium

Test IT позволяет загружать результаты автоматизированных тестов из Appium. Appium — это утилита для тестирования мобильных приложений, которая поддерживает множество языков программирования: JavaScript/NodeJS, Java, C#, Python, Ruby и другие. Вы можете использовать любой тестовый фреймворк, который поддерживается Appium . Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Необходимо использовать тестовый фреймворк, который поддерживает генерацию отчета в формате JUnit XML. В данной инструкции в качестве тестового фреймворка используется TestNG.

- Для генерации отчета воспользуйтесь командой:

```sh
$ java org.testng.TestNG testng1.xml
```

  После выполнения тестов отчет будет автоматически сгенерирован в директории `test-output` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Appium test run" \
  --results test-output/test/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты успешно загружены в систему управления тестированием Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Behave

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Behave. Существуют следующие способы интеграций:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Подробнее можно прочитать в официальной документации
.

- Для генерации отчета воспользуйтесь командой:

```sh
$ behave --junit
```

После выполнения тестов отчеты будут автоматически сгенерированы в директории `reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit -h

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit \
  --url <YOUR_INSTANCE_URL>\
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Behave test run" \
  --results reports
```

- <YOUR_TOKEN> - Приватный токен, сгенерированный в вашем профиле Test IT,
- <YOUR_INSTANCE_URL> - адрес вашей системы Test IT, к примеру, https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```
console

Результаты загружены в систему Test IT.

Адаптер

Предоставляет более гибкую возможность интеграции. С помощью адаптера можно загружать не только базовую информацию об автотестах, но и другие метаданные, например, шаги, вложения, ссылки и многое другое.

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Behave

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Adapter.addLinks` — ссылки на результаты автотестов
- `Adapter.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.addMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты выполнения с помощью нашего публичного API. Существуют два способа взаимодействия:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



BrowserStack

Test IT позволяет загружать результаты автоматизированных тестов с использованием Selenium и BrowserStack. BrowserStack использует Selenium в качестве библиотеки автоматизации работы с браузерами. Selenium часто используют совместно с JavaScript/NodeJS, Java, C#, Python, Ruby и другими языками. Вы можете использовать любой тестовый фреймворк, который поддерживается Selenium . Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Необходимо использовать тестовый фреймворк, который поддерживает генерацию отчета в формате JUnit XML. В данной инструкции в качестве тестового фреймворка используется TestNG.

- Для генерации отчета воспользуйтесь командой:

```sh
$ java org.testng.TestNG testng1.xml
```

  После выполнения тестов отчет будет автоматически сгенерирован в директории `test-output` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "BrowserStack test run" \
  --results test-output/test/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Catch2

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Catch2. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Catch поддерживает генерацию JUnit XML. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ test --reporter junit --out test-results.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в файле `test-results.xml` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Catch test run" \
  --results test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
```



```
| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

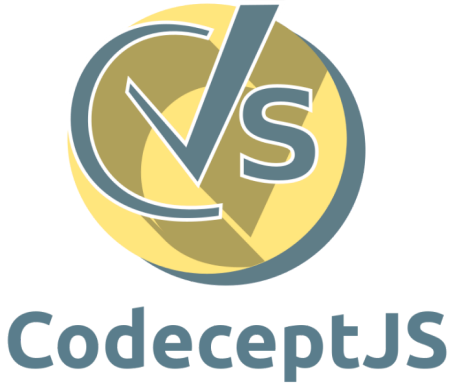
1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



CodeceptJS

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка CodeceptJS. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. CodeceptJS поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов mocha-junit-reporter .

1. Для подключения пакета выполните команду:

```sh
$ npm install mocha-junit-reporter
```

2. Для подключения пакета к проекту добавьте его в `codecept.conf.js` :

```conf
"mocha": {
    "reporterOptions": {
        "mochaFile": "output/result.xml"
    }
},
```

3. Для генерации отчета воспользуйтесь командой:

```sh
$ codeceptjs run --reporter mocha-junit-reporter
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `output` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "CodeceptJS test run" \
  --results output/result.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для CodeceptJS

- `workItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `displayName` — имя автотеста, используемое в Test IT
- `externalId` (не путать с `globalId` или `id` автотеста) — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `description` — описание автотеста, указанное в его карточке
- `labels` — лейблы, указанные в карточке автотеста
- `link` — ссылки, указанные в карточке автотеста
- `namespace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `classname` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `addLinks` — ссылки на результаты автотестов
- `addAttachments` — вложения, прикрепляемые к результатам автотестов
- `addMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Cucumber

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Cucumber. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ cucumber --format junit --out results/test-results.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Cucumber test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 6 result file with a total of 14 tests
Created new testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
Sending test results to Test IT ...
Uploading: 100%|██████████| 14/14 [00:06<00:00, 2.09it/s]
Successfully sent test results
Completed testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Cucumber

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Adapter.addLinks` — ссылки на результаты автотестов
- `Adapter.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.addMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Cypress

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Cypress. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Cypress поддерживает генерацию JUnit XML отчета. Подробнее можно прочитать в официальной документации .

1. Для подключения функции необходимо указать ее в `cypress.config.js` :

```conf
reporter: 'junit',
  reporterOptions: {
    mochaFile: 'results/my-test-output-[hash].xml',
},
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ cypress run
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Cypress test run" \
  --results results
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Cypress

- `tms.addWorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `tms.addDisplayName` — имя автотеста, используемое в Test IT
- `tms.addTitle` — — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName` .
- `tms.addDescription` — описание автотеста, указанное в его карточке
- `tms.addLabels` — лейблы, указанные в карточке автотеста
- `tms.addtags` — теги, указанные в карточке автотеста
- `tms.addLinks` — ссылки на результаты автотестов
- `tms.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `tms.addMessage` — сообщения о результатах автотестов
- `tms.addNameSpace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `tms.addClassName` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `tms.addParameter` — параметр, указанный в результате автотеста
- `tms.step` — шаг автотеста

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Emcee

Test IT позволяет загружать результаты автоматизированных тестов из Emcee. Emcee — это утилита для тестирования мобильных приложений, которая поддерживает параллельный запуск тестов. Вы можете использовать любой тестовый фреймворк, который поддерживается Emcee . Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Emcee поддерживает генерацию JUnit XML отчета. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
./emceeFree runTests \
 --queue "ssh://emcee:qwerty@ios-build-machine77" \
 --worker "ssh://emcee:qwerty@ios-build-machine77" \
 --worker "ssh://emcee:qwerty@ios-build-machine78" \
 --device "com.apple.CoreSimulator.SimDeviceType.iPhone-14" \
 --runtime "com.apple.CoreSimulator.SimRuntime.iOS-16-4" \
 --test-bundle derivedData/Build/Products/Debug-iphonesimulator/EmceeSampleTestsWithoutHost.xctest \
 --junit test-output/tests_without_host_junit.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `test-output` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Emcee test run" \
  --results test-output/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты успешно загружены в систему управления тестированием Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Flutter

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Flutter. Загрузка осуществляется с помощью Адаптера Test IT.

Смотрите также: Дополнительные ресурсы.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Flutter

- `description` — описание автотеста, указанное в его карточке
- `externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `links` — ссылки, указанные в карточке автотеста
- `tags` — лейблы, указанные в карточке автотеста
- `workItemsIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `addAttachment` — одно вложение, прикрепляемое к результатам автотестов
- `addAttachments` — все вложения, прикрепляемые к результатам автотестов
- `addLink` — одна ссылка на результат автотестов
- `addLinks` — все ссылки на результаты автотестов
- `addMessage` — сообщения о результатах автотестов
- `step` — обозначение шага автотеста в коде

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Golang

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Golang. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Golang поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов gotest.tools/gotestsum .

1. Для подключения пакета выполните команду:

```sh
$ go install gotest.tools/gotestsum@latest
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ gotestsum --junitfile results/unit-tests.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `reports` .

Загрузка результатов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Golang test run" \
  --results results/unit-tests.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Golang

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Step` — обозначение шага автотеста в коде
- `tms.AddLinks` — ссылки на результаты автотестов
- `tms.AddAttachments` — вложения, прикрепляемые к результатам автотестов
- `tms.AddAttachmentsFromString` — вложения из строки результата автотеста
- `tms.AddMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



GoogleTest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка GoogleTest. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Подробнее можно прочитать в официальной документации
.

- Для генерации отчета воспользуйтесь командой:

```sh
$ test --gtest_output="xml:results/test-results.xml"
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "GoogleTest test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 6 result file with a total of 14 tests
Created new testrun (ID: 4eeead72-b88a-4845-b258-
3d077065561b)
Sending test results to Test IT ...
Uploading: 100%|██████████| 14/14 [00:06<00:00,
2.09it/s]
Successfully sent test results
Completed testrun (ID: 4eeead72-b88a-4845-b258-
3d077065561b)
```
console

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Jasmine

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Jasmine. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Jasmine поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных

пакетов jasmine-reporters .

1. Для подключения пакета выполните команду:

```sh
$ npm i jasmine-reporters
```

2. Для подключения пакета к проекту зарегистрируйте его в Jasmine:

```js
var path = require('path');
var reporters = require('jasmine-reporters');
var junitReporter = new reporters.JUnitXmlReporter({
    savePath: path.join(__dirname, '/results'),
    consolidateAll: false
});
jasmine.getEnv().addReporter(junitReporter)
```

3. Для генерации отчета воспользуйтесь командой:

```sh
$ npx jasmine
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Jasmine test run" \
  --results results
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



JBehave

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка JBehave. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы.

JBehave не поддерживает генерацию JUnit XML отчета. Вы можете воспользоваться нашим пакетом для генерации jbehave-reporter-junitxml .

1. Для подключения пакета добавьте его в зависимости проекта:
   - Maven:

```xml
<dependency>
    <groupId>ru.testit</groupId>
    <artifactId>jbehave-reporter-junitxml</artifactId>
    <version>LATEST_VERSION</version>
    <scope>compile</scope>
</dependency>
```

   - Gradle:

```groovy
implementation "ru.testit:jbehave-reporter-junitxml:LATEST_VERSION"
```

2. Добавьте его в конфигурацию JBehave:

```java
import org.jbehave.core.ConfigurableEmbedder;
import org.jbehave.core.configuration.Configuration;
import org.jbehave.core.configuration.MostUsefulConfigurati
import org.jbehave.core.embedder.Embedder;
import org.jbehave.core.embedder.EmbedderControls;
import org.jbehave.core.embedder.NullEmbedderMonitor;
import org.jbehave.core.io.LoadFromClasspath;
import org.jbehave.core.reporters.NullStoryReporter;
import org.jbehave.core.reporters.StoryReporterBuilder;
import org.jbehave.core.steps.InjectableStepsFactory;
import org.jbehave.core.steps.InstanceStepsFactory;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.io.TempDir;
import ru.testit.reporter.JunitXmlReporter;

import java.io.File;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.List;

public class TestStoriesRunner extends ConfigurableEmbedder
    public Embedder embedder;
    @TempDir
    Path temp;

    @Override
    @Test
    public void run() {
        embedder = new Embedder();
        embedder.useEmbedderMonitor(new NullEmbedderMonitor
        embedder.useEmbedderControls(new EmbedderControls()
                .doGenerateViewAfterStories(false)
                .doFailOnStoryTimeout(false)
                .doBatch(false)
                .doIgnoreFailureInStories(true)
                .doIgnoreFailureInView(true)
                .doVerboseFailures(false)
                .doVerboseFiltering(false)
        );
        embedder.useConfiguration(configuration());

embedder.useCandidateSteps(stepsFactory().createCandidateSt
```

```
43
44          File dir = new File("./src/test/resources/stories")
45          List<String> stories = new ArrayList<>();
46
47          for (File file : dir.listFiles()){
48              if ( file.isFile() )
49                  stories.add("stories/" + file.getName());
50          }
51
52          embedder.runStoriesAsPaths(stories);
53      }
54
55      public Configuration configuration() {
56          final JunitXmlReporter reporter = new JunitXmlRepor
57
58          return new MostUsefulConfiguration()
59                  .useStoryLoader(new
60  LoadFromClasspath(this.getClass()))
61                  .useStoryReporterBuilder(
62                          new
63  TestStoryReporterBuilder(temp.toFile())
64                                  .withReporters(reporter))
65                  .useDefaultStoryReporter(new NullStoryRepor
66      }
67
68      public InjectableStepsFactory stepsFactory() {
69          return new InstanceStepsFactory(embedder.configurat
70  new SampleSteps());
71      }
72
73      static class TestStoryReporterBuilder extends
74  StoryReporterBuilder {
75
76          private final File outputDirectory;
77
78          TestStoryReporterBuilder(final File outputDirectory
79              this.outputDirectory = outputDirectory;
80          }
81
82          @Override
            public File outputDirectory() {
```

```
            return outputDirectory;
        }
    }
}
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `output` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL>\
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "JBehave test run" \
  --results output/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, к примеру, https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```
console

Результаты успешно загружены в систему Test IT. Вы также можете воспользоваться адаптером.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для JBehave

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `NameSpace` — директория в системе управления тестированием (по умолчанию используется имя файла теста)
- `ClassName` — субдиректория в системе управления тестированием (по умолчанию используется имя класса теста)
- `testit.addLinks` — ссылки на результаты автотестов
- `testit.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `testit.addMessage` — сообщения о результатах автотестов
- `testit.step` — использование в конструкции *with* для обозначения шага в теле теста

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API

2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Jest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Jest. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Jest поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов jest-junit

.

1. Для подключения пакета добавьте его в `jest.config.js` :

```js
module.exports = {
  reporters: [
    "default",
    ["jest-junit", {
      outputFile: "results/test-results.xml",
      ancestorSeparator: " > ",
    }]
  ],
};
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ jest
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Jest test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией в GitHub.

Список метаданных, передаваемых адаптером для Jest

- `testit.workItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `testit.displayName` — имя автотеста, используемое в Test IT
- `testit.externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `testit.title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `testit.description` — описание автотеста, указанное в его карточке
- `testit.labels` — лейблы, указанные в карточке автотеста
- `testit.link` — ссылки, указанные в карточке автотеста
- `testit.namespace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `testit.classname` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `testit.addLinks` — ссылки на результаты автотестов
- `testit.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `testit.addMessage` — сообщения о результатах автотестов
- `testit.step` — шаг автотеста

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API

2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



JUnit

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка JUnit. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. JUnit поддерживает несколько утилит для сборки. Руководство по настройке используемой вами утилиты смотрите в официальной документации:

- Gradle
- Maven
- Ant
- Kotlin
- Android

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Junit test run" \
  --results test-output/test/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 6 result file with a total of 14 tests
Created new testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
Sending test results to Test IT ...
Uploading: 100%|██████████| 14/14 [00:06<00:00, 2.09it/s]
Successfully sent test results
Completed testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для JUnit

- `RunWith(BaseJunit4Runner.class)` — соединение адаптера, необходимое для запуска автотестов (является обязательным)
- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Step` — шаг автотеста
- `ClassName` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `NameSpace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `Adapter.addLinks` — ссылки на результаты автотестов
- `Adapter.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.addMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Kotest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Kotest. Загрузка осуществляется с помощью Адаптера Test IT.

Смотрите также: Дополнительные ресурсы.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.). Пример проекта с тестами .

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Kotest

- `externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `links` — ссылки, указанные в карточке автотеста
- `workItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `attachments` — список вложений автотестов
- `name` — имя автотеста, используемое в Test IT
- `title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `message` — сообщение автотеста
- `itemStatus` — свойство автотеста, относящееся к статусу объекта
- `description` — описание в карточке автотеста
- `labels` — лейблы, указанные в карточке автотеста

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Marathon Cloud

Test IT позволяет загружать результаты автоматизированных тестов из Marathon Cloud — облачного сервиса для запуска тестов мобильных приложений на базах Android и iOS.
Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Один из артефактов, оставляемых Marathon Cloud при прогоне автотестов — отчет JUnit XML.

- Для генерации отчета воспользуйтесь командой:

```sh
marathon-cloud run android
 --application=app.apk
 --test-application=appTest.apk
 --output=report
```

Отчет `marathon_junit_report.xml` будет создан в директории `report/tests/omni/` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Marathon Cloud test run" \
  --results report/tests/omni/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты успешно загружены в систему управления тестированием Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API
- Пользовательская документация Marathon Cloud



Minitest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Minitest. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Minitest поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных

пакетов minitest-reporters/minitest-reporters .

1. Для подключения пакета выполните команду:

```sh
$ gem install minitest-reporters
```

2. Добавьте в `test_helper.rb` следующие строки:

```rb
require "minitest/reporters"
Minitest::Reporters.use!
Minitest::Reporters::JUnitReporter.new
```

3. Для генерации отчета воспользуйтесь командой:

```sh
$ rake test
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `test/reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Minitest test run" \
  --results test/reports/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Mocha

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Mocha. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Mocha поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов mocha-junit-reporter .

1. Для подключения пакета добавьте его `mocharc.js` :

```conf
"use strict";

module.exports = {
  reporter: 'mocha-junit-reporter',
  reporterOptions: [
    "mochaFile=results/test-results.xml",
    "jenkinsMode=1",
    "testCaseSwitchClassnameAndName=1",
    "outputs=1"
  ]
}
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ npx mocha
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `reports` .

Загрузка результатов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Mocha test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Mocha

- `this.workItemsIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `this.displayName` — имя автотеста, используемое в Test IT
- `this.externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `this.title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `this.description` — описание автотеста, указанное в его карточке
- `this.labels` — лейблы, указанные в карточке автотеста
- `this.links` — ссылки, указанные в карточке автотеста
- `this.nameSpace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `this.className` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `this.addLinks` — ссылки на результаты автотестов
- `this.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `this.addMessage` — сообщения о результатах автотестов
- `this.addSteps` — информация о шаге автотеста

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



MSTest

MSTest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка MSTest. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. MSTest поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов spekt/junit.testlogger .

1. Для подключения пакета выполните команду:

```sh
$ dotnet add package JunitXml.TestLogger
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ dotnet test --logger:"junit;LogFilePath=results/test-results.xml"
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "MSTest test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для MSTest

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Step` — шаг автотеста
- `Adapter.AddLinks` — ссылки на результаты автотестов
- `Adapter.AddAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.AddMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Nose2

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Nose2. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Nose2 поддерживает генерацию JUnit XML отчета с помощью встроенного плагина. Подробнее можно прочитать в

официальной документации Nose2 .

1. Для подключения плагина добавьте его в `nose2.cfg` :

```conf
[unittest]
plugins = nose2.plugins.junitxml
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ nose2 --junit-xml
```

После выполнения тестов отчет будет автоматически сгенерирован в файл `nose2-junit.xml` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Nose2 test run" \
  --results nose2-junit.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



NUnit

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка NUnit.net. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. NUnit поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов spekt/junit.testlogger .

1. Для подключения пакета выполните команду:

```sh
$ dotnet add package JunitXml.TestLogger
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ dotnet test --logger:"junit;LogFilePath=results/test-results.xml"
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "NUnit test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для NUnit

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Step` — шаг автотеста
- `Adapter.AddLinks` — ссылки на результаты автотестов
- `Adapter.AddAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.AddMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



PHPUnit

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка PHPUnit. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Подробнее можно прочитать в официальной документации
.

- Для генерации отчета воспользуйтесь командой:

```sh
$ phpunit --log-junit results/test-results.xml
```

  После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "PHPunit test run" \
  --results results/test-results.xml
```

   - <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
   - <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ Пример выполнения команды

```
Collecting log files ...
Found 6 result file with a total of 14 tests
Created new testrun (ID: 4eeead72-b88a-4845-b258-
3d077065561b)
Sending test results to Test IT ...
Uploading: 100%|██████████| 14/14 [00:06<00:00,
2.09it/s]
Successfully sent test results
Completed testrun (ID: 4eeead72-b88a-4845-b258-
3d077065561b)
```
console

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Playwright

- **Для этого раздела мы подготовили видео:**

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Playwright. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Playwright поддерживает генерацию JUnit XML отчета. Подробнее можно прочитать в официальной документации .

1. Для подключения функции добавьте ее в `playwright.config.js` :

```conf
"reporter: [['junit', {
    outputFile: 'results/test-results.xml',
    embedAnnotationsAsProperties: true,
  }]],
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ npx playwright test
```

   После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-name "Playwright test run" \
  --results results
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Playwright

- `testit.workItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `testit.displayName` — имя автотеста, используемое в Test IT (может быть заменено строкой документации)
- `testit.externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `testit.title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `testit.description` — описание автотеста, указанное в его карточке
- `testit.labels` — лейблы, указанные в карточке автотеста
- `testit.links` — ссылки, указанные в карточке автотеста
- `testit.namespace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `testit.classname` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `testit.addLinks` — ссылки на результаты автотестов
- `testit.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `testit.addMessage` — сообщения о результатах автотестов
- `testit.step` — информация о шаге автотеста, указанная в его результате

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Postman

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Postman. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Postman поддерживает генерацию JUnit XML отчета при помощи исполнителя командной строки Newman. Подробнее можно

прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ newman run collection.json -r cli,junit --reporter-junit-export results/test-results.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results`.

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-name "Postman test run" \
  --results results
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



pytest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка pytest. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. pytest поддерживает генерацию JUnit XML отчета. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ pytest --junit-xml=reports/results.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Pytest test run" \
  --results reports
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для pytest

- `testit.workItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `testit.displayName` — имя автотеста, используемое в Test IT
- `testit.externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `testit.title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `testit.description` — описание автотеста, указанное в его карточке
- `testit.labels` — лейблы, указанные в карточке автотеста
- `testit.link` — ссылки, указанные в карточке автотеста
- `testit.step` — шаг теста, вызываемый в теле теста или другого шага
- `testit.nameSpace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `testit.className` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `testit.addLinks` — ссылки на результаты автотестов
- `testit.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `testit.addMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



pytest-BDD

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка pytest-BDD. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. pytest-BDD поддерживает генерацию JUnit XML отчета. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ pytest --junit-xml=reports/results.xml
```

  После выполнения тестов отчет будет автоматически сгенерирован в директории `reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Pytest-BDD test run" \
  --results reports
```

   - <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
   - <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

   ▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
```



```
| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Robot Framework

Robot Framework

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Robot Framework. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Robot Framework поддерживает генерацию JUnit XML отчета. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ robot --xunit reports/test-results.xml examples
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL>\
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Robot Framework test run" \
  --results reports
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Robot Framework

- `testit.workItemsId` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `testit.displayName` — имя автотеста, используемое в Test IT
- `testit.externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `testit.title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `testit.description` — описание автотеста, указанное в его карточке
- `testit.links` — ссылки, указанные в карточке автотеста
- `testit.labels` — лейблы, указанные в карточке автотеста
- `testit.nameSpace` — расположение автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `testit.className` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `Add Links` — ссылки на результаты автотестов
- `Add Link` — добавляет одну ссылку к результату автотеста
- `Add Attachments` — вложения, прикрепляемые к результатам автотестов
- `Add Attachment` — upload given content with given filename in the autotest result
- `Add Message` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



RSpec

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка RSpec. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Rspec поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных

пакетов sj26/rspec_junit_formatter .

1. Для подключения пакета выполните команду:

```sh
$ gem install rspec_junit_formatter
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ rspec --format RspecJunitFormatter --out results/test-results.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `reports`.

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-name "Rspec test run" \
  --results results/unit-tests.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Sauce Labs

Test IT позволяет загружать результаты автоматизированных тестов с использованием Selenium и Sauce Labs. Sauce Labs использует Selenium в качестве библиотеки автоматизации работы с браузерами. Selenium часто используют совместно с JavaScript/NodeJS, Java, C#, Python, Ruby и другими языками. Вы можете использовать любой тестовый фреймворк, который поддерживается Selenium . Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Необходимо использовать тестовый фреймворк, который поддерживает генерацию отчета в формате JUnit XML. В данной инструкции в качестве тестового фреймворка используется Mocha.

1. Для подключения пакета выполните команду:

```sh
$ npm install mocha-junit-reporter
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ npx mocha --reporter node_modules/mocha-junit-reporter --reporter-options mochaFile=results/mocha-test-results.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results`.

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Sauce Labs test run" \
  --results results
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Selenide

Test IT позволяет загружать результаты автоматизированных тестов с использованием Selenide. Вы можете использовать любой тестовый фреймворк, который поддерживается Selenide . Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Необходимо использовать тестовый фреймворк, который поддерживает генерацию отчета в формате JUnit XML. В данной инструкции в качестве тестового фреймворка используется TestNG.

- Для генерации отчета воспользуйтесь командой:

```sh
$ java org.testng.TestNG testng1.xml
```

  После выполнения тестов отчет будет автоматически сгенерирован в директории `test-output` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Selenide test run" \
  --results test-output/test/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для Selenide

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Step` — шаг автотеста
- `Adapter.addLinks` — ссылки на результаты автотестов
- `Adapter.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.addMessage` — сообщения о результатах автотестов
- `saveScreenshots(true)` — сохранение скриншотов, содержащих ошибки автотестов
- `savePageSource(true)` — сохраняет источник страницы проваленного автотеста
- `saveLogs(LogType.BROWSER, Level.All)` — сохраняет логи проваленных автотестов
- `includeSelenideSteps(true)` — шаг автотеста

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Selenium

Test IT позволяет загружать результаты автоматизированных тестов с использованием Selenium. Selenium часто используют совместно с JavaScript/NodeJS, Java, C#, Python, Ruby и другими языками. Вы можете использовать любой тестовый фреймворк, который поддерживается Selenium . Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Необходимо использовать тестовый фреймворк, который поддерживает генерацию отчета в формате JUnit XML. В данной инструкции в качестве тестового фреймворка используется Mocha.

1. Для подключения пакета выполните команду:

```sh
$ npm install mocha-junit-reporter
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ npx mocha --reporter node_modules/mocha-junit-reporter --reporter-options mochaFile=results/mocha-test-results.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results`.

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Selenium test run" \
  --results results
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



SoapUI

Test IT позволяет загружать результаты автоматизированных тестов с использованием SoapUI. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

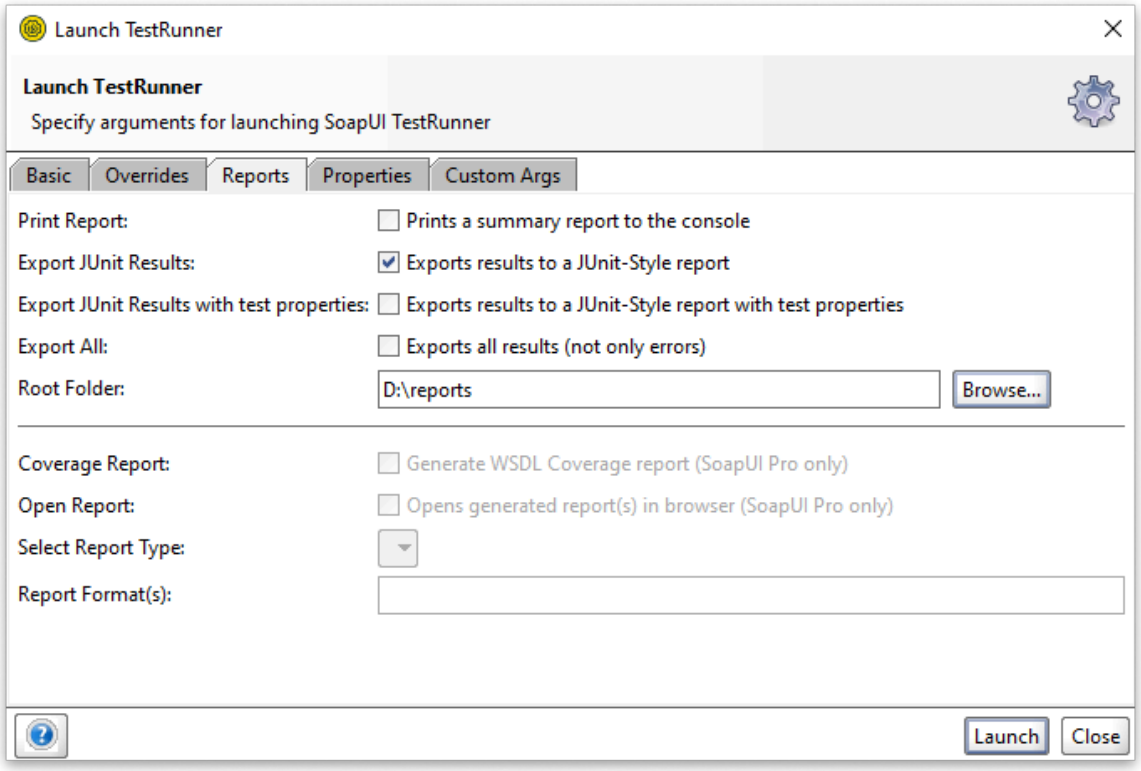
Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

SoapUI поддерживает генерацию отчета JUnit XML. Подробнее можно прочитать в официальной документации .

- Для генерации отчета установите следующие настройки:



После выполнения тестов отчет будет автоматически сгенерирован в директории `reports`.

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "SoapUI test run" \
  --results reports
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



SpecFlow

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка SpecFlow. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. SpecFlow поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов spekt/junit.testlogger .

1. Для подключения пакета выполните команду:

```sh
$ dotnet add package JunitXml.TestLogger
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ dotnet test --logger:"junit;LogFilePath=results/test-results.xml"
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "SpecFlow test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для SpecFlow

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Adapter.AddLinks` — ссылки на результаты автотестов
- `Adapter.AddAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.AddMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Telerik Test Studio

Test IT позволяет загружать результаты автоматизированных тестов с использованием Telerik Test Studio. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Telerik Test Studio поддерживает генерацию JUnit XML отчета при помощи исполнителя командной строки. Подробнее можно

прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ Telerik.ApiTesting.Runner test <YOUR_PROJECT_PATH> -f junit
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `<YOUR_PROJECT_PATH>/.testresults` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-name "Telerik test run" \
  --results <YOUR_PROJECT_PATH>/.testresults
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



TestCafe

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка TestCafe. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. TestCafe поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов testcafe-reporter-junit .

1. Для подключения пакета выполните команду:

```sh
$ npm install testcafe-reporter-junit
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ npx testcafe chrome tests --reporter
junit:reports/report.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "TestCafe test run" \
  --results reports
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для TestCafe

- `workItemsIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `displayName` — имя автотеста, используемое в Test IT
- `externalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `description` — описание автотеста, указанное в его карточке
- `labels` — лейблы, указанные в карточке автотеста
- `links` — ссылки, указанные в карточке автотеста
- `nameSpace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `className` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `adapter.addLinks` — ссылки на результаты автотестов
- `adapter.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `adapter.addMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



TestComplete

Test IT позволяет загружать результаты автоматизированных тестов с использованием TestComplete. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. TestComplete поддерживает генерацию JUnit XML. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ TestComplete.exe <YOUR_PROJECT_PATH> /run /p:
<YOUR_PROJECT_NAME>
/ExportSummary:"C:\TestLogs\report.xml"
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `C:\TestLogs` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "TestComplete test run" \
  --results "C:\TestLogs"
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
```



```
| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Testim

Test IT позволяет загружать результаты автоматизированных тестов с использованием Testim. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Testim поддерживает генерацию JUnit XML отчета при помощи исполнителя командной строки. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ testim --token <YOUR_ACCESS_TOKEN> --project <YOUR_PROJECT_ID> --grid <Your_grid_name> --report-file reports/testim-tests-report.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-name "Testim test run" \
  --results reports
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
```



```
| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



TestNG

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка TestNG. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Подробнее можно прочитать в официальной документации .

- Для генерации отчета воспользуйтесь командой:

```sh
$ java org.testng.TestNG testng1.xml
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `test-output` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
  --testrun-name "TestNG test run" \
  --results test-output/test/
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```
Collecting log files ...
Found 6 result file with a total of 14 tests
Created new testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
Sending test results to Test IT ...
Uploading: 100%|██████████| 14/14 [00:06<00:00,  2.09it/s]
Successfully sent test results
Completed testrun (ID: 4eeead72-b88a-4845-b258-3d077065561b)
```
console

Результаты загружены в систему Test IT.

Адаптер

- **Для этого раздела мы подготовили видео:**

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для TestNG

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Step` — шаг автотеста
- `Namespace` — расположение автотеста в секции Test IT (по умолчанию используется имя директории автотеста)
- `Classname` — субдиректория автотеста в секции Test IT (по умолчанию используется имя файла автотеста)
- `Adapter.addLinks` — ссылки на результаты автотестов
- `Adapter.addAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.addMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

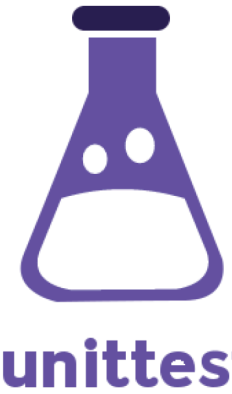
1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



Unittest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Unittest. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. Unittest поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных

пакетов unittest-xml-reporting .

1. Для подключения пакета выполните команду:

```sh
$ pip install unittest-xml-reporting
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ python -m xmlrunner discover -t ~/mycode/tests -o
/junit-reports
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `junit-reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "Unittest test run" \
  --results junit-reports/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
| 7/7 [00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

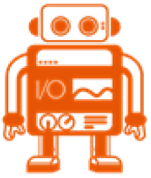
1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



WebdriverIO

Test IT позволяет загружать результаты автоматизированных тестов с использованием WebdriverIO. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. WebdriverIO поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь официальным пакетом

@wdio/junit-reporter .

1. Для подключения пакета выполните команду:

```sh
$ npm install @wdio/junit-reporter --save-dev
```

2. Для подключения пакета к проекту добавьте его в `wdio.conf.js`:

```conf
module.exports = {
    // ...
    reporters: [
        'dot',
        ['junit', {
            outputDir: './results/',
            outputFileFormat: function(options) {
                return 'test-results.xml';
            }
        }]
    ],
    // ...
};
```

3. Для генерации отчета воспользуйтесь командой:

```sh
$ npx wdio run wdio.conf.js
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results`.

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL>\
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "WebdriverIO test run" \
  --results results
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



XCTest

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка XCTest. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. XCTest поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов tuist/xcbeautify .

1. Для подключения пакета выполните команду:

```sh
$ brew install xcbeautify
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ set -o pipefail && xcodebuild [flags] test |
xcbeautify --report junit
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `build/reports` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "XCTest test run" \
  --results build/reports/junit.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для xUnit

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Classname` — имя класса
- `Namespace` — имя поставки
- `AddLinks` — добавление ссылок к результатам автотестов
- `Attachments` — вложения, прикрепляемые к результатам автотестов
- `Message` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM



xUnit.net

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка xUnit.net. Существуют следующие способы интеграции:

- CLI
- Адаптер
- Public API

Смотрите также: Дополнительные ресурсы.

CLI

Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

1. Запустить тесты и сгенерировать отчет JUnit XML
2. Загрузить результаты тестов в Test IT

Запуск тестов и генерация отчета JUnit XML

Вы можете запускать тесты через командную строку или как часть вашей CI/CD системы. xUnit.net поддерживает генерацию JUnit XML отчета с помощью сторонних пакетов. Воспользуйтесь одним из популярных пакетов spekt/junit.testlogger .

1. Для подключения пакета выполните команду:

```sh
$ dotnet add package JunitXml.TestLogger
```

2. Для генерации отчета воспользуйтесь командой:

```sh
$ dotnet test --logger:"junit;LogFilePath=results/test-results.xml"
```

После выполнения тестов отчет будет автоматически сгенерирован в директории `results` .

Загрузка результатов тестов в Test IT

Загрузка осуществляется при помощи Test IT CLI. Утилита распространяется как пакет для Python и устанавливается в системе.

1. Для установки выполните команду:

```sh
$ pip install testit-cli
$ testit --help

usage: testit [options]
[...]
```

2. Добавьте переменную окружения **TMS_TOKEN** и выполните команду:

```sh
$ export TMS_TOKEN=<YOUR_TOKEN>
$ testit results import \
  --url <YOUR_INSTANCE_URL> \
  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
  --configuration-id 15dbb164-c1aa-4cbf-830c-
8c01ae14f4fb \
  --testrun-name "xUnit test run" \
  --results results/test-results.xml
```

- <YOUR_TOKEN> — приватный токен, сгенерированный в вашем профиле Test IT
- <YOUR_INSTANCE_URL> — адрес вашей системы Test IT, например https://tms.testit.software/

▼ **Пример выполнения команды**

```console
Collecting log files ...
Found 3 result file with a total of 7 tests
Created new testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
Sending test results to Test IT ...
Uploading: 100%|
|| 7/7
[00:01<00:00,  3.98it/s]
Successfully sent test results
Completed testrun (ID: 18725f28-f054-40d6-b30e-
b05cf36deaf9)
```

Результаты загружены в систему Test IT.

Адаптер

Адаптер предоставляет гибкие возможности интеграции и позволяет загружать не только базовую информацию об автотестах, но и другие метаданные (шаги, вложения, ссылки и т.д.).

Для подключения адаптера воспользуйтесь инструкцией на GitHub.

Список метаданных, передаваемых адаптером для xUnit

- `WorkItemIds` — идентификатор (ID) рабочего элемента, позволяет привязывать автотесты к ручным тестам в Test IT. Получает массив ID ручных тестов.
- `DisplayName` — имя автотеста, используемое в Test IT
- `ExternalId` — уникальный внешний идентификатор автотеста в Test IT, связывающий автотест с его результатами. Если не указать `externalId`, он будет сгенерирован автоматически на основе *Fully Qualified Name* тестового метода, и если автотеста с таким *externalId* нет в проекте, для его результата будет создана новая карточка автотеста.
- `Title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `DisplayName`.
- `Description` — описание автотеста, указанное в его карточке
- `Labels` — лейблы, указанные в карточке автотеста
- `Links` — ссылки, указанные в карточке автотеста
- `Step` — шаг автотеста
- `Adapter.AddLinks` — ссылки на результаты автотестов
- `Adapter.AddAttachments` — вложения, прикрепляемые к результатам автотестов
- `Adapter.AddMessage` — сообщения о результатах автотестов

На нашей странице GitHub вы можете подписаться на обновления адаптера и узнать о совместимости версий адаптера и системы Test IT.

Public API

Вы можете загружать тесты и результаты их выполнения с помощью нашего публичного API. Существует два способа загрузки:

1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT с другими системами, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Плагин Test IT Менеджмент для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 8/17/26, 5:27 PM

Загрузка результатов автотестов из Vanessa Automation

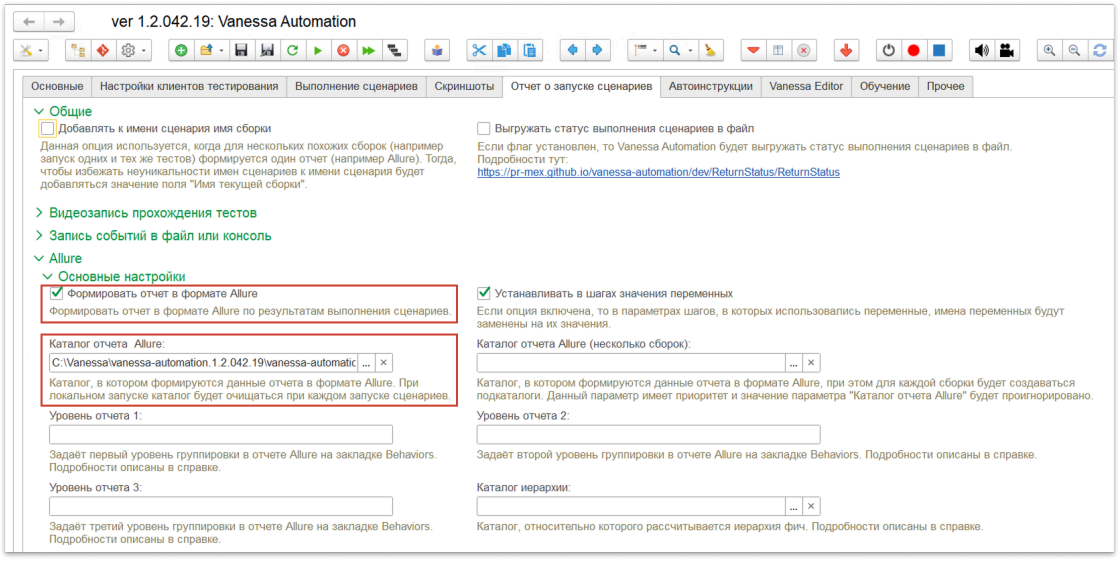
Вы можете загружать результаты автотестов в Test IT из Vanessa Automation — инструмента тестирования прикладных решений на базе 1С:Предприятие. Результаты загружаются в форме отчетов Allure. Настройка интеграции включает:

- Настройку создания отчетов Allure
- Установку и настройку Python
- Установку и настройку импортера Allure

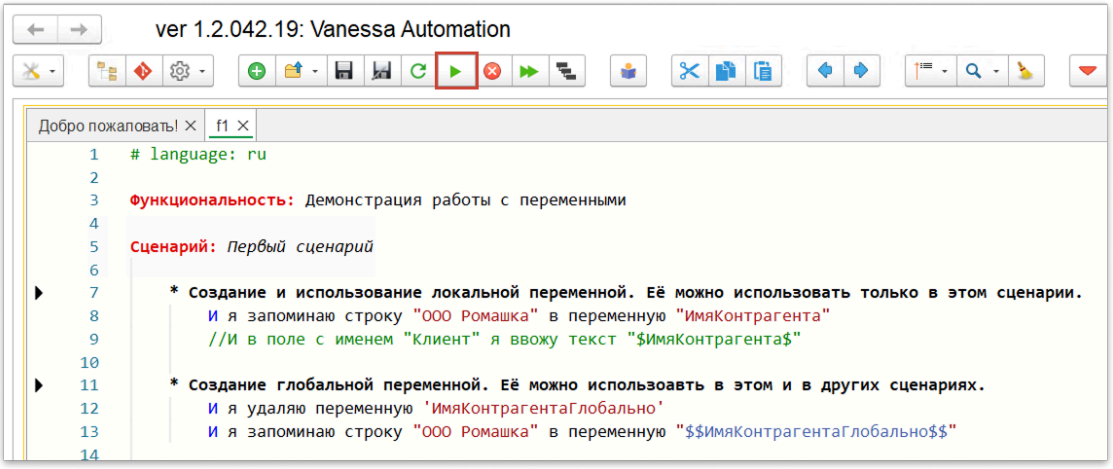
Настройка создания отчетов Allure

Настройка создания отчетов Allure в формате `.json` производится в Vanessa Automation.

1. В Vanessa Automation откройте директорию: **Сервис → Отчет о запуске сценариев → Основные настройки**.
2. Выберите **Формировать отчет в формате Allure**.
3. В поле **Каталог отчета Allure** укажите директорию в которой будут формироваться отчеты о запусках автотестов.



4. Проверьте корректность настройки, запустив пробный автотест из Vanessa Automation. Для этого нажмите кнопку запуска в верхней панели или клавишу `F5`.



В указанной директории будут созданы файлы отчетов.

Установка и настройка Python

Установка Python производится на компьютер с развернутой Vanessa Automation для получения отчетов Allure. В ином случае придется копировать отчеты на другой компьютер с установленным Python.

1. Скачайте Python с официального сайта Python .
2. Установите Python. При установке рекомендуется выбрать опции:
   - **Standalone installer**
   - **Add Python to PATH** (добавление Python в PATH)
3. Убедитесь, что Python установлен успешно:
   - Откройте Powershell
   - Убедитесь, что доступен менеджер пакетов pip

Установка и настройка импортера Allure

> **Выберите совестимую версию импортера**
>
> Выберите версию импортера, совместимую с вашей версией Test IT. Информация о совместимости версий доступна на странице GitHub Test IT:
>
> - Для **Test IT Про** смотрите номер версии.
> - Для тарифов **Базовый** и **Стандарт** используйте версию **Cloud**.

Импортер — утилита Test IT для переноса отчетов Allure в Test IT.

1. Скачайте импортер со страницы GitHub Test IT: https://github.com/testit-tms/importers .
2. Установите импортер с помощью pip. Используйте команду вида:

```sh
pip install testit-importer-allure==1.14.4
```

3. Настройте конфигурацию для загрузки документов:

```sh
testit --url <url>
testit --privatetoken <token>
testit --projectid <id>
testit --configurationid <id>
```

4. **Опционально:** Для доступа к полному списку параметров используйте команду:

```sh
testit --help
```

Отправка отчетов Allure в Test IT

В качестве новых результатов тестов в Test IT отправляются отчеты Allure. Вы можете отправлять отчеты в существующий прогон или создать новый. Чтобы отправить отчет Allure в Test IT:

1. Используйте команду:

```sh
testit --resultsdir %Путь_к_папке_с_отчетами%
```

2. **Опционально:** Укажите относительный путь, открыв pip в папке более высокого уровня чем папка с отчетами.
3. **Опционально:**
   - Укажите пользовательское имя прогона, используя параметр `--testrunname <name>`
   - Загрузите результаты тестов в существующий прогон, используя параметр `--testrunid <id>` В Test IT в разделе **Прогоны** появится новый прогон, содержащий тестовые сценарии (если тесты запускались без параметра `--testrunid`).

Устранение ошибки сертификата

При возникновении ошибки сертификата:

```sh
    certificate verify failed: self-signed certificate in
certificate chain (_ssl.c:1077)')))
```

- Установите пакет pip-system-certs, используя команду:

```sh
pip install pip-system-certs
```

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Плагин Test IT Менеджмент для систем IDE
- Примеры интеграции с тестовыми фреймворками
- Взаимодействие с REST API

Обновлено: 4/22/26, 2:57 PM

Что такое Test IT Менеджмент?

> **Пользователям Test IT Про: требуется версия 5.0 или более поздняя**
>
> Плагин **Test IT Менеджмент 1.0.9** совместим с версией **Test IT 5.0** и выше. Актуальная версия Test IT Про доступна на официальном сайте .

Плагин Test IT Менеджмент — это утилита для интеграции ручных тестов (рабочих элементов Test IT) в код среды разработки.

Плагин Test IT Менеджмент позволяет:

- Создавать шаблоны и сценарии автотестов на основе ручных тестов из Test IT
- Хранить тестовую документацию в интегрированной среде разработки (*integrated development environment — IDE*)
- Для JetBrains IDE: Просматривать ручные тесты с привязанными автотестами и работать с кодом автотестов

Плагин поддерживает:

- IDE-системы Jetbrains:
  - IntelliJ IDEA
  - OpenIDE (используется плагин для IntelliJ IDEA)
  - PyCharm
  - Rider
  - WebStorm
- Visual Studio Code (VS Code)

Поддерживаемые языки программирования и тестовые фреймворки (список будет расширяться)

Java:

- Cucumber
- JUnit

JavaScript:

- CodeceptJS
- Cucumber
- Jest
- Mocha
- Playwright
- TestCafe

Python:

- Behave
- Nose2
- pytest
- Robot Framework

.NET:

- MSTest
- NUnit
- SpecFlow
- xUnit.net

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Взаимодействие с REST API

Обновлено: 4/22/26, 2:57 PM

Test IT Менеджмент для JetBrains IDE

> **Пользователям Test IT Про: требуется версия 5.0 или более поздняя**
>
> Плагин **Test IT Менеджмент 1.0.9** совместим с версией **Test IT 5.0** и выше. Актуальная версия Test IT Про доступна на официальном сайте .

Плагин Test IT Менеджмент позволяет:

- Создавать шаблоны и сценарии автотестов на основе ручных тестов из Test IT
- Хранить тестовую документацию в интегрированной среде разработки
- Просматривать ручные тесты с привязанными автотестами и работать с кодом автотестов

Плагин поддерживает IDE-системы Jetbrains :

- IntelliJ IDEA
- OpenIDE (используется плагин для IntelliJ IDEA)
- PyCharm
- Rider
- WebStorm

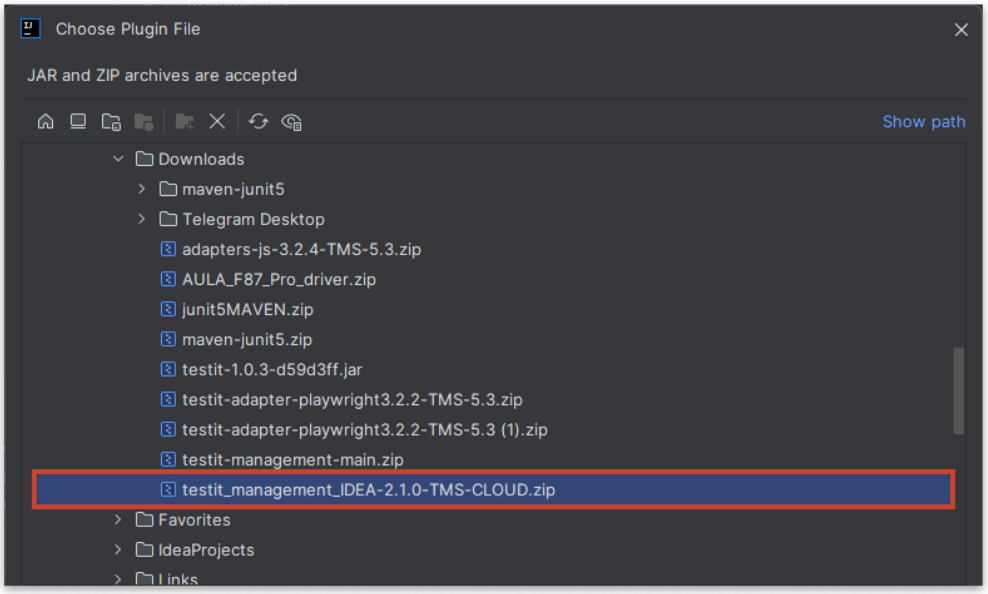
Установка

Установка плагина для различных продуктов JetBarins осуществляется по единому процессу с незначительными различиями. Для установки нужно скачать плагин на странице GitHub Test IT и установить его, следуя документации выбранной системы IDE.

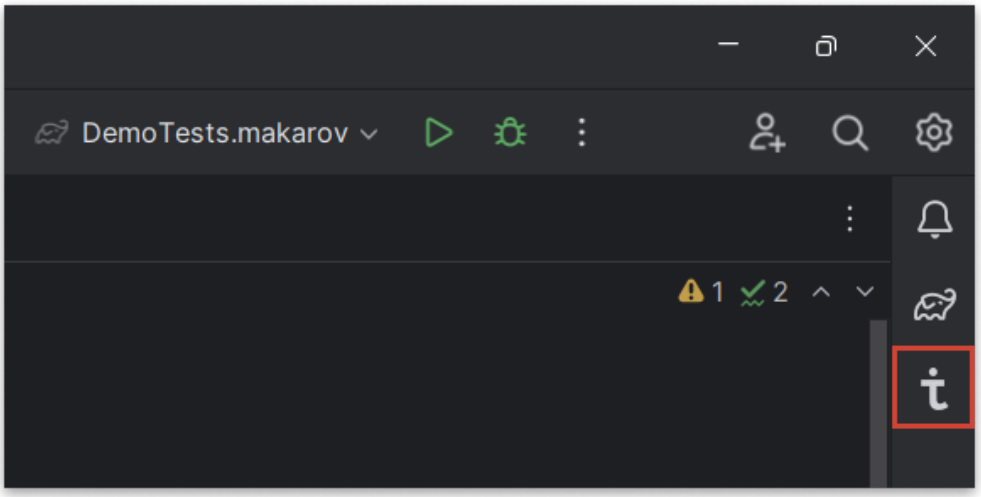
IntelliJ IDEA и OpenIDE

Плагин для IntelliJ IDEA используется в двух сервисах: IntelliJ IDEA и Open IDE. Процесс установки для них аналогичен.

1. Скачайте плагин на странице GitHub Test IT . Выберите и скачайте полный архив с плагином для **IntelliJ IDEA**.

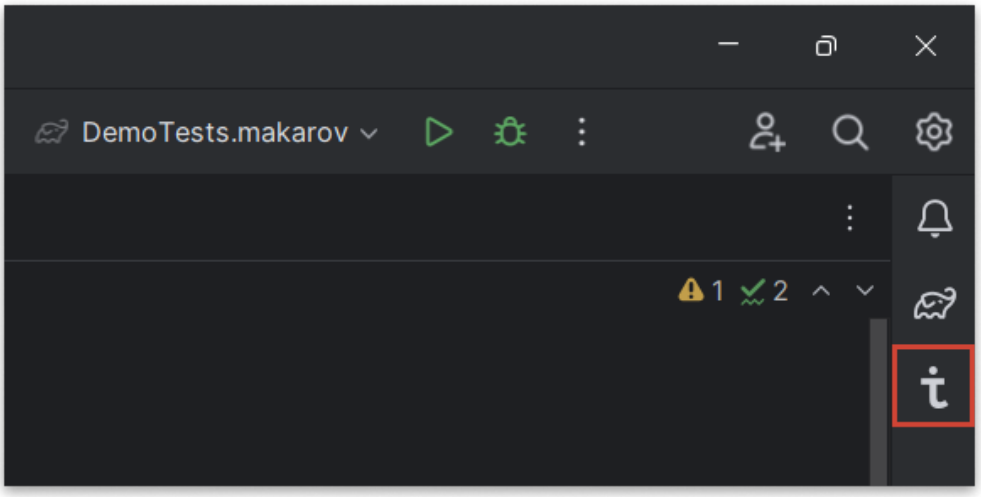


2. Установите плагин, следуя инструкции из документации IntelliJ IDEA . После установки значок плагина появится в навигационной панели вашей системы IDE.



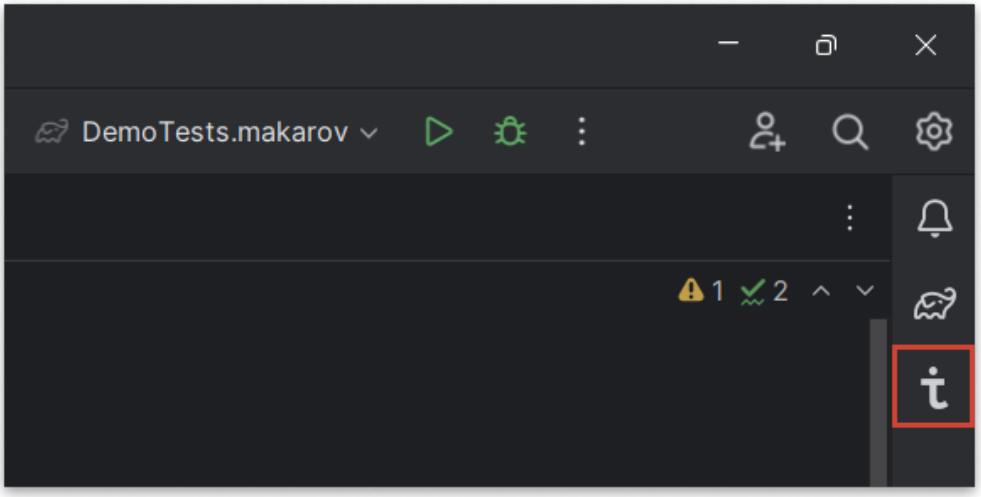
PyCharm

1. Скачайте плагин на странице GitHub Test IT . Выберите архив с плагином для **PyCharm**.
2. Установите плагин, следуя инструкции из документации PyCharm . После установки значок плагина появится в правом навигационном меню вашей системы IDE.



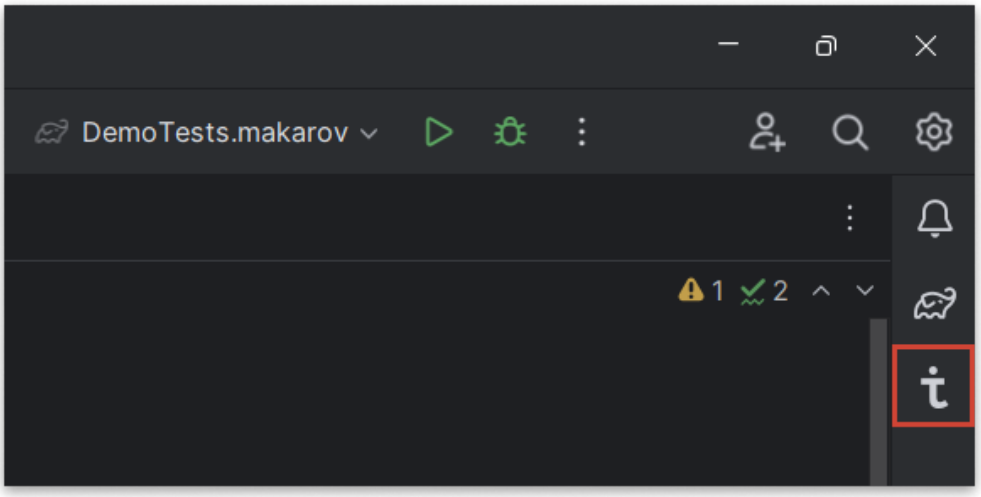
Rider

1. Скачайте плагин на странице GitHub Test IT . Выберите архив с плагином для **Rider**.
2. Установите плагин, следуя инструкции из документации Rider . После установки значок плагина появится в правом навигационном меню вашей системы IDE.



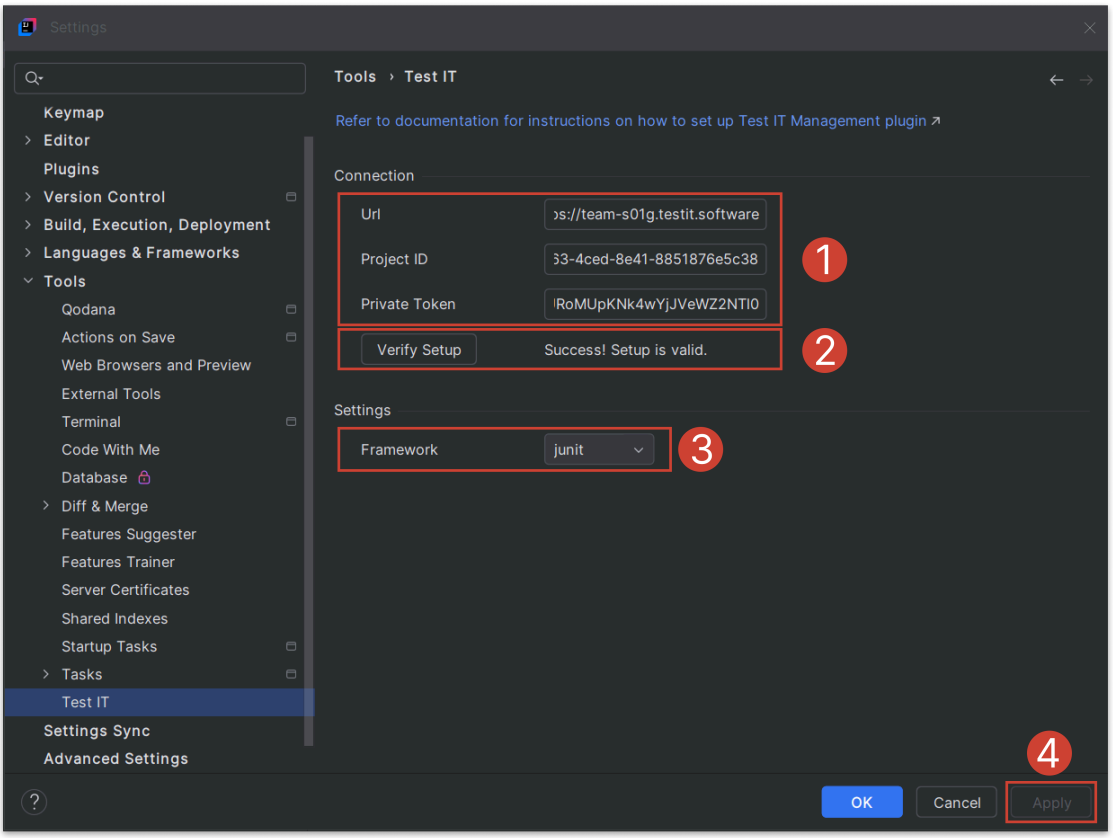
WebStorm

1. Скачайте плагин на странице GitHub Test IT . Выберите архив с плагином для **Rider**.
2. Установите плагин, следуя инструкции из документации WebStorm . После установки значок плагина появится в правом навигационном меню вашей системы IDE.



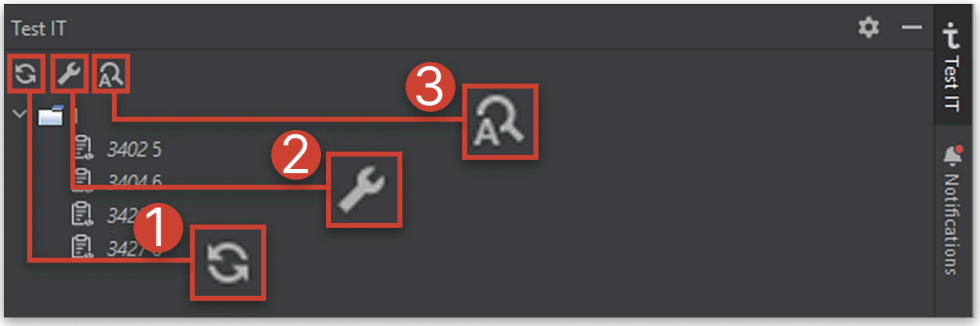
Настройка

1. В директории `File → Settings → Tools → Test IT` настройте соединение между Test IT и вашей системой IDE. Заполните поля:
   - **URL** — URL вашей системы Test IT
   - **Project ID** — ID проекта Test IT
   - **Private token** — приватный API-токен Test IT
2. В меню **Framework** выберите фреймворк: **JUnit** (Java) или **pytest** (Python). Изменять фреймворк можно в процессе работы, переустановка плагина не требуется.
3. Проверьте соединение, для этого нажмите **Verify Setup**.
   При успешной установке соединения отобразится текст: *label = "Success! Setup is valid."*
4. Примените настройки, для этого нажмите **Apply**.



Работа с проектом

- Откройте дерево проектов, перейдя в директорию **View → Tool Windows → Test IT**. Вам доступны опции:
  - Синхронизируйте ваш проект Test IT, нажав значок синхронизации (**Sync**) — 1.
  - Откройте настройки плагина, нажав значок настроек (**Test IT Settings**) — 2.
  - **Функциональность плагина для pytest, Behave и Robot Framework:** Используйте опцию поиска и парсинга аннотаций Allure-pytest и замены декораторов Allure, нажав значок поиска аннотаций (**Search Allure Methods**) — 3.



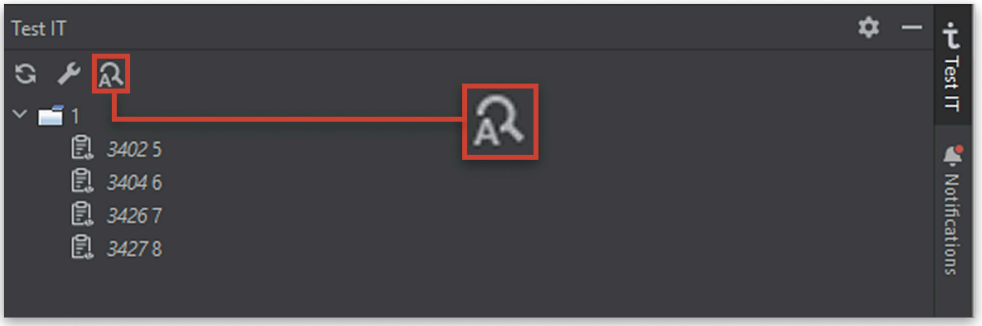
Замена декораторов Allure

Один из способов адаптации ваших автотестов для работы с Test IT — замена декораторов Allure на декораторы Test IT. Опция доступна для работы с фреймворками:

- Behave
- JUnit
- Playwright
- pytest
- Robot Framework

Подробную инструкцию смотрите на GitHub Test IT .
Для замены декораторов:

1. В проекте нажмите значок **Search Allure Methods**.

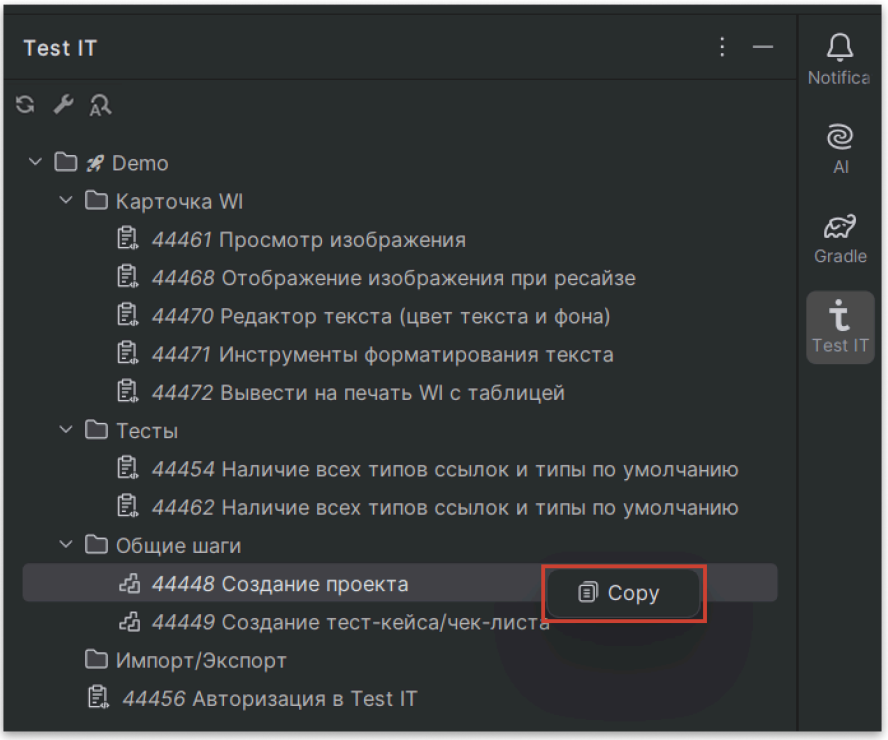


   Откроется меню найденных декораторов.
2. Выберите декораторы для замены:
   - Чтобы заменить один или несколько декораторов, выберите их и нажмите **Replace selected**.
   - Чтобы заменить все декораторы, нажмите **Replace all**. Откроется окно просмотра замен.
3. Нажмите **Replace**.

Создание шаблонов автотестов

Шаблон автотеста — это сниппет кода ручного теста, отображенный в системе IDE. Шаблоны используются для последующего написания автотестов. Чтобы получить сниппет кода:

- В дереве проектов скопируйте тест, для этого нажмите правую кнопку мыши, затем **Copy**.



Полученный шаблон можно использовать для создания автотестов (затем автотесты можно, например, развернуть на сервере, запустить локально или с помощью системы CI/CD).

Пример код сниппета:

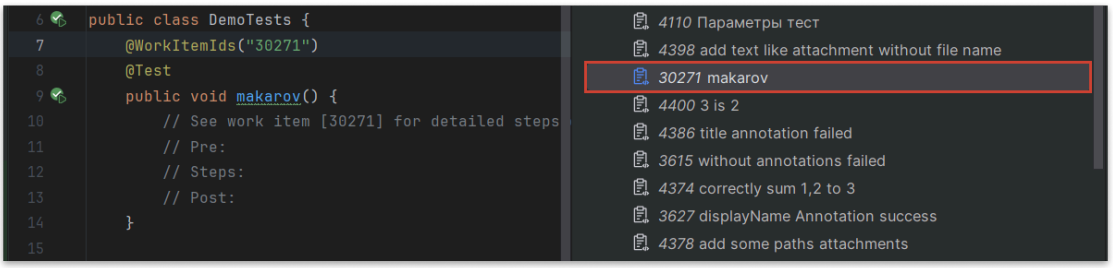
```
public class DemoTests {

    @WorkItemIds("44456")
    @Test
    public void авторизацияВTestIT() {
        // See work item [44456] for detailed steps description
        // Pre:
        //   Создание проекта
        // Steps:
        //   Перейти на страницу авторизации Test IT
        //   Ввести %Login
        //   Ввести %Password
        //   Нажать кнопку Войти
        // Post:
        //   Удалить локального пользователя
    }
}
```

Просмотр связанных автотестов

В дереве проекта тест-кейсы со связанными автотестами подсвечиваются синим. Чтобы открыть код автотеста, привязанного к ручному тесту в Test IT,

- Дважды нажмите на ручной тест в дереве проектов. Привязанный автотест отобразится в редакторе.



Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Взаимодействие с REST API

Обновлено: 8/13/26, 12:57 PM

Test IT Менеджмент для Visual Studio Code

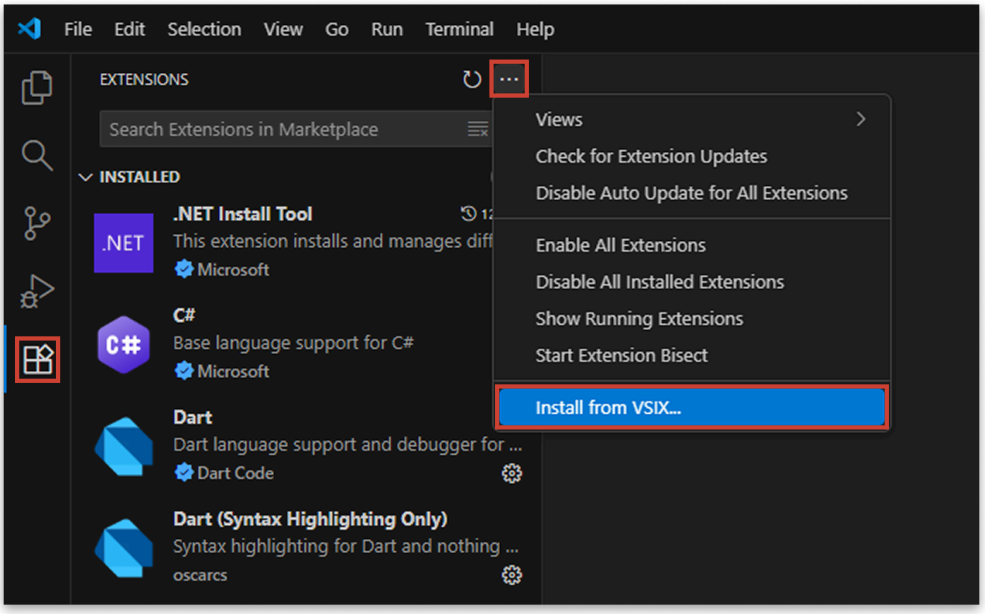
> **Пользователям Test IT Про: требуется версия 5.0 или более поздняя**
>
> Плагин **Test IT Менеджмент 1.0.0** совместим с версией **Test IT 5.0** и выше. Актуальная версия Test IT Про доступна на официальном сайте .

Плагин Test IT Менеджмент для Visual Studio Code (VS Code) позволяет:

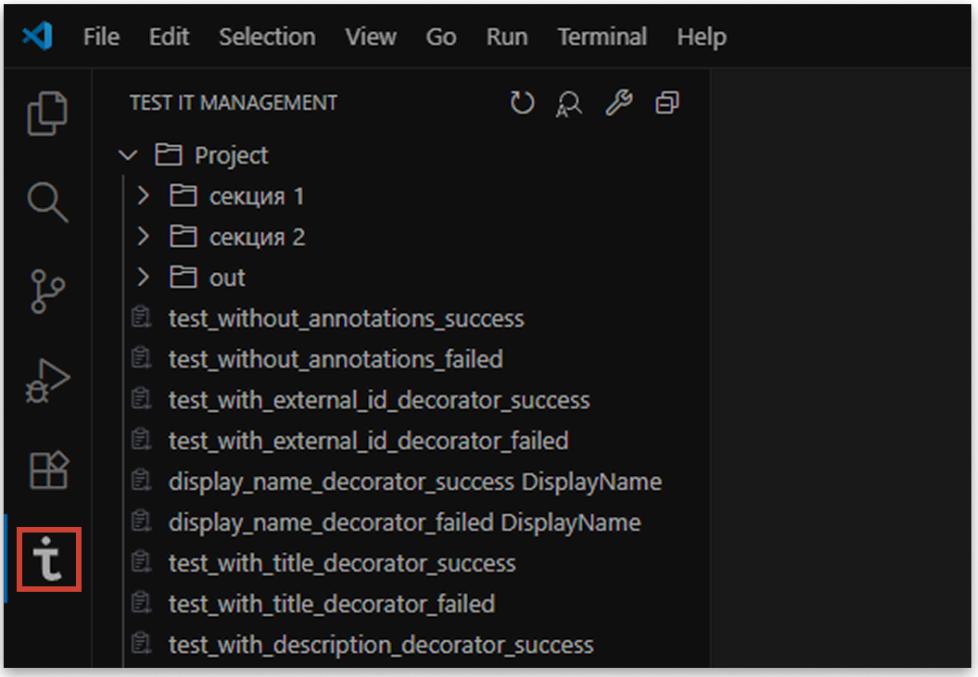
- Создавать шаблоны и сценарии автотестов на основе ручных тестов из Test IT
- Хранить тестовую документацию в интегрированной среде разработки

Установка

1. Скачайте `.vsix` -файл с плагином на странице GitHub Test IT .
2. В VS Code перейдите в раздел **Extensions**, откройте меню **Views and more actions** и выберите **Install from VSIX**.



3. Перейдите в директорию со скачанным `.vsix` -файл с плагином и установите его вручную.
   После установки значок плагина появится в левой навигационной панели VS Code.

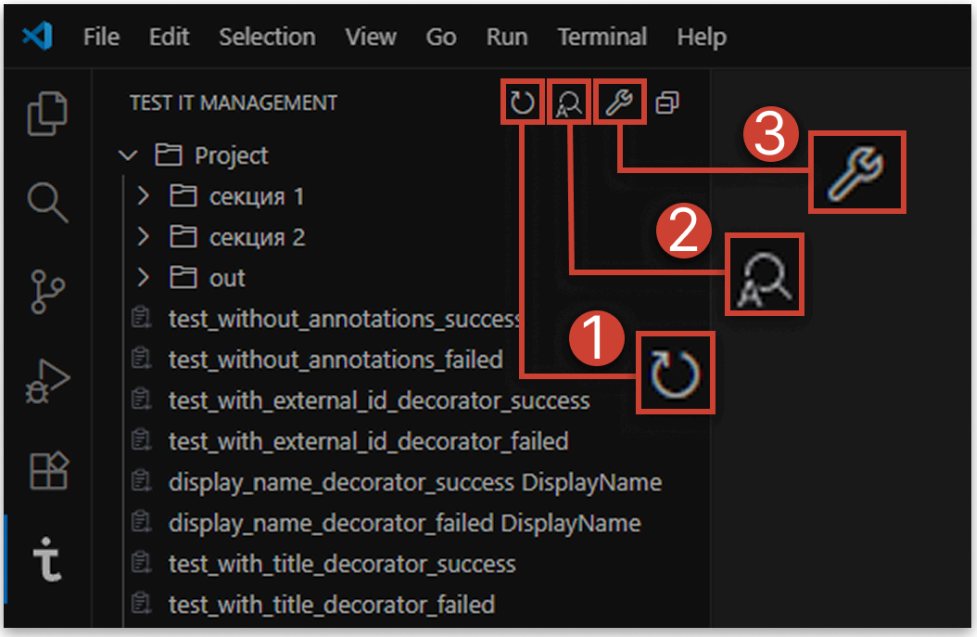


Настройка

1. В VS Code откройте директорию **Manage → Settings → Extensions → Test IT**.
2. Настройте соединение между Test IT и VS Code, заполнив поля:
   - **Framework** — ваш тестовый фреймворк
   - **Project ID** — ID проекта Test IT
   - **URL** — URL вашей системы Test IT
   - **Private token** — приватный API-токен Test IT

Работа с проектом

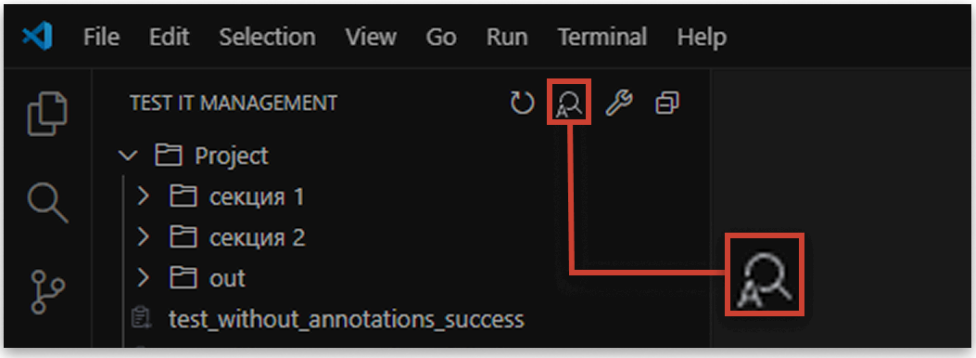
- Откройте дерево проектов, перейдя в директорию **View → Tool Windows → Test IT**. Вам доступны опции:
  - Синхронизируйте ваш проект Test IT, нажав значок синхронизации (**Sync**) — 1.
  - Откройте настройки плагина, нажав значок настроек (**Test IT Settings**) — 2.
  - **Функциональность плагина для Pytest и Robot Framework:** Используйте опцию поиска и парсинга аннотаций Allure-pytest и замены декораторов Allure, нажав значок поиска аннотаций (**Search Allure Methods**) — 3.



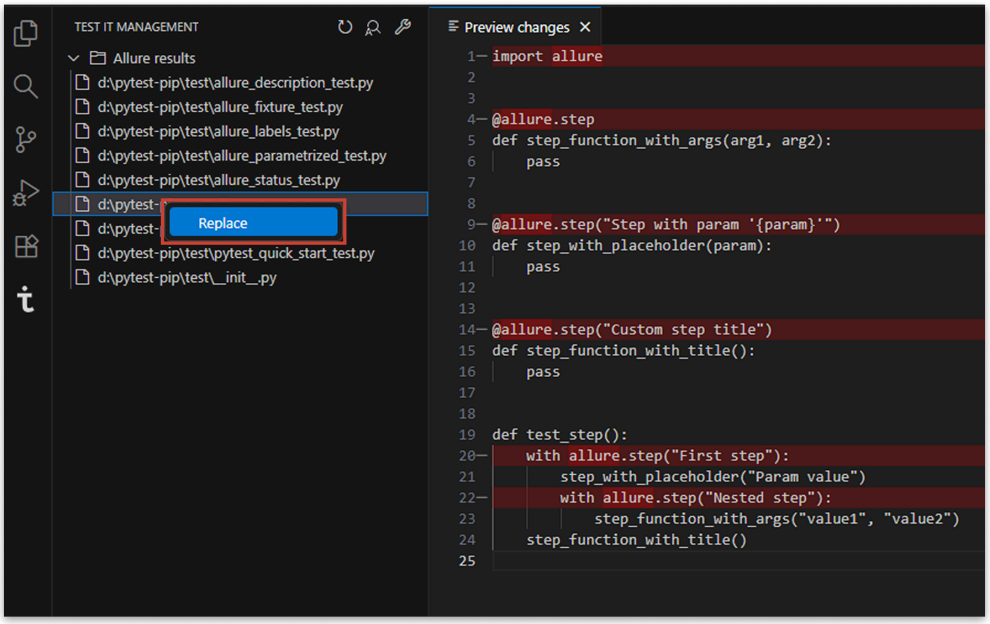
Замена декораторов Allure

Один из способов адаптации ваших автотестов для работы с Test IT — замена декораторов Allure на декораторы Test IT. Опция доступна для работы с фреймворками pytest и Robot Framework (список будет расширяться). Подробную инструкцию смотрите на GitHub Test IT .
Для замены декораторов:

1. В проекте нажмите значок **Search Allure Methods**.



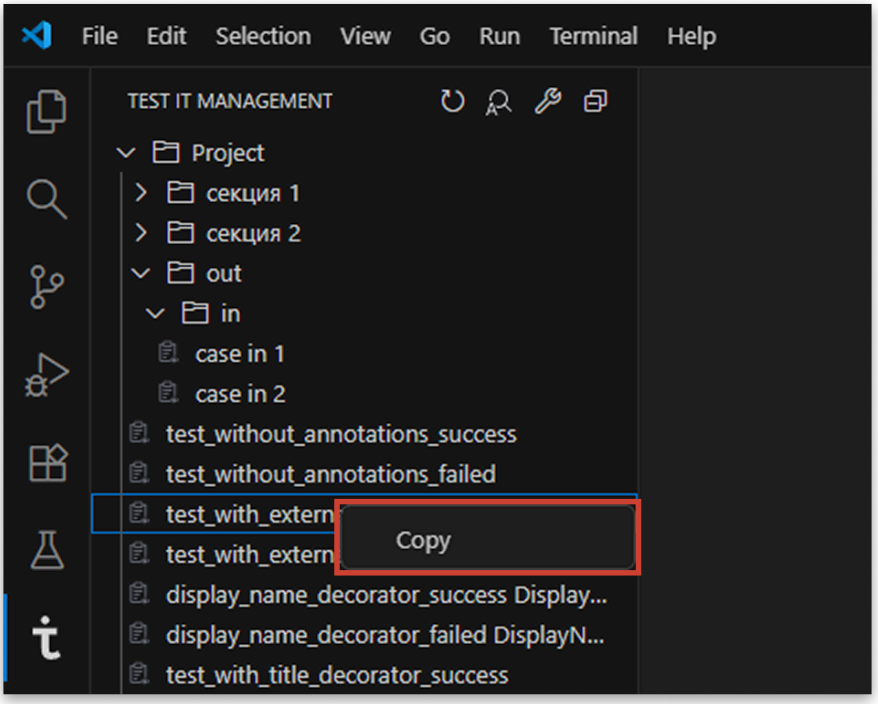
   Откроется меню найденных декораторов и окно просмотра изменений **Preview Changes**.
2. Правой кнопкой мыши нажмите на файл или его директорию.
3. Нажмите **Replace**, чтобы заменить декораторы в выбранном файле и **Replace all**, чтобы заменить декораторы во всех файлах.
4. В проекте нажмите значок **Search Allure Methods**.



Создание шаблонов автотестов

Шаблон автотеста — это сниппет кода ручного теста, отображенный в системе IDE. Шаблоны используются для последующего написания автотестов. Чтобы получить сниппет кода:

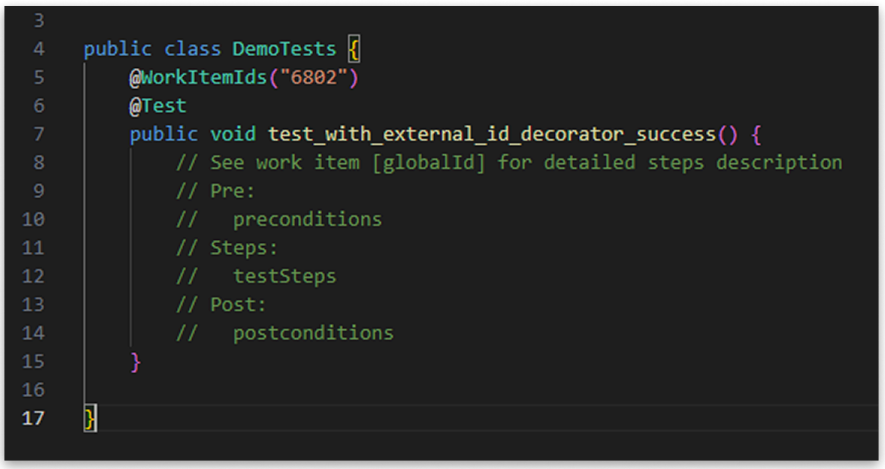
- В дереве проектов скопируйте тест, для этого нажмите правую кнопку мыши, затем **Copy**.



Полученный шаблон можно использовать для создания автотестов (затем автотесты можно, например, развернуть на сервере, запустить

локально или с помощью системы CI/CD).

Пример код сниппета:



```
public class DemoTests {
    @WorkItemIds("6802")
    @Test
    public void test_with_external_id_decorator_success() {
        // See work item [globalId] for detailed steps description
        // Pre:
        //   preconditions
        // Steps:
        //   testSteps
        // Post:
        //   postconditions
    }

}
```

Дополнительные ресурсы

Чтобы узнать больше о возможностях интеграции Test IT, прочтите наши статьи:

- Использование утилиты Test IT CLI
- Интеграции с CI/CD-системами
- Примеры интеграции с тестовыми фреймворками
- Взаимодействие с REST API

Обновлено: 4/22/26, 2:57 PM

Использование API-клиентов

Вы можете создавать собственные интеграции с Test IT, используя готовые API-клиенты. Они поддерживают все публичные методы, которые предоставляет система.

Примеры из практики

С использованием API-клиентов наши клиенты создали следующие интеграции:

- Загрузка и использование коллекций Postman в Test IT
- Импорт тестов из Cypress и TestLink в Test IT

Подробности смотрите на GitHub Test IT .

Список API-клиентов:

- C# (фреймворк .NET)
- Dart и Flutter
- Golang
- Java
- JavaScript
- Kotlin
- Python
- Swift

Обновлено: 4/15/26, 7:08 PM

Миграция в Test IT: возможности и процесс

- **Для этого раздела мы подготовили видео:**

Test IT поддерживает автоматическую миграцию тест-кейсов из других систем управления тестированием (TMS). Эта опция позволяет сэкономить время вашей команды, легко и аккуратно перенести всю вашу тестовую документацию из сторонней TMS, чтобы продолжить работу в Test IT.

Помимо тест-кейсов, в Test IT переносятся шаги тестов, общие шаги, ссылки, вложения, секции, итерации и многое другое (переносимые данные зависят от используемой TMS).

В Test IT поддерживается миграция из следующих TMS:

- Allure TestOps
- Azure DevOps
- DoQA
- HP ALM
- PractiTest
- Qase
- SpiraTest
- TestCollab
- TestLink

- TestRail
- Xray
- Zephyr Scale
- Zephyr Squad

Процесс миграции

Перенос документации из сторонних систем управления тестированием (TMS) в Test IT осуществляется с помощью набора утилит (мигратора), который включает в себя:

- **Экспортер** (exporter) — утилита для выгрузки файлов из сторонних систем
- Импортер (importer) — утилита для обработки экспортируемых файлов и их переноса в Test IT

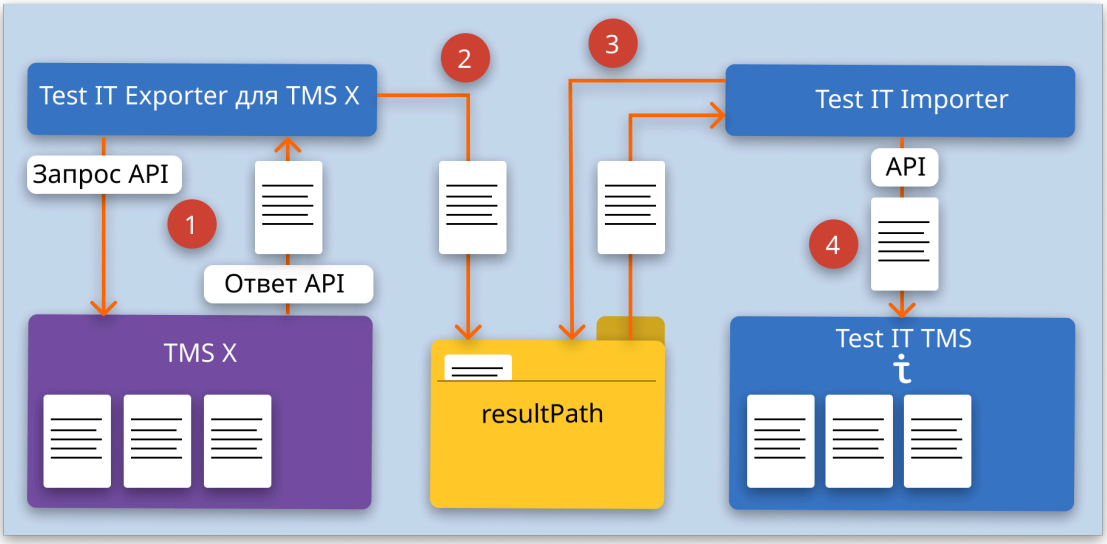
Импортер — это общая утилита для всех сторонних систем. Экспортер используется для каждой системы индивидуально. При работе с мигратором нужно выбрать импортер и экспортер для своей операционной системы (Windows, macOS или Linux).

> **Учитывайте версию TMS**
>
> Экспортер разрабатывается и тестируется с учетом текущих версий сторонних TMS. С некоторыми версиями (например, устаревшими) экспортер может работать некорректно.

Процесс миграции включает в себя следующие этапы (1-2 — выгрузка данных из сторонней TMS; 3-4 — загрузка данных в Test IT):

1. Экспортер выгружает тестовую документацию из сторонней TMS посредством API-запроса.
2. Экспортер сохраняет тестовую документацию в заданной директории (`resultPath`).
3. Импортер загружает и обрабатывает тестовую документацию из директории `resultPath` .

4. Импортер передает тестовую документацию в Test IT посредством API.



Мы непрерывно работаем над созданием новых утилит для миграции из сторонних тестовых систем. Вы можете предложить интеграцию на официальном сайте Test IT .

Обновлено: 9/30/26, 10:59 AM

Миграция из Allure TestOps

- **Для этого раздела мы подготовили видео:**

Вы можете автоматически перенести вашу тестовую документацию из Allure TestOps в Test IT. Вам потребуется учетная запись администратора Allure TestOps для получения API-токена.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов и их вложения
- Пользовательские поля
- Уровни тестирования (test layers)
- Ссылки

Миграция протестирована на версии **Allure TestOps 4.25.1**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из Allure TestOps
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер AllureExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из Allure TestOps

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `allure.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `allure.config.json`. Вы можете настроить соединение с помощью API-токена или токена на предъявителя (*bearer token*) от вашей системы Allure.
   - Пример настройки соединения в конфигурационном файле с помощью API-токена:

```json
{
    "resultPath": "/Users/user01/Documents/importer",
    "allure": {
        "url": "https://allure.ru/",
        "apiToken": "49a1238b-b0a6-4ebb-a47c-
acb2b7a9c4e9",
       "projectName": "ProjectName",
      "migrateAutotests": "true"
    }
}
```

- Пример настройки соединения в конфигурационном файле с помощью токена на предъявителя:

```json
{
    "resultPath": "/Users/user01/Documents/importer",
    "allure": {
        "url": "https://allure.ru/",
        "bearerToken":
"MmEwMjd4OWYtNGMxMC0zYjhkLWExMmUtNTZjYmY0OTE0MGExIDcxMjAyMCU
       "projectName": "ProjectName",
      "migrateAutotests": "true"
    }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `allure.url` — URL сервера Allure
- `allure.apiToken` — API-токен для доступа на сервер Allure
- `allure.bearerToken` — токен на предъявителя для доступа на сервер Allure
- `allure.projectName` — название проекта на сервере Allure
- `allure.migrateAutotests` — переменная для миграции автотестов. По умолчанию используется значение `false`. При использовании значения `true` в Test IT будут перенесены автотесты из Allure TestOps по аналогии с ручными тестами.

2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./AllureExporter{OS}-{version_number}
./AllureExporter{OS}-{version_number}
```

В команде вместо `AllureExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера.
Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `allure.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `allure.config.json`. Вы можете настроить соединение с помощью API-токена или токена на предъявителя (*bearer token*) от вашей системы Allure.
   - Пример настройки соединения в конфигурационном файле с помощью API-токена:

     ```json
     {
         "resultPath": "/Users/user01/Documents/importer",
         "allure": {
             "url": "https://allure.ru/",
             "apiToken": "49a1238b-b0a6-4ebb-a47c-
     acb2b7a9c4e9",
            "projectName": "ProjectName",
           "migrateAutotests": "true"
         }
     }
     ```

   - Пример настройки соединения в конфигурационном файле с помощью токена на предъявителя:

     ```json
     {
         "resultPath": "/Users/user01/Documents/importer",
         "allure": {
             "url": "https://allure.ru/",
             "bearerToken":
     "MmEwMjd4OWYtNGMxMC0zYjhkLWExMmUtNTZjYmY0OTE0MGExIDcxMjAyMCU
            "projectName": "ProjectName",
           "migrateAutotests": "true"
         }
     }
     ```

     Значения параметров:
     - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
     - `allure.url` — URL сервера Allure
     - `allure.apiToken` — API-токен для доступа на сервер Allure
     - `allure.bearerToken` — токен на предъявителя для доступа на сервер Allure
     - `allure.projectName` — название проекта на сервере Allure
     - `allure.migrateAutotests` — переменная для миграции автотестов. По умолчанию используется значение `false`. При использовании

значения `true` в Test IT будут перенесены автотесты из Allure TestOps по аналогии с ручными тестами.

2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 8/12/26, 4:43 PM

Миграция из Azure DevOps

Вы можете автоматически перенести вашу тестовую документацию из Azure DevOps в Test IT. Вам потребуется учетная запись администратора Azure DevOps для получения токена.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов и их вложения
- Общие шаги
- Итерации
- Ссылки

Миграция протестирована на версиях **Azure DevOps Server 2022** и **Azure DevOps Cloud**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из Azure DevOps
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Выберите и скачайте экспортер AzureExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы и версии Azure DevOps в зависимости от типа поставки (облачная или серверная) и года релиза (для серверной версии):

#	Версия Azure DevOps	Экспортер для Windows	Экспортер для macOS	Экспортер для Linux

1	Облачная	AzureExporter-win-x64-2.1.9.exe	AzureExporter-osx-arm64-2.1.9 AzureExporter-osx-x64-2.1.9	AzureExporte linux-x64-2.1.
2	Серверная от 2022 и позднее	AzureExporter-win-x64-2.1.9.exe	AzureExporter-osx-arm64-2.1.9 AzureExporter-osx-x64-2.1.9	AzureExporte linux-x64-2.1.
3	Серверная от 2020-2021	AzureExporter-2020-win-x64-2.1.9.exe	AzureExporter-2020-osx-arm64-2.1.9 AzureExporter-2020-osx-x64-2.1.9	AzureExporte 2020-linux-x64-2.1.9
4	Серверная от 2019 и ранее	AzureExporter-2019-win-x64-2.1.9.exe	AzureExporter-2019-osx-arm64-2.1.9 AzureExporter-2019-osx-x64-2.1.9	AzureExporte 2019-linux-x64-2.1.9

Сохраните экспортер в отдельной папке.

2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из Azure DevOps

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `azure.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `azure.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "azure": {
    "url": "https://dev.azure.com/testit",
    "token":
"ovequpv4qsgi5fv7tb5eysqddodkux2cwp3failevеd56qandckq",
    "projectName": "ProjectName"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `azure.url` — URL сервера Azure с названием организации
   - `azure.token` — токен для доступа на сервер Azure
   - `azure.projectName` — название проекта на сервере Azure
2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./AzureExporter{OS}-{version_number}
./AzureExporter{OS}-{version_number}
```

В команде вместо `AzureExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера. Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `azure.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `azure.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "azure": {
    "url": "https://dev.azure.com/testit",
    "token":
"ovequpv4qsgi5fv7tb5eysqddodkux2cwp3failevеd56qandckq",
    "projectName": "ProjectName"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `azure.url` — URL сервера Azure с названием организации
- `azure.token` — токен для доступа на сервер Azure
- `azure.projectName` — название проекта на сервере Azure

2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В

случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 7/7/26, 2:09 PM

Миграция из DoQA

Вы можете автоматически перенести вашу тестовую документацию из DoQA в Test IT. Вам потребуется учетная запись администратора DoQA.

В процессе миграции переносятся следующие данные:

- Информация о тесте:
  - Название
  - Описание
  - Шаги
  - Статус
  - Приоритет
  - Предусловия
  - Встроенные изображения base64
  - Вложенные файлы
  - Ссылки: Яндекс Трекер (тип ссылки *Issue*), Figma, Google Docs, Wiki (связанные)
- Секции
  Материалы в формате HTML переводятся в текст с соответствующими разрывами строк.

При экспорте частота API- запросов ограничивается, чтобы не превышать лимиты запросов DoQA.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из DoQA
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер DoqaExporter на странице GitHub Test IT.
   Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.

2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из DoQA

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `doqa.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `doqa.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "doqa": {
    "url": "https://your-workspace.doqa.app",
    "email": "user@example.com",
    "password": "your_password",
    "spaceId": 1
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `doqa.url` — URL рабочего пространства DoQA
- `doqa.email` — электронный адрес, используемый при авторизации в DoQAэкспорт (по умолчанию: `1`)
- `doqa.password` — пароль, используемый при авторизации в DoQA
- `doqa.spaceId` — ID рабочего пространства DoQA, из которого производится экспорт

2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./DoqaExporter
./DoqaExporter
```

Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `doqa.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `doqa.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "doqa": {
    "url": "https://your-workspace.doqa.app",
    "email": "user@example.com",
    "password": "your_password",
    "spaceId": 1
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `doqa.url` — URL рабочего пространства DoQA
- `doqa.email` — электронный адрес, используемый при авторизации в DoQAэкспорт (по умолчанию: `1`)
- `doqa.password` — пароль, используемый при авторизации в DoQA
- `doqa.spaceId` — ID рабочего пространства DoQA, из которого производится экспорт

2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.

3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 9/30/26, 10:59 AM

Миграция из HP ALM

Вы можете автоматически перенести вашу тестовую документацию из HP ALM в Test IT.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов
- Ссылки
- Вложения

Миграция протестирована на версии **HP ALM 12.60**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из HP ALM
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер HPALMExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из HP ALM

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `hpalm.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `hpalm.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/hpalm",
  "hpalm": {
    "url": "[URL of your HP ALM Server]",
    "clientId": "apikey-egaoktntfhidsjoddmcp",
    "secret": "dpojbgifjifbjplf",
    "domainName": "DEFAULT",
    "projectName": "Migration"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `hpalm.url` — URL сервера HP ALM с названием организации
- `hpalm.clientId` — идентификатор клиента (client ID) для доступа на сервер HP ALM
- `hpalm.secret` — секретный ключ для доступа на сервер HP ALM
- `hpalm.domainName` — доменное имя сервера HP ALM
- `hpalm.projectName` — название проекта на сервере HP ALM

2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./HPALMExporter{OS}-{version_number}
./HPALMExporter{OS}-{version_number}
```

В команде вместо `HPALMExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера. Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `hpalm.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `hpalm.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/hpalm",
  "hpalm": {
    "url": "[URL of your HP ALM Server]",
    "clientId": "apikey-egaoktntfhidsjoddmcp",
    "secret": "dpojbgifjifbjplf",
    "domainName": "DEFAULT",
    "projectName": "Migration"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `hpalm.url` — URL сервера HP ALM с названием организации

- `hpalm.clientId` — идентификатор клиента (client ID) для доступа на сервер HP ALM
- `hpalm.secret` — секретный ключ для доступа на сервер HP ALM
- `hpalm.domainName` — доменное имя сервера HP ALM
- `hpalm.projectName` — название проекта на сервере HP ALM

2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 7/7/26, 2:09 PM

Миграция из PractiTest

Вы можете автоматически перенести вашу тестовую документацию из PractiTest в Test IT. Вам потребуется учетная запись администратора PractiTest для получения токена.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов и их вложения
- Общие шаги
- Предусловия
- Теги
- Пользовательские атрибуты

Миграция протестирована на версии **PractiTest Cloud 3.0.16**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из PractiTest
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер PractiTestExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из PractiTest

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `practitest.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `practitest.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "practiTest": {
    "url": "[URL of your PractiTest server]",
    "token": "a7b9bea53f1e7a6ef7d3052969b55926680bc13b",
    "projectId": "15467"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `practiTest.url` — URL API системы PractiTest
- `practiTest.token` — токен для доступа в систему PractiTest
- `practiTest.projectId` — идентификатор проекта PractiTest

2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./PractiTestExporter{OS}-{version_number}
./PractiTestExporter{OS}-{version_number}
```

В команде вместо `PractiTestExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера.
Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `practitest.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `practitest.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "practiTest": {
    "url": "[URL of your PractiTest server]",
    "token": "a7b9bea53f1e7a6ef7d3052969b55926680bc13b",
    "projectId": "15467"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `practiTest.url` — URL API системы PractiTest
- `practiTest.token` — токен для доступа в систему PractiTest
- `practiTest.projectId` — идентификатор проекта PractiTest

2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.

3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 8/12/26, 4:43 PM

Миграция из Qase

Возможны ограничения на стороне Qase

Qase ограничил доступ пользователей на территории РФ. Возможна некорректная работа мигратора. Для уточнения информации обратитесь в техническую поддержку: support@testit.software .

- **Для этого раздела мы подготовили видео:**

Вы можете автоматически перенести вашу тестовую документацию из Qase в Test IT. Вам потребуется учетная запись администратора Qase для получения токена.

В процессе миграции переносятся следующие данные:

- Информация о тесте:
  - Название
  - Описание
  - Статус
  - Приоритет
- Шаги теста:
  - Основные (*action*, *expected*, *data*)
  - Gherkin-шаги (*step action*)
  - Общие шаги
- Предусловия (*setup step actions*)
- Постусловия (*teardown step actions*)
- Секции и подсекции (*suites*)
- Вложения
- Теги
- Параметры
- Атрибуты: *Type*, *Layer*, *Is flaky*, *Severity*, *Behavior*, *To be automated*, *Automation status*, а также пользовательские атрибуты

Миграция протестирована на версии **Qase Cloud Q3 2024**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из Qase
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер QaseExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из Qase

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `qase.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `qase.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer/qase",
  "qase": {
    "url": "http://api.qase.io/",
    "token":
"e62b232d32e69d82h53e8f0211209ccf652efdccddf3d1fb11ef764ghi998
    "projectKey": "KEY"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `qase.url` — URL API-сервера Qase
   - `qase.token` — токен для доступа на сервер Qase
   - `qase.projectKey` — ключ проекта (Project code) на сервере Qase
2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./QaseExporter{OS}-{version_number}
./QaseExporter{OS}-{version_number}
```

   В команде вместо `QaseExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название

операционной системы и версию экспортера.
Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `qase.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `qase.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer/qase",
  "qase": {
    "url": "http://api.qase.io/",
    "token":
"e62b232d32e69d82h53e8f0211209ccf652efdccddf3d1fb11ef764ghi998
    "projectKey": "KEY"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `qase.url` — URL API-сервера Qase
   - `qase.token` — токен для доступа на сервер Qase
   - `qase.projectKey` — ключ проекта (Project code) на сервере Qase
2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 8/12/26, 4:43 PM

Миграция из SpiraTest

Вы можете автоматически перенести вашу тестовую документацию из SpiraTest в Test IT. Вам потребуется учетная запись администратора SpiraTest для получения токена.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов
- Общие шаги
- Вложения

Миграция протестирована на версии **SpiraTest 7.9.0.1**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из SpiraTest
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер SpiraTestExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из SpiraTest

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `spiratest.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `spiratest.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/spiratest",
  "spiraTest": {
    "url": "[SpiraTest server URL with company name]",
    "username": "administrator",
    "token": "{620F2362-41CB-4F87-ACDA-0CF223C72737}",
    "projectName": "Library Information System
(Sample)"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `spiraTest.url` — URL сервера SpiraTest с названием организации
   - `spiraTest.username` — имя пользователя для доступа на сервер SpiraTest
   - `spiraTest.token` — токен для доступа на сервер SpiraTest
   - `spiraTest.projectName` — название проекта в системе SpiraTest

2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./SpiraTestExporter{OS}-{version_number}
./SpiraTestExporter{OS}-{version_number}
```

В команде вместо `SpiraTestExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера. Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `spiratest.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `spiratest.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/spiratest",
  "spiraTest": {
    "url": "[SpiraTest server URL with company name]",
    "username": "administrator",
    "token": "{620F2362-41CB-4F87-ACDA-0CF223C72737}",
    "projectName": "Library Information System
(Sample)"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `spiraTest.url` — URL сервера SpiraTest с названием организации

- `spiraTest.username` — имя пользователя для доступа на сервер SpiraTest
- `spiraTest.token` — токен для доступа на сервер SpiraTest
- `spiraTest.projectName` — название проекта в системе SpiraTest

2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 8/12/26, 4:43 PM

Миграция из TestCollab

Вы можете автоматически перенести вашу тестовую документацию из TestCollab в Test IT. Вам потребуется учетная запись администратора TestCollab для получения токена.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов
- Общие шаги
- Вложения
- Пользовательские поля

Миграция протестирована на версии **TestCollab 2.0.20**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из TestCollab
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер TestCollabExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из TestCollab

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `testcollab.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `testcollab.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/testcollab",
  "testcollab": {
    "url": "https://api.testcollab.io/",
    "token": "bPM3ji6XEw64fUHq",
    "projectName": "ProjectName"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `testcollab.url` — URL сервера TestCollab с названием организации
   - `testcollab.token` — токен для доступа на сервер TestCollab
   - `testcollab.projectName` — название проекта в системе TestCollab
2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./TestCollabExporter{OS}-{version_number}
./TestCollabExporter{OS}-{version_number}
```

В команде вместо `TestCollabExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера.
Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `testcollab.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `testcollab.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/testcollab",
  "testcollab": {
    "url": "https://api.testcollab.io/",
    "token": "bPM3ji6XEw64fUHq",
    "projectName": "ProjectName"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `testcollab.url` — URL сервера TestCollab с названием организации
- `testcollab.token` — токен для доступа на сервер TestCollab
- `testcollab.projectName` — название проекта в системе TestCollab

2. Запустите `.exe` -файл экспортера и дождитесь окончания экспорта.

3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 7/7/26, 2:09 PM

Миграция из TestLink

Вы можете автоматически перенести вашу тестовую документацию из TestLink в Test IT. Вам потребуется учетная запись администратора TestLink для получения токена.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов
- Секции
- Вложения

Миграция протестирована на версии **TestLink 1.9.20**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из TestLink
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер TestLinkExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из TestLink

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `testlink.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `testlink.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer/",
  "testLink": {
    "url": "[URL of your TestLink API]",
    "token": "15fb632cdd8b606561a8b60d69a7149e",
    "projectName": "ProjectName"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `testlink.url` — URL API системы TestLink
   - `testlink.token` — токен для доступа в систему TestLink
   - `testlink.projectName` — название проекта в системе TestLink
2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./TestLinkExporter{OS}-{version_number}
./TestLinkExporter{OS}-{version_number}
```

В команде вместо `TestLinkExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера. Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `testlink.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `testlink.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer/",
  "testLink": {
    "url": "[URL of your TestLink API]",
    "token": "15fb632cdd8b606561a8b60d69a7149e",
    "projectName": "ProjectName"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `testlink.url` — URL API системы TestLink
- `testlink.token` — токен для доступа в систему TestLink
- `testlink.projectName` — название проекта в системе TestLink

2. Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В

случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 7/7/26, 2:09 PM

Миграция из TestRail

Вы можете автоматически перенести вашу тестовую документацию из TestRail в Test IT. При миграции переносятся данные, содержащиеся в файлах TestRail XML.

В процессе миграции переносятся следующие данные:

- Тест-кейсы с указанием приоритета в описании
- Статусы тестов
- Шаги тестов
- Общие шаги тестов
- Предусловия
- Вложения
- Время выполнения
- Атрибуты

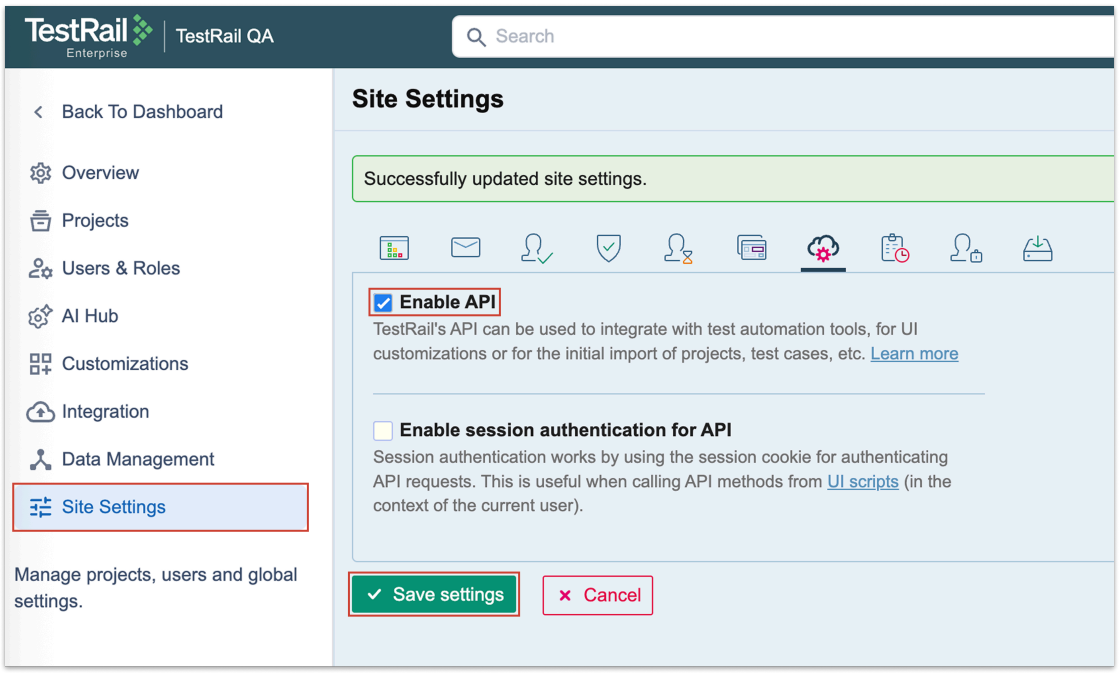
Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из TestRail
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Убедитесь, что в вашей системе TestRail включен API (требуется роль **Administrator**):
   а. Перейдите в раздел **Admin** → **Site Settings** → **API**.
   б. Проверьте, выбрана ли опция **Enable API**, если нет — выберите ее и сохраните изменения.



Подробности — в официальной документации TestRail .

2. Скачайте экспортер TestRailExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
3. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из TestRail

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `testrail.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `testrail.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "testrail": {
    "url": "[your TestRail URL]",
    "login": "user",
    "password": "pass",
    "projectName": "ProjectName"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `url` — URL доступа к учетной записи TestRail
- `login` — логин учетной записи TestRail
- `password` — пароль учетной записи TestRail
- `projectName` — название вашего проекта TestRail

2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./TestRailExporter{OS}-{version_number}
./TestRailExporter{OS}-{version_number}
```

В команде вместо `TestRailExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера. Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `testrail.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `testrail.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "testrail": {
    "url": "[your TestRail URL]",
    "login": "user",
    "password": "pass",
    "projectName": "ProjectName"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `url` — URL доступа к учетной записи TestRail
- `login` — логин учетной записи TestRail
- `password` — пароль учетной записи TestRail
- `projectName` — название вашего проекта TestRail

2. Запустите `.exe` -файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 7/7/26, 2:09 PM

Миграция из Xray

Вы можете автоматически перенести вашу тестовую документацию из Xray в Test IT. Вам потребуется учетная запись администратора Xray для получения токена.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов
- Секции
- Вложения

Миграция протестирована на версии **Xray Enterprise 1.0.1 (server)**.

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из Xray
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер XrayExporter на странице GitHub Test IT. Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из Xray

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `xray.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `xray.config.json` . Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/xray",
  "xray": {
    "url": "[URL of your Jira server]",
    "token":
"MTg1Mzk4NDQxMTE3Ojy/3+hs+mkQ8cqhpKh0bMUPwU3e",
    "projectKey": "XRAYT"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `xray.url` — URL сервера Xray с названием организации
   - `xray.token` — токен для доступа на сервер Xray
   - `xray.projectKey` — ключ проекта на сервере Xray
2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:

```sh
sudo chmod +x ./XRayExporter{OS}-{version_number}
./XRayExporter{OS}-{version_number}
```

В команде вместо `XRayExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера. Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `xray.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `xray.config.json`. Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath":
"/Users/user01/Documents/importer/xray",
  "xray": {
    "url": "[URL of your Jira server]",
    "token":
"MTg1Mzk4NDQxMTE3Ojy/3+hs+mkQ8cqhpKh0bMUPwU3e",
    "projectKey": "XRAYT"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `xray.url` — URL сервера Xray с названием организации
- `xray.token` — токен для доступа на сервер Xray
- `xray.projectKey` — ключ проекта на сервере Xray

2. Запустите `.exe` -файл экспортера и дождитесь окончания экспорта.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В

случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 7/7/26, 2:09 PM

Миграция из Zephyr Scale

Вы можете автоматически перенести вашу тестовую документацию из Zephyr Scale в Test IT. Вам потребуется учетная запись администратора Zephyr Scale для получения токена.

Учитывайте форму поставки (облачная или серверная)

Ссылки на скачивание экспортера, параметры выгрузки тестов и команда для запуска экспортера для серверной (Data Center) и облачной (Cloud) поставок различаются.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов
- Атрибуты тестов
- Вложения тестов*
- Секции
- Ссылки
- Архивные тест-кейсы

---

*В облачной версии Zephyr Scale возможны проблемы с миграцией вложений тестов.

Миграция протестирована на версиях:

- Серверная поставка: **Zephyr Scale Data Center 7.0.1**
- Облачная поставка: **Zephyr Scale 1.0.62-AC Cloud**

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из Zephyr Scale
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер на странице GitHub Test IT . Выберите файл для вашей версии поставки:
   - Для **серверной** поставки — с названием **ZephyrScaleServerExporter**
   - Для **облачной** поставки — с названием **ZephyrScaleExporter**
     Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из Zephyr Scale

> **Используйте экспортер для своей операционной системы**
>
> Для различных операционных систем требуется свой экспортер. В этой инструкции содержатся шаги для различных систем (шаги 2—3)

Серверная версия

1. Создайте конфигурационный файл `zephyr.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `zephyr.config.json` .
   Вы можете настроить соединение с помощью токена или с помощью логина и пароля. Также доступна возможность частичного экспорта, позволяющая перенести крупные проекты в несколько итераций.
   - Пример настройки соединения в конфигурационном файле с помощью токена:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "zephyr": {
    "url": "Link to your JIRA instance",
    "exportArchived": true,
    "confluence": "URL of your Confluence server",
    "token":
"MDc2MjIxNjVzNjg40kJCA43J4AfsIRBXomRs8bKw81+D",
    "confluenceToken":
"MDc2MjIxNjVzNjg40kJCA43J4AfsIRBXomRs8bKw81+D",
    "projectKey": "PK"
    }
 }
```

- Пример настройки соединения в конфигурационном файле с помощью логина и пароля:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "zephyr": {
    "url": "Link to your JIRA instance",
    "exportArchived": true,
    "confluence": "https://confluence.instance.ru",
    "login": "user",
    "password": "pass",
    "confluenceLogin": "User",
    "confluencePassword": "pass",
    "projectKey": "PK"
    }
 }
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `zephyr.url` — URL сервера Zephyr с названием организации
- `zephyr.exportArchived` — экспорт архивных тест-кейсов
- `zephyr.confluence` — URL сервера Confluence.
- `zephyr.token` — ключ (токен) доступа на сервер Jira
- `zephyr.login` — ваш логин для входа на сервер Jira
- `zephyr.password` — ваш пароль для входа на сервер Jira
- `zephyr.confluenceToken` — токен доступа к API Confluence

- `zephyr.confluenceLogin` — логин сервера Confluence
- `zephyr.confluencePassword` — пароль сервера Confluence
- `zephyr.projectKey` — ключ проекта на сервере Zephyr

  Параметры `zephyr.confluence`, `zephyr.confluenceToken`, `zephyr.confluenceLogin` и `zephyr.confluencePassword` используются, когда необходимо перенести ссылки из Confluence. Для частичного экспорта используются дополнительные параметры.

2. **Если вы используете macOS или Linux:** Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер:
   - **Для Windows:** Запустите `.exe`-файл экспортера и дождитесь окончания экспорта.
   - **Для macOS или Linux:** Используйте команду:

     ```sh
     sudo chmod +x ./ZephyrScaleServerExporter{OS}-{version_number}
     ./ZephyrScaleServerExporter{OS}-{version_number}
     ```

     В команде вместо `ZephyrScaleServerExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера.

     Дождитесь окончания экспорта.
4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Частичный экспорт и объединение частей проекта

Функции частичного экспорта (Partial export) и объединения частей проекта (Merge functionality for batches) опциональны и используются для миграции крупных проектов по частям, чтобы перенести проект в несколько итераций и избежать простоев в работе. Эти функции применяются последовательно: проект экспортируется по частям, затем эти части объединяются.

1. Для частичного экспорта добавьте в конфигурационный файл блок:

```json
{
  "zephyr": {
    "partial": true,
    "partialFolderName": "batch",
    "count": 1000
  }
}
```

   Таким способом можно перенести тест-кейсы проекта в несколько итераций. Сохраненные тест-кейсы выгружаются во временный файл `$"{config.Value.Zephyr.ProjectKey}-batch.txt"`, расположенном в папке с файлом `.exe`, который используется как фильтр при очередном запуске.

2. Когда все части проекта, выгружены, объедините проект, добавив в конфигурационный файл блок:

```json
{
  "zephyr": {
    "merge": true
  }
}
```

   Объединение частей проекта происходит с учетом значений `resultPath` и `zephyr.projectKey`.

   Значения параметров:

   - `partial` — активация частичной выгрузки
   - `partialFolderName` — название папки, куда будет перенесена часть проекта
   - `count` — количество тест-кейсов в переносимой части проекта (batch)
   - `merge` — объединение частей проекта

Пример

Для переноса проекта, содержащего 50000 тест-кейсов, указывается `count = 10000`. Затем производится 5 итераций без изменения параметров. Тест-кейсы выгружаются в директорию `resultPath/ProjectKey/batch_1, batch_2, batch_x .`

Облачная версия

1. Создайте конфигурационный файл `zephyr.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `zephyr.config.json .`
   Пример настройки соединения в конфигурационном файле:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "zephyr": {
    "url": "https://api.zephyrscale.smartbear.com/v2/",
    "token":
"ovequpv4qsgi5fv7tb5eysqddodkux2cwp3faileved56qandckq",
    "projectName": "ProjectName"
  }
}
```

   Значения параметров:
   - `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
   - `zephyr.url` — URL сервера Zephyr с названием организации
   - `zephyr.token` — токен для доступа на сервер Zephyr
   - `zephyr.projectName` — название проекта на сервере Zephyr
2. **Если вы используете macOS или Linux:** Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер:
   - **Для Windows:** Запустите `.exe` -файл экспортера и дождитесь окончания экспорта.
   - **Для macOS или Linux:** Используйте команду:

```sh
sudo chmod +x ./ZephyrScaleExporter{OS}-{version_number}
./ZephyrScaleExporter{OS}-{version_number}
```

В команде вместо `ZephyrScaleExporter{OS}-{version_number}` укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера.

Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 8/13/26, 12:57 PM

Миграция из Zephyr Squad

Вы можете автоматически перенести вашу тестовую документацию из Zephyr Squad в Test IT.

Учитывайте форму поставки (облачная или серверная)

Ссылки на скачивание экспортера, параметры выгрузки тестов и команда для запуска экспортера для серверной (Server) и облачной (Cloud) поставок различаются.

В процессе миграции переносятся следующие данные:

- Тест-кейсы
- Шаги тестов
- Вложения тестов*
- Ссылки

---

*В облачной версии Zephyr Squad возможны проблемы с миграцией вложений из шагов тестов.

Миграция протестирована на версиях:

- Серверная поставка: **Jira Server 9.0.0 - 9.14.1**; **Jira Data Center 9.0.0 - 9.14.1**
- Облачная поставка: **Zephyr Squad 1.3.25-AC Cloud**

Миграция тестовой документации включает в себя:

- Подготовку (загрузку экспортера и импортера Test IT)
- Выгрузку тестов из Zephyr Squad
- Загрузку тестов в Test IT с помощью импортера

Подробнее: Миграция в Test IT: возможности и процесс.

Подготовка

1. Скачайте экспортер на странице GitHub Test IT . Выберите файл для вашей версии поставки:
   - Для **серверной** поставки — с названием **ZephyrSquadServerExporter**
   - Для **облачной** поставки — с названием **ZephyrSquadExporter**
     Выберите экспортер для своей операционной системы. Сохраните экспортер в отдельной папке.
2. Скачайте импортер на странице GitHub Test IT. Выберите импортер для своей операционной системы и версии Test IT. Если импортер у вас уже есть (например, вы производили миграцию из других TMS), пропустите этот шаг. Сохраните импортер в отдельной папке.

Выгрузка тестов из Zephyr Squad

Используйте экспортер для своей операционной системы

Для различных операционных систем требуется свой экспортер. Выберите инструкцию по миграции тестов для своей системы в соответствующей вкладке.

macOS & Linux

1. Создайте конфигурационный файл `zephyr.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `zephyr.config.json` . Пример настройки соединения в конфигурационном файле:
   - Для **серверной** поставки вы можете настроить соединение с помощью токена или с помощью логина и пароля.
     - Пример настройки соединения в конфигурационном файле с помощью токена:

```json
{
  "resultPath":
"/Users/user01/Documents/importer",
 "zephyr": {
    "url": "Link to your JIRA instance",
    "token":
"MDc2MjIxNjVzNjg40kJCA43J4AfsIRBXomRs8bKw81+D",
   "projectKey": "PK"
 }
}
```

- Пример настройки соединения в конфигурационном файле с помощью логина и пароля:

```json
{
  "resultPath":
"/Users/user01/Documents/importer",
  "zephyr": {
    "url": "Link to your JIRA instance",
    "login": "user",
    "password": "pass",
    "projectKey": "PK"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `zephyr.url` — URL сервера Zephyr с названием организации
- `zephyr.token` — ключ (токен) доступа на сервер Jira
- `zephyr.login` — ваш логин для входа на сервер Jira
- `zephyr.password` — ваш пароль для входа на сервер Jira
- `zephyr.projectKey` — ключ к проекту на сервер Jira

- Для **облачной** поставки:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "zephyr": {
    "url": "https://prod-api.zephyr4jiracloud.com/",
    "accessKey":
"MmEwMjd4OWYtNGMxMC0zYjhkLWExMmUtNTZjYmY0OTE0MGExIDcxM
    "secretKey": "GkxCBle-7ib1mBOf9Oeiy8HZjJDw4ESnTZzY
    "accountId": "712021:3c9f032d-eeea-4dc0-8622-9b0f6
    "projectId": 10000,
    "projectName": "ProjectName"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `zephyr.url` — URL сервера Zephyr с названием организации
- `zephyr.accessKey` — ключ доступа на сервер Zephyr
- `zephyr.secretKey` — секретный ключ доступа на сервер Zephyr
- `zephyr.accountId` — идентификатор (ID) аккаунта для доступа на сервер Zephyr
- `zephyr.projectId` — идентификатор проекта на сервере Zephyr
- `zephyr.projectName` — название проекта на сервере Zephyr

2. Выдайте разрешение на запуск экспортера на вашем устройстве Mac или Linux.
3. Запустите экспортер с помощью команды:
   - Для **серверной** поставки:

```sh
sudo chmod +x .\ZephyrSquadServerExporter{OS}-
{version_number}
.\ZephyrSquadServerExporter{OS}-{version_number}
```

   - Для **облачной** поставки:

```sh
sudo chmod +x ./ZephyrSquadExporter{OS}-
{version_number}
./ZephyrSquadExporter{OS}-{version_number}
```

В команде вместо `ZephyrSquadServerExporter{OS}-{version_number}` (для серверной версии) или `ZephyrSquadExporter{OS}-{version_number}` (для облачной версии) укажите имя файла экспортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию экспортера.
Дождитесь окончания экспорта.

4. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Windows

1. Создайте конфигурационный файл `zephyr.config.json` и сохраните его в папке с экспортером. Настройте соединение в файле `zephyr.config.json`. Пример настройки соединения в конфигурационном файле:
   - Для **серверной** поставки вы можете настроить соединение с помощью токена или с помощью логина и пароля.
     - Пример настройки соединения в конфигурационном файле с помощью токена:

```json
{
  "resultPath":
"/Users/user01/Documents/importer",
 "zephyr": {
    "url": "Link to your JIRA instance",
    "token":
"MDc2MjIxNjVzNjg40kJCA43J4AfsIRBXomRs8bKw81+D",
   "projectKey": "PK"
 }
}
```

     - Пример настройки соединения в конфигурационном файле с помощью логина и пароля:

```json
{
  "resultPath":
"/Users/user01/Documents/importer",
  "zephyr": {
    "url": "Link to your JIRA instance",
    "login": "user",
    "password": "pass",
    "projectKey": "PK"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `zephyr.url` — URL сервера Zephyr с названием организации
- `zephyr.token` — ключ (токен) доступа на сервер Jira
- `zephyr.login` — ваш логин для входа на сервер Jira
- `zephyr.password` — ваш пароль для входа на сервер Jira
- `zephyr.projectKey` — ключ к проекту на сервер Jira

- Для **облачной** поставки:

```json
{
  "resultPath": "/Users/user01/Documents/importer",
  "zephyr": {
    "url": "https://prod-api.zephyr4jiracloud.com/",
    "accessKey":
"MmEwMjd4OWYtNGMxMC0zYjhkLWExMmUtNTZjYmY0OTE0MGExIDcxM
    "secretKey": "GkxCBle-7ib1mBOf9Oeiy8HZjJDw4ESnTZzY
    "accountId": "712021:3c9f032d-eeea-4dc0-8622-9b0f6
    "projectId": 10000,
    "projectName": "ProjectName"
  }
}
```

Значения параметров:

- `resultPath` — директория папки, в которой будут сохранены файлы с тестовой документацией
- `zephyr.url` — URL сервера Zephyr с названием организации
- `zephyr.accessKey` — ключ доступа на сервер Zephyr

- `zephyr.secretKey` — секретный ключ доступа на сервер Zephyr
- `zephyr.accountId` — идентификатор (ID) аккаунта для доступа на сервер Zephyr
- `zephyr.projectId` — идентификатор проекта на сервере Zephyr
- `zephyr.projectName` — название проекта на сервере Zephyr

2. Запустите `.exe` -файл экспортера.
3. По окончании экспорта убедитесь, что экспортируемые файлы находятся в директории, указанной в параметре `resultPath`. В случае ошибки миграции устраните проблему самостоятельно или свяжитесь с технической поддержкой (support@testit.software).

Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 7/7/26, 2:09 PM

Загрузка тест-кейсов в Test IT с помощью импортера

Перед импортом используйте экспортер

Чтобы загрузить тесты с помощью импортера, вначале экспортируйте файлы с тестовой документацией из сторонней TMS-системы.

После того, как вы выгрузили тестовую документацию из сторонней системы управления тестированием (TMS) с помощью экспортера, нужно загрузить ее в Test IT. Для этого применяется импортер — утилита для переноса файлов с тестовой документацией из любых сторонних TMS-систем в Test IT.

- Импортер доступен для скачивания на странице GitHub Test IT.

Используйте импортер для своей операционной системы и версии Test IT

Для различных операционных систем и версий Test IT требуется свой импортер. Если вы используете Test IT Про версии 5.2 или более ранней, скачайте импортер на странице Migrators .

macOs & Linux

Чтобы загрузить тестовую документацию в Test IT с помощью импортера:

1. Создайте конфигурационный файл `tms.config.json` и сохраните его в папке с импортером. Настройте соединение в файле `tms.config.json` .
   Пример настройки конфигурационного файла:

```json
{
    "resultPath" : "/Users/user01/Documents/importer",
    "tms" : {
        "url" : "https://testit.software/",
        "privateToken" : "cmZzWDkYTfBvNvVMcXhzN3Vy",
        "certValidation" : true
    }
}
```

Значения параметров:

- resultPath — директория папки, в которой будут сохранены файлы с тестовой документацией
- tms.url — URL вашего сервера Test IT
- tms.privateToken — токен для доступа на сервер Test IT
- tms.certValidation — разрешение / запрет на валидацию сертификатов

2. При необходимости выдайте разрешение на запуск импортера на вашем устройстве Mac или Linux.
3. Запустите импортер с помощью команды:

```sh
sudo chmod +x ./Importer{OS}-{version_number}
./Importer{OS}-{version_number}
```

В команде вместо `Importer{OS}-{version_number}` укажите имя файла импортера, скачанного с Git. Как правило, имя файла содержит название операционной системы и версию импортера.
Дождитесь окончания импорта. По окончании импорта тестовая документация будет перенесена в Test IT.

Windows

Чтобы загрузить тестовую документацию в Test IT с помощью импортера:

1. Создайте конфигурационный файл `tms.config.json` и сохраните его в папке с импортером. Настройте соединение в файле `tms.config.json` .
Пример настройки конфигурационного файла:

```json
{
    "resultPath" : "/Users/user01/Documents/importer",
    "tms" : {
        "url" : "https://testit.software/",
        "privateToken" : "cmZzWDkYTfBvNvVMcXhzN3Vy",
        "certValidation" : true
    }
}
```

Значения параметров:

- resultPath — директория папки, в которой будут сохранены файлы с тестовой документацией
- tms.url — URL вашего сервера Test IT
- tms.privateToken — токен для доступа на сервер Test IT
- tms.certValidation — разрешение / запрет на валидацию сертификатов

2. Запустите `.exe` -файл экспортера и дождитесь окончания импорта. По окончании импорта тестовая документация будет перенесена в Test IT.

Обновлено: 4/15/26, 7:08 PM

Работа с дашбордами

Тарифные ограничения

В тарифе **Базовый** доступен 1 дашборд с 3 виджетами. В тарифах **Стандарт** и **Про** дашборды и виджеты создаются без ограничений.
Подробнее: о тарифах Test IT .

- **Для этого раздела мы подготовили видео:**

Вы можете отслеживать работу на проекте с помощью дашбордов. Test IT позволяет использовать два вида дашбордов:

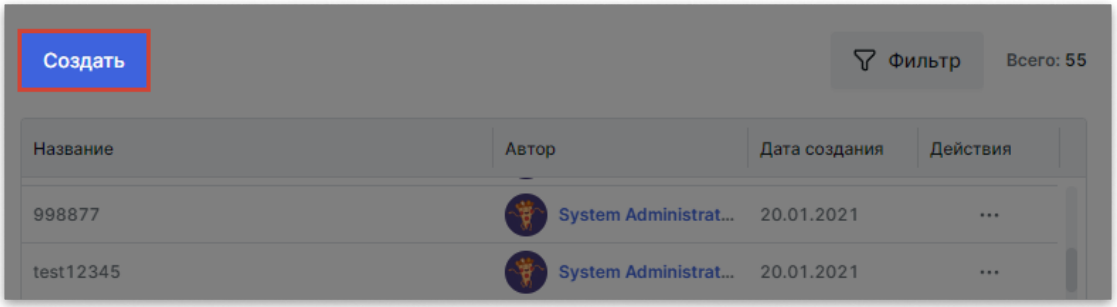
- Пользовательские дашборды, создаваемые пользователем проекта с соответствующими правами доступа. Данный вид дашбордов доступен для просмотра только их создателю и пользователям, которым создатель выдает доступ к ним.
- Проектные дашборды, создаваемые автоматически вместе с проектом. Данный тип дашбордов доступен всем пользователям проекта.

Дашборды позволяют визуализировать данные аналитики по проекту и настраивать формат их отображения с помощью виджетов. Вы можете настроить неограниченное количество виджетов для отслеживания необходимых вам метрик, а также фильтровать данные внутри самих виджетов по времени, проекту, секциям и пользовательским атрибутам.

Работа с пользовательскими дашбордами

Test IT позволяет создавать пользовательские дашборды, в которых вы можете группировать и визуализировать данные аналитики. Вы можете давать доступ к созданному вами дашборду как конкретным пользователям, так и группам пользователей, добавленным в систему Test IT. Чтобы создать дашборд:

1. Находясь на главной странице, перейдите в раздел **Дашборды**.
2. Нажмите **Создать**.



3. Введите название дашборда.
4. В шагах **Группы** и **Пользователи** выберите пользователей и/или группы пользователей, которым хотите дать доступ к вашему дашборду.
5. Нажмите **Сохранить**.

После сохранения дашборд отобразится в общей таблице. Воспользуйтесь фильтром, чтобы отсортировать дашборды. Вы можете редактировать и удалять дашборды, а также копировать ссылки на них с помощью меню в колонке **Действия**.

Чтобы визуализировать данные, вам необходимо создать в дашборде виджеты.

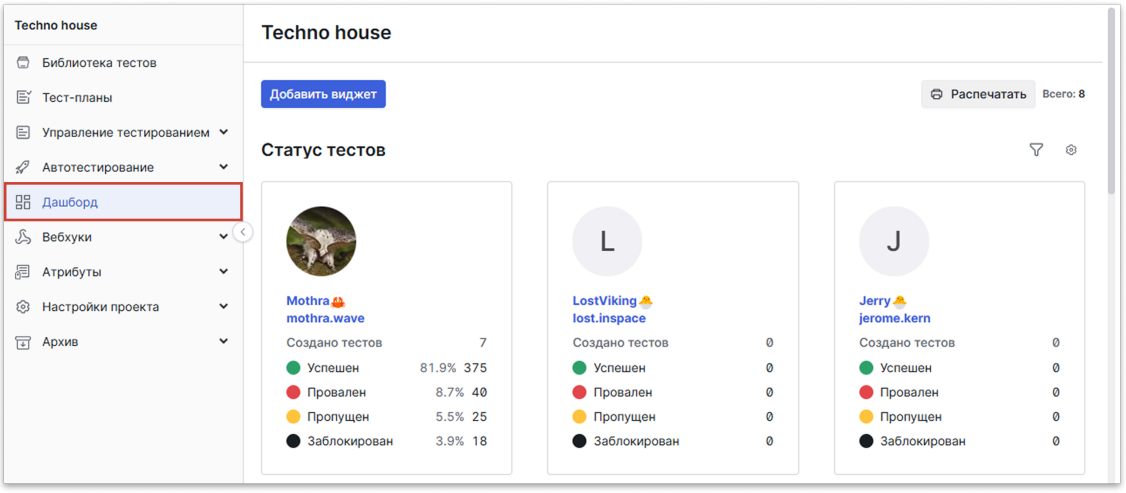
Работа с проектными дашбордами

В каждом из созданных вами проектов автоматически создается дашборд, позволяющий всем пользователям на проекте отслеживать аналитические данные по нему.

Чтобы перейти к проектному дашборду:

1. Откройте проект.

2. Перейдите в раздел **Дашборд**.



В данном разделе по умолчанию собраны следующие виджеты:

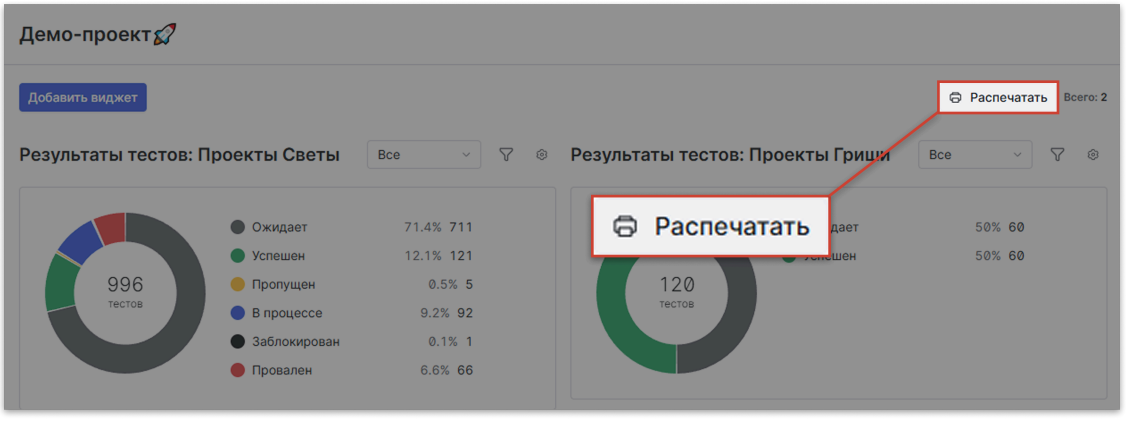
- Тренды результатов по прохождению тестов
- Распределение тестов по приоритетам
- Распределение тестов по статусу готовности
- Распределение тестов по типу тестирования
- Результаты тест-планов
- Команда
- Таблица результатов по секциям

Чтобы отображать больше данных, создавайте собственные виджеты, где можете визуализировать и группировать данные по вашему проекту.

Печать дашбордов

Чтобы распечатать дашборд:

1. Откройте дашборд.
2. Нажмите **Распечатать**.



Форма для печати дашборда откроется в новой вкладке.

3. Распечатайте дашборд или сохраните его на вашем устройстве.

Обновлено: 8/12/26, 4:43 PM

Действия с виджетами

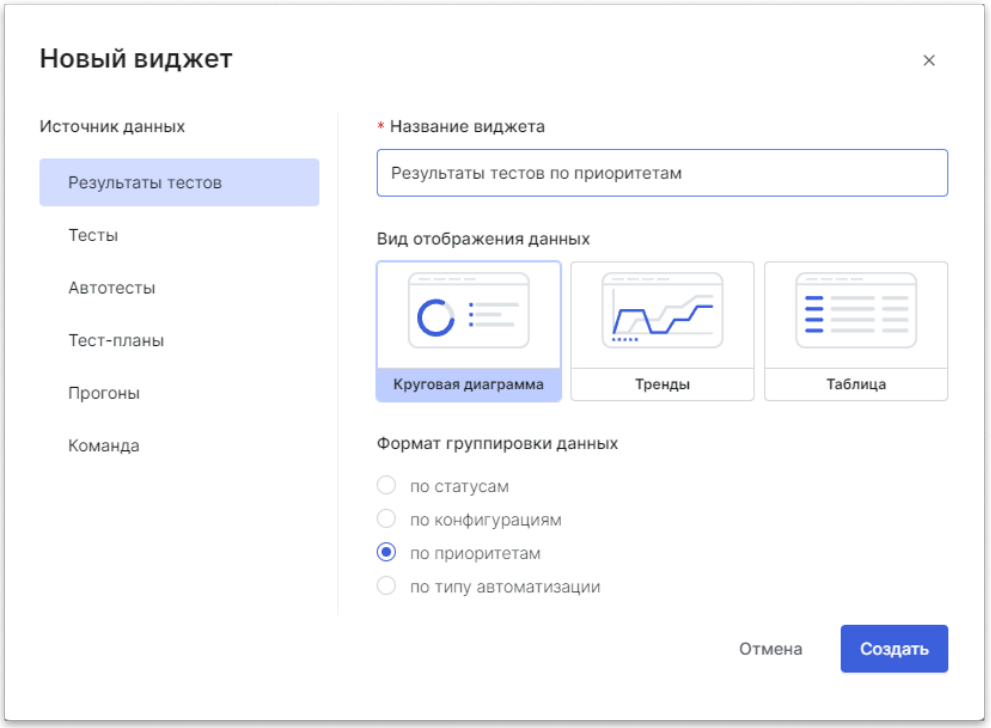
Виджеты позволяют визуализировать данные аналитики в дашбордах. Вы можете создавать неограниченное количество виджетов в дашборде. Настраивайте формат и тип отображаемых данных, а также способ их группировки.

С помощью виджетов вы можете визуализировать следующие аналитические данные:

- Результаты тестов
- Тесты
- Автотесты
- Тест-планы
- Прогоны
- Активность команды

Создание виджетов

1. Откройте дашборд.
2. Нажмите **Создать**, если дашборд пуст, или **Добавить виджет**, если в дашборде уже есть виджеты.
3. Заполните данные виджета:
    - Источник данных (информацию, которую вы хотите визуализировать)
    - **Опционально:** Измените название виджета (по умолчанию задается автоматически в зависимости от выбранных данных)
    - Вид отображения данных: *Круговая диаграмма*, *Тренды*, *Линейчатая диаграмма*, *Таблица*, *Команда*, *Таймлайн* (для некоторых типов визуализации некоторые типы отображаемых данных могут быть недоступны)
    - Формат группировки данных
4. Нажмите **Создать**.

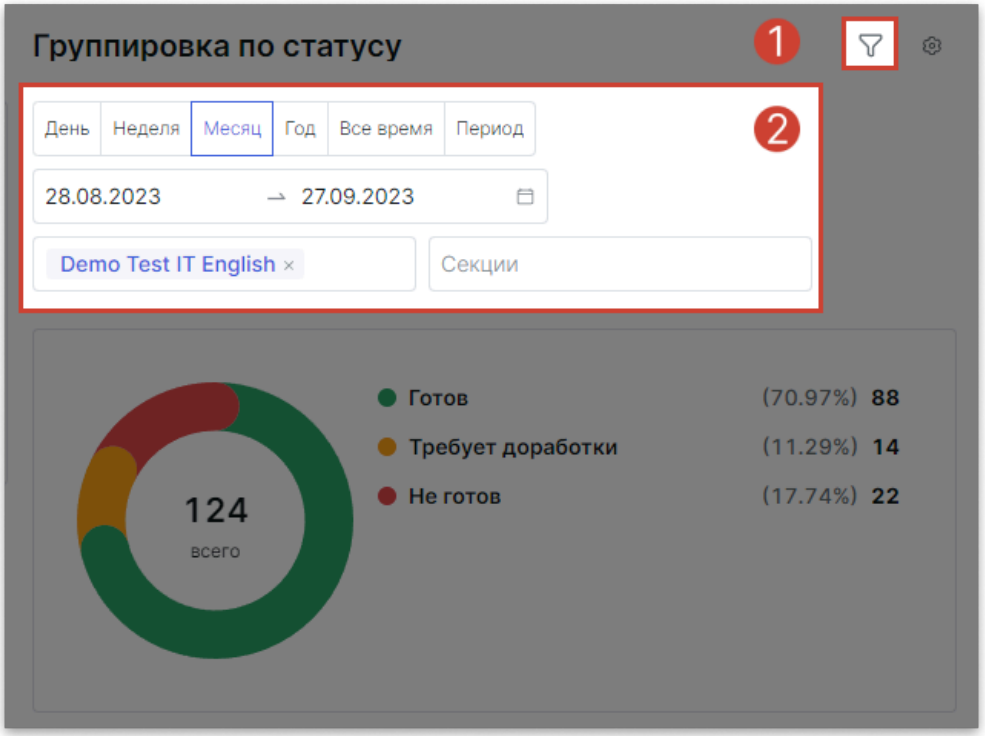


Новый виджет появится на вашем дашборде.

Фильтр данных

Вы можете воспользоваться фильтром данных, отображаемых в виджете, чтобы собрать аналитику за период времени, а также отфильтровать данные виджета по проекту, секции, приоритету, тест-плану, и глобальным атрибутам. Для этого:

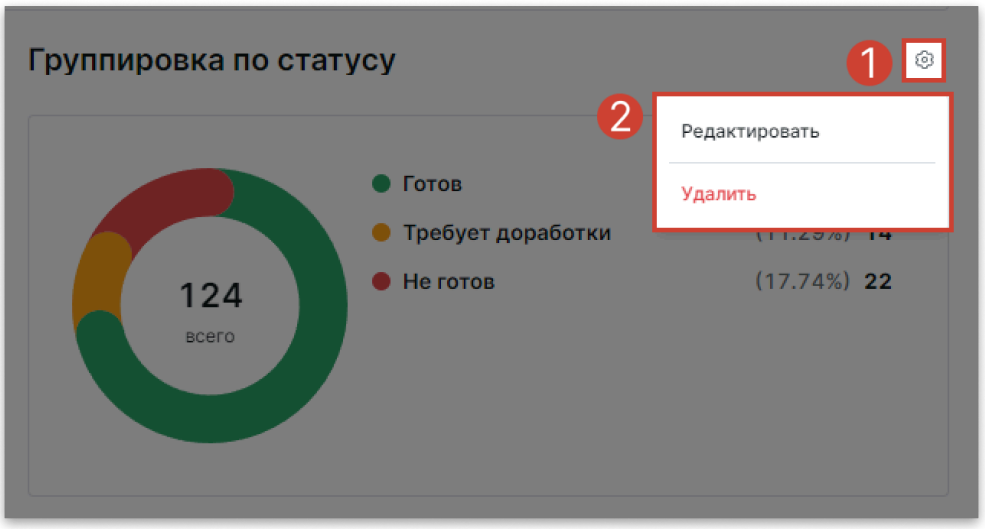
1. Нажмите значок **Фильтр** в правом верхнем углу виджета.
2. В открывшемся фильтре выберите требуемые значения.



Редактирование и удаление виджетов

Чтобы открыть окно редактирования виджета или удалить виджет:

1. Откройте настройки виджета.
2. В открывшемся меню выберите **Редактировать** или **Удалить**.



3. В зависимости от выбранной опции продолжите редактирование виджета или подтвердите удаление.

Обновлено: 4/15/26, 7:08 PM

Анализ результатов тестов

Вы можете отслеживать информацию о прохождении тестов. Тесты без прохождений не учитываются. Для результатов тестов доступны следующие способы группировки данных:

- Группировка по статусу
- Группировка по конфигурациям
- Группировка по приоритету
- Группировка по типу автоматизации
- Группировка по специалистам
- Группировка по тест-поинтам

Группировка по статусу

Вы можете визуализировать аналитические данные по результатам тестов, распределяя их по статусам готовности. Вы можете отобразить аналитику в форматах **круговой диаграммы** и **трендов**.

Группировка по конфигурациям

Вы можете визуализировать аналитические данные по тестам с проставленным результатом, распределяя их по конфигурациям, для которых были проставлены результаты. Архивные конфигурации также отображаются, если они использовались при прохождении теста. Вы можете отобразить аналитику в форматах **круговой диаграммы** и **трендов**.

Группировка по приоритету

Вы можете визуализировать аналитические данные по тестам с проставленным результатом, распределяя их по присвоенным им приоритетам. У автотеста, который не привязан к тест-кейсу, нет приоритета. Результаты автотестов, включенные в прогон (тест-ран) методом `POST /api/v2/testRuns/byAutoTests` , не будут отображаться на

этой диаграмме. Для отображения таких автотестов в диаграмме используйте для создания прогона методы `/api/v2/testRuns/byConfigurations`, `/api/v2/testRuns/byWorkItems` или графический интерфейс. Вы можете отобразить аналитику в форматах **круговой диаграммы** и **трендов**.

Группировка по типу автоматизации

Вы можете визуализировать аналитические данные по тестам с проставленным результатом, разделяя их на автоматизированные и ручные. Если к тест-кейсу привязан автотест, тест-кейс считается автоматизированным. Вы можете отобразить аналитику в форматах **круговой диаграммы** и **трендов**.

Группировка по специалистам

Вы можете визуализировать аналитические данные по тестам с проставленным результатом, распределяя их по специалистам, проставившим результаты. Вы можете отобразить аналитику в формате **трендов**.

Группировка по тест-поинтам

Вы можете визуализировать аналитические данные по тестам с проставленным результатом, распределяя их по секциям библиотеки тестов. Вы можете отобразить аналитику в формате **таблицы**.

Обновлено: 4/15/26, 7:08 PM

Анализ тестов, хранящихся в библиотеке

Вы можете визуализировать и отслеживать информацию о тестах, хранящихся в библиотеке. Для тестов доступны следующие возможности:

- Группировка по статусу
- Группировка по приоритету
- Группировка по типу автоматизации
- Группировка по авторам
- Анализ покрытия тестов ссылками

Группировка по статусу

Вы можете визуализировать аналитические данные по тестам в библиотеке, распределяя их по статусам готовности. Вы можете отобразить аналитику в форматах **круговой диаграммы** или **трендов**.

Группировка по приоритету

Вы можете визуализировать аналитические данные по тестам в библиотеке, распределяя их по присвоенным им приоритетам. Данный формат группировки отображает автотесты только если они привязаны к тест-кейсу. Вы можете отобразить аналитику в форматах **круговой диаграммы** или **трендов**.

Группировка по типу автоматизации

Вы можете визуализировать аналитические данные по тестам в библиотеке, разделяя их на автоматизированные и ручные. Если к тест-кейсу привязан автотест, тест-кейс считается автоматизированным. Вы можете отобразить аналитику в форматах **круговой диаграммы**, **трендов** или **таблицы**. Данный тип группировки позволяет отслеживать процент автоматизированных тестов от общего количества тестов в секциях библиотеки.

Группировка по авторам

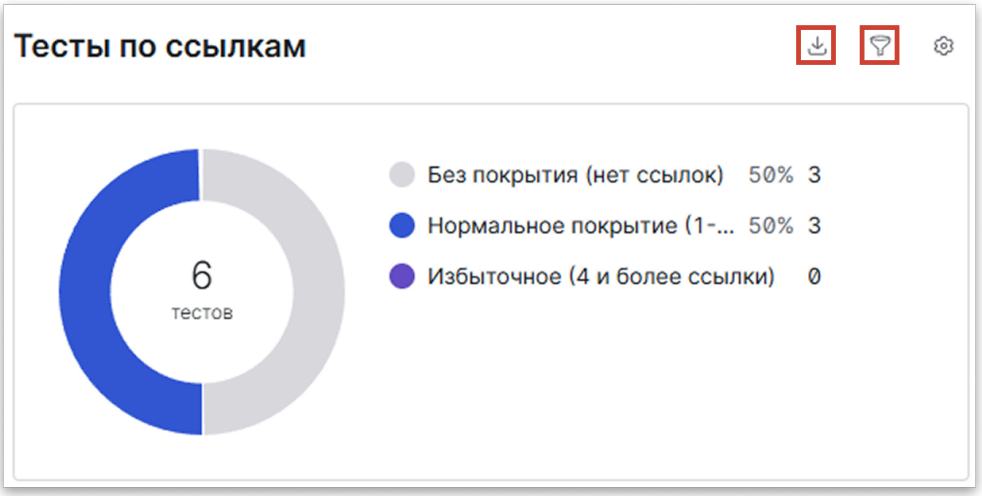
Вы можете визуализировать аналитические данные по тестам в библиотеке, распределяя их по создателям. Вы можете отобразить аналитику в формате **круговой диаграммы**.

Анализ покрытия тестов ссылками

Вы можете просматривать и выгружать информацию о покрытии тестов ссылками. Виджет содержит данные по тестам из библиотеки, к которым прикреплены ссылки различного типа — из Test IT или вашего таск-трекера.
Данные отображаются:

- В виде круговой диаграммы — общая информация
- В виде таблицы — полная информация, доступно для скачивания в формате `.xlsx`. Чтобы получить данные о покрытии тестов:

1. Создайте виджет **Тесты**. В секции **Формат группировки данных** выберите **по ссылкам**.
2. **Опционально:** Выберите данные для отображения, нажав значок **Фильтр** в правой верхней части виджета. используя выпадающие меню: **Проект**, **Секция**, **Тип теста** и др. По умолчанию отображаются все данные по вашим проектам Test IT.
   Отобразится круговая диаграмма с данными.
3. **Опционально:** Чтобы загрузить полную информацию по вашим критериям в виде таблицы, нажмите **Скачать** в правой верхней части виджета.



Загрузка `.xlsx`-файла начнется в фоновом режиме. По окончании загрузки файл будет доступен в Фоновых задачах.

Пример из практики

Виджет **Тесты по ссылкам** можно использовать как инструмент трассировки требований. Каждый функциональный тест, выполняемый командой, должен содержать 1 ссылку на требования (тип ссылки в Test IT — **Требование**).
Чтобы визуализировать данные о покрытии:

1. Создайте тег `функциональное тестирование`.
2. Присваивайте его соответствующим тест-кейсам (дымовые тесты, юнит-тесты и т.д.) в процессе подготовки.
3. Создайте виджет **Покрытие тестов ссылками**.
4. Выберите данные для отображения:
   - Тег **функциональное тестирование**
   - Тип ссылки в TI: **Требование**
   - Другие критерии для отображения данных
     Данные отобразятся на диаграмме и будут доступны для скачивания.

Обновлено: 5/12/26, 8:25 AM

Анализ автотестов

Вы можете визуализировать и отслеживать аналитические данные по работе автотестов. Для автотестов доступны следующие способы группировки данных:

- Группировка по стабильности
- Группировка по времени прохождения

Группировка по стабильности

Вы можете визуализировать аналитику по автотестам, разделяя их на стабильные и нестабильные. Вы можете отобразить аналитику в форматах **круговой диаграммы** или **трендов**.

Группировка по продолжительности

Вы можете визуализировать аналитические данные по времени прохождения автотестов. Время прохождения рассчитывается как медиана (значение, находящееся в середине числового ряда) времени прохождения от всех завершенных запусков. Вы можете отбирать тесты по медианной продолжительности, уровню стабильности, конфигурации, проекту, тест-плану, прогону, лейблам, категории ошибок и результату (для автотестов в статусе *Успешен*, *Пропущен*, *Заблокирован* и *Провален*). Доступный формат отображения — **таблица**.

Обновлено: 4/15/26, 7:08 PM

Анализ прогонов

Вы можете визуализировать и отслеживать аналитические данные по прогонам (тест-ранам) автотестов. Для прогонов доступны следующие способы группировки данных:

- Группировка по статусу
- Группировка по конфигурации
- Группировка по специалисту
- Группировка по причине результата
- Таймлайн

Группировка по статусу

Вы можете визуализировать аналитические данные по прогонам (прогонам автотестов), распределяя их по статусам прохождения. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Группировка по конфигурации

Вы можете визуализировать аналитические данные по прогонам, распределяя их по конфигурациям, на которых они были запущены. Архивные конфигурации также отображаются, если они использовались при прохождении теста. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Группировка по специалисту

Вы можете визуализировать аналитические данные по прогонам, распределяя их по специалистам, инициировавшим запуск. Если тест-поинт не был назначен, результат отобразится в виде **Нет тестировщика**. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Группировка по причине результата

Вы можете визуализировать аналитические данные по прогонам, распределяя их по причинам результатов и категориям ошибок. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Таймлайн

Визуализируйте прогоны в разрезе времени. Чтобы посмотреть детальную информацию о прогоне, наведите на него курсор. Вы можете отобразить аналитику в формате **таймлайна**.

Обновлено: 4/15/26, 7:08 PM

Анализ тест-планов

Вы можете визуализировать и отслеживать аналитические данные по тест-планам. Для тест-планов доступны следующие способы группировки данных:

- Группировка по статусу
- Группировка по конфигурациям
- Группировка по специалистам
- Группировка по типу автоматизации

Группировка по статусу

Вы можете визуализировать аналитические данные по тест-планам, распределяя их по статусам прохождения. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Группировка по конфигурациям

При такой группировке вы можете визуализировать аналитические данные по прогонам, распределяя их по конфигурациям, на которых они были запущены. Архивные конфигурации также отображаются, если они использовались при прохождении теста. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Группировка по специалистам

Вы можете визуализировать аналитические данные по прогонам, распределяя их по специалистам, инициировавшим запуск. Если тест-поинт не был назначен, результат отобразится в виде **Нет тестировщика**. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Группировка по типу автоматизации

Вы можете визуализировать аналитические данные по прогонам автотестов, распределяя их по способу проставления результатов тестов. Результаты, проставленные через пользовательский графический интерфейс, считаются ручными прохождениями, результаты, проставленные через API — автоматизированными прохождениями. Статус автоматизации теста при этом значения не имеет. Вы можете отобразить аналитику в формате **линейчатой диаграммы**.

Обновлено: 4/15/26, 7:08 PM

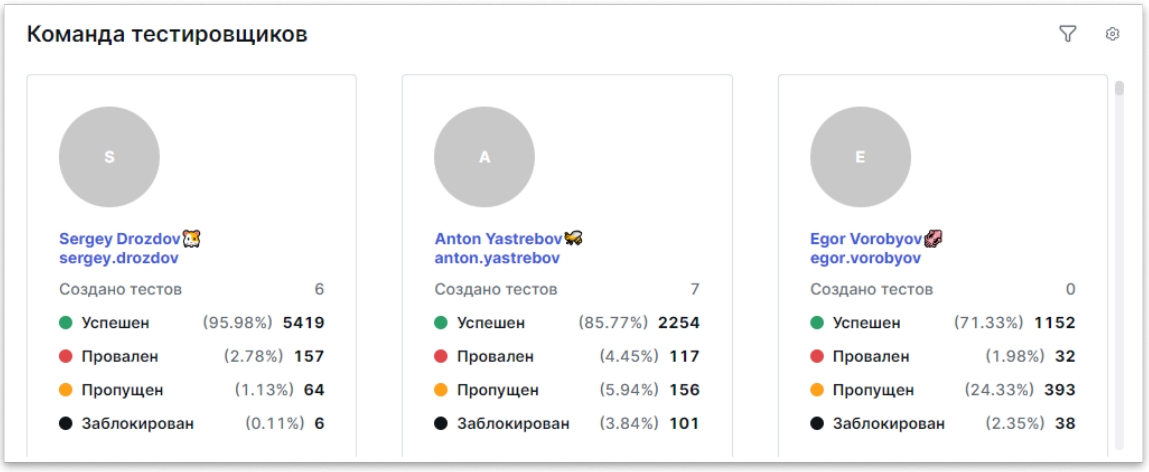
Анализ активности команды

Вы можете визуализировать и отслеживать общее количество созданных пользователями тестов и количество пройденных тестов с фильтром по проекту. С помощью данного виджета вы можете быстро оценить нагрузку ваших тестировщиков.

За работу над проектами тестировщикам начисляются баллы:

- За создание теста — 30 баллов
- За пройденный тест-поинт — 10 баллов
- За проваленный тест-поинт — 20 баллов

Тестировщики ранжируются по общему количеству набранных баллов в порядке убывания.



Количество созданных пользователями тестов учитывается из всех проектов независимо от примененного фильтра по проектам, а также удаленные тесты. Скопированные пользователем тесты не считаются созданными.

В данном виджете доступна группировка данных по **Специалистам**. Вы можете отобразить аналитические данные в формате **Команды**.

Обновлено: 4/15/26, 7:08 PM

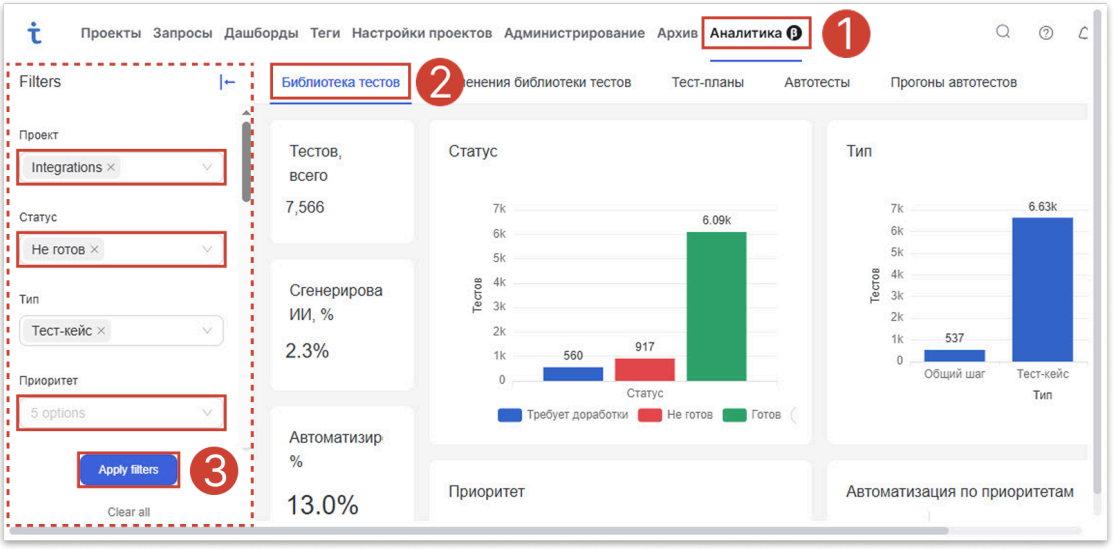
Аналитика+

> **Тарифные ограничения**
>
> Опция доступна в платных тарифах. В тарифе Стандарт она приобретается в качестве дополнительного модуля. Подробнее: о тарифах Test IT .

Вы можете получать подробную отчетность о результатах вашей работы в Test IT в разделе **Аналитика+**. Аналитика Test IT работает на базе сервиса Apache Superset. Данные представлены на виджетах.

Просмотр аналитики

1. В верхней навигационной панели откройте раздел **Аналитика+**.
2. В верхней части окна аналитических данных выберите нужную вкладку (например, **Библиотека тестов** или **Тест-планы**).
3. Чтобы выбрать нужные данные (например, по проекту, периоду и т.д.), слева в области фильтров выберите нужные фильтры и нажмите **Apply filters**. На всех виджетах отобразятся данные по выбранным критериям.



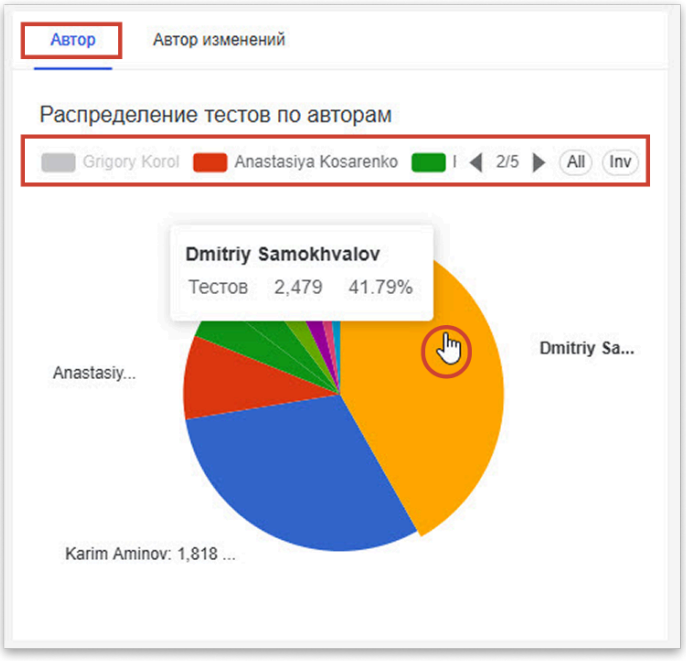
4. **Опционально:** Воспользуйтесь интерактивными виджетами для более точного отбора данных.

Работа с виджетами раздела Аналитика+

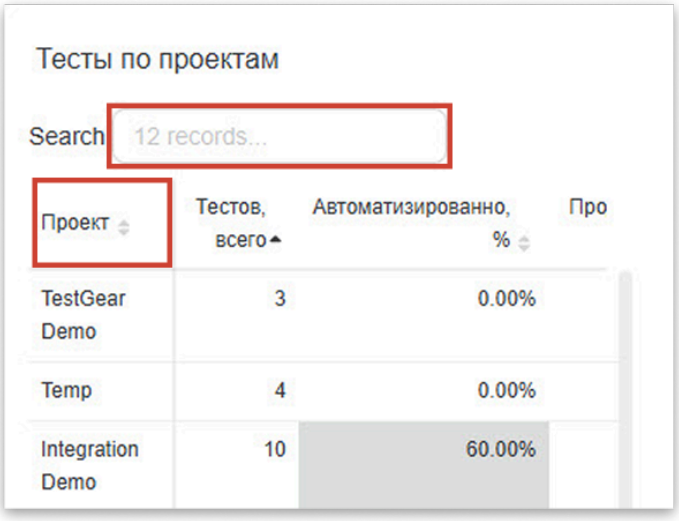
Данные раздела **Аналитика+** представлены на виджетах в таблицах, диаграммах и в определенных значениях. Многие виджеты интерактивны.

В зависимости от типа виджета вы можете:

- Отобразить или скрыть группы данных, нажав на них в легенде диаграммы
- Просмотреть детализированные данные по отдельной метрике, наведя курсор на столбец/сегмент диаграммы
- Если виджет содержит несколько вкладок по типам данных: переключаться между вкладками
- Если легенда диаграммы содержит много значений: пролистывать значения легенды
- Показать все значения — **All** или обратные значения **Inv**



- Воспользоваться поиском — полем **Search** внутри виджета
- Сортировать данные по столбцам таблиц. Для сортировки по нескольким столбцам используется сочетание Shift + Click



.

Обновлено: 10/29/26, 1:42 PM

Настройка аналитических данных

- Вход в Apache Superset
- Управление пользователями
- Копирование дашборда
- Редактирование дашборда
- Добавление дашборда
  - Получение идентификатора дашборда в Apache Superset
  - Добавление дашборда Apache Superset в Test IT с помощью идентификатора
  - Добавление дашборда Apache Superset в Test IT с помощью SDK
- Редактирование виджета
- Подключение сторонних источников данных через Airbyte
- Сохранение и замена пользовательских настроек при обновлении Test IT
- Рекомендации по настройке аналитических данных

> **Опция тарифов Про**
>
> Эта опция доступна в тарифах Про и Про Макс (серверная поставка). Подробнее: о тарифах Test IT .

Вы можете настроить дашборды, их вкладки и виджеты раздела **Аналитика+**. Настройки производятся в сервисе Apache Superset, связанном с вашим доменом Test IT.

В разделе **Аналитика+** Test IT может отображаться только 1 дашборд, при этом в Apache Superset можно создать несколько. После установки Test IT отображается дашборд по умолчанию. Настройки дашбордов производятся в сервисе Apache Superset, связанном с вашим доменом Test IT.

Данные для дашбордов поступают по ETL-процессу: репликация из рабочей базы Test IT (PostgreSQL) с помощью Airbyte передается в аналитическую базу ClickHouse. Дашборд Superset подключен к базе ClickHouse. ETL-процессы настраиваются в Airbyte.

Вы можете:

- Копировать, править и заменять отображаемые дашборды в Apache Superset
- Настраивать ETL-процессы, добавляя свои интеграции в Airbyte Производя настройки, учитывайте Рекомендации по настройке аналитических данных.

Смотрите также:

- Документация Apache Superset
- Документация Airbyte

Вход в Apache Superset

Создание, копирование и редактирование дашбордов и виджетов (charts) осуществляется в Apache Superset. Вход в него осуществляется через URL вашей платформы Test IT.

1. Откройте страницу Apache Superset, используя ваш домен Test IT с постфиксом `/superset/`. Например: `https://my-domain.testit.ru/superset/`.
   Откроется окно авторизации.
2. Введите логин и пароль для Apache Superset. По умолчанию логин — `admin`; пароль — `password`.
   Откроется главная страница Apache Superset (по умолчанию — вкладка **Dashboards**).

Управление пользователями

Изменить пароль и добавить новых пользователей можно внутри Apache Superset .

Не удаляйте пользователей TMS User и Superset Admin

Не удаляйте пользователей **TMS User** и **Superset Admin**.

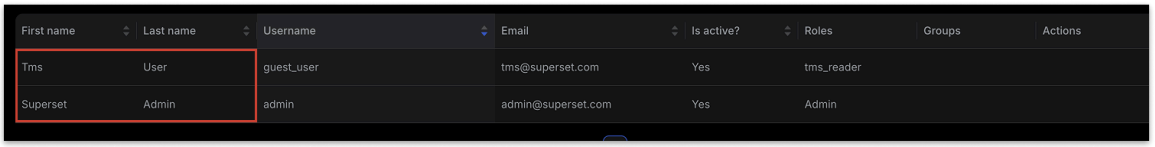
Если эти пользователи были удалены по ошибке, примените команду установки Test IT:

- В Docker Compose:

```sh
docker compose -f docker-compose.etl.yml up -d
```

- В Kubernetes:

```sh
helm upgrade --install -f testit_unichart/values-override.yaml -n <my-namespace> --create-namespace testit testit_unichart/ --wait --timeout 20m
```



Копирование дашборда

Особенность копирования дашборда

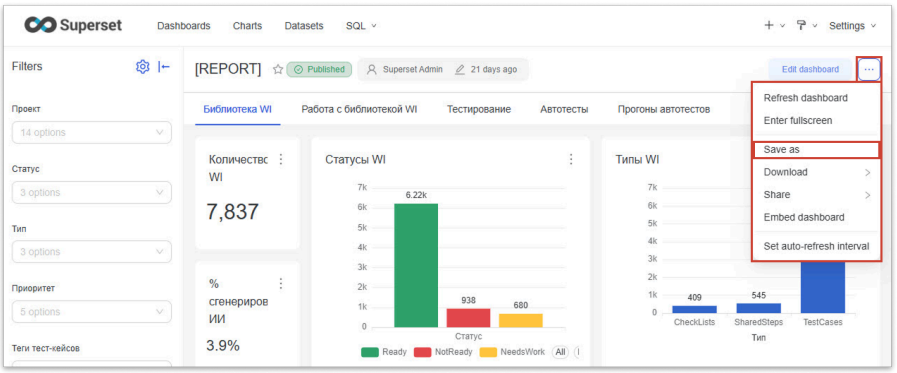
Дашборд, который вы скопируете, будет подключен к той же базе ClickHouse, что и оригинальный дашборд.

Если вы хотите изменить / добавить виджеты, изменить SQL-запросы, при этом сохранив ваш дашборд по умолчанию, структуру базы данных и ETL-процессы, вы можете скопировать дашборд и сделать нужные правки.

Чтобы скопировать дашборд:

1. Войдите в Apache Superset.
2. Во вкладке **Dashboards** в списке дашбордов нажмите на нужный дашборд.

3. В правом верхнем углу откройте меню действий (значок [⋯]) и выберите *(Save as)*.



4. **Опционально:** Укажите название нового дашборда (по умолчанию будет использовано прежнее название с добавлением **[copy]**).
5. **Опционально:** Чтобы скопировать виджеты, выберите **also copy (duplicate) charts**. При этом наборы данных для оригинальных и скопированных виджетов останутся общими (вы можете переключить их вручную).
6. Нажмите **Save**.
   Новый дашборд появится в списке.

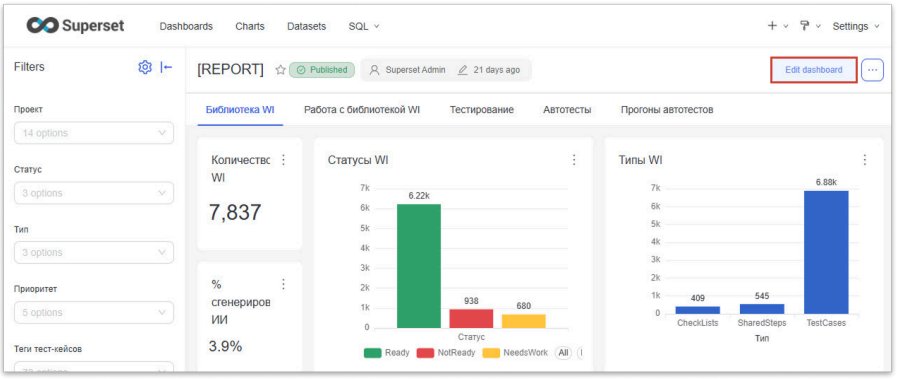
Редактирование дашборда

> **Риск потери данных при обновлении Test IT**
>
> При обновлении Test IT все изменения в виджетах и датасетах по умолчанию (“из коробки“) теряются. Чтобы вносить свои изменения, нужно создавать свои виджеты и датасеты или копировать существующие.

Если вы хотите изменить / добавить виджеты, изменить SQL-запросы, при этом сохранив структуру базы данных и ETL-процессы, вы можете отредактировать дашборд.

Чтобы отредактировать дашборд:

1. Войдите в Apache Superset.
2. Во вкладке **Dashboards** в списке дашбордов нажмите на нужный дашборд.
3. В правом верхнем углу нажмите **Edit dashboard**.



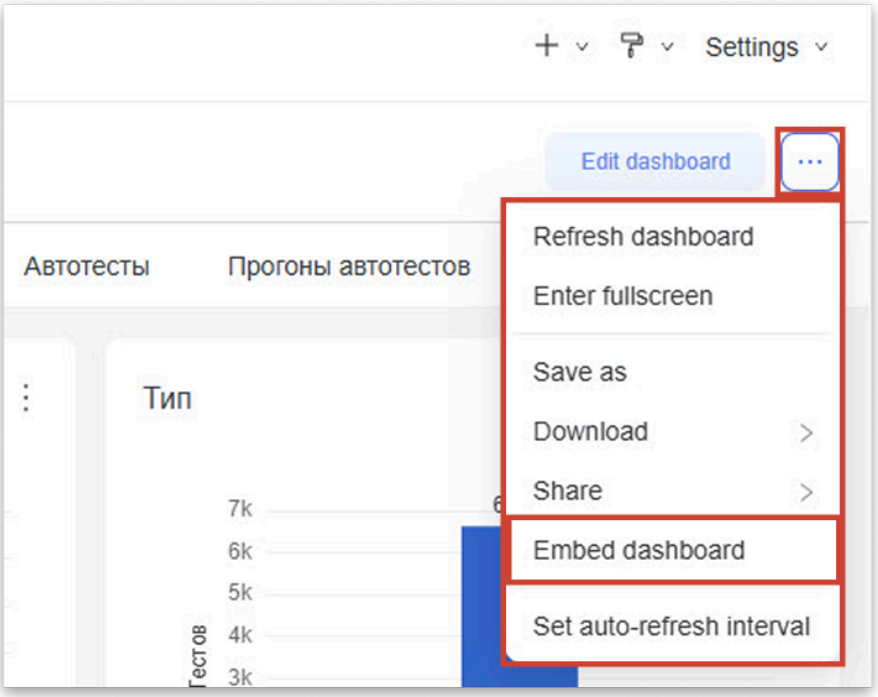
4. Внесите изменения.
5. **Опционально:** Чтобы вернуться на страницу дашборда без сохранения, нажмите **Discard**.
6. Сохраните изменения, нажав **Save**.
   Новый дашборд появится в списке.

Добавление дашборда

В Apache Superset вы можете создать несколько дашбордов, при этом в разделе **Аналитика+** Test IT может отображаться только 1 дашборд. Чтобы вставить дашборд, необходимо получить его идентификатор в Apache Superset и внести в Test IT в разделе **Администрирование**.

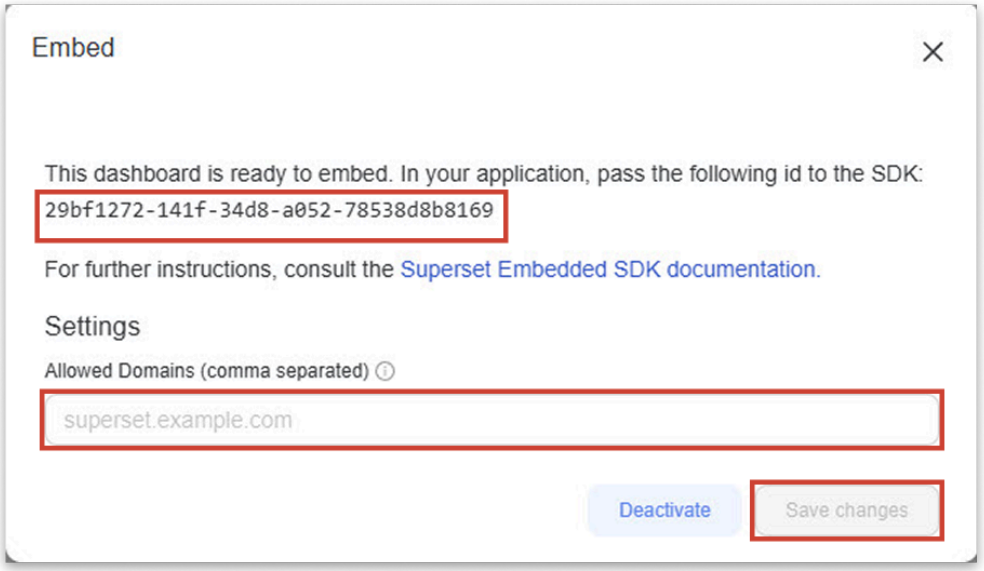
Получение идентификатора дашборда в Apache Superset

1. Откройте Apache Superset.
2. Во вкладке **Dashboards** выберите нужный дашборд и откройте его.
3. В правом верхнем углу окна нажмите значок действий и выберите **Embed Dashboard** (справа от кнопки **Edit dashboard**).



Откроется окно встраивания дашборда.

4. Скопируйте идентификатор дашборда.



Добавление дашборда Apache Superset в Test IT с помощью идентификатора

1. Откройте Test IT, затем в верхней навигационной панели нажмите **Администрирование**.
2. В левой панели нажмите **Системные настройки**.
3. В поле **ID дашборда** вставьте идентификатор дашборда, скопированный в Apache Superset.
4. Нажмите **Сохранить**.

Добавление дашборда Apache Superset в Test IT с помощью SDK

Вы можете добавить нужную версию дашборда с помощью SDK (Software Development Kit — набор средств разработки ПО).

Для этого:

1. Установите SDK с помощью команды:

```sh
npm install @superset-ui/embedded-sdk
```

2. Встройте дашборд:

```sh
import { embedDashboard } from '@superset-ui/embedded-sdk';

embedDashboard({
  id: 'dashboard-uuid-here', // from Dashboard → Embed
  supersetDomain: 'https://superset.example.com',
  mountPoint: document.getElementById('superset-container'),
  fetchGuestToken: () => fetchTokenFromYourBackend(),
  dashboardUiConfig: {
    hideTitle: true,
    filters: { expanded: false },
  },
});
```

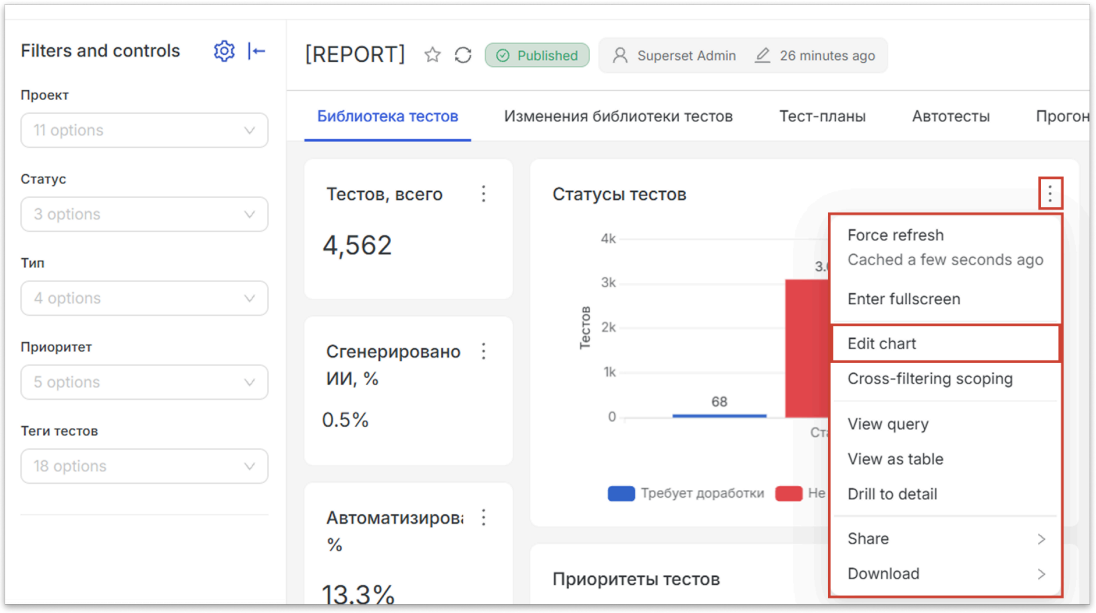
Функция `fetchGuestToken` возвращает гостевой токен (guest token), который вы получаете на своем сервере, обращаясь к эндпоинту Superset `/api/v1/security/guest_token/` через сервисный аккаунт. Этот эндпоинт нельзя вызывать напрямую из браузерного (клиентского) кода. Подробнее: официальная документация Superset .

Редактирование виджета

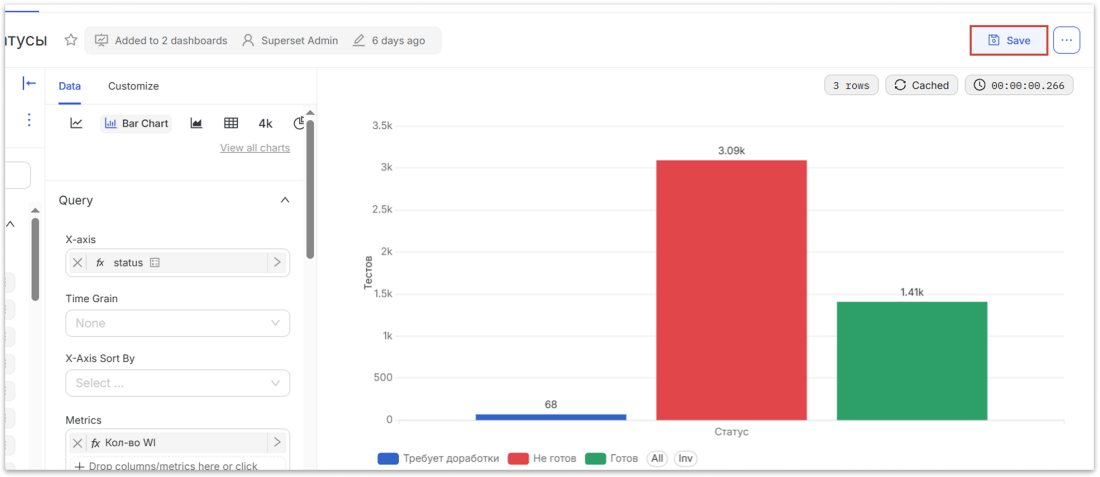
Вы можете изменять виджеты (charts) по умолчанию. Чтобы изменения сохранились после обновления Test IT, виджеты необходимо сохранять как копию, используя опцию **Save as** (сохранить как).

Чтобы отредактировать виджет:

1. Откройте Apache Superset.
2. Откройте окно изменения виджета одним из способов:
   а. Откройте дашборд, затем в нужном виджете откройте меню действий ⋮, затем выберите **Edit Chart**.

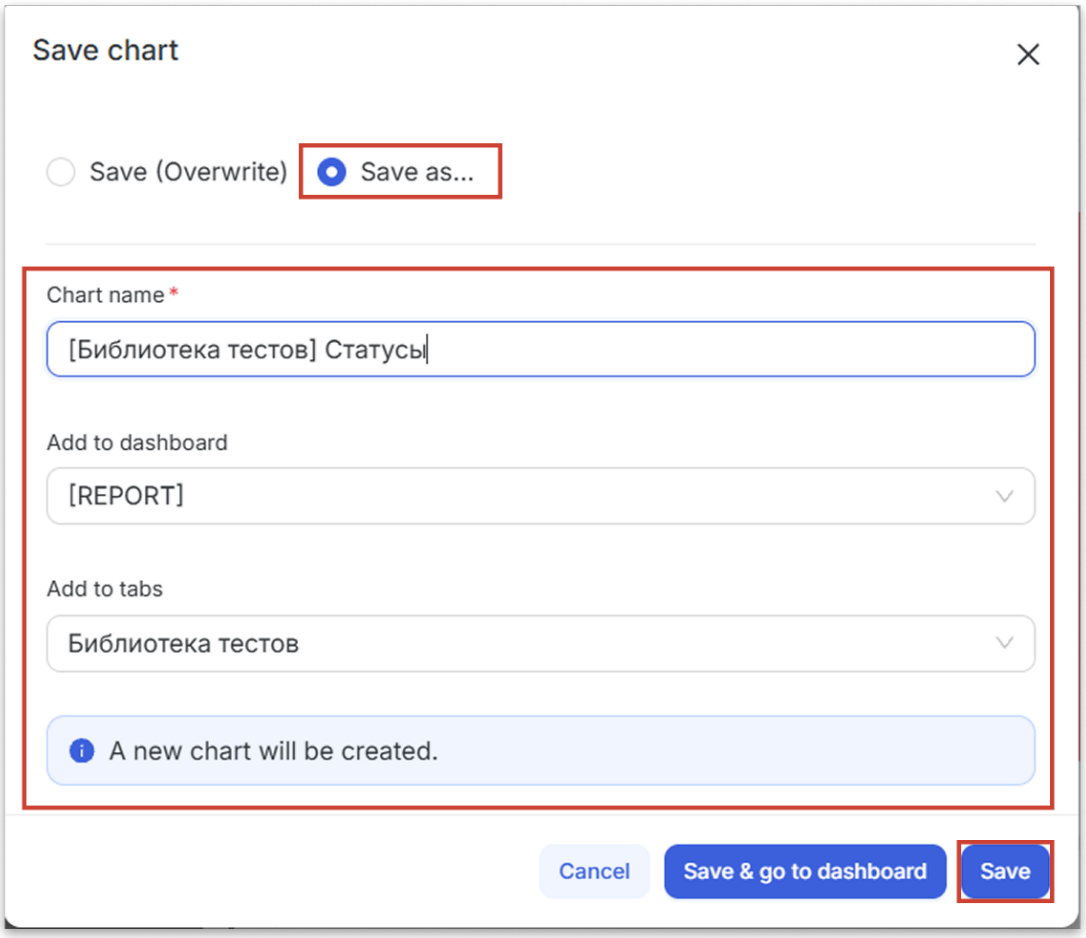


   б. Откройте вкладку **Charts**, в нужном виджете справа нажмите значок редактирования (“карандаш”).
3. Внесите необходимые изменения.
4. В правом верхнем углу окна нажмите **Save**.



Откроется окно сохранения дашборда.

5. Для сохранения выберите опцию **Save as...** (сохранить как).



6. Завершите сохранение виджета.

Подключение сторонних источников данных через Airbyte

Управление ETL-процессами выполняется в интерфейсе Airbyte. Вы можете добавить дополнительные интеграции (например, подключить Jira) через Airbyte путем создания новой связи (connection). Вносить правки в оригинальный Connection не рекомендуется, при обновлении Test IT нарушится работа интеграции, при этом данные сохранятся.

Официальная документация Airbyte:

- Создание новой связи (connection)
- Настройка ETL-процессов

Сохранение и замена пользовательских настроек при обновлении Test IT

При обновлении Test IT вы сможете выбрать, что сделать с вашими настройками в дашборде по умолчанию и ETL-процессах. Вы можете сохранить их или заменить новой версией “из коробки“ или оставить в текущем виде.

Особенности обновления Test IT

При обновлении Test IT вместе с платформой обновляется весь ETL-процесс: репликация из рабочей базы Test IT (PostgreSQL) с помощью Airbyte передается в аналитическую базу ClickHouse, а также на дашборд по умолчанию.

Поэтому редактирование дашборда по умолчанию и любые дополнительные интеграции, добавленные в базу ClickHouse по умолчанию, подвержены рискам, описанным ниже.

#	Пользовательские настройки	Замена на обновленную версию “из коробки“	Сохранение пользовательских настроек
1	Дашборд по умолчанию обновляется как обычно. Созданные вами копии не удаляются и не изменяются.	Дашборд по умолчанию обновляется как обычно. Созданные вами копии не удаляются и не изменяются.	Неприменимо: копия не управляется обновлением. **Важно:** В копии используется общая БД ClickHouse, поэтому при изменении схемы данных запросы в дашборде-копии могут перестать работать — их нужно будет изменить вручную. При копировании дашборда вы можете: • Не выбирать опцию копирования виджетов (also copy (duplicate) charts). В этом случае копируется только дашборд, виджеты и датасеты остаются общими вместе с дашбордом по умолчанию. • Выбрать опцию копирования

#	Пользовательские настройки	Замена на обновленную версию “из коробки“	Сохранение пользовательских настроек
			виджетов (also copy (duplicate) charts). Виджеты будут скопированы, но датасеты останутся общими. Необходимы ручные правки.
2	Дашборд по умолчанию отредактирован (копирование не производилось)	Дашборд по умолчанию и его данные заменяются новой версией “из коробки”. Все изменения, внесенные вами, теряются.	Дашборд остается видимым, но реплика данных останавливается. Данные «замораживаются» на старой схеме. При каждом обновлении создается дополнительная копия базы ClickHouse, что приводит к соответствующему расходу дискового пространства.
3	Изменен ETL-процесс, к базе по умолчанию (ClickHouse) через Airbyte подключены внешние источники (например, Jira)	Изменения, внесенные в интеграцию по умолчанию, теряются безвозвратно.	ETL-процесс необходимо изменять путем создания нового Connection.

Рекомендации по настройке аналитических данных

Инструмент аналитики продолжает развиваться — описанный в этом документе процесс кастомизации и обновления является временным и будет упрощен в последующих релизах. В текущей версии для эффективной и безопасной работы мы рекомендуем следующее:

- Вы можете добавлять на существующий дашборд новые виджеты (см. документацию Superset).
- Виджеты и датасеты, созданные ранее, можно добавить на дашборд, полученный после обновления Test IT.
- Для точечных изменений в виджетах и датасетах копируйте дашборд и работайте в копии. **Не рекомендуется** вносить изменения в оригинальные виджеты и датасеты: при обновлении Test IT эти изменения будут потеряны.
- Если вы редактируете виджеты или датасеты без копирования, при обновлении Test IT они будут потеряны. Для сохранения данных создавайте свои виджеты и датасеты (создавайте новые или копируйте).
- Удаление старого дашборда по умолчанию (если вы выбрали оставить, а затем решили от него отказаться) на текущий момент выполняется вручную.
- При необходимости дашборд можно переключить на нужную версию, используя ID или SDK дашбода в Apache Superset.
- Если вы подключаете внешние источники через Airbyte в базу ClickHouse по умолчанию, добавьте новый Connection, чтобы не потерять данные при обновлении Test IT.

Обновлено: 10/29/26, 1:42 PM

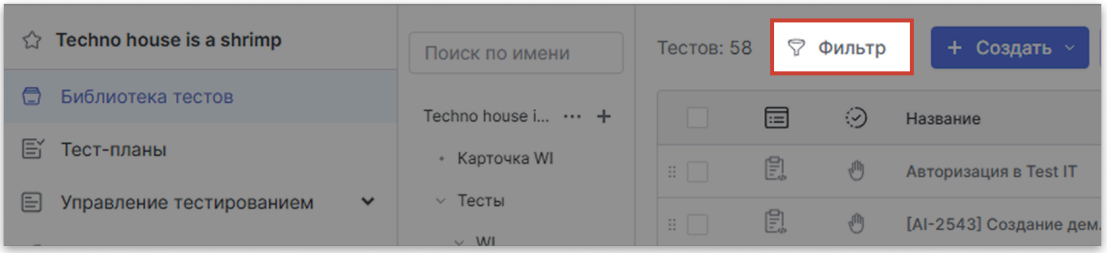
Использование фильтров

Вы можете найти необходимые вам рабочие элементы, автотесты, прогоны (тест-раны) и другие объекты с помощью фильтров. Во всех разделах продукта, содержащих таблицы с элементами, доступен фильтр по содержимому колонок.

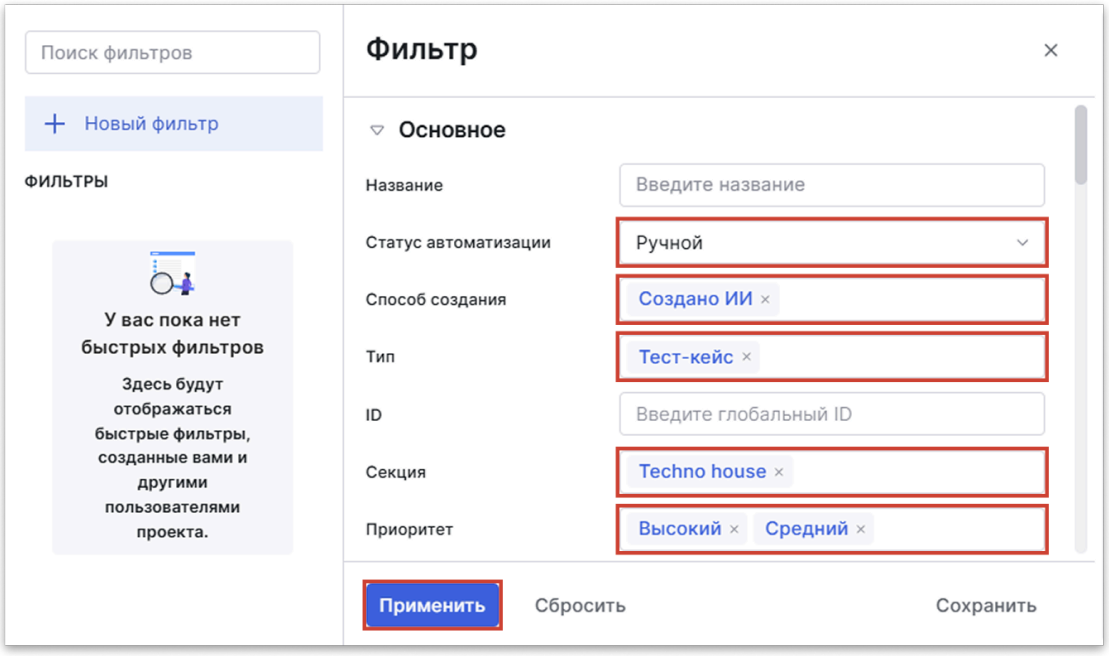
Настройка фильтра данных

Данная инструкция описывает настройку фильтра на примере рабочих элементов. Остальные фильтры работают аналогично.

1. Откройте модуль системы, в котором присутствует таблица.
2. Нажмите **Фильтр**.



3. Задайте критерии, по которым хотите отфильтровать содержимое таблицы. Поля фильтров совпадают с колонками таблиц.



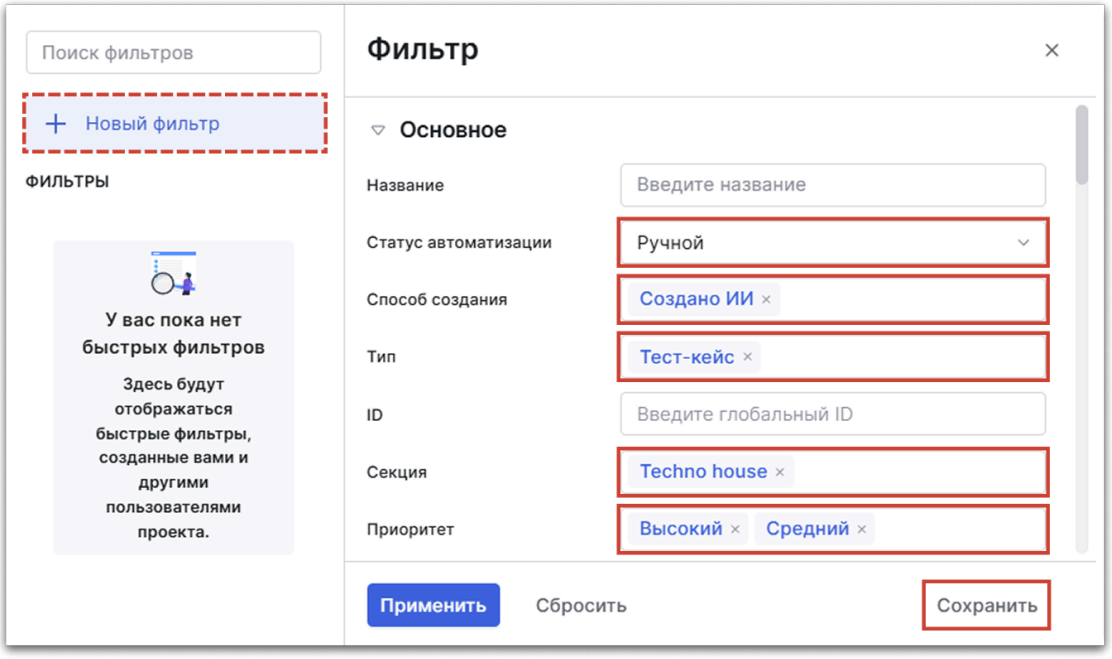
4. Нажмите **Применить**. Таблица переформируется по заданным в фильтре критериям.

Создание фильтров

> **Важно**
>
> Сохранение и редактирование фильтров возможно только в разделе **Библиотека тестов**.

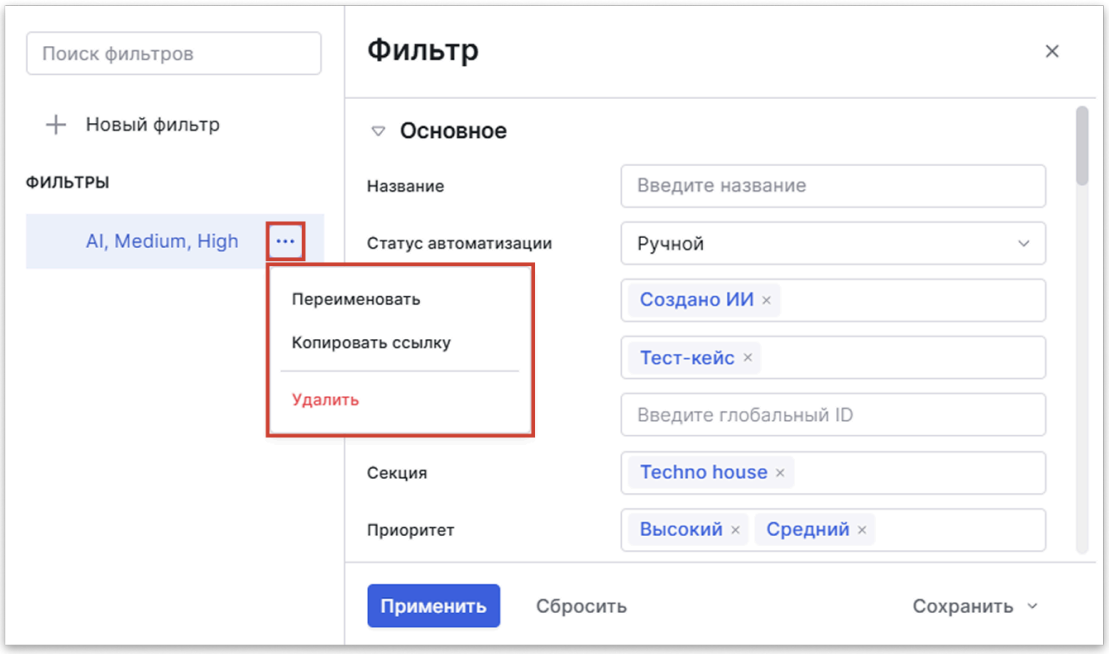
Test IT позволяет сохранять настроенные фильтры библиотеки тестов. Просмотр, редактирование и удаление сохраненных фильтров становится доступным всем пользователям системы, у которых есть право доступа к библиотеке тестов и к фильтрам. Чтобы сохранить фильтр:

1. Перейдите в раздел **Библиотека тестов**.
2. Нажмите **Фильтр** над таблицей рабочих элементов.
3. Убедитесь, что в библиотеке фильтров выбрана корневая папка **Новый фильтр**.



4. Настройте фильтр.
5. Нажмите **Сохранить**.
6. В открывшемся окне введите название нового фильтра и сохраните изменения.
7. **Опционально:** Чтобы переименовать, скопировать ссылку или удалить фильтр, выберите в списке нужный фильтр, откройте меню действий и

выберите нужную опцию.



Фильтрация тест-планов

Настроенные фильтры в разделе **Тест-планы** автоматически сохраняются при переходе между вкладками тест-плана.

Обновлено: 4/15/26, 7:08 PM

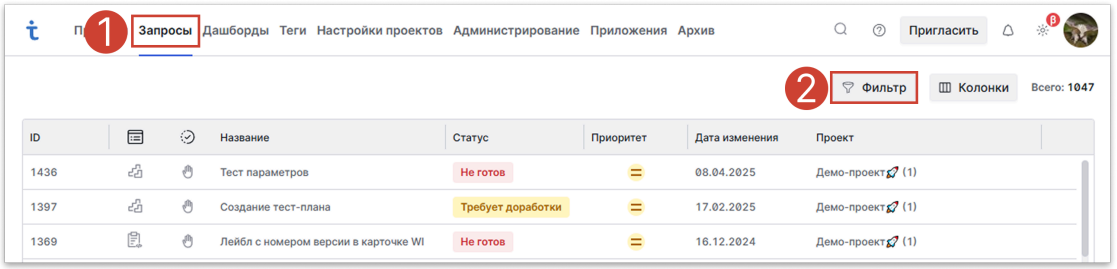
Работа с запросами

Создание запросов доступно для всех пользователей системы. Вы можете фильтровать рабочие элементы по проектам, секциям и пользовательским атрибутам. Test IT позволяет создавать запросы и сохранять их как:

- Личные запросы, доступные только вам
- Публичные запросы, доступные другим пользователям системы

Создание запросов

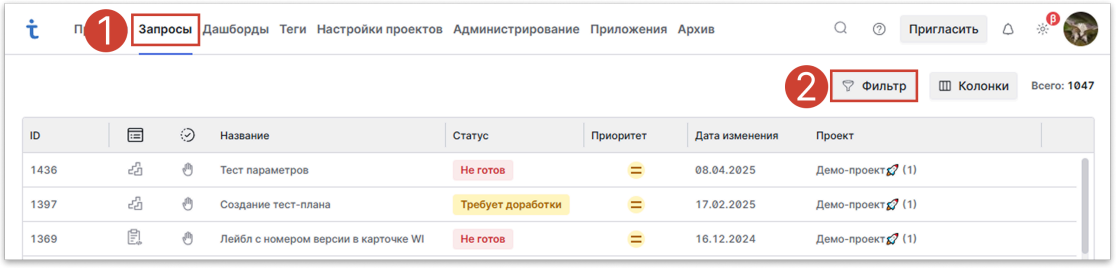
1. В верхней навигационной панели нажмите **Запросы**.
2. Нажмите **Фильтр**.



3. Задайте критерии запроса.
4. Нажмите **Применить**.
   Данные, найденные по запросу, отобразятся в таблице.

Сохранение запросов

1. В верхней навигационной панели нажмите **Запросы**.
2. Нажмите **Фильтр**.



3. Задайте критерии запроса.
4. Нажмите **Сохранить**.
5. В открывшемся окне укажите название запроса и его тип — **Общий** или **Личный**. Созданный запрос сохранится в общем списке. В нем запросы

разделены на личные — доступные только вам, и общие — доступные всем пользователям системы. Пользователи системы могут делиться ссылками на общие запросы.

Вы можете переводить созданные вами запросы из личных в общие и обратно. Для этого:

- Аналогично работе с фильтрами воспользуйтесь меню действий с запросом и выберите **Сделать личным** или **Сделать общим** соответственно. Общие запросы, созданные другими пользователями, невозможно сделать личными.

Обновлено: 4/15/26, 7:08 PM

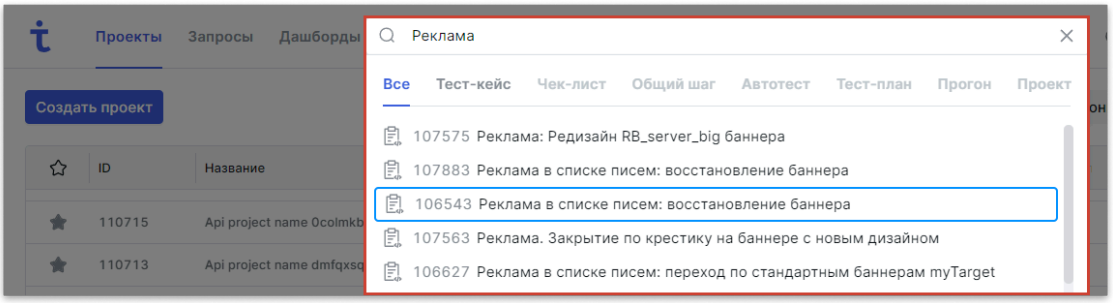
Глобальный поиск

Вы можете осуществлять поиск по всему содержимому вашего пространства Test IT с помощью инструмента **Глобальный поиск**. Поиск осуществляется по названию или глобальному идентификатору объекта (тест-кейс, чек-лист и т. д.). Также вы можете быстро перейти к искомому объекту из меню поиска.

Вы можете просматривать все результаты поиска одновременно или отображать их по типу: **Тест-кейс**, **Чек-лист**, **Общий шаг**, **Автотест**, **Тест-план**, **Прогон** (тест-ран) или **Проект**.

Чтобы воспользоваться поиском:

1. Нажмите значок **Поиск** в верхнем левом углу окна.
2. Начните вводить название или глобальный идентификатор объекта. Результаты поиска отобразятся в выпадающем меню.



3. Для быстрого перехода к найденному объекту, нажмите на соответствующий результат поиска в выпадающем меню. Вы также можете переходить к требуемым пунктам меню с помощью клавиш ↑ и ↓ на клавиатуре и открывать найденные объекты нажатием Enter.

Обновлено: 4/15/26, 7:08 PM

Уведомления

Просмотр уведомлений

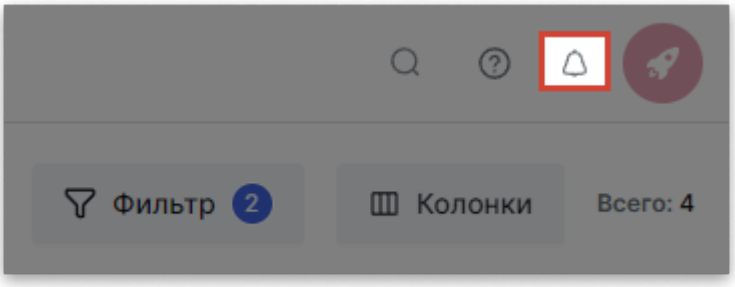
> **Настройте уведомления в мессенджере**
>
> Когда пользователь получает уведомления, вы можете узнавать об этом из сообщений в мессенджере. Для этого нужно настроить вебхук, указав необходимые переменные.

В Test IT есть 4 типа уведомлений:

- Вас назначили исполнителем тестов
- Вас упомянули
- Вас выбрали в тесте
- Вас попросили разблокировать тест-план

Вы можете посмотреть все отправленные вам уведомления в центре уведомлений:

1. Нажмите на значок центра уведомлений.

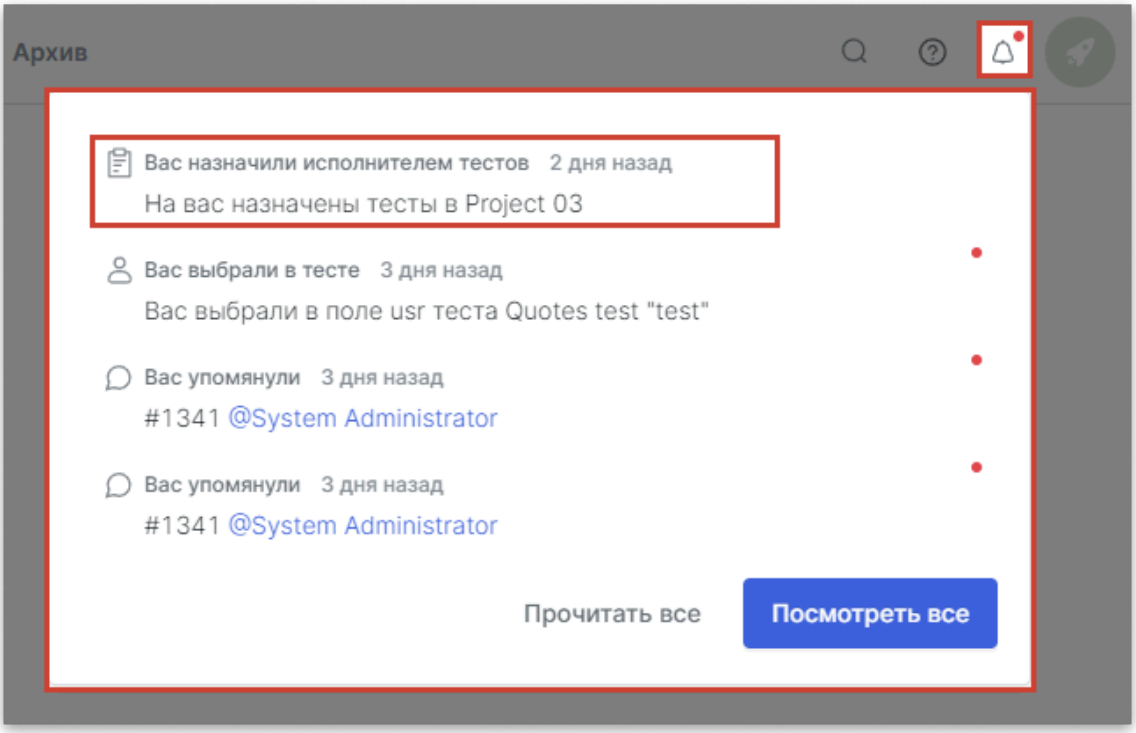


   Откроются последние отправленные уведомления.
2. Нажмите **Посмотреть все**.

По умолчанию открываются все уведомления. Чтобы фильтровать их по типам, нажмите **Фильтр** и укажите необходимые вам параметры поиска.

Вас назначили исполнителем тестов

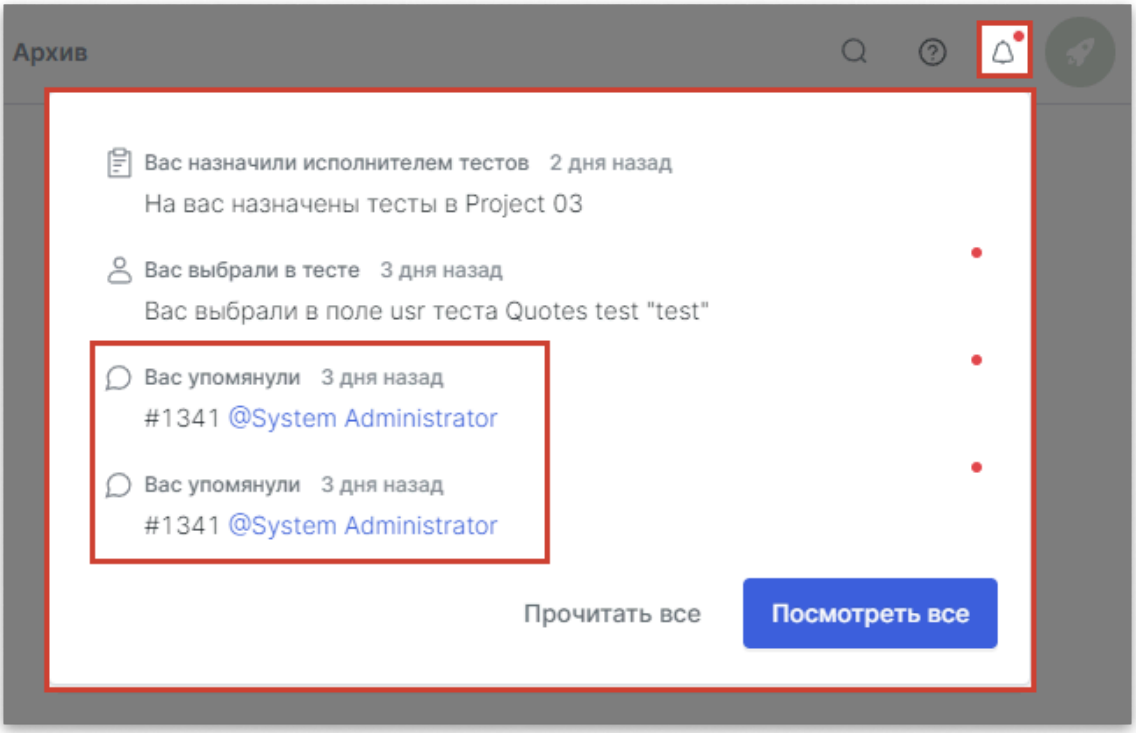
Отправляется при назначении тестов на пользователя.



При нажатии на уведомление о назначенных тестах будет открыт тест-план с назначенными на данного пользователя тестами.

Вас упомянули

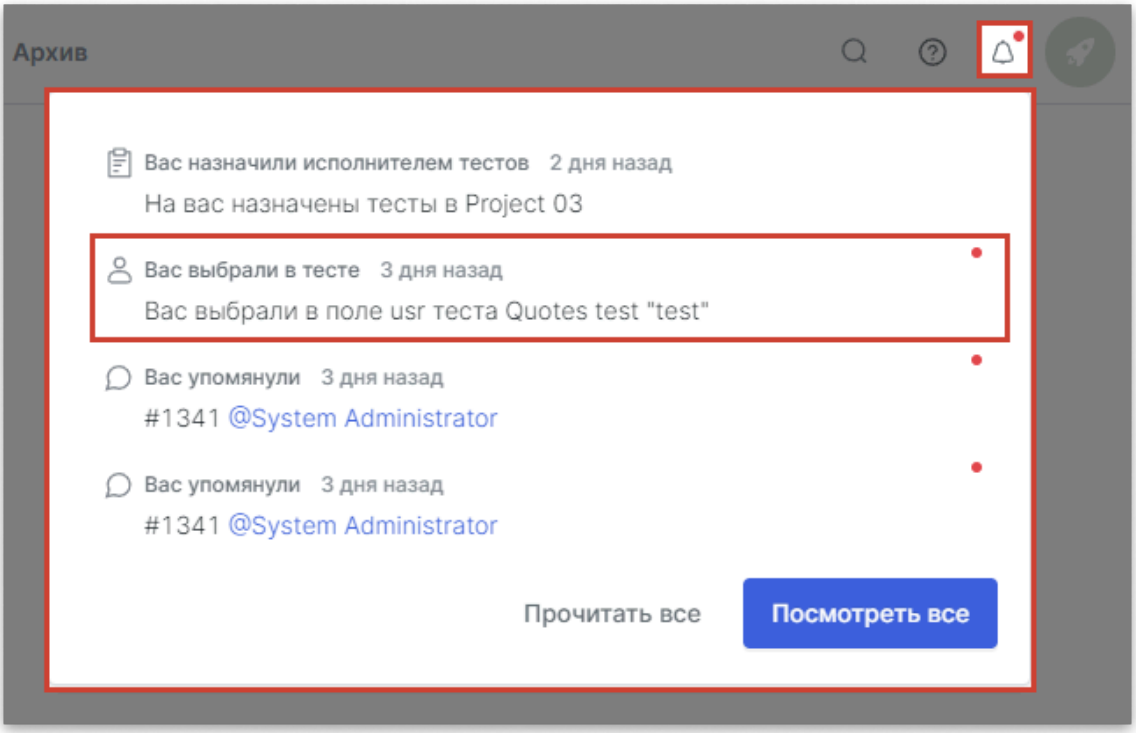
Отправляется при упоминании пользователя в комментарии к тесту.



При нажатии на уведомление об упоминании в комментарии будет открыта страница с тест-кейсом и подсвеченным комментарием, в котором упомянули пользователя.

Вас выбрали в тесте

Отправляется при выборе пользователя в атрибуте типа **Пользователь**.



Уведомление содержит информацию о том, в каком тесте и в каком поле выбрали пользователя.

Вас попросили разблокировать тест-план

Если вы заблокировали тест-план, другой пользователь может отправить вам запрос на разблокировку тест-плана. В этом случае вам придет уведомление с запросом на разблокировку. В центре уведомлений данный тип можно найти в разделе **Все**.

Чтобы перейти к тест-плану, который вас просят разблокировать:

- Нажмите на уведомление с запросом о разблокировке тест-плана.

Обновлено: 4/15/26, 7:08 PM

Подключение Test IT MCP

> **₽ Тарифные ограничения**
>
> Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

- Test IT MCP: общие принципы и данные для подключения
- Подключение к AI-агентам
- Подключение в Cursor
- Подключение в VS Code (Kilo Code)
- Подключение в Zed
- Устранение ошибок

Test IT MCP: общие принципы и данные для подключения

При настройке Test IT MCP нужно указать в IDE или ИИ-агенте адрес MCP-сервера (уже развернут в Test IT) и передать учетные данные для доступа к вашей платформе Test IT.

Работа ведется в **MCP-клиенте** — Cursor, VS Code с расширением Kilo Code, Zed или другом агенте с поддержкой MCP по HTTP. В настройках часто указывают транспорт (transport): **Streamable HTTP** или `type: "streamable-http"` . Подключение через локальный stdio в этом сценарии не используется.

Данные, которые указываются при настройке:

- **URL Test IT** — URL, по которому вы заходите в Test IT, без `/` в конце. Пример: `https://qaspace.testit.software` . При обращении к серверу MCP-клиент отправляет URL Test IT в заголовке `X-baseurl` .
- **URL MCP-сервера** — URL Test IT с постфиксом `/mcp` . Пример: `https://qaspace.testit.software/mcp` (без `/` в конце).
- API-токен Test IT. При обращении к серверу MCP-клиент отправляет API-токен в заголовке `X-token` .

Условия безопасной обработки данных

- Некоторые команды MCP изменяют данные в Test IT (создание кейсов, архив, смена статуса). В чате с агентом подтверждайте действия, которые меняют проект.
- Для проверок удобно создавать отдельный проект в Test IT и отдельный токен с ограниченными правами.

Подключение к AI-агентам

Предварительные условия

Перед настройкой, убедитесь, что:

- MCP поддерживается по HTTP (не только локальный stdio).
- В интерфейсе можно задать произвольные HTTP-заголовки на каждый запрос.
- Поддерживаются MCP Prompts (готовые сценарии Test IT).
- На вашем компьютере есть сетевой доступ к URL MCP-сервера.

Test IT MCP можно использовать в AI-агентах, поддерживающих **Model Context Protocol** и подключение по **HTTP**. При настройке в конфигурации указываются данные:

- Endpoint / URL — URL платформы Test IT + `/mcp`
- Transport — HTTP, Streamable HTTP — как в документации клиента
- Custom headers — `X-baseurl` и `X-token` Конфигурация обычно расположена в файле `mcp.json` , в разделе `mcpServers` настроек IDE или в панели интеграций агента.

Пример конфигурации:

```json
{
  "mcpServers": {
    "test-it": {
      "url": "https://xxxx.testit.software/mcp",
      "headers": {
        "X-baseurl": "https://xxxx.testit.software",
        "X-token": "<ваш_api_token>"
      }
    }
  }
}
```

Подключение в Cursor

1. В Cursor откройте настройки MCP одним из способов:
   - Нажмите F1 или Ctrl + Shift + P (на macOS — Cmd + Shift + P).
   - Примените команду **Open MCP Settings**.
   - Откройте директорию **Settings → MCP → Add Custom Server**. Конфигурация может быть глобальной или в файле `.cursor/mcp.json` в корне рабочей папки.
2. В конфигурационном файле укажите данные сервера:

```json
{
  "mcpServers": {
    "test-it": {
      "url": "https://xxxx.testit.software/mcp",
      "headers": {
        "X-baseurl": "https://xxxx.testit.software",
        "X-token": "<ваш_api_token>"
      }
    }
  }
}
```

Имя `test-it` можно заменить на другой. В строки "url" и "X-baseurl" подставьте свой адрес Test IT. В строке `url` добавьте постфикс `/mcp` . Сохраните файл.

3. Проверьте статус MCP-сервера. Убедитесь, что MCP-сервер в списке имеет статус Connected. В случае ошибки обратитесь к разделу Устранение ошибок.
4. Проверьте работу MCP-сервера в чате с помощью тестового запроса. Например: «Покажи список проектов в Test IT».
   При корректной настройке ассистент обратится к MCP и отобразит данные вашей платформы Test IT.

В Cursor доступны готовые сценарии через MCP Prompts — команды вида `/имя-сервера/имя-сценария` (имя сервера совпадает с ключом в `mcpServers`, например `test-it`).

Подключение в VS Code (Kilo Code)

В VS Code MCP настраивается в расширении Kilo Code.

1. Установите Kilo Code . Если расширение уже установлено, пропустите этот шаг.
2. Откройте настройки MCP в директории: **Settings → Agent Behavior → MCP Servers**. Затем выберите сферу применения MCP:
   - **Edit Global MCP** — для всех проектов в VS Code, или
   - **Edit Project MCP** — только для текущей папки.
3. Добавьте конфигурацию:

```json
{
  "mcpServers": {
    "test-it": {
      "type": "streamable-http",
      "url": "https://xxxx.testit.software/mcp",
      "headers": {
        "X-baseurl": "https://xxxx.testit.software",
        "X-token": "<ваш_api_token>"
      },
      "disabled": false
    }
  }
}
```

- Укажите поле `"type": "streamable-http"`, как показано в примере.
- **Опционально:** Используйте параметр `alwaysAllow`.

О параметре `alwaysAllow`

Параметр `alwaysAllow` со списком имен команд позволяет делать вызовы без подтверждения действия. Его не рекомендуется использовать для команд, изменяющих данные в Test IT.

4. Проверьте работу сервера:
   а. Если сервер не отображается, перезапустите окно VS Code.
   б. В чате Kilo Code запросите список проектов Test IT. При ошибке авторизации проверьте `X-baseurl` (убедитесь, что указан префикс `https://`) и актуальность токена.

Подключение в Zed

В Zed внешние MCP-серверы называются **context servers**. Чтобы подключить MCP-сервер в Zed:

1. Откройте настройки, для этого нажмите `F1` , затем примените команду `zed: open settings file` .
2. Добавьте данные сервера (context server):

```json
{
  "context_servers": {
    "test-it-mcp": {
      "enabled": true,
      "url": "https://xxxx.testit.software/mcp",
      "headers": {
        "X-baseurl": "https://xxxx.testit.software",
        "X-token": "<ваш_api_token>"
      }
    }
  }
}
```

   Вместо `test-it-mcp` можно использовать другое имя.
   Сохраните файл и при необходимости перезапустите Zed.
3. Проверьте работу:
   а. В AI-панели включите context server.
   б. Задайте контрольный вопрос по данным Test IT, например: «Какие у

меня проекты в Test IT?».
В случае ошибки обратитесь к разделу Устранение ошибок.

Устранение ошибок

Ниже описаны наиболее распространенные ошибки и способы их устранения:

- Нет подключения / timeout: Проверьте URL: адрес пространства + `/mcp`, HTTPS, VPN и доступ в сеть организации.
- Ошибка, связанная с заголовками или контекстом безопасности (security context): В конфигурационном файле проверьте `X-baseurl` и `X-token`.
- Ошибка 401 или отказ Test IT: Перевыпустите API-токен Test IT, проверьте адрес платформы Test IT в `X-baseurl` и права пользователя в проекте.
- Ассистент вносит некорректные изменения или изменяет некорректные данные: Просмотрите историю вызовов MCP в клиенте; при необходимости отмените изменения в Test IT вручную.

Если проблема сохраняется, напишите на **support@testit.software**. В письме укажите: клиент (Cursor / Kilo Code / Zed / другой), текст или скриншот ошибки, версию клиента.
Не отправляйте API-токен — при необходимости поддержка попросит проверить доступ иным способом.

Обновлено: 8/18/26, 4:54 PM

Test IT MCP: возможности и способы работы

₽ Тарифные ограничения

Опция доступна в платных тарифах. Подробнее: о тарифах Test IT .

Безопасность работы с MCP: рекомендации

Чтобы ваша работа с MCP была продуктивной и безопасной, вот несколько рекомендаций:

- Запросы к ИИ не должны содержать чувствительных данных, подлежащих NDA.
- Запросы на поиск и анализ данных можно совершать свободно.
- Запросы на изменение данных (например, создание или архивацию тестов) рекомендуется выполнять с подтверждением.
- Не передавайте API-токен в общие чаты; в IDE он указывается только в настройках MCP.
- Для тестовых запросов и экспериментов используйте отдельный проект в Test IT.

Test IT MCP дает ИИ-ассистенту в IDE доступ к платформе Test IT. Получив доступ, ИИ-ассистент может читать проекты и кейсы, искать прогоны, формировать отчеты и вносить изменения.

После подключения MCP вы можете использовать его возможности и применять сценарии .

Просмотр возможностей MCP-сервера

Просмотр команд и промптов в MCP Inspector

MCP Inspector — отдельное веб-приложение для просмотра инструментов, запросов и ресурсов вне чата в IDE. Это может быть удобно, чтобы 1 раз проверить корректность ответов сервера и полноту списка команд.

1. На компьютере с установленным Node.js выполните в терминале команду:

```sh
npx @modelcontextprotocol/inspector
```

2. В открывшемся интерфейсе укажите:
   - **URL MCP-сервера:** адрес вашего пространства + `/mcp` (например `https://xxxx.testit.software/mcp`);
   - **Transport:** Streamable HTTP (или HTTP — как предлагает Inspector);
   - **Headers:** `X-baseurl` и `X-token`. Подробнее: Test IT MCP: общие принципы и данные для подключения
3. Подключитесь и просмотрите вкладки **Tools** и **Prompts**. Отдельные команды можно вызвать вручную с тестовыми параметрами.

Inspector не заменяет IDE: в повседневной работе удобнее чат с ассистентом и готовые prompts ниже.

Обзорный промпт в чате с ИИ-ассистентом

Обзорный промпт может быть удобен, когда Inspector недоступен и нужно быстро получить простое описание возможностей сервера.

- Вызовите обзорный промпт одним из способов:
  - В чате IDE введите команду: `testit_mcp_capabilities_manual`. — Ассистент получит с сервера MCP полную инструкцию, включая:
    - Доступные инструменты и их назначение
    - Доступные промпты и их типовые цепочки. Пример команды в Cursor (если сервер в конфиге назван `test-it`):

```text
/test-it/testit_mcp_capabilities_manual
```

  - В чате с ИИ-ассистентом отправьте запрос: «Вызови prompt testit_mcp_capabilities_manual и кратко перечисли, что умеет Test IT MCP». Затем задайте уточняющий вопрос — ассистент подберет нужный инструмент или сценарий из обзора.

Список подключенных инструментов в IDE

В Cursor, Kilo Code и других клиентах с поддержкой MCP обычно есть список подключенных инструментов (tools). Новые инструменты отображаются после успешного соединения.
Если список пуст:

- Проверьте статус сервера и заголовки авторизации.

Способы работы с MCP

- **Инструменты (tools)**. Предназначены для выполнения конкретных действий, например найти прогоны автотестов, сравнить прогоны, создать тест-кейс и т.д.
- **Готовые сценарии (prompts)**. Подробная инструкция для ИИ-ассистента (рекомендуется для решения типовых задач): анализ библиотеки, сравнение прогонов, генерация кейсов по техническому заданию и т.д.
- **Ресурсы (resources)**. Тексты сценариев в виде справочных материалов, аналогичные готовым сценариям (prompts) — для клиентов, где готовые сценарии недоступны.

Использование готовых сценариев

Использование сценариев в Cursor

В Cursor сценарии инициируются с помощью команд. В префиксе команды указывается имя сервера.

Например, если имя сервера `test-it` , в чате доступны команды вида:

- `/test-it/testit_library_actuality`
- `/test-it/testit_compare_test_runs`
- `/test-it/testit_manual_test_scenarios`

Использование сценариев в других клиентах

- Используйте раздел **MCP Prompts**
- В чат с ИИ-ассистентом отправьте запрос: «Следуй сценарию `testit_library_actuality` ». Ассистент отобразит инструкцию с сервера.

Обновлено: 10/19/26, 6:15 PM

Сценарии работы с MCP

Для типовых задач, с которыми сталкиваются команды тестирования, есть готовые сценарии. Применять их можно как с помощью простого текстового запроса или специального промпта.

Простые запросы без промптов (с использованием разговорного языка) могут применяться при использовании корректных, четких формулировок, например:

- Покажи проекты в Test IT
- Найди прогоны проекта ... за последнюю неделю
- Найди тест-кейсы в секции ...
- Создай секцию ... в проекте ...
- Создай пустой прогон в проекте ...

Test IT MCP предлагает готовые типовые сценарии для:

- Ручного тестирования
- Автоматизированного тестирования
  Каталог сценариев будет расширяться.

Возможности Test IT MCP не ограничиваются этими сценариями, вы можете предлагать и создавать свои. Для экспериментальных сценариев и тестовых запросов рекомендуем создать отдельный проект.

Работа по сценариям возможна после подключения Test IT MCP.

> **Внимание**
>
> Для операций, изменяющих данные в Test IT (создание / изменение / удаление объекта, обновление статуса), четко обозначьте в запросе:
>
> 1. Необходимость сначала показать план
> 2. Необходимость вашего подтверждения перед выполнением действия

Сценарии для ручного тестирования

Анализ актуальности библиотеки

Сценарий включает аудит библиотеки ручных тестов и автотестов, в том числе поиск:

- Устаревших ручных тестов, дубликатов ручных тестов, неиспользуемых общих шагов, тестов, которые давно не включались в тест-планы

- Нестабильных автотестов и автотестов, которые давно не запускались

Промпт: `testit_library_actuality`

Параметры

Параметр	Обязательный	Пояснение
`projectId`	да	GUID проекта в Test IT
`sectionId`	нет	Анализ отдельной секции библиотеки (включая все ее дочерние секции)
`duplicateSampleSize`	нет	Количество тестов для проверки на дубликаты (по умолчанию 120)

Пример запроса в чате:

```text
/test-it/testit_library_actuality
// Вводить промпт необязательно, для запроса можно использовать разговорный язык.

Проект 01955ac0-bce1-7e95-9508-d6c9295ef89f.
Секция ce9305c3-1946-4f07-bd9e-0dea63cc750b — только эта ветка библиотеки.
```

Реализация сценария

1. Инициация анализа библиотеки на сервере MCP.
2. Отображение таблицы, содержащей тест-кейсы, общие шаги, автотесты, их статус (*OK* или *Проблема*), количество, примеры, рекомендации по улучшению библиотеки.
3. Предоставление отдельной сводки по типам проблем и групп дубликатов.
4. Предложение действия: массовой архивации или изменения статуса тестов с пометкой *Проблема* (требуется подтверждение).

Типичные действия по результатам анализа

Результат анализа	Типичные действия
Дубликаты ручных тестов	Архивация дубликатов по id
Устаревшие кейсы	Перевод в статус *Требует доработки* или архивация
Нет связи тестов с требованием	Связь тестов с требованиями в интерфейсе Test IT или архивация

Результат анализа	Типичные действия
Проблемные автотесты	Прогон, доработка автотестов, архивация неактуальных автотестов

Часть проверок по автотестам и тест-планам выполняется на уровне всего проекта, даже если задан `sectionId` — эта информация указывается в шапке отчета.

Ограничения

- Сценарий не позволяет определить, была ли связь с требованиями удалена или не была создана.
- Перезапуски автотестов оцениваются по статистике, доступной в API, а не по всей истории.

Генерация ручных тест-кейсов

Сценарий позволяет быстро получить набор ручных функциональных тест-кейсов по описанию функциональности, требованиям или техническому заданию с учетом уже существующих кейсов в библиотеке. После генерации тест-кейсы можно доработать.

Промпт: `testit_manual_test_scenarios`

Параметры

Параметр	Обязательный	Пояснение
`projectId`	да	Идентификатор проекта, в который будут сохранены тесты
`taskDescription`	да	Описание функциональности / задачи
`specificationText`	нет	Текст технического задания или требований
`librarySectionId`	нет	Идентификатор секции библиотеки, в которой содержатся тесты, используемые в качестве контекста
`libraryWorkItemIds`	нет	Идентификаторы тестов для контекста

Пример запроса в чате:

```text
/test-it/testit_manual_test_scenarios
// Вводить промпт необязательно, для запроса можно использовать
разговорный язык.

Проект …, задача: авторизация по SSO.
ТЗ: <вставить текст или приложить файл>.
Учти кейсы из секции …
```

Реализация сценария

1. Сбор контекста (проект, при необходимости — существующие кейсы из Test IT).
2. Подготовка **промпта для генерации** и правил форматирования (название, предусловия, шаги, ожидаемые результаты).
3. Генерация кейсов, из которых вы можете выбрать тесты для сохранения.
4. Сохранение в Test IT (требуется подтверждение и идентификатор секции).

Подготовка общих шагов из дубликатов шагов тестов

Сценарий позволяет найти повторяющиеся шаги тестов (дубликаты), подготовить из них общие шаги и заменить дубликаты в тест-кейсах ссылками на созданные общие шаги.

Промпт: `testit_create_shared_steps`

Параметры

- `projectId` — идентификатор проекта
- Область поиска: вся библиотека, секция или список кейсов (требуется указать в чате с ИИ-ассистентом)

Пример запроса в чате:

```text
/test-it/testit_create_shared_steps
// Вводить промпт необязательно, для запроса можно использовать
разговорный язык.

В проекте … найди повторяющиеся шаги в секции «Регресс» и предложи
вынести в общие шаги.
```

Реализация сценария

ИИ-ассистент проводит следующие шаги:

1. **Поиск** кейсов с шагами в указанной области
2. **Анализ** — группировка семантически одинаковых шагов
3. **План** (без записи в Test IT): какие ОШ создать, какие кейсы обновить, таблица соответствий «старый шаг → общий шаг»
4. После вашего **«да»** — создание ОШ и обновление шагов в кейсах.

Уникальные шаги не удаляются. В спорных случаях ИИ-ассистент делает пометку **на проверку**.

Перенос общих шагов в предусловия секции

Часть повторяющихся шагов логичнее хранить не в каждом тест-кейсе или общем шаге, а в предусловиях секции. В этом случае шаг будет указан в секции 1 раз, а использоваться во всех тестах секции. В отличие от общих шагов, предусловия секции применяются ко всем кейсам в папке автоматически.

Промпт: `testit_section_preconditions`

Параметры

- `projectId` — идентификатор проекта
- При необходимости указываются конкретные секции или весь проект по секциям.

Пример запроса в чате:

```text
/test-it/testit_section_preconditions
// Вводить промпт необязательно, для запроса можно использовать разговорный язык.

Проект …: по каждой секции найди шаги, которые повторяются у большинства кейсов, и предложи перенести в предусловия секции.
```

Реализация сценария

1. Загрузка дерева секций проекта.
2. Анализ тестов по каждой секции (удобнее и точнее, чем один запрос на весь проект).
3. Поиск шагов, которые повторяются у нескольких кейсов и соответствуют уровню секции (подготовка окружения, общий вход), а не уникальной проверки.
4. План: предлагаются дополнения в предусловия секции, а также список кейсов, из которых можно убрать общие шаги.
5. Обновление предусловий секции и правка шагов тестов (требуется подтверждение).

Сценарии для автоматизированного тестирования

Сравнение двух прогонов

Сценарий позволяет сопоставить прогоны автотестов: количество тестов, количество тестов с ошибками и изменения в их диагностике, выявленные улучшения и проблемы.

Промпт: `testit_compare_test_runs`

Параметры

Параметр	Обязательный	Пояснение
`projectId`	да	GUID проекта
`testRunId1` ; `testRunId2`	нет	Идентификаторы прогонов для сопоставления

Для сравнения нужно указать 2 идентификатора прогонов либо не указывать ни одного. Если идентификаторы не указаны, сравниваются два последних прогона проекта.

Примеры запроса в чате:

- Запрос с использованием сценария:

```text
/test-it/testit_compare_test_runs

Проект 01955ac0-….
Сравни прогоны aaaaaaaa-… и bbbbbbbb-….
```

- Свободный текстовый запрос:

```text
Сравни два последних прогона в проекте …
```

Реализация сценария

1. Сравнение возраста прогонов по дате создания.
2. Расчет длительности прогонов, количества тестов и тестов с ошибками, дельт.

3. Отображение исправленных результатов (*Failed* → *Passed*), новых падений автотестов, новых или измененных текстов ошибок.
4. Если общее число результатов или упавших тестов не совпадает — показ списка расхождений по отдельным тестам (`onlyInOlder` , `onlyInNewer` , смена результата).
5. Отображение результата анализа в виде таблицы. По запросу можно просмотреть подробности отдельного результата в Test IT.

Сценарии для управления тестированием

Подготовка тест-плана

Сценарий помогает создавать тест-план из тестов, расположенных в библиотеке по вашим критериям, например:

- Время прохождения
- Приоритет
- Тест-кейсы, в которые часто вносились изменения
- Нестабильные автотесты
- Теги

Промпты:

- `propose_smart_test_plan` — предлагает проект тест-плана по вашему запросу, не создавая его. Проект содержит список тестов со ссылками на них в Test IT
- `create_smart_test_plan` — создает готовый тест-план по вашему запросу по проекту, подготовленному с помощью промпта `propose_smart_test_plan` либо без него. Ссылка на созданный тест-план содержится в ответе MCP.
- `search_project_test_plans` – поиск тест-планов проекта, сбор контекста
- `search_test_runs_of_test_plan` – просмотр прогонов тест-плана

Параметры

- `projectId` — идентификатор проекта (необязательный параметр, можно запросить поиск проекта по критериям пользователя)
- Область поиска: вся библиотека, секция или список кейсов (требуется указать в чате с ИИ-ассистентом)

Примеры запроса в чате:

```text
// Пример 1
/test-it/propose_smart_test_plan
// Вводить промпт необязательно, для запроса можно использовать разговорный язык.

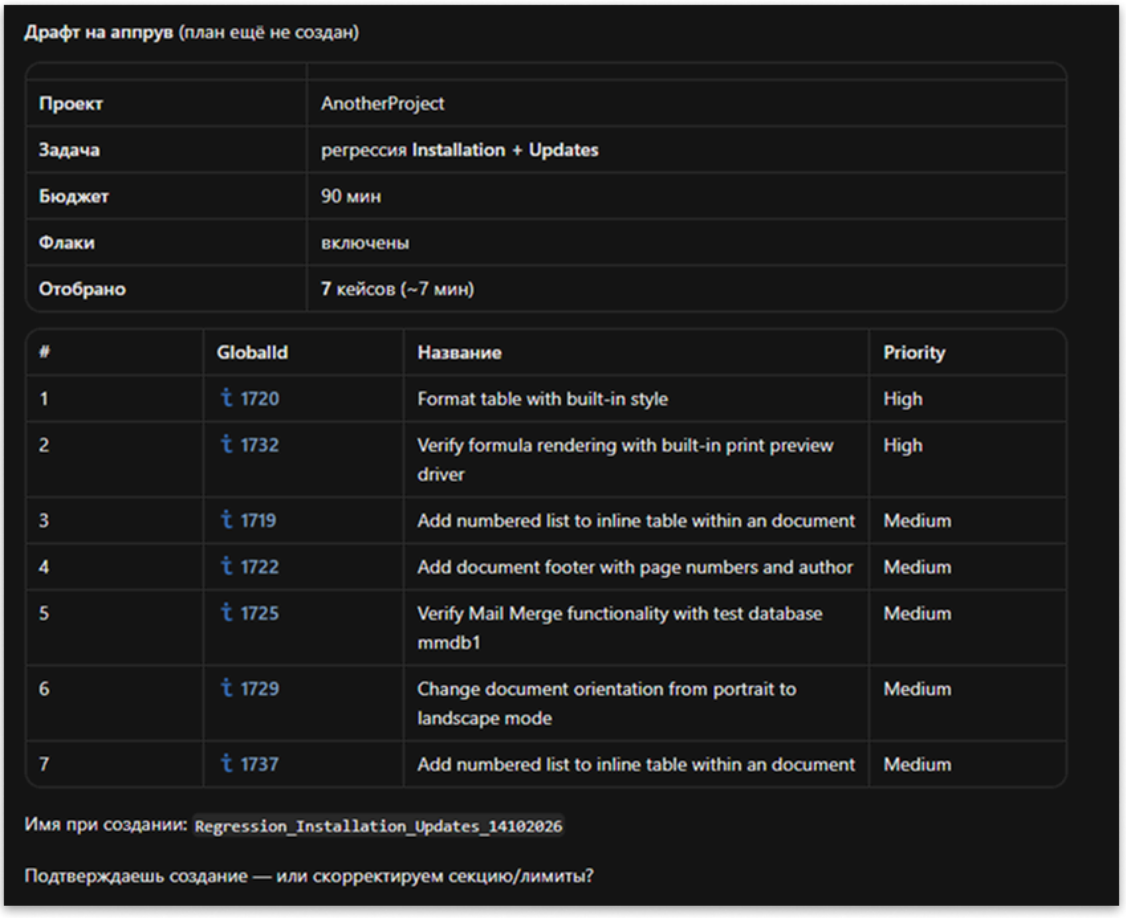
В проекте 01955ac0-bce1-7e95-9508-d6c9295ef89f предложи тест-план для тестов с высоким приоритетом, который часто изменяли.

// Пример 2
/test-it/create_smart_test_plan
// Вводить промпт необязательно, для запроса можно использовать разговорный язык.

Выбери проект, где много тест-кейсов, создай тест-план, объединив тесты по общей задаче. Flaky-тесты тоже включай в план.
```

Реализация сценария

1. Если указано в запросе: поиск проекта по заданным критериям.
2. Анализ тестов проекта по критериям, заданным пользователем.
3. Ответ на запрос (в зависимости от промпта): * Подготовка проекта тест-плана со ссылками на тесты в Test IT



Драфт на аппрув (план ещё не создан)

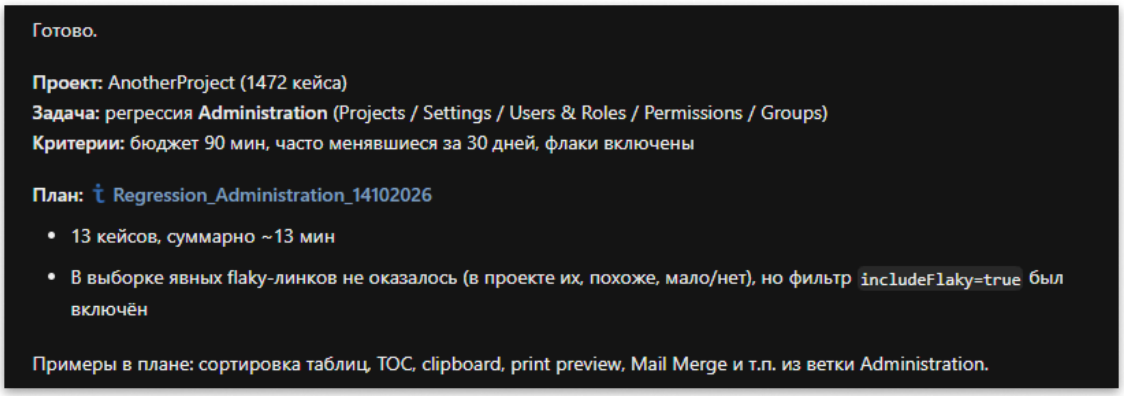
Проект	AnotherProject
Задача	регрессия **Installation + Updates**
Бюджет	90 мин
Флаки	включены
Отобрано	**7** кейсов (~7 мин)

#	GlobalId	Название	Priority
1	t 1720	Format table with built-in style	High
2	t 1732	Verify formula rendering with built-in print preview driver	High
3	t 1719	Add numbered list to inline table within an document	Medium
4	t 1722	Add document footer with page numbers and author	Medium
5	t 1725	Verify Mail Merge functionality with test database mmdb1	Medium
6	t 1729	Change document orientation from portrait to landscape mode	Medium
7	t 1737	Add numbered list to inline table within an document	Medium

Имя при создании: `Regression_Installation_Updates_14102026`

Подтверждаешь создание — или скорректируем секцию/лимиты?

* Создание тест-плана со ссылкой на него в Test IT



Промпты Test IT MCP

Промпт можно использовать как альтернативу текстовому запросу — как готовый сценарий или его часть.

#	Задача	Рекомендуемый промпт	Результат
1	Обзор функциональности MCP	`testit_mcp_capabilities_manual`	Обзор возможностей ассистента
2	Проверить состояние библиотеки тестов	`testit_library_actuality`	Отчет с устаревшими и дублированными ручными тестами, автотестами, а также рекомендации по улучшению
3	Сгенерировать ручные тесты по ТЗ	`testit_manual_test_scenarios`	Подготовка промпта и сохранение тестов в проекте
4	Вынести повторяющиеся шаги в общие шаги (ОШ)	`testit_create_shared_steps`	План → создание ОШ → обновление кейсов
5	Перенести общие шаги в предусловия секции	`testit_section_preconditions`	План → предусловия секции → очистка дублей в тест-кейсах

6	Сравнить два прогона автотестов	`testit_compare_test_runs`	Метрики, дельты, улучшения / ошибки и падения автотестов

Устранение ошибок

- Ассистент не распознает сценарий: Вызовите промпт непосредственно (`/test-it/testit_…`) или напишите: «следуй сценарию `testit_library_actuality` ».
- Отсутствуют нужные инструменты (tools): Проверьте подключение MCP; откройте список в MCP Inspector.
- Ошибка доступа к Test IT: В настройках проверьте заголовок `X-baseurl` и API-токен.
- Ассистент изменяет данные без подтверждения: Напомните: `Сначала только план, без записи в Test IT` .

По вопросам подключения, ошибок сервера, а также с предложениями по развитию MCP обратитесь в техническую поддержку: support@testit.software .

Обновлено: 10/20/26, 6:49 PM

Работа с API

Создание API-токенов

API-токены можно создать в профиле пользователя. Вы можете создать один или несколько API-токенов.

Вы можете взаимодействовать с системой Test IT посредством открытого API. Методы API позволяют создавать и запускать автотесты, создавать тесты в библиотеке и многое другое. Описание методов API хранится в Swagger. Вы можете взаимодействовать с открытым API любым способом, например с помощью Postman .

Изменения в API-методах для управления параметрами

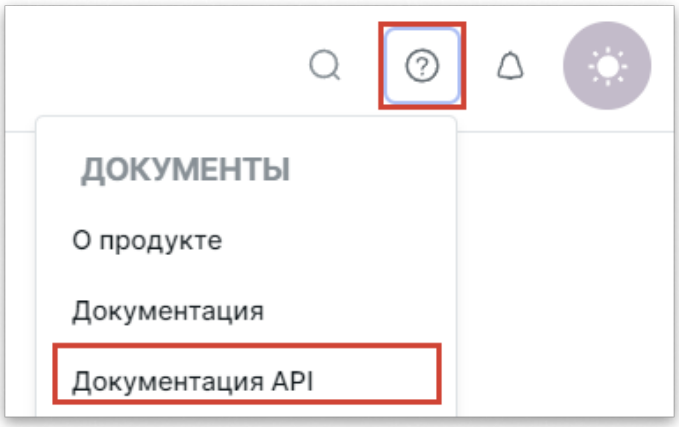
С 27.08.2026 параметры разделены на различные типы. Что изменилось в API-методах:

- Методы группы `/api/v2/Parameters` доступны только для работы с **параметрами ручных тестов**.
- Для работы с **параметрами конфигураций** в Swagger Test IT используйте API-методы группы `/api/v2/ConfigurationParameters` .

Переход в Swagger из интерфейса Test IT

Вы можете открыть Swagger из интерфейса Test IT двумя способами:

- Нажмите значок **Информация** в правом верхнем углу окна, затем в выпадающем меню выберите **Документация API**.



- В адресной строке браузера добавьте постфикс `/swagger` к адресу системы Test IT, затем перейдите по получившемуся адресу. Например, *testit.software/swagger*, где *testit.software* необходимо заменить на адрес системы вашей компании.

Если вы авторизованы в Test IT, авторизация с помощью API-токена не требуется. Вы можете выполнять запросы из Swagger.

Авторизация с помощью API-токена

1. Скопируйте сохраненный или создайте новый API-токен.
2. Откройте Swagger. В открывшемся окне нажмите **Authorize**.
3. В открывшемся диалоговом окне введите `PrivateToken {API Secret Key}` . Вставьте скопированный ранее секретный ключ API вместо `{API Secret Key}` , не используя фигурные скобки. Например: `PrivateToken NxNBZ3A5d6h3bU5JLk40Q3VM` .
4. Нажмите **Authorize**.

Обновлено: 8/27/26, 9:18 AM

Горячие клавиши

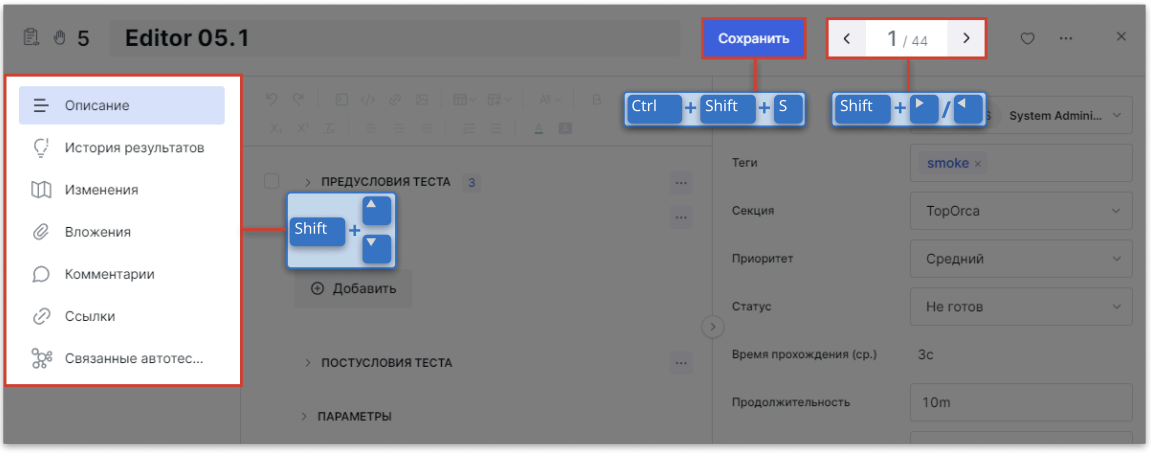
Горячие клавиши помогут вам сэкономить время и сделать работу в Test IT легкой и интуитивной. Вы можете использовать горячие клавиши для работы с модальными окнами всех разделов системы, рабочими элементами и тест-поинтами во вкладке Выполнение тест-плана. В горячих клавишах применяется латинский алфавит.

Горячие клавиши для работы с модальными окнами

При работе с модальными окнами (например с окном восстановления проекта из Архива) вы можете использовать следующие клавиши:

- C — отмена
- Esc — выход
- S — сохранить и выйти

Горячие клавиши для действий с рабочими элементами



Чтобы воспользоваться горячими клавишами, откройте окно рабочего элемента.

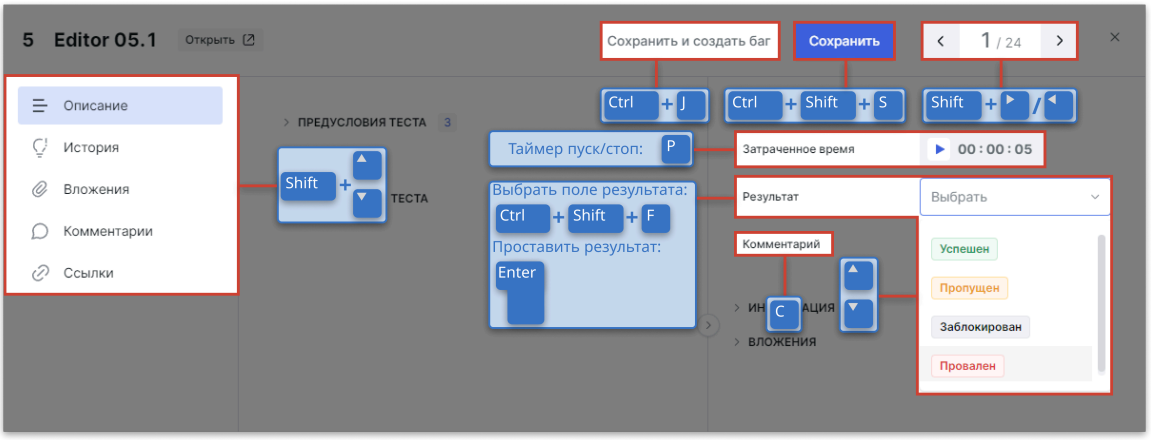
Важно

Для использования горячих клавиш не должны быть выбраны поля для ввода данных (в полях окна не должно быть фокуса).

Вы можете использовать следующие сочетания клавиш:

- Shift + → ; Shift + ← — переключение между рабочими элементами, идентично переключению с помощью стрелок в интерфейсе Test IT
- Shift + ↓ ; Shift + ↑ — переключение между пунктами левого навигационного меню: **Описание**, **История результатов**, **Изменения** и т. д.
- Ctrl / Cmd + Shift + S — сохранение рабочего элемента после внесения изменений
- Ctrl / Cmd + Z — отмена последнего действия, связанного с вводом данных (шаги, условия, указание продолжительности теста)
- Ctrl / Cmd + Y — повтор отмененного действия, связанного с вводом данных

Горячие клавиши для работы с тест-поинтами



Чтобы воспользоваться горячими клавишами, откройте тест-план, затем откройте окно тест-поинта. Вы можете использовать следующие клавиши и их сочетания:

- Shift + → ; Shift + ← — переключение между тест-поинтами. Идентично переключению с помощью стрелок в интерфейсе Test IT.

- Shift + ↓ ; Shift + ↑ — переключение между пунктами левого навигационного меню: **Описание**, **История**, **Вложения** и т. д.
- P — запуск / остановка таймера прохождения теста
- C — выбор поля **Комментарий**
- Ctrl / Cmd + Shift + S — сохранение рабочего элемента после внесения изменений
- Ctrl / Cmd + J — создание задачи в таск-трекере (если результат теста *Пропущен*, *Заблокирован* или *Провален*; требуется предварительно настроенная интеграция и подключение к проекту)
- Ctrl / Cmd + Z — отмена последнего действия, связанного с вводом данных (например, написание комментария)
- Ctrl / Cmd + Y — повтор отмененного действия, связанного с вводом данных

Указание результата прохождения тест-поинта

Для проставления результатов тест-поинтов вы можете использовать следующие клавиши и их сочетания:

- Ctrl / Cmd + Shift + F — выбор меню результата
- ↓ / ↑ — переключение между пунктами меню выбора результата
- Enter — проставление тест-поинту выбранного результата

Обновлено: 4/15/26, 7:08 PM

Глоссарий

API (Application Programming Interface) — программный интерфейс приложения, позволяющий связывать между собой различные приложения. Вы можете взаимодействовать с системой Test IT посредством открытого API. Подробнее: Работа с API.

CI/CD (Continuous Integration / Continuous Deployment) — набор инструментов, практик и процессов, которые помогают автоматизировать и упростить непрерывную интеграцию и непрерывное развертывание. Test IT поддерживает интеграцию с многими системами CI/CD для загрузки результатов автотестов. Подробнее: Интеграция с системами CI/CD.

CLI (Command Line Interface) — в системе Test IT — утилита командной строки, которая помогает взаимодействовать с Test IT. Позволяет собирать и загружать результаты автоматизированных тестов, работать с любыми тестовыми фреймворками и интегрировать их. Использует формат JUnit XML. Подробнее: Test IT CLI.

CWM (Collaborative Work Management) — система совместной работы над проектами (например, Jira или TeamStorm). См. Таск-трекер.

MCP (Model Context Protocol) — сервис, связывающий ИИ-ассистента в редакторе кода с данными в Test IT. Подробнее: Что такое Test IT MCP?.

Test IT Про — версия системы Test IT, устанавливаемая на сервере клиента. Объем хранилища и количество пользователей не ограничены. Подробнее .

Test IT Базовый — облачная версия системы Test IT с ограниченным объемом хранилища, количеством пользователей и набором функций. Содержание инфраструктуры не требуется.

Test IT Менеджмент — плагин, для интеграции рабочих элементов Test IT в код IntelliJ IDEA. Подробнее: Плагин Test IT Менеджмент.

Test IT Стандарт — облачный тариф Test IT с ограниченным объемом хранилища, количеством пользователей, при этом доступен полный набор функций. Содержание инфраструктуры не требуется.

TMS (Test Management System) — система управления тестированием.

Автотест — автоматизированный тест ПО, который создается и проводится в сторонних системах. Автотест содержит сценарий для автоматизированного тестирования. Подробнее: Работа с автоматизированными тестами.

Адаптер — утилита для переноса автотестов и их результатов из тестовых фреймворков в Test IT, устанавливаемая в проект с автотестами. Адаптер разрабатывается для тестовых фреймворков индивидуально. Позволяет выводить дополнительные метаданные, использовать фильтры и различные способы конфигурирования. Подробнее: Адаптер Test IT.

Атрибут — свойство тест-кейса (например приоритет, статус) или тест-плана (например версия продукта), отображаемое в виде отдельной колонки. В Test IT доступны глобальные атрибуты, относящиеся ко всей системе и проектные атрибуты, относящиеся к отдельному проекту. Глобальные атрибуты можно объединять в наборы для массового использования.
В Test IT можно создать **шаблон атрибутов** — готовый набор глобальных атрибутов для массового использования в проектах.
Также доступны **атрибуты тест-планов** — глобальные атрибуты, применяемые к тест-планам. Добавляют соответствующие им поля во все тест-планы проекта. Подробнее: Настройка глобальных атрибутов; Настройка атрибутов в проекте.

Архив — хранилище неиспользуемых ручных тестов, тест-планов, конфигураций и проектов. Объекты, помещенные в архив, недоступны для редактирования и использования. Объекты могут быть восстановлены в течение установленного периода, по истечению которого они удаляются безвозвратно. Период хранения объектов в архиве устанавливается в настройках системы.

Библиотека тестов — раздел системы, где хранятся рабочие элементы проекта (тест-кейсы, чек-листы, общие шаги). Подробнее: Библиотека тестов.

Вебхук (англ. *webhook*) — способ автоматизированной передачи данных между приложениями при помощи HTTP-запросов (запрос отправляется при наступлении определенного события). Подробнее: Настройка вебхуков.

Дашборд — аналитический инструмент, позволяющий визуализировать выполнение тестов, загруженность инженеров и контролировать процесс выполнения работ. Подробнее: Работа с дашбордами.

Дефект (баг) — нарушение в работе ПО, вызванное ошибками в программном коде. Test IT позволяет отмечать результаты тестов и документировать (заводить) баги прямо из системы Test IT.

Запуск автотеста — процесс запуска скриптов или сценариев для автоматической проверки функциональности программного обеспечения. В Test IT запуски автотестов входят в прогон (тест-ран). Подробнее: Отслеживание запусков автотестов.

Импортер — утилита для автоматического экспорта тестов из сторонних систем управления тестированием (TMS). Для импорта тестов из любых TMS используется один импортер. Подробнее: Загрузка тест-кейсов в Test IT с помощью импортера

Категория ошибок — опция Test IT, позволяющая обучать систему Test IT распознавать типичные ошибки в автотестах с помощью регулярных выражений и категорий ошибок, ассоциирующихся с ними. Подробнее: Определение причины падения автотестов.

Конфигурация — технические характеристики окружения, в условиях которого проводится тестирование ПО (например операционная система определенной версии, браузер, и т. д.). Подробнее: Работа с параметрами и конфигурациями.

Лицензионный ключ (ключ продукта) — уникальный набор символов, вводимый пользователем в **Test IT Про** для проверки подлинности лицензии продукта и получения доступа к его функциональности.

Мигратор — набор утилит (экспортер и импортер), позволяющий автоматически перенести данные тестов из сторонних систем управления тестированием (TMS) в Test IT.

Нестабильный автотест (также *flaky autotest*) — автотест, показавший неодинаковый результат при отсутствии изменений в коде или тестовой среде. Такие тесты считаются ненадежными. Подробнее: Работа с нестабильными автотестами.

Общий шаг — шаг теста, который можно использовать в нескольких тест-кейсах, что позволяет сэкономить время на создание тестовых сценариев с типичными шагами. В Test IT общий шаг является рабочим элементом.

Отчет по тест-плану — инструмент для просмотра, сортировки и анализа данных. Информация отображается в виде таблиц, диаграмм и настраиваемых сводок. Отчет используется тест-менеджером для просмотра результатов тестирования, распределения задач и проверки выявленных дефектов. Отчет может быть сохранен в формате `PDF`, распечатан и отправлен по электронной почте. Подробнее: Отчет по тест-плану.

Падение теста — ошибка, непрохождение теста. Как правило, относится к автотестам. Test IT позволяет определять и анализировать причины падения автотестов.

Параметр — свойство или переменная ПО, подлежащее тестированию (например, имя пользователя или пароль) или характеристика тестового окружения (например, операционная система или браузер). Подробнее: Создание параметров.

Прогон (тест-ран) — запуск одного или нескольких автотестов при определенной конфигурации с указанием результата. Подробнее: Анализ результатов автотестов.

Проект — совокупность всех тестов, связанных с отдельным IT-продуктом (например приложением). Подробнее: Структура и компоненты проекта.

Проектная роль — набор прав пользователей для работы с проектами Test IT. В системе доступен набор преднастроенных проектных ролей (пользователь с ролью **ProjectAdmin** имеет полный доступ ко всем функциям проекта и его настройкам, а **TestExecutor** может только выполнять ручные и автоматизированные тесты). Подробнее: Настройка проектных ролей.

Рабочий элемент — основные элементы ручного тестирования: тест-кейс, чек-лист и общий шаг.

Результат теста — отметка об итоге проведения тест-поинта или прогона (тест-рана) автотеста (*Ожидает*, *Успешен*, *Заблокирован*, *Пропущен*, *Провален*). Подробнее: Результаты тестов.

Ручной тест — тест ПО, проводимый вручную. Подробнее: Ручное тестирование.

Секция — группа рабочих элементов внутри проекта, создаваемая для удобства навигации. В Test IT вы можете создать секцию и указывать пред- и постусловия для всех входящих в нее тестов.

Система — программная платформа Test IT.

Системная роль — набор системных прав пользователей для работы в системе. В Test IT доступны следующие системные роли:

- **Администратор** — позволяет добавлять лицензий, пользователей (в Test IT Про) и группы, создавать подключения, интеграции, проектные роли, глобальные атрибуты и проекты.
- **Руководитель проектов** — позволяет создавать и редактировать проекты, создавать проектные атрибуты, присваивать проектные роли.
- **Пользователь** — предусматривает минимальный уровень прав в системе. По умолчанию пользователь может только войти в свою учетную запись. Для совершения действий в системе пользователю нужна проектная роль.
- **Наблюдатель** (также **просмотровая лицензия**) — роль в **Test IT Про**, позволяющая просматривать данные в системе без возможности внесения изменений. Плата за просмотровые лицензии не взимается, свободных лицензий для них не требуется. При назначении пользователю роли **Наблюдатель** остальные системные роли снимаются. Подробнее: Присвоение системных ролей.

Таск-трекер (также багтрекер, CWM) — программное обеспечение для фиксации дефектов (багов), а также организации и автоматизации работ по их устранению (например Jira или Azure DevOps). Вы можете настроить интеграцию с таск-трекерами и документировать (заводить) задачи прямо из системы Test IT или создавать тест-кейсы по задачам с помощью ИИ-моделей.

Тег — слово-метка, сообщающее дополнительную информацию о рабочем элементе, тест-плане или автотесте. В Test IT вы можете сортировать и фильтровать рабочие элементы по тегам. Подробнее: Использование тегов.

Температура (ИИ) — это параметр искусственного интеллекта, определяющий случайность генерируемого ответа. От температуры зависит предсказуемость или вариативность поведения ИИ-модели. В Test IT применяется при генерации тестов. Можно выбрать температуру от 0 до 1, где 0 — результат, максимально приближенный к тексту задачи, а 1 — вариативный и менее предсказуемый результат.

Тестирование программного обеспечения — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. Test IT предоставляет полный набор инструментов для управления тестированием ПО.

Тестировщик — специалист по тестированию программного обеспечения. Test IT позволяет назначать тестировщиков вручную и автоматически, а также отслеживать их загруженность и участие в проектах.

Тест-кейс — тестовый сценарий для определенной функциональности (например запуск приложения, авторизация, и т. д.). В Test IT тест-кейс является рабочим элементом.

Тестовый набор — часть тест-плана, как правило включающая совокупность ручных тестов (тест-кейсов и тест-поинтов) одного компонента системы. В Test IT вы можете создавать тестовые наборы различных типов: пользовательские, секций библиотеки, вложенные и динамические тестовые наборы.

Тест-план — группа ручных тестов проекта, на которую назначаются тестировщики и в которой указываются результаты тестов и добавляются атрибуты. В тест-план также могут входить автотесты, привязанные к ручным тестам. Список тест-планов доступен в разделе **Тест-планы**.

Тест-поинт — ручной тест, проводимый по сценарию, описанному в тест-кейсе с определенными значениями параметров или конфигурациями. Подробнее: Структура и компоненты проекта.

Чек-лист — список проверок, позволяющий добавить шаги ручного теста, но не описывать предусловия и постусловия и ожидаемые результаты. Может быть конвертирован в тест-кейс. В Test IT чек-лист является рабочим элементом.

Шаблоны атрибутов — наборы глобальных атрибутов, предназначенные для массового использования в проектах. Подробнее: Создание шаблонов атрибутов.

Экспортер — утилита для автоматического экспорта тестов из сторонних TMS-систем. Для каждой TMS применяется отдельный экспортер. Подробнее: Миграция в Test IT: возможности и процесс.

Обновлено: 8/12/26, 4:43 PM