

# Что такое Test IT?

## О Руководстве пользователя Test IT

Это руководство описывает работу с Test IT в следующих поставках:

-  **Test IT Enterprise** — последняя версия
-  **Test IT Lite и Standard** (ранее Test IT Cloud) — 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 TMS с помощью **мигрантора**.

## 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 Management**.

## 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 task-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**.

# Администрирование Test IT

Данный раздел описывает работу администратора Test IT по настройке системы после установки. 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 лицензий
-  Приглашение пользователей
- Добавление пользователей и групп
- Присвоение системных ролей
- Настройка проектных ролей
- Настройка очистки хранилища
- Настройка атрибутов
- Настройка интеграций с таск-трекерами
- Настройка интеграций с ИИ-моделями

# Вход в систему после установки

## Функциональность Test IT Enterprise

Данная инструкция описывает вход в систему после установки **Test IT Enterprise**.

Если вы используете **Test IT Lite** или **Standard**, обратитесь к соответствующей инструкции в **Личном кабинете**.

## Вход

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

1. Введите данные служебного пользователя ( `admin` / `Qwerty123` ).
2. 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: 16.10.2025, 18:35:17

# Внешние подключения в Test IT

## Функциональность Test IT Enterprise

Данная инструкция описывает настройку подключений в **Test IT Enterprise**.  
Если вы используете облачную версию **Test IT — Lite** или **Standard**,  
пропустите этот раздел.

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

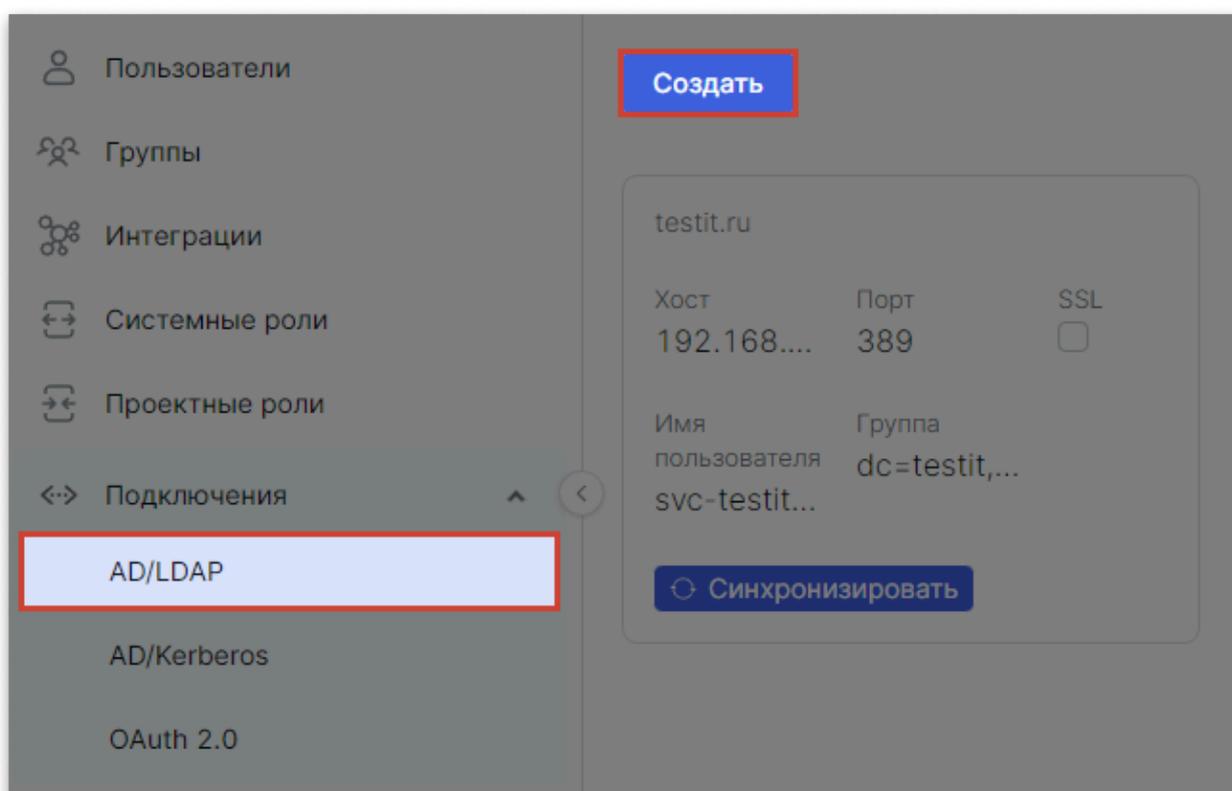
- AD/LDAP
- AD/Kerberos
- OpenID Connect (OAuth 2.0)

# 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**. В случае 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:

1	<pre>(&amp;(objectCategory=Person)(sAMAccountName=*)(memberOf=CN=Group-Sonya,OU=Sonya-test,OU=podrazdelenie2,OU=Podrazdelenie1,DC=mtest,DC=ru))</pre>	text
---	---	------

Пример фильтра, который добавляет пользователей из вложенных групп:

1	<pre>(&amp;(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))</pre>	text
---	--	------

## Group Schema

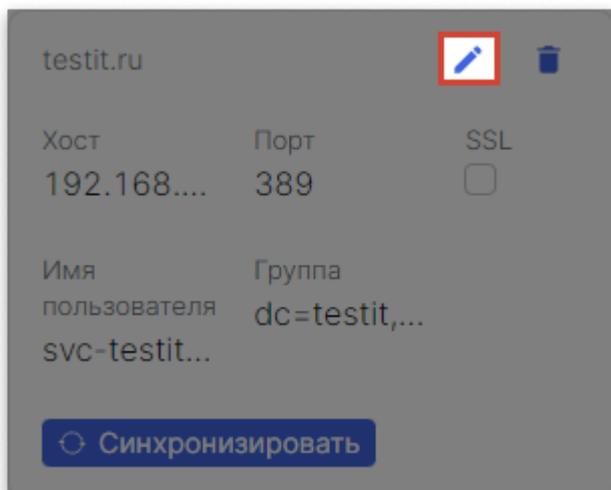
Фильтр для одной группы (имя начинается с):

1	<pre>(&amp;(objectCategory=group)(name=Group-Sonya*))</pre>	text
---	---	------

## 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: 27.02.2025, 20:31:47

## Поля настроек подключения

#	Поле	Значение	Пример
1	Имя	Имя соединения в разделе <b>Администрирование</b> Test IT	My connection
<b>Settings</b>			
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 (чекбокс)	-
8	Paging Mode	Режим пагинации (выпадающее меню)	-
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.	-
11	Bind Referral Requests	Включает повторное связывание реферальных запросов (referral requests)	-

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: 06.10.2025, 11:55:48

# 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).

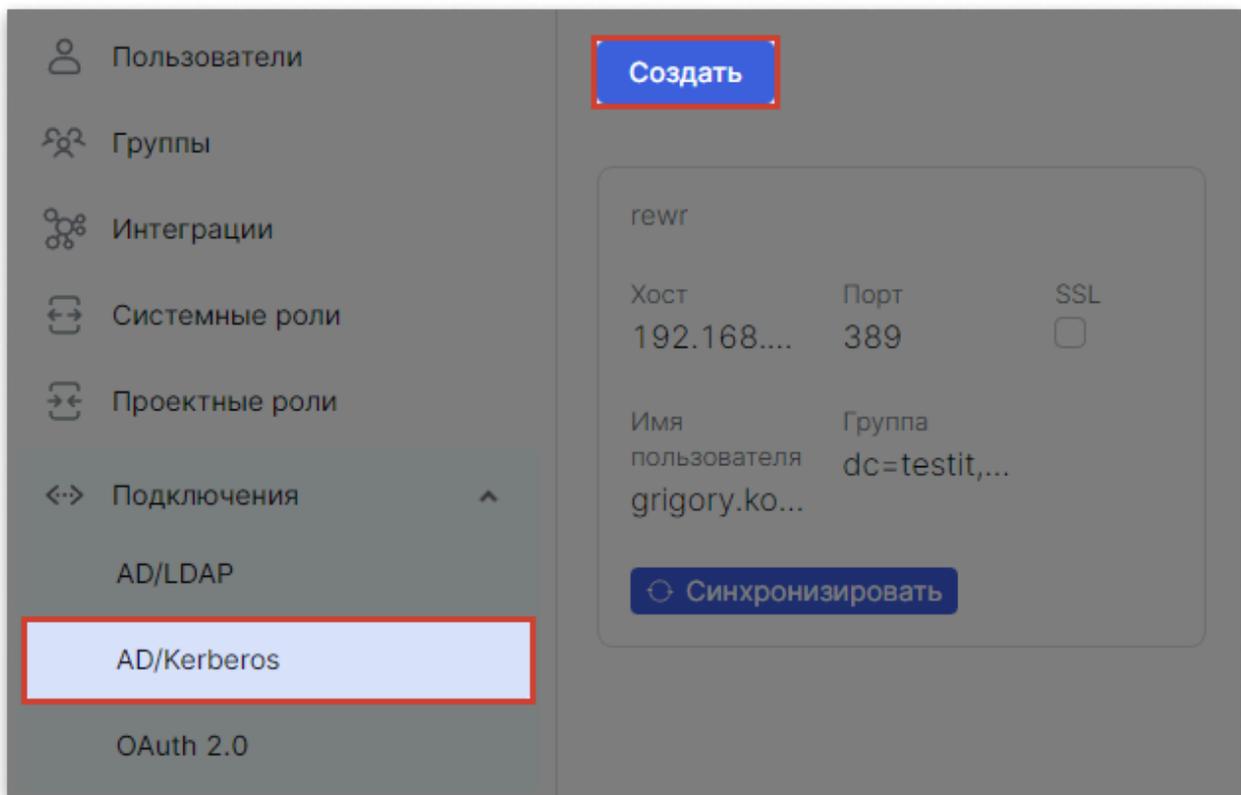
1. Добавьте пользователя в контроллер домена, от имени которого будет происходить синхронизация с системой дистрибуции ключей (KDC).
2. Сгенерируйте файл `.keytab` на контроллере домена, используя утилиту `ktpass`:

```
1 | ktpass.exe /princ HTTP/application_url@YOURDOMAIN /mapuser text  
   | 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.

# 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.

# Добавление лицензий

## Функциональность Test IT Enterprise

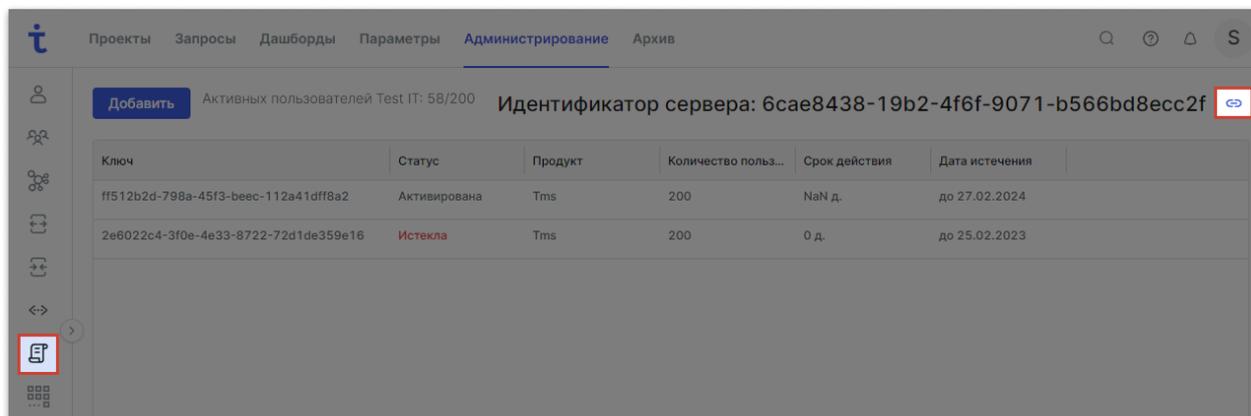
- Данная инструкция описывает добавление лицензий в **Test IT Enterprise**. Если вы используете **Test IT Lite** или **Standard**, обратитесь к инструкции по оформлению подписки и отслеживанию лицензий и платежей в Личном кабинете.
- Если активная лицензия отсутствует, Test IT работает преимущественно в режиме просмотра. Подробности читайте в соответствующем разделе.

## Добавление лицензий

Чтобы добавить лицензию, вам потребуется 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.

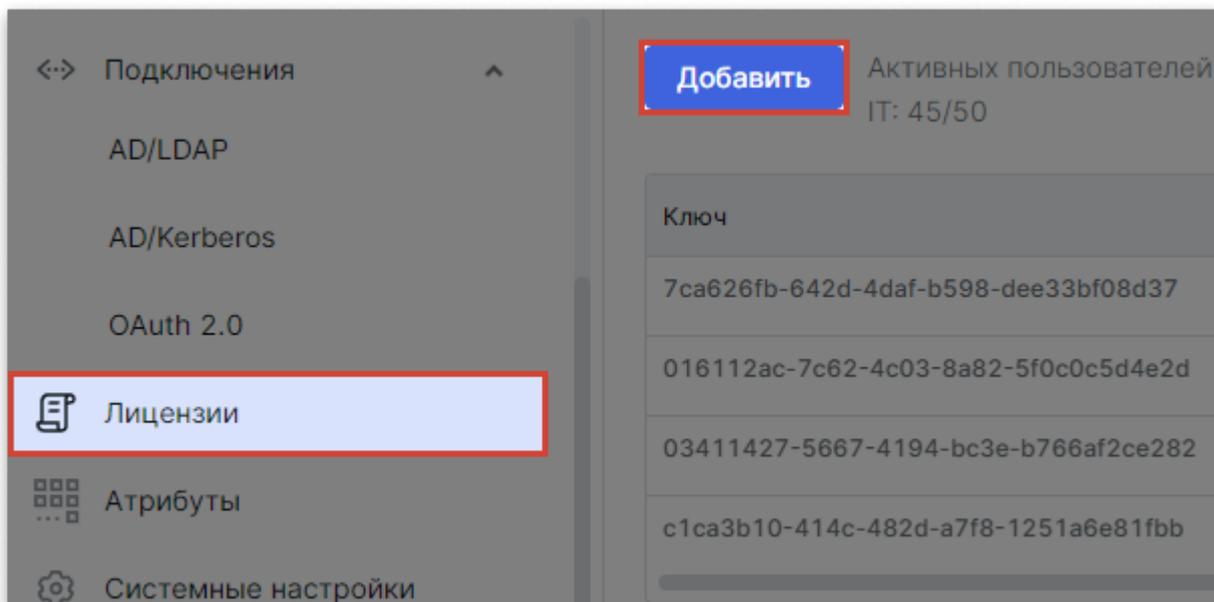


3. Отправьте сообщение на электронный адрес отдела продаж Test IT: **order@testit.software**. При обращении укажите идентификатор сервера в теме письма или в его тексте.  
Ключи лицензий будут отправлены в ответном сообщении.

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 сохраняются. При входе в аккаунт администратора вы получите уведомление об оставшемся сроке пробного периода.

## Просмотр информации о добавленных лицензиях

Чтобы найти информацию о лицензиях:

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

The screenshot shows the 'Лицензии' section of the software interface. The left sidebar has 'Лицензии' highlighted. The main content area shows a table of licenses. The table has columns: Ключ, Статус, Продукт, Количество поль..., and Срок действия. The table contains three rows of license data.

Ключ	Статус	Продукт	Количество поль...	Срок действия
b709374f-43a0-4662-b9bb-9efb788e55cb	Активирована	Tms	50	192 д.
44815cf0-0125-4414-bb1e-57af681a05b0	Истекла	Tms	3	0 д.
c50ece26-b849-46c9-bf18-a95af27d98c3	Истекла	Tms	20	0 д.

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

# ☁ Приглашение пользователей

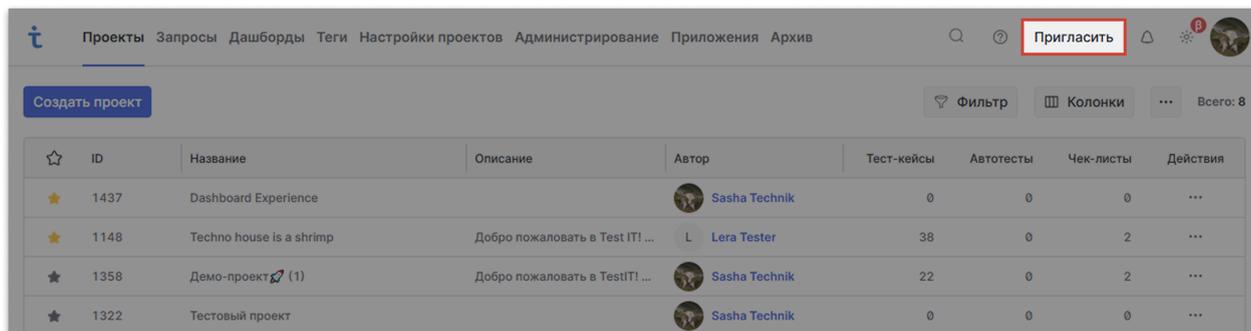
## ☁ Функциональность Test IT Lite & Standard

- Данная инструкция описывает работу с пользователями в облачной версии **Test IT — Lite** и **Standard**. Если вы используете **Test IT Enterprise**, обратитесь к инструкции по добавлению пользователей.
- Вы можете добавлять в одно пространство до 100 пользователей (тариф **Standard**).

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.

## Пригласить коллег



Сила и удобство – в командной работе: пригласите коллег (инженеров, руководителей, администраторов) и начните тестировать проекты вместе!

alien.starling@space2001.com, sveta.medoed@testit.com

Руководитель проектов ▾

▶ Пригласить

Вы сможете назначить коллегам проектные роли после того, как они примут приглашения

### 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 статье **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.

Логин	Имя пользователя	Администратор Test IT	Руководитель проектов	Пользователь Test IT
yuriy.gorbanev	Юрий	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
elena.skvortsova	AlienStarling	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
valeriya.goldberg	Lera Tester	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
aaron.schneider	Ari Schneider	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

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.

Логин	Имя пользователя	Администратор Test IT	Руководитель проектов	Пользователь Test IT
yuriy.gorbanev	Юрий	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
valeriya.goldberg	Lera Tester	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
aaron.schneider	Johhny Mandel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
alexander.panov	Sasha Technik	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

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

# Добавление пользователей и групп

- Для этого раздела мы подготовили видео:

## Работа с пользователями

---

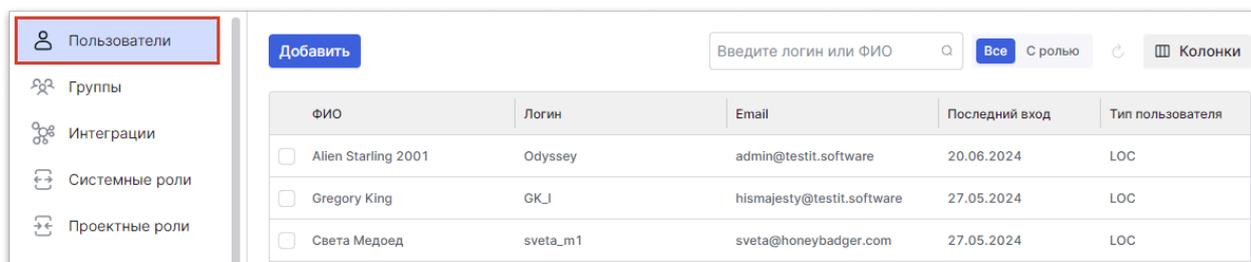
Используйте инструкции для своей поставки Test IT

-  Инструкции по добавлению пользователей, редактированию их данных и удалению описывают функциональность **Test IT Enterprise**.
  -  Если вы используете **Test IT Lite** или **Standard**, обратитесь к инструкции по приглашению пользователей из **Личного кабинета** или **рабочего пространства Test IT**.
-   Инструкции по работе с группами пользователей и присвоению группам системных ролей актуальны для пользователей всех тарифов.

Вы можете добавлять, удалять и редактировать данные пользователей, добавленных в Test IT Enterprise 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**.

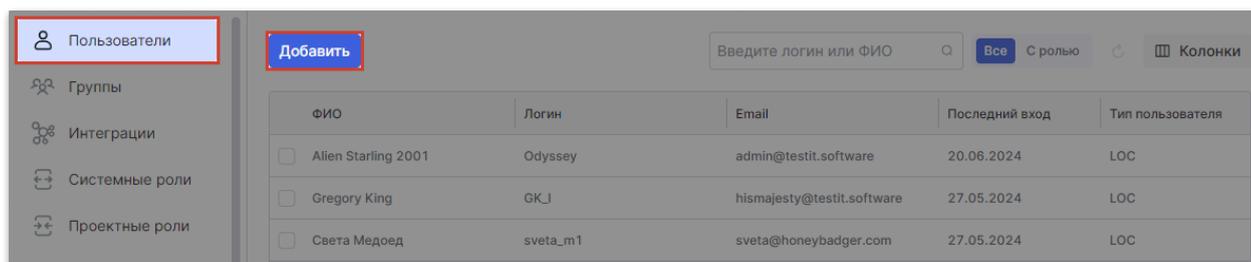
2. В навигационном меню слева выберите **Пользователи**.



## Добавление пользователей

Вы можете добавлять локальных пользователей в систему Test IT Enterprise:

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 пользователя

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

Чтобы редактировать данные пользователя:

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

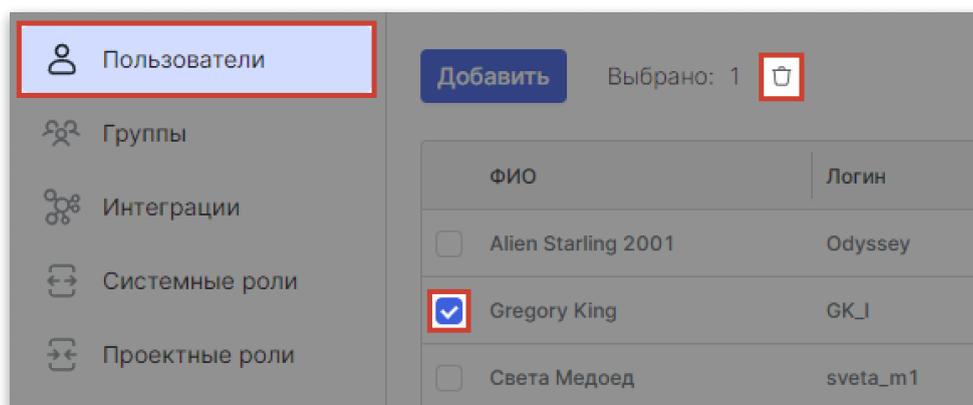
## Важно

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

## 🗑️ Удаление пользователей

Чтобы удалить локальных пользователей из системы Test IT Enterprise:

1. Поставьте флажок напротив локального пользователя (пользователей), которого хотите удалить из системы.
2. Нажмите **Удалить**.
3. Подтвердите удаление в диалоговом окне.



## Работа с группами пользователей

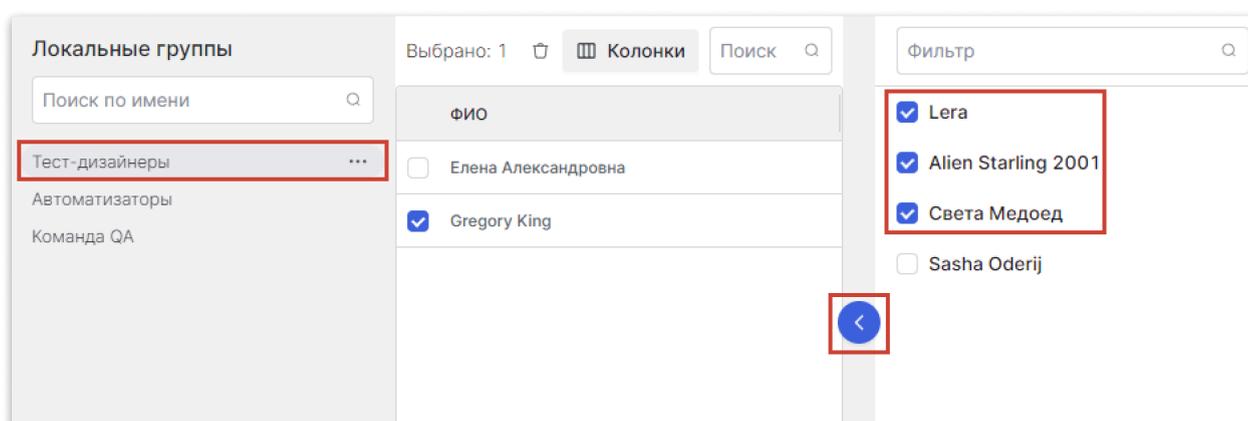
- Для этого раздела мы подготовили видео:

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**.

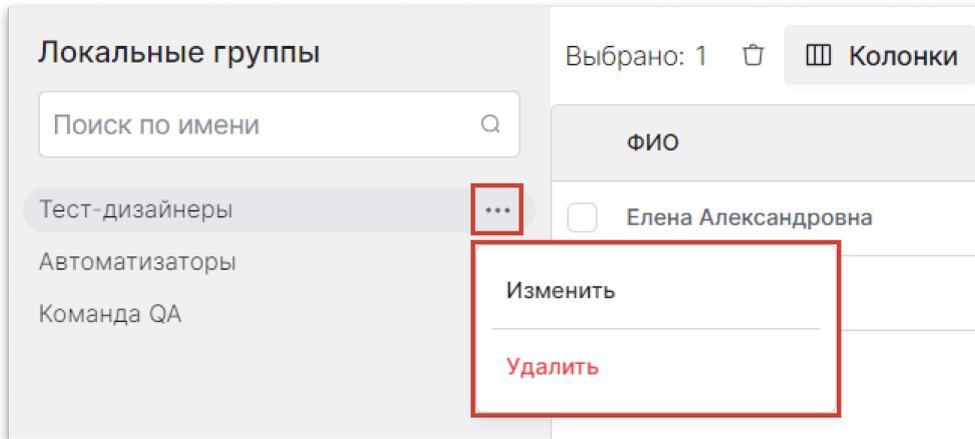
## Добавление, редактирование и удаление локальных групп

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: Администратор, Руководитель проектов или Пользователь, сделав отметку в нужном столбце.

# Присвоение системных ролей

- Для этого раздела мы подготовили видео:

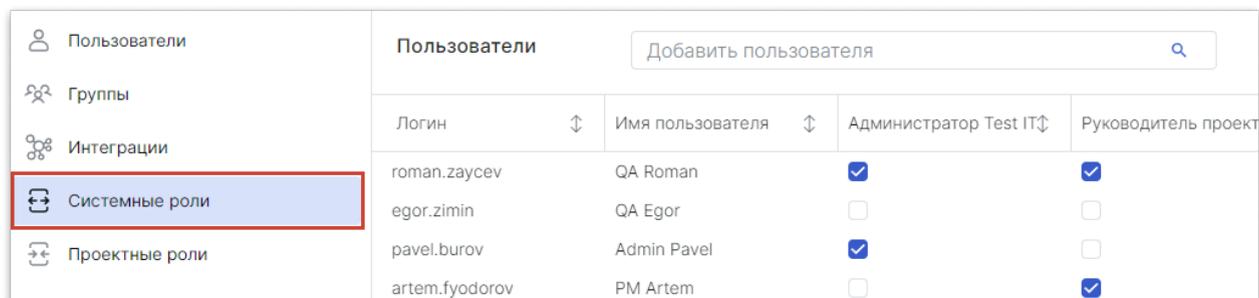
Для входа в систему пользователю необходима системная роль. Test IT предусматривает следующие системные роли, отличающиеся уровнем прав в системе:

- **Администратор** — данная роль предусматривает самые полные права в системе. Администратору доступно добавление лицензий, пользователей (в версии **Test IT Enterprise**, в **Test IT Lite** и **Standard** 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 просмотрные лицензии не взимается, свободных лицензий для них не требуется. При назначении пользователю роли **Наблюдатель** остальные системные роли снимаются. В **Test IT Lite** и **Standard** (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.



Пользователи		Добавить пользователя	
Логин	Имя пользователя	Администратор Test IT	Руководитель проекта
roman.zaycev	QA Roman	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
egor.zimin	QA Egor	<input type="checkbox"/>	<input type="checkbox"/>
pavel.burov	Admin Pavel	<input checked="" type="checkbox"/>	<input type="checkbox"/>
artem.fyodorov	PM Artem	<input type="checkbox"/>	<input checked="" type="checkbox"/>

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

**Пользователям Test IT Enterprise:** 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**.

# 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](#).

## 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.

<ul style="list-style-type: none"> <li>Пользователи</li> <li>Группы</li> <li>Интеграции</li> <li>Системные роли</li> <li><b>Проектные роли</b></li> </ul>	<b>Создать роль</b>				
		ProjectAdmin	TestManager	TestDesigner	TestEngineer
	Фильтры	Полный доступ	Полный доступ	Полный доступ	Полный доступ
	Редактирование проекта	Полный доступ	Запрещено	Запрещено	Запрещено
Библиотека тестов	Полный доступ	Полный доступ	Редактирование	Редактирование	

В системе доступны следующие преднастроенные проектные роли

- **ProjectAdmin** — полный доступ ко всей функциональности проекта, включая его настройку.
- **TestManager** — доступны все опции проекта кроме его настроек.
- **TestDesigner** — 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**.
- **TestEngineer** — доступ ко всем разделам без права удаления сущностей.
- **TestExecutor** — роль для прохождения ручных тестов и запуска автотестов. Зона ответственности — раздел **Тест-планы**, вкладка **Выполнение**.
- **Observer** — минимальный доступ к проекту, только просмотр элементов. Вы можете удалить все преднастроенные роли, кроме **ProjectAdmin** — данная роль обязательна для любого проекта. Чтобы удалить роль, нажмите значок удаления рядом с ее названием.

3. Нажмите **Создать роль**.

<ul style="list-style-type: none"> <li>Пользователи</li> <li>Группы</li> <li>Интеграции</li> <li>Системные роли</li> <li><b>Проектные роли</b></li> </ul>	<b>Создать роль</b>	
		ProjectAdmin
	Фильтры	Полный доступ
	Редактирование проекта	Полный доступ
	Библиотека тестов	Полный доступ

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

5. Включите режим редактирования роли.

	ProjectAdmin	TestManager	 
Архив	Полный доступ	Полный доступ	

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, в котором хранятся карточки автотестов
  - **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.

### 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**.

# 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**.

Пользователи

Группы

Интеграции

Системные роли

Проектные роли

Системные настройки

АДМИНИСТРИРОВАНИЕ/СИСТЕМНЫЕ НАСТРОЙКИ

При включенных настройках объекты будут безвозвратно удалены из системы после установленного количества дней

Тест-кейсы, чек-листы, общие шаги удаляются из архива через: 15 дней

Вложения из результатов тестов удаляются через: 30 дней

Автоматическое очищение архивов через: 30 дней

Автоматическое очищение архива автотестов через: [ ] дней

Автоматическое очищение архива прогонов через: 30 дней

Включить геймификацию для всех пользователей

Использовать "Умеренный" режим геймификации  
Пользователи будут видеть уровни, баллы и статистику достижений, но не будут видеть названия и изображения достижений и уровней

Выключить геймификацию для всех пользователей

Сохранить

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.

<b>Объект / действие</b>	 <b>Enterprise</b>	 <b>Lite &amp; Standard</b>
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: 11.11.2025, 21:27:49

# Настройки геймификации

## ₽ Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тариф **Lite** она не входит.

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.

Пользователи

Группы

Интеграции

Системные роли

Проектные роли

**Системные настройки**

### ADMINISTRATION/SYSTEM SETTINGS

При включенных настройках объекты будут безвозвратно удалены из системы после установленного количества дней

- Тест-кейсы, чек-листы, общие шаги удаляются из архива через: 15 дней
- Вложения из результатов тестов удаляются через: 30 дней
- Автоматическое очищение архивов через: 30 дней
- Автоматическое очищение архива автотестов через: [ ] дней
- Автоматическое очищение архива прогонов через: 30 дней

Включить геймификацию для всех пользователей

Использовать "Умеренный" режим геймификации  
Пользователи будут видеть уровни, баллы и статистику достижений, но не будут видеть названия и изображения достижений и уровней

Выключить геймификацию для всех пользователей

**Сохранить**

4. Нажмите **Сохранить**.

Обновлено: 17.10.2025, 13:38:51

# 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.

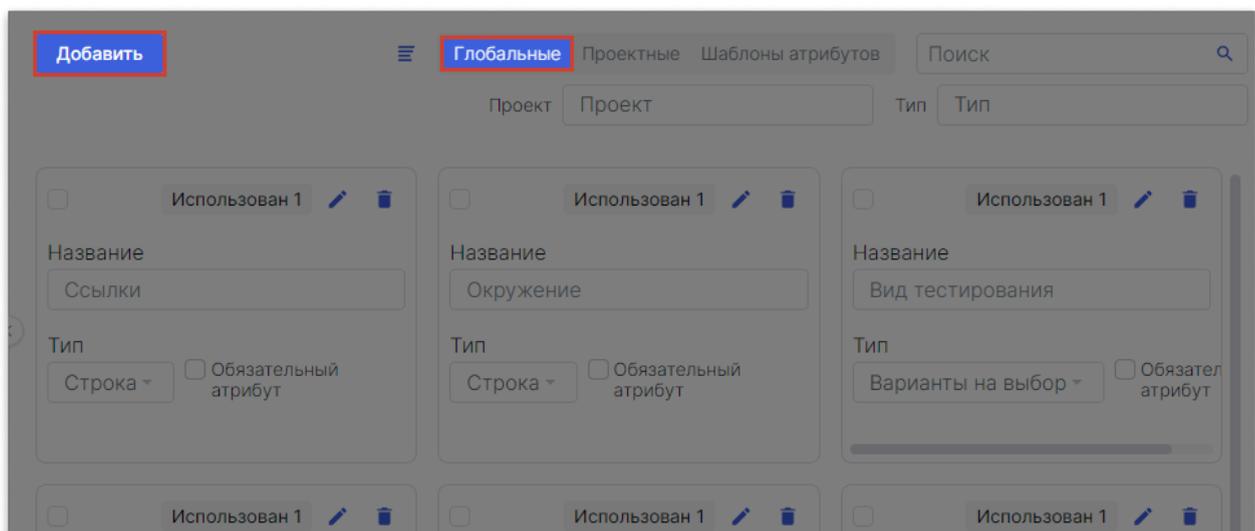
## Пример из практики

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**.

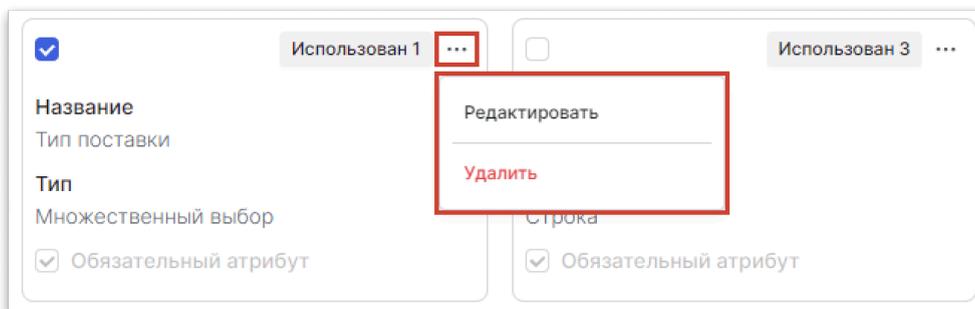


3. В открывшемся окне введите название нового атрибута, выберите его тип: **Строка, Дата, Варианты на выбор, Множественный выбор, Пользователь и Чекбокс**. Для типов **Варианты на выбор** и **Множественный выбор** введите необходимые значения. Например, для атрибута "Команда" значениями могут быть команды,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

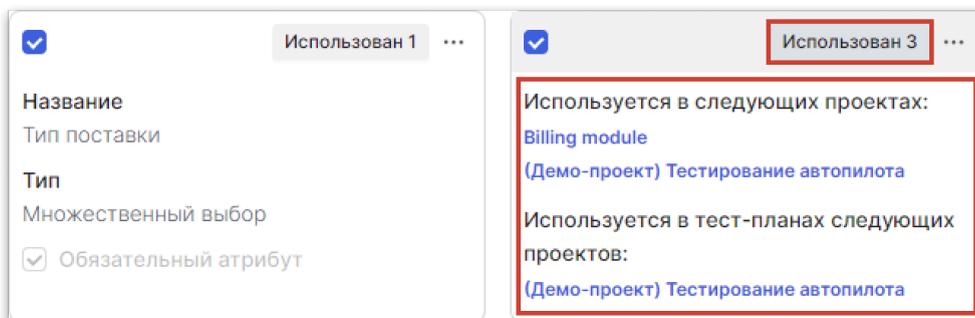
Чтобы использовать глобальные атрибуты в тестах проекта, их нужно **добавить в проект**. Для использования в тест-планах их необходимо **добавить в атрибуты тест-плана**.

Вы можете:

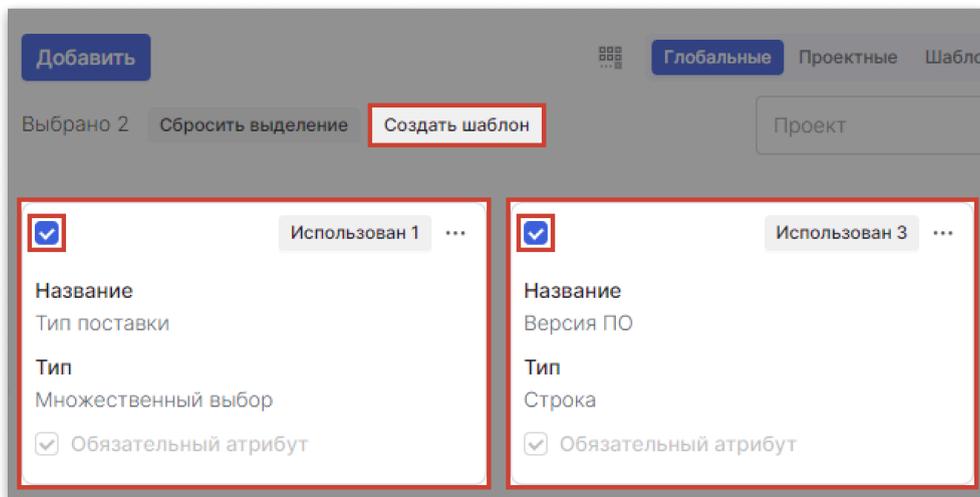
- Отредактировать атрибут или удалить его из системы, используя меню действий в карточке или строке атрибута.



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

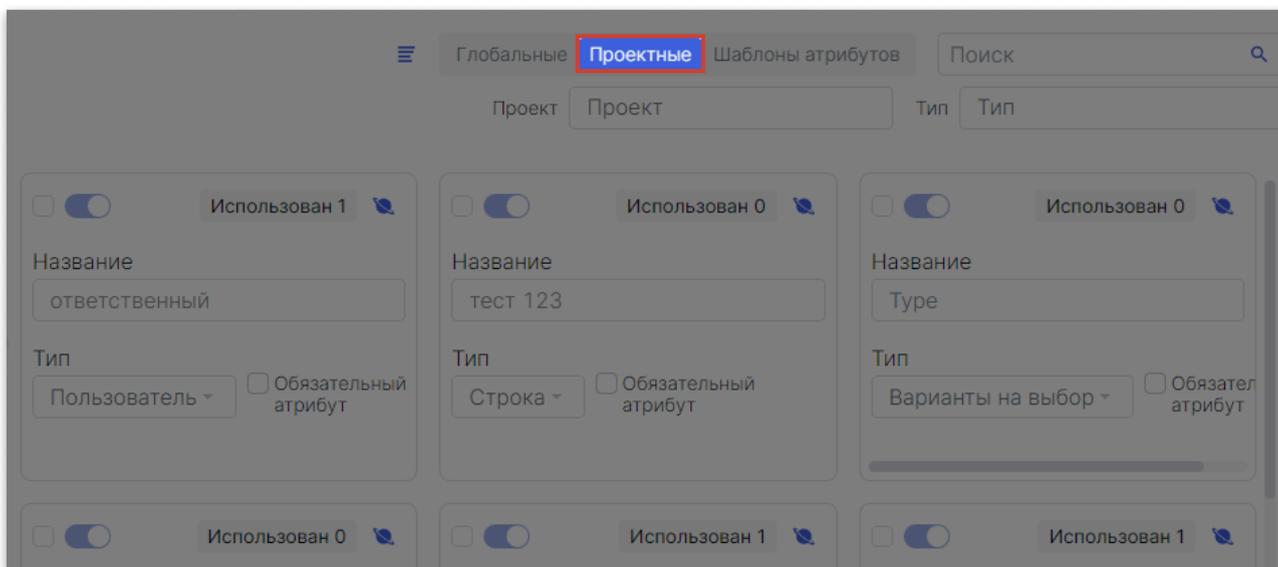


- Создать шаблон атрибутов, выбрав несколько карточек глобальных атрибутов и нажав **Создать шаблон** в левом верхнем углу окна атрибутов.



## Работа с проектными атрибутами

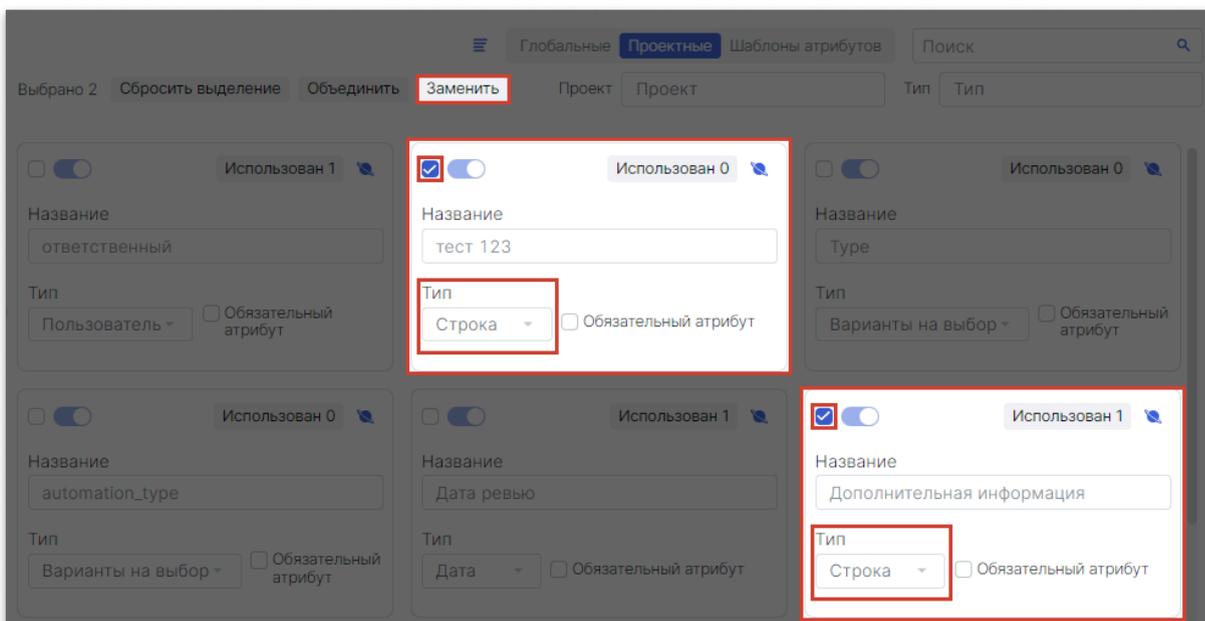
В разделе **Атрибуты** отображаются карточки всех проектных атрибутов, созданных в Test IT. Для этого необходимо перейти на вкладку **Проектные**.



Вы можете:

- Сделать проектный атрибут глобальным. В этом случае атрибут 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.



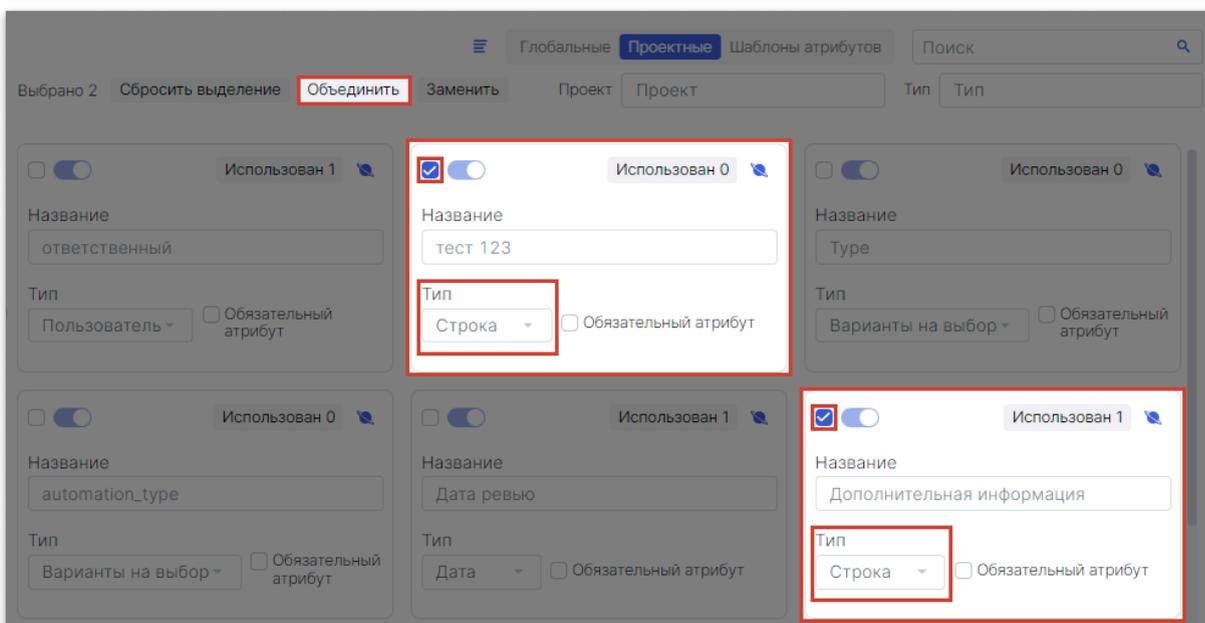
3. Выберите глобальный атрибут из открывшегося списка и нажмите **Сохранить**.

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](#).

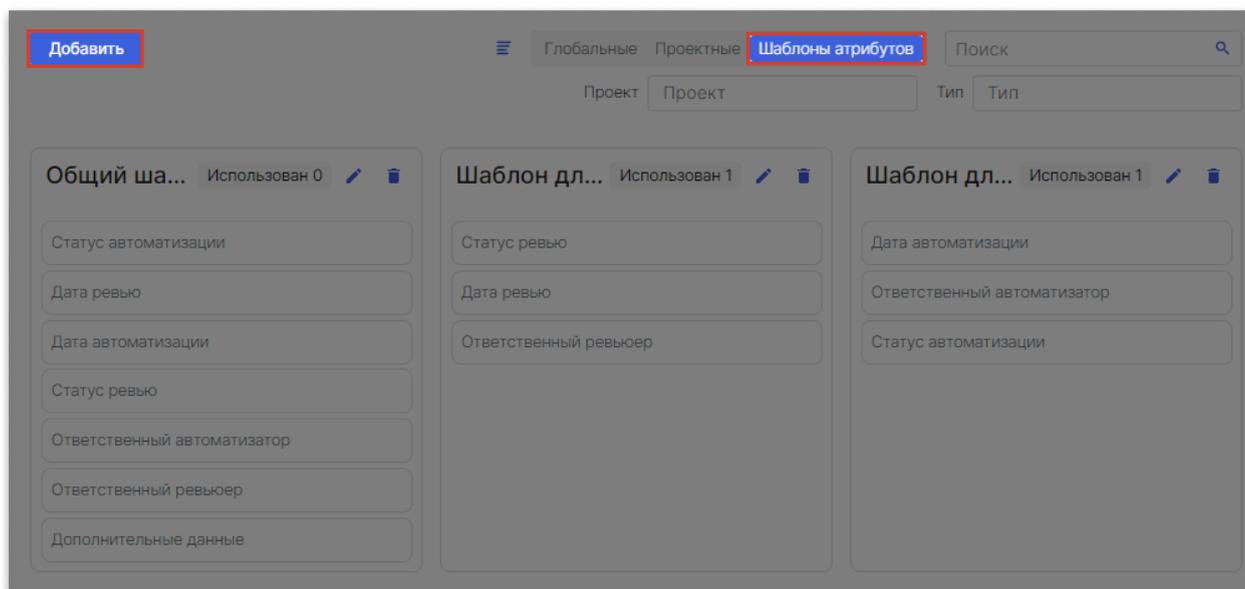
## Создание шаблонов атрибутов

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.

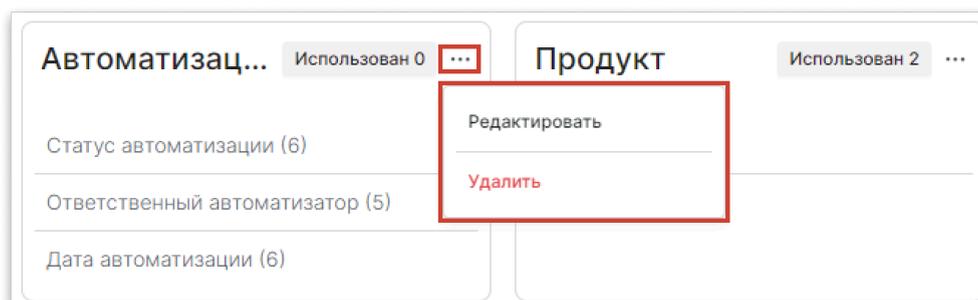
1. Используя аккаунт с правами администратора, нажмите **Настройки проектов** на верхней навигационной панели.
2. В навигационном меню слева выберите **Атрибуты**, перейдите на вкладку **Шаблоны атрибутов** и нажмите **Добавить**.



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: 11.11.2025, 21:27:49

# Интеграция с таск-трекерами

- 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 Yandex Tracker
- Интеграция с YouTrack

Обновлено: 12.10.2025, 20:50:48

# Интеграция с Azure DevOps

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

### Create a new personal access token ✕

Name

Organization

Expiration (UTC)

Scopes  
Authorize the scope of access associated with this token  
Scopes  Full access  
 Custom defined

---

**Work Items**  
Work items, queries, backlogs, plans, and metadata  
 Read  Read & write  Read, write, & manage

---

**Code**  
Source code, repositories, pull requests, and notifications  
 Read  Read & write  Read, write, & manage  Full  Status

---

**Build**  
Artifacts, definitions, requests, queue a build, and update build properties  
 Read  Read & execute

---

**Release**  
Read, update, and delete releases, release pipelines, and stages  
 Read  Read, write, & execute  Read, write, execute, & manage

[Show all scopes \(28 more\)](#)

6. Нажмите **Create**.

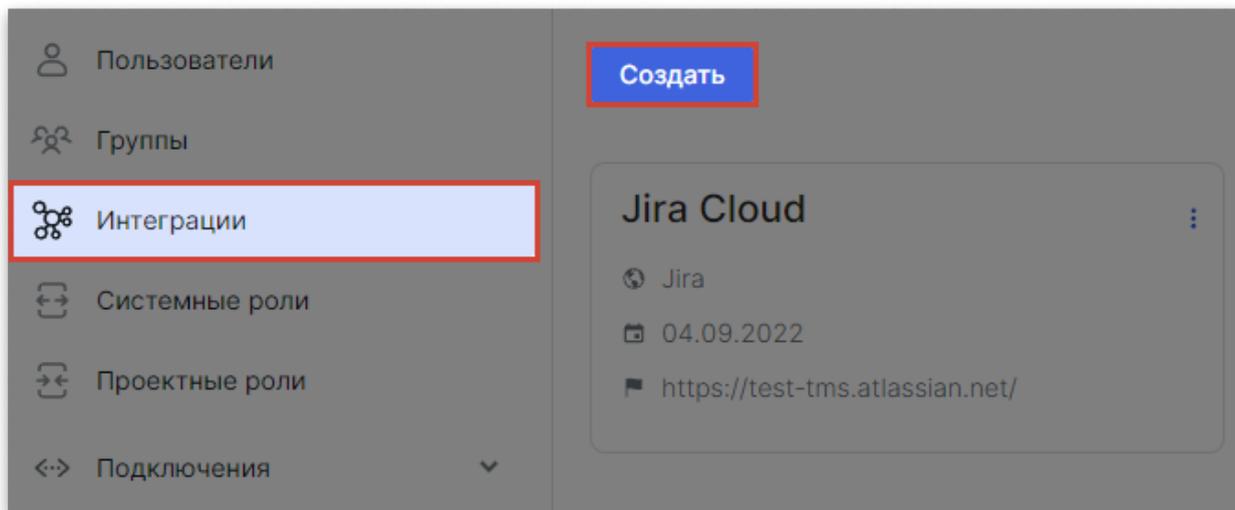
7. Скопируйте сгенерированный токен. Убедитесь, что вы сохранили токен, так как Azure DevOps не хранит его. Вы можете потерять токен при закрытии окна создания.

## 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: 21.12.2025, 20:53:46

# Интеграция с ClickUp

## 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@yoonion.ru](mailto:support@yoonion.ru).
- 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

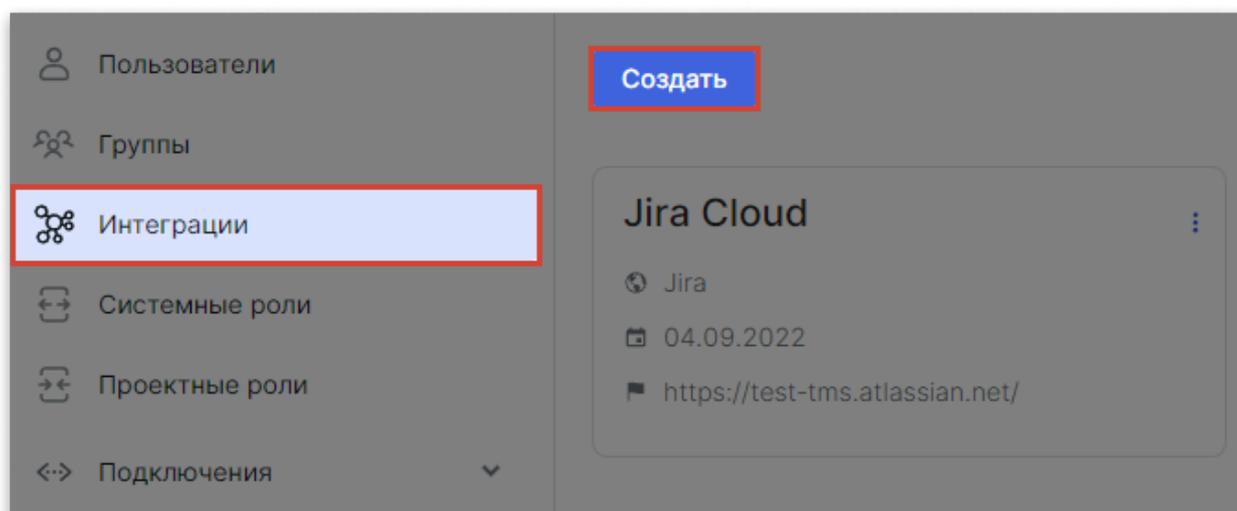
## Получение токена доступа к API в ClickUp

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 страницу 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.

### Редактировать подключение ✕

**Сервис для интеграции**

 ClickUp ▾

Тип установки:  Cloud  Self-hosted

**Данные для подключения**

\* Имя интеграции

Click UP

\* URL

\* Token

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**.

# 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.

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:

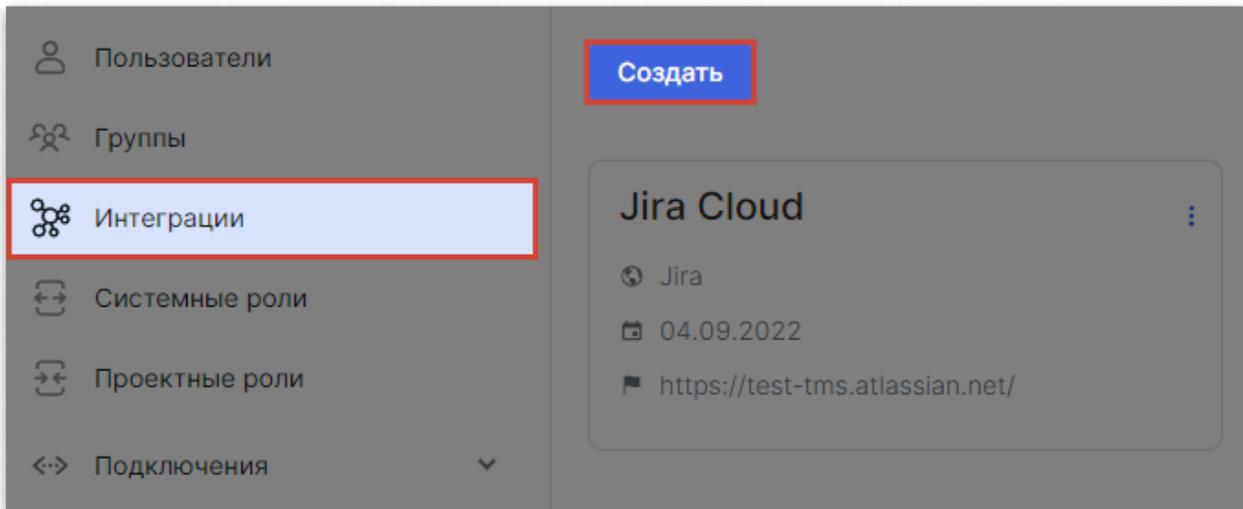
1. Войдите в аккаунт Easy Redmine.
2. Нажмите на значок профиля пользователя в правом верхнем углу окна, затем нажмите **Редактировать**. Откроется окно редактирования профиля.
3. На вкладке **Общее** пролистайте страницу. Ключ расположен в правой нижней части страницы.
4. Скопируйте ключ доступа в поле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

# Интеграция с EvaProject

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

---

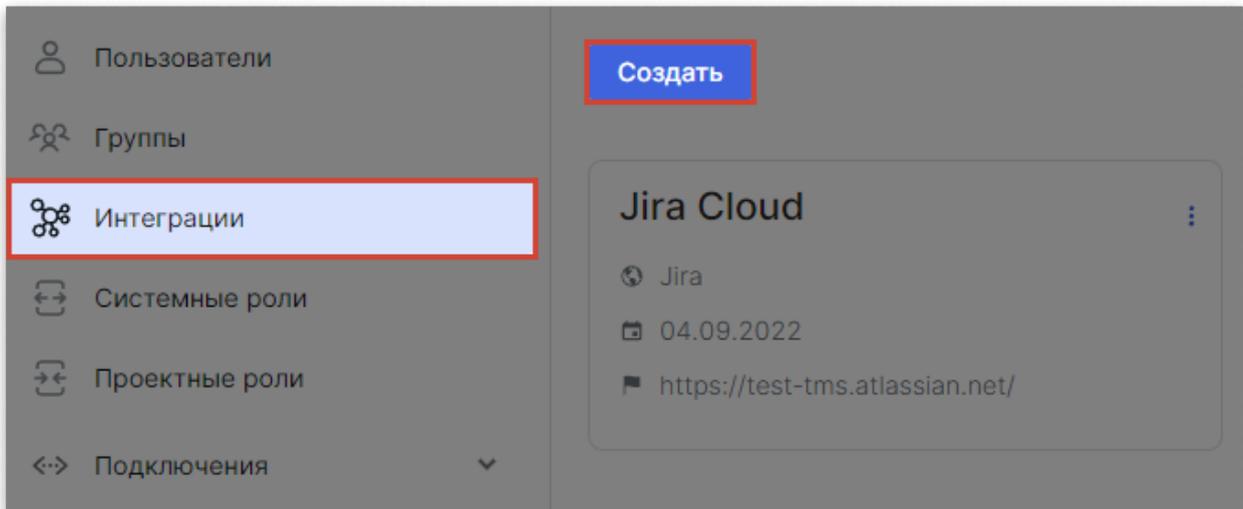
1. Создайте специального пользователя для работы с API, например `apiuser@example.com`.
2. Авторизуйтесь под созданной учетной записью.
3. Нажмите **Моя страница**.
4. Откройте вкладку **Безопасность**.
5. Нажмите **Сгенерировать API-токен**.
6. Сохраните полученный токен. Подробности — в [официальной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

# 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.

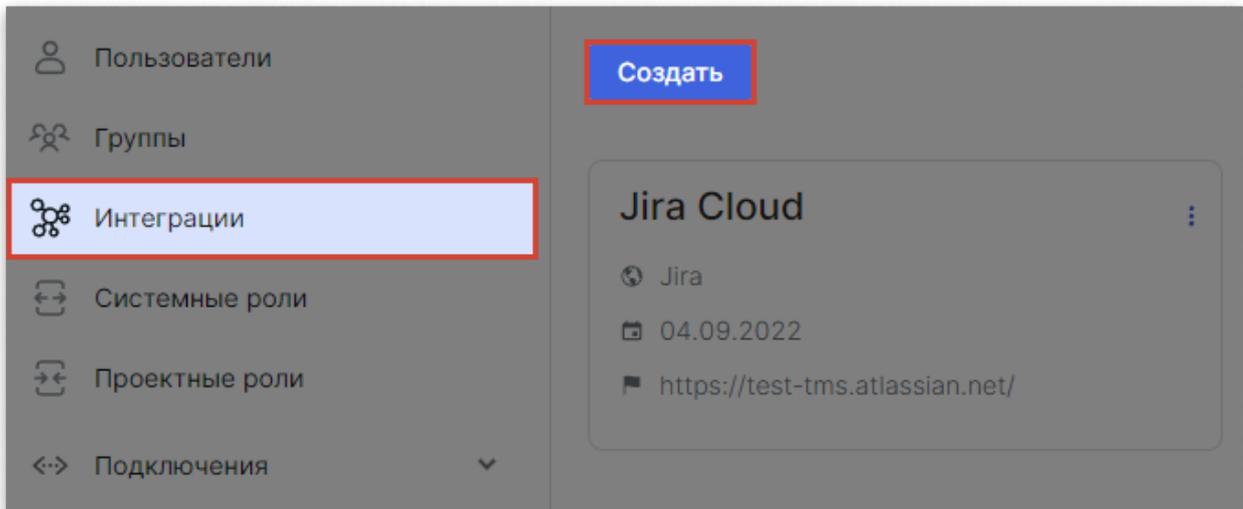
## Создание интеграции с Test IT

### 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**.

### Test Issue

📎 Attach
📄 Create subtask
🔗 Link issue
▼
⋮

Description

**Build:** 1.0.2322

**Test case:** [6894 Ссылка на лицензионное соглашение](#)

**Test result:** <http://best-tms.testit.ru/projects/38/test-plans/6575/results?testResultId=4757f59a-0faa-42b5-9713-64835bc60b4c>

**Configuration:** macOS + Firefox v2

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**.

Preconditions				
#	Action	Expected	StepLink	Last result
1	12			● [Passed]
2	13			● [Passed]
3	14			● [Passed]
Steps				
#	Action	Expected	StepLink	Last result
1	15			● [Passed]
2	16			● [Passed]
Postconditions				
#	Action	Expected	StepLink	Last result
1	11			● [Passed]

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

Обновлено: 27.08.2025, 20:37:19

# 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

---

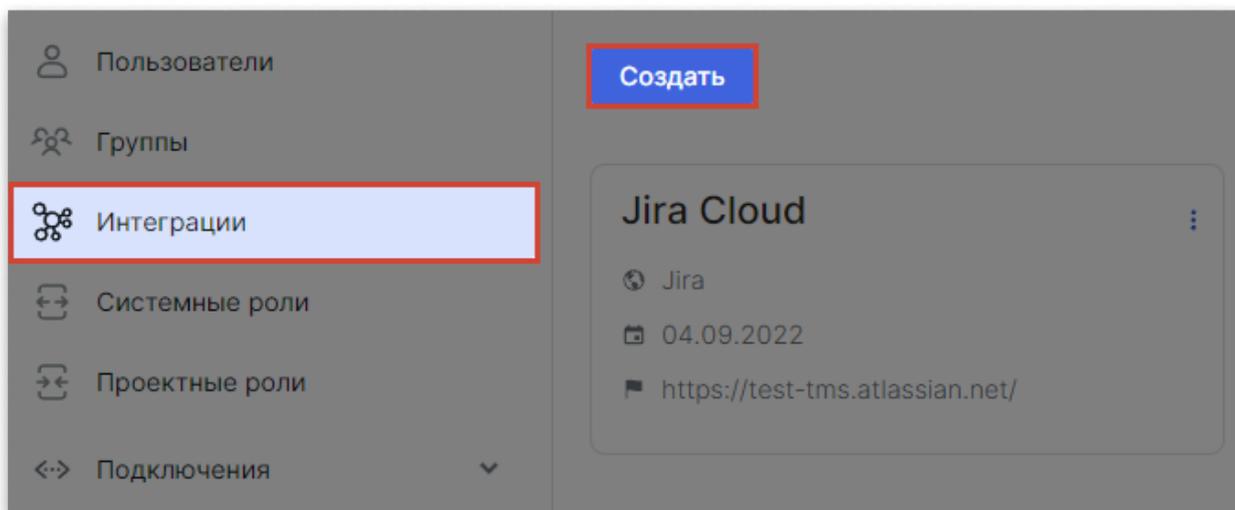
На данном этапе создается Consumer Key (в Jira — *Ключ покупателя, Ключ потребителя*) и копируется Публичный ключ, а также указывается URL учетной записи Jira, с которой создается подключение.

## 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

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: VHIVT1YhkRREuhITJKULeJ6yfnmAu6fb Access token secret: wf4veCDUi9Guwk8XQ0CE7RefwEDuT4Vd.*
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: 27.08.2025, 20:37:19

# Интеграция с Kaiten

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

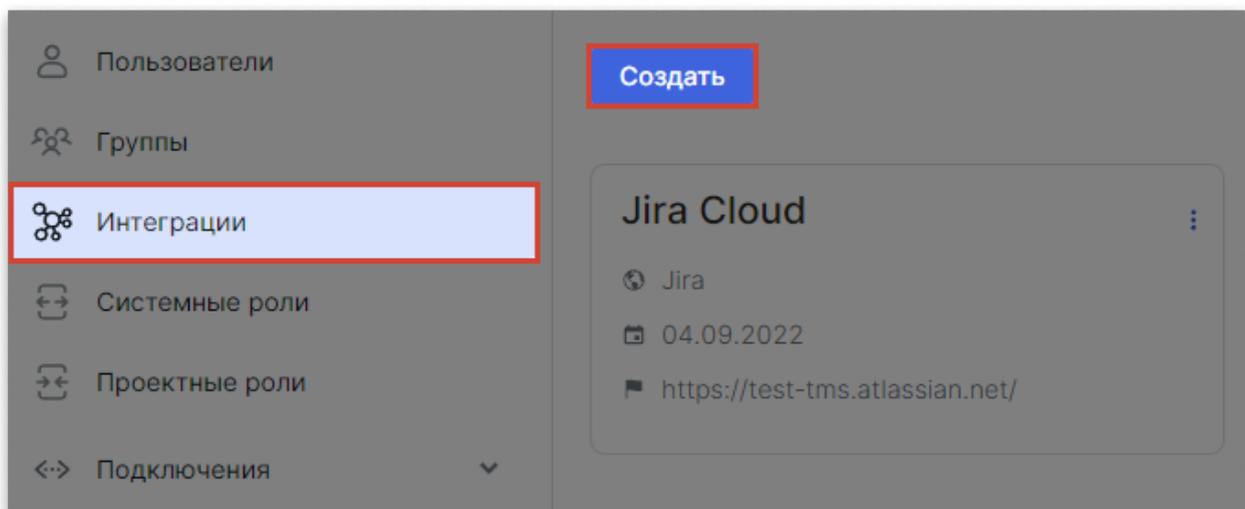
- Получить API-ключ Kaiten
- Создать интеграцию в Test IT

## 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**.

# Интеграция с Monday

## Доступна интеграция с облачной версией

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

## 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@yoonion.ru](mailto:support@yoonion.ru).

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

---

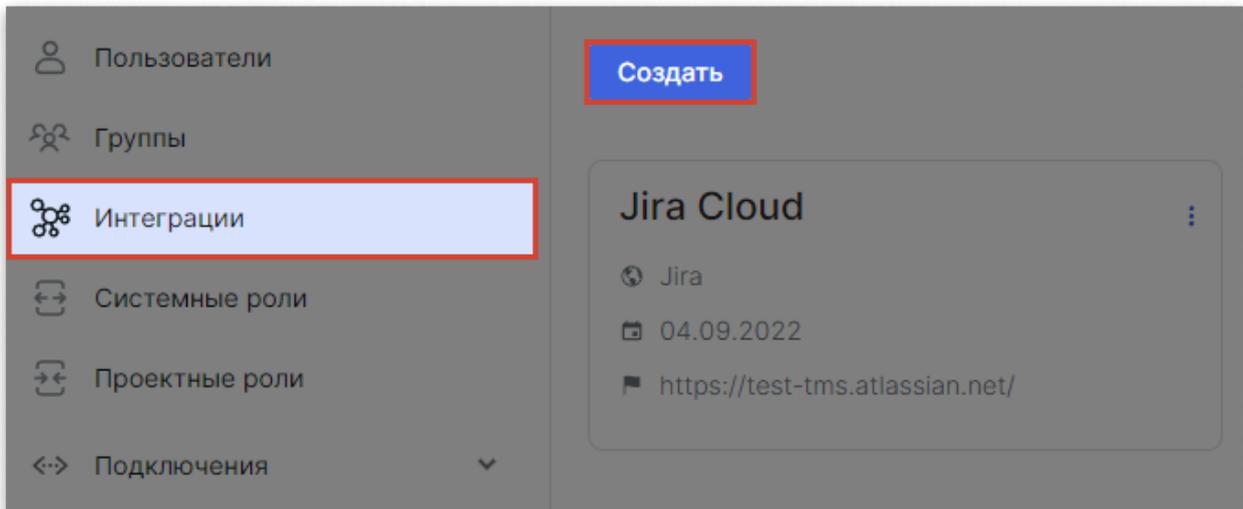
1. Войдите в аккаунт Monday.
2. Нажмите на значок профиля пользователя в правом верхнем углу окна, затем выберите **Администрирование (Administration)**. Откроется окно администрирования.
3. В меню, расположенном слева, выберите **API**.
4. Нажмите значок копирования, чтобы скопировать API-токен.

## 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**.

# Интеграция с Redmine

## 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

---

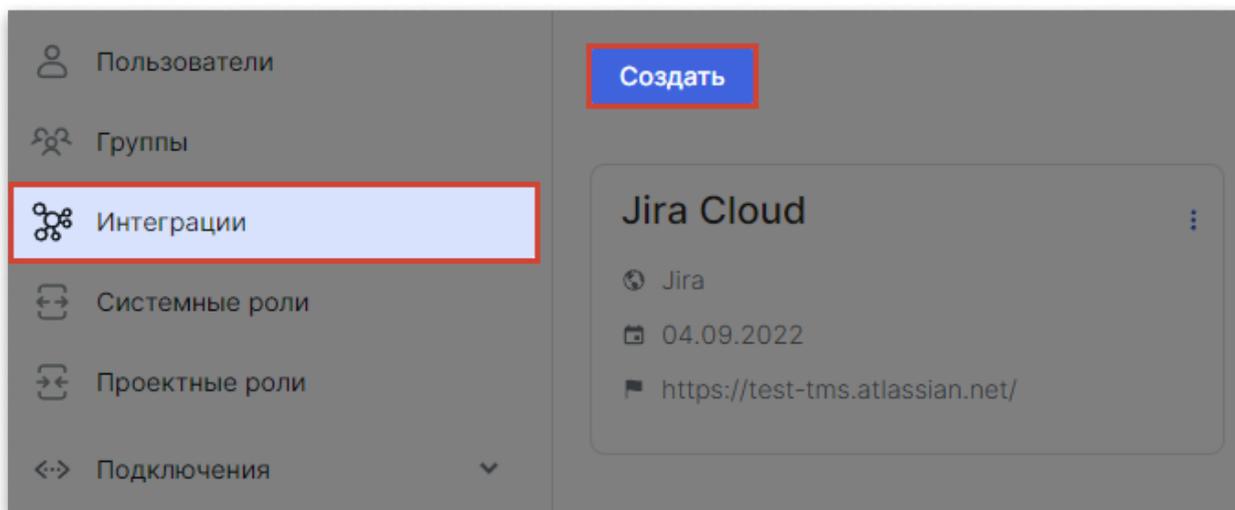
1. Войдите в Redmine по вашей учетной записи.
2. Войдите в ваш профиль Redmine. Для этого нажмите **Моя учетная запись** (*My account*) в правой верхней части окна.
3. Скопируйте **Ключ доступа к API** (*API access key*).

## 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 создаваться 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**.

# Интеграция с TeamStorm

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:

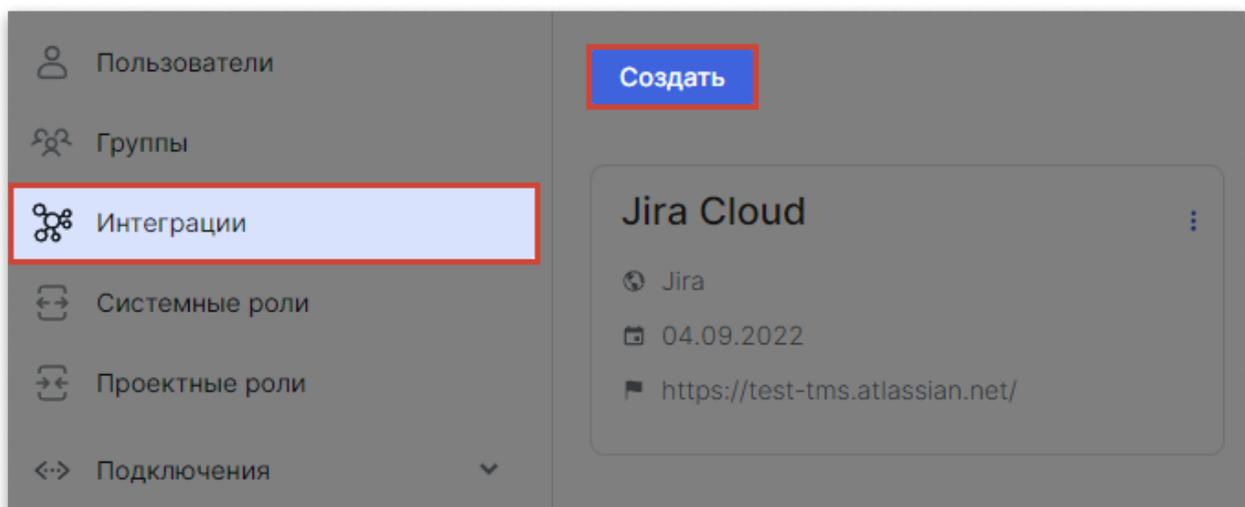
- Доступ к учетной записи TeamStorm для **получения токена**
- Учетная запись Test IT с доступом к модулю **Администрирование** для **создания интеграции**

## 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** (колонок **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 (колонок **Ссылки**)
- **История результатов** на странице тест-кейса (колонок **Ссылки**)

# Интеграция с Yandex Tracker

Вы можете настроить интеграцию с Yandex Tracker 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 Yandex Tracker
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 Яндекса](#).

## 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**.

Яндекс ID

Мои приложения

Создать приложение

Документация

Про Яндекс ID

### Общие данные

Название вашего сервиса

Test IT ✓

Иконка сервиса (не более 1Мб)

Прикрепить иконку

### Платформы приложения

Выберите хотя бы одну платформу

Веб-сервисы

iOS-приложение

Android-приложение

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.

Яндекс ID

Мои приложения

Создать приложение

Документация

Про Яндекс ID

Android-приложение

### Доступ к данным

Чтобы добавить доступ, укажите его название

Название доступа

Чтение из трекера  
tracker:read

Запись в трекер  
tracker:write

### Почта для связи

Укажите почту компании или свою. Будем оповещать об изменениях во внешней авторизации

claricestarl@yandex.ru ✓

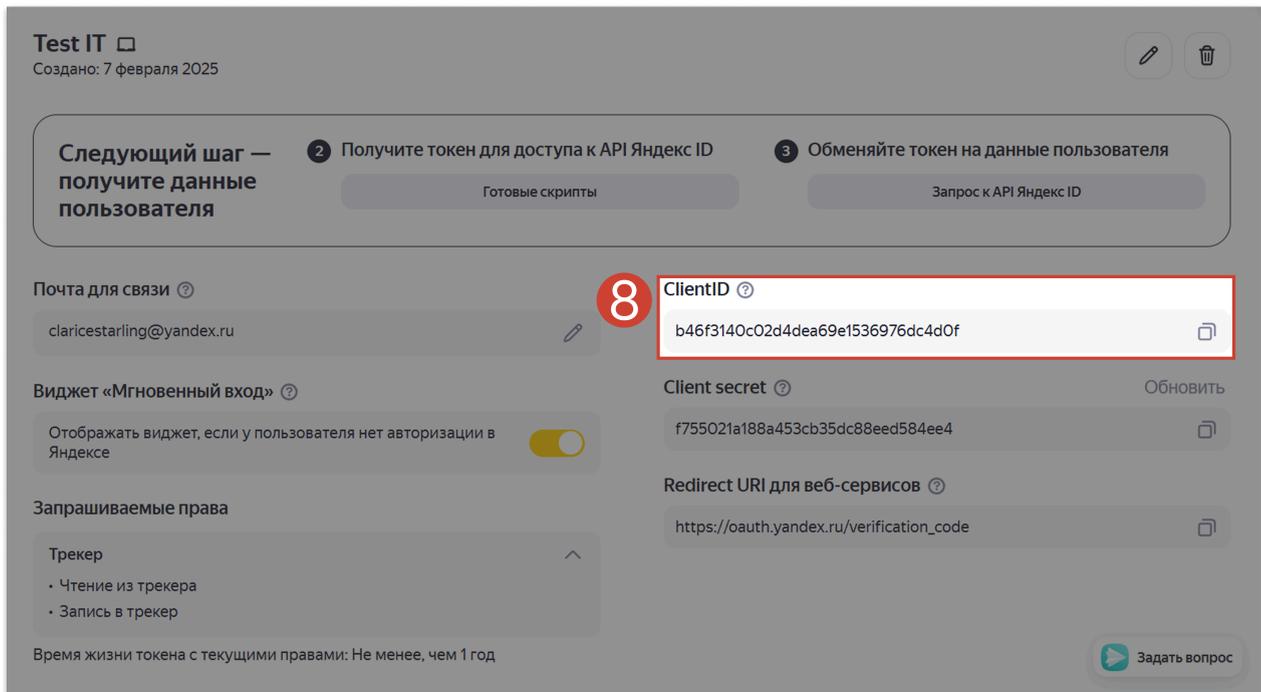
Задать вопрос

Справка и помощь

© 2001-2025, Яндекс

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:

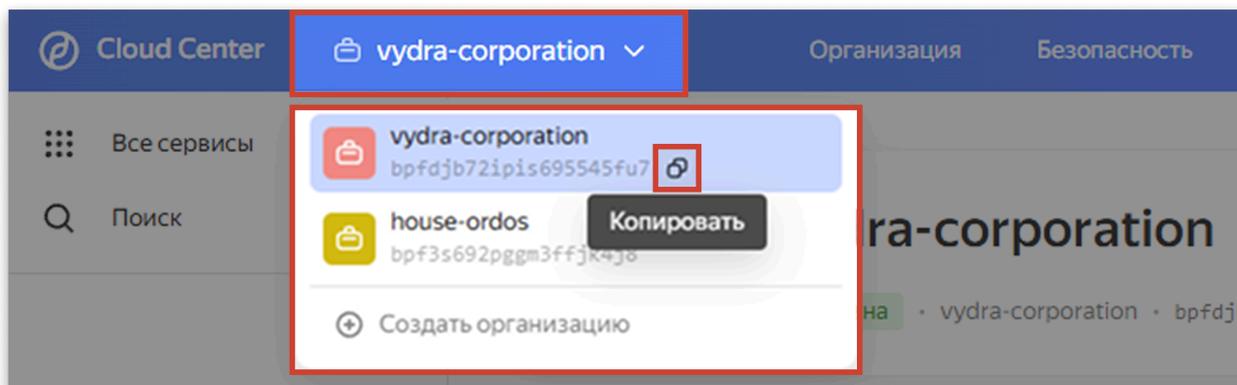
1. В Яндекс ID откройте страницу получения токена, пройдя по ссылке:  
`https://oauth.yandex.ru/authorize?response_type=token&client_id=<идентификатор_приложения>` . Откроется страница с сообщением об ошибке (404), так как в ссылке не указан идентификатор приложения.
2. В адресной строке замените текст `<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 Yandex Tracker

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

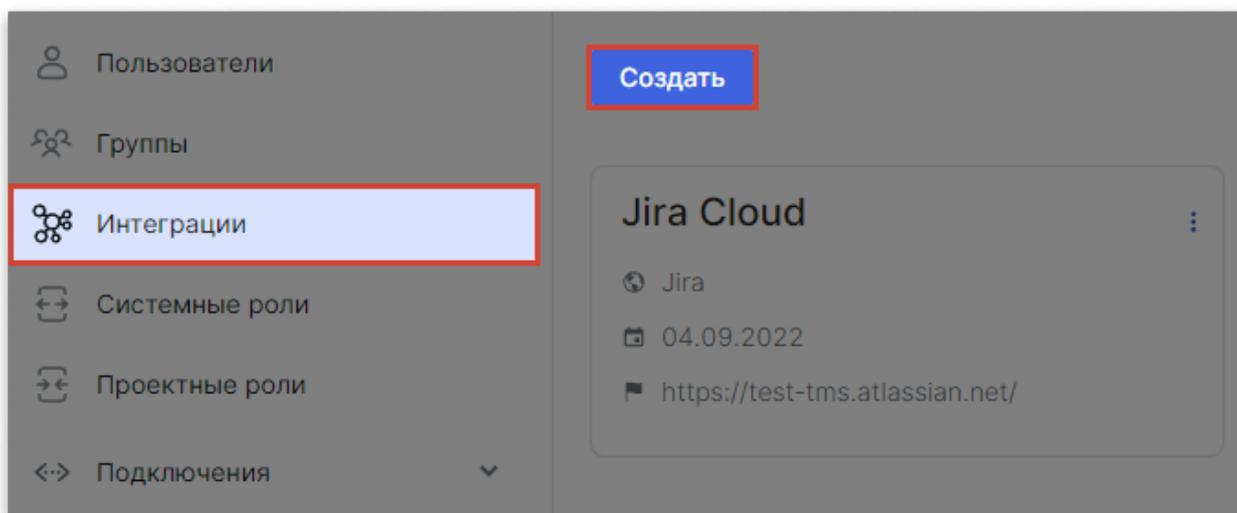
Чтобы скопировать идентификатор организации:

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 **Yandex Tracker**.
4. Заполните данные для подключения:
  - Имя интеграции
  - URL системы Yandex Tracker. Он общий для всех аккаунтов:  
<https://api.tracker.yandex.net>
  - Токен. Вставьте скопированный ранее OAuth-токен из Yandex Tracker.

- Тип организации: **Yandex Cloud Organization** или **Яндекс 360**
- Идентификатор организации. Вставьте идентификатор, скопированный из **Yandex Tracker**.

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 Yandex Tracker для создания дефекта и тип задачи. Откроется окно создания дефекта в Yandex Tracker. Убедитесь, что все поля содержат корректную информацию о проваленном тесте.
6. Нажмите **Сохранить**.  
В Yandex Tracker должна автоматически создаваться 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**.

# Интеграция с YouTrack

## 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@yoonion.ru](mailto:support@yoonion.ru).

## 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

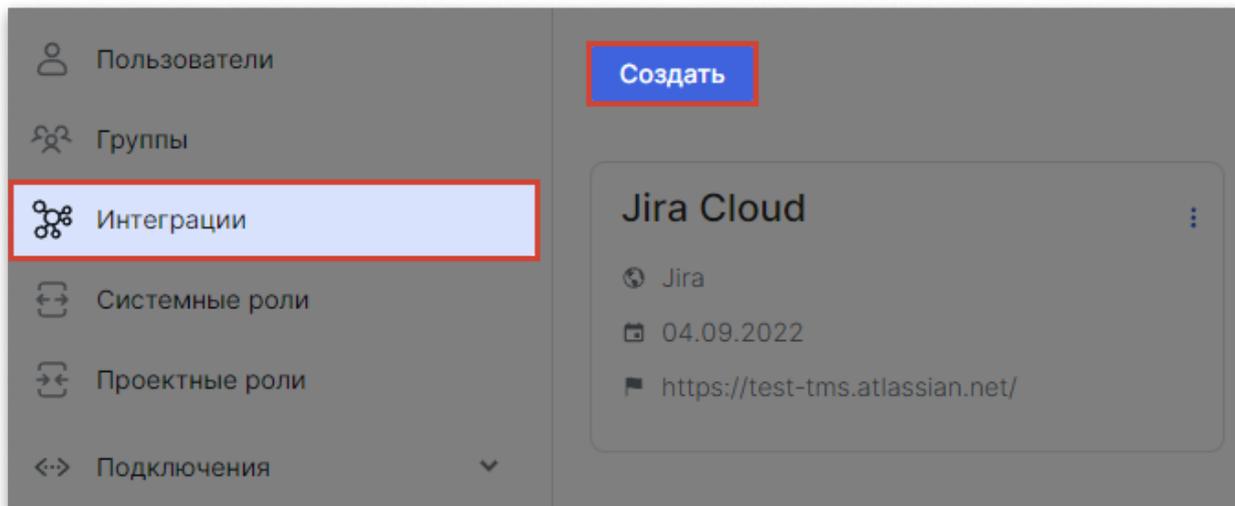
---

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**.

# Интеграция Test IT с ИИ-моделями

В Библиотеке Test IT вы можете **создавать тесты с помощью искусственного интеллекта (ИИ)**. Тесты создаются по задачам из вашего task-трекера. Из одной задачи создается несколько тестовых сценариев — до 10 (это значение предустановлено и не может быть изменено). 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 task-трекере нет.

Подробнее читайте в [блоге Test IT](#) .

Для генерации тестов вам потребуется:

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

Для интеграции с Test IT доступны следующие сервисы:

- **GigaChat**
- **YandexGPT**
- **Сервисы, совместимые с OpenAI**

Обновлено: 28.10.2025, 14:25:16

# Интеграция с GigaChat

## Требуется настроенная интеграция с таск-трекером

Чтобы создавать тест-кейсы по задачам из вашего таск-трекера, нужно настроить и протестировать интеграцию с таск-трекером и добавить ее в проект.

Для ИИ-генерации тестов с помощью 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**.

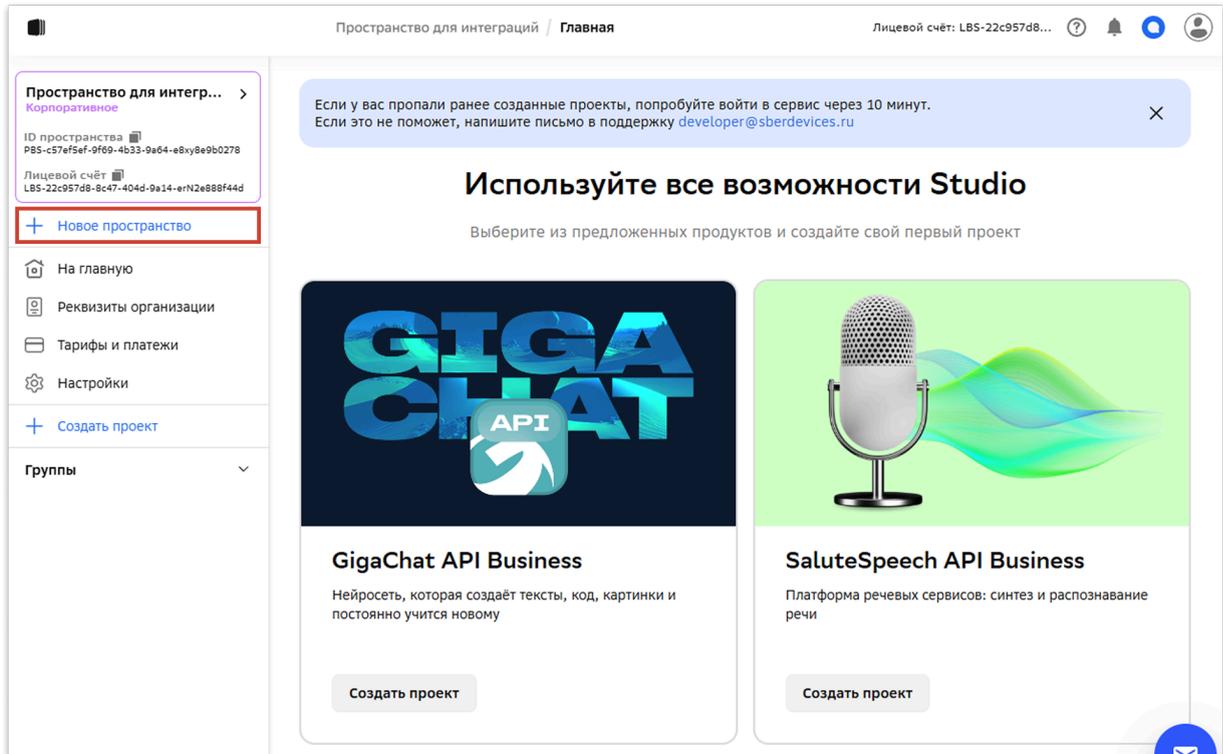
Использование GigaChat API платное.

## 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:

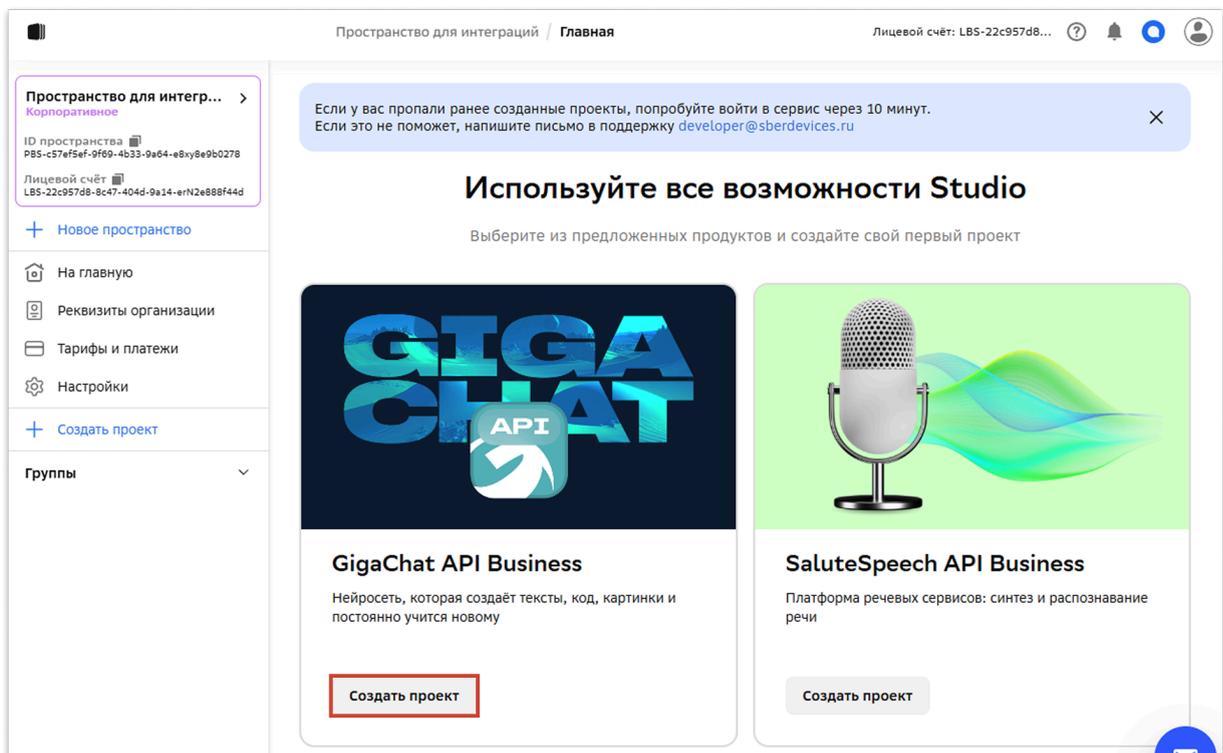
- Нажмите **+ Новое пространство**.



- В окне создания пространства введите имя пространства, выберите его тип, затем нажмите **Создать**.

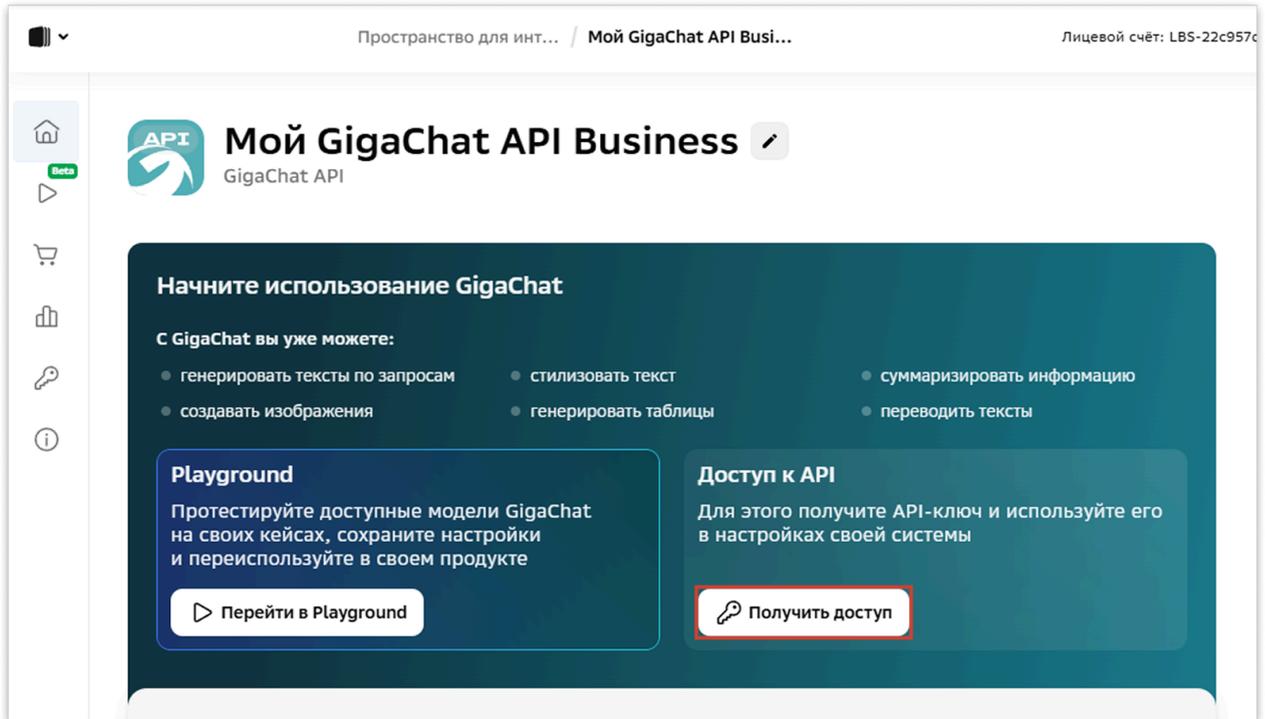
### 3. Создайте проект:

- 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.

### Сохраните ваш Authorization Key ×

Authorization Key передается в заголовке запроса на получение Access Token.

Значение поля является результатом кодирования пары Client ID:Client Secret в base64

Authorization Key

▼ Вы также можете использовать пару Client ID:Client Secret

Закодируйте пару Client ID:Client Secret в base64 и передавайте ее в заголовке запроса на получение Access Token

Client Secret

Client ID

**Не закрывайте это окно, пока не сохраните данные**  
Данные отображаются только один раз и не хранятся в системе  
При компрометации или утере вы сможете получить новое значение

**Готово**

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 GigaChat отобразится на странице интеграций.  
Для создания тестов с помощью ИИ потребуется добавить интеграцию в проект.

Обновлено: 25.09.2025, 14:10:25

# Интеграция с YandexGPT

Требуется настроенная интеграция с таск-трекером

Чтобы создавать тест-кейсы по задачам из вашего таск-трекера, нужно настроить и протестировать интеграцию с таск-трекером и добавить ее в проект.

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.

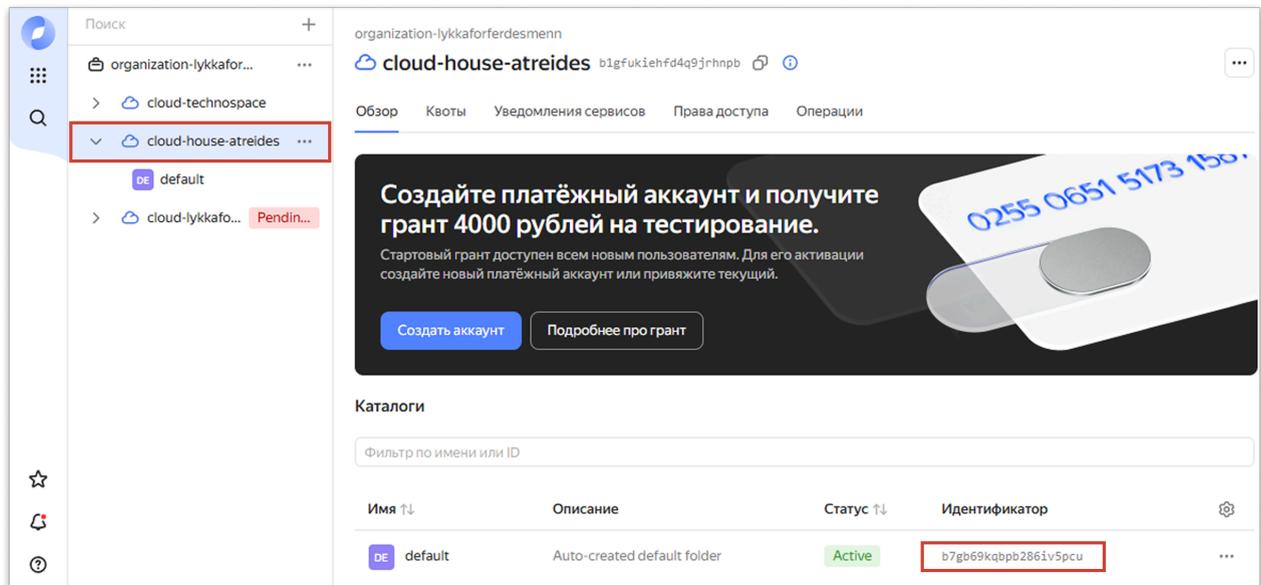
1. Перейдите на сайт [Yandex Cloud](#) и нажмите **Подключиться**.
2. Зарегистрируйтесь, используя ваш Yandex ID (если у вас его нет, [создайте его](#) ).
3. После авторизации войдите в консоль Yandex Cloud, нажав **Консоль** в правом верхнем углу окна справа от значка аккаунта.
4. Создайте организацию, 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

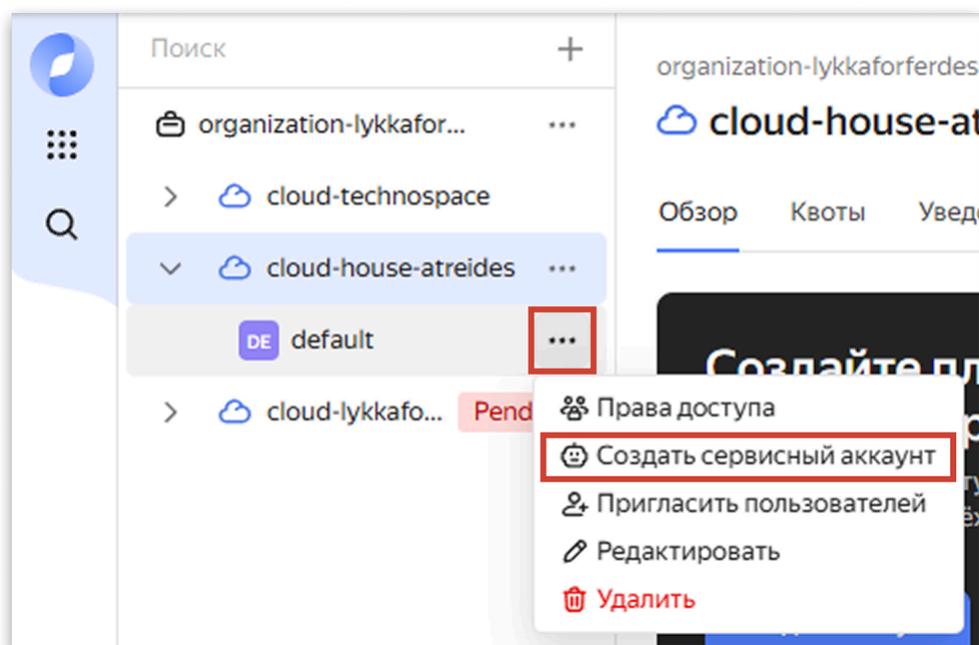
### Создание сервисного и платежного аккаунтов Yandex Cloud

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:

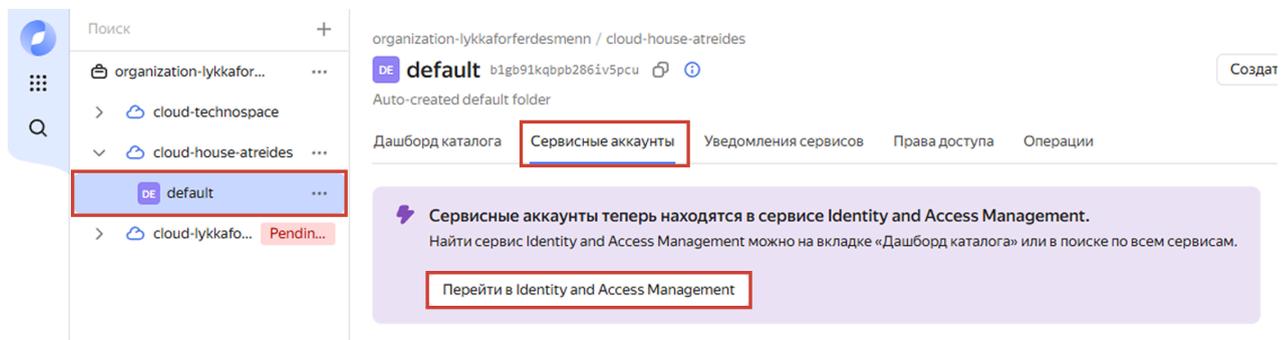
1. В **консоли управления** в списке слева выберите облако. Откроется выпадающий список каталогов.
2. В требуемом каталоге нажмите значок **Действия** и выберите **Создать сервисный аккаунт**. Откроется окно создания аккаунта.



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: `ус.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**.

# Интеграция с сервисами, совместимыми с OpenAI

## ₽ Опция Standard и Enterprise

Эта опция доступна в тарифах Standard и Enterprise. В тариф Lite она не входит.

Подробнее: [о тарифах Test IT](#) .

## Требуется настроенная интеграция с task-трекером

Чтобы создавать тест-кейсы по задачам из вашего task-трекера, нужно настроить и протестировать интеграцию с task-трекером и добавить ее в проект.

## Об интеграции с сервисами, совместимыми с OpenAI

Для генерации тест-кейсов на основе задач из вашего task-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.

# Авторизация

## Функциональность Test IT Enterprise

Данная инструкция описывает авторизацию в **Test IT Enterprise**.

Если вы используете **Lite** или **Standard**, 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.

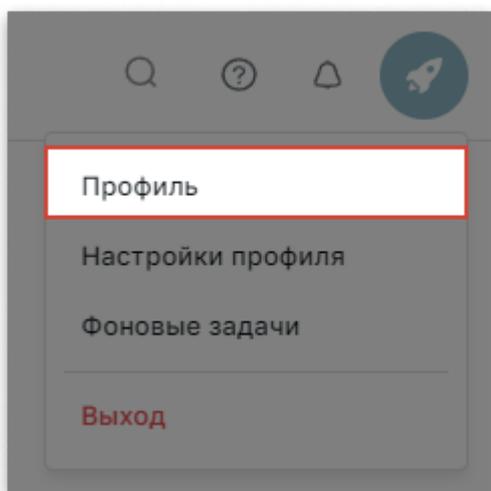
Обновлено: 16.10.2025, 18:35:17

# 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.

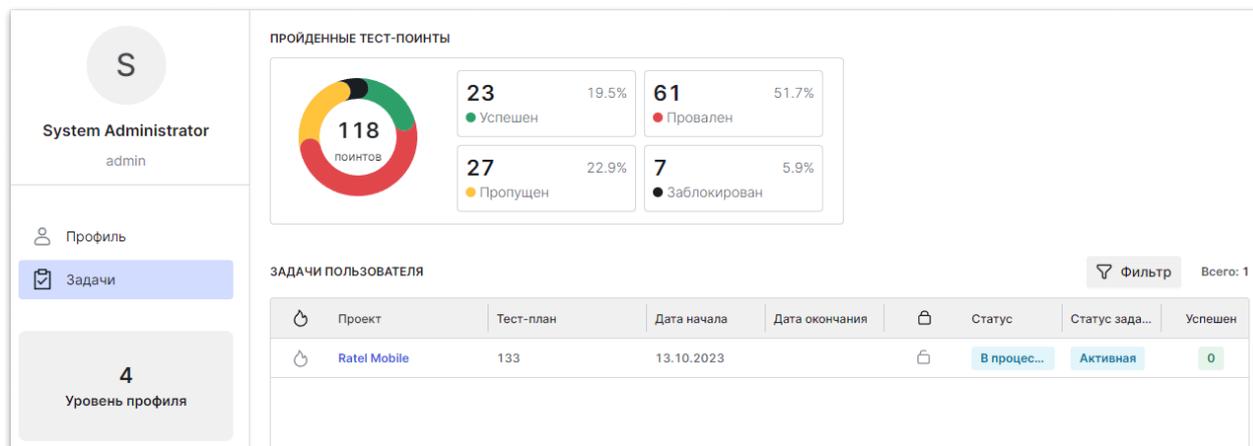


В профиле отображается информация о вашей активности в Test IT и актуальных задачах.

A screenshot of the user profile page in the Test IT system. The page is divided into several sections. On the left, there is a sidebar with the user's name 'System Administrator' and 'admin', and a list of menu items: 'Профиль' (Profile), 'Задачи' (Tasks), '1 Уровень профиля' (Profile level), '0 Лайки профиля' (Profile likes), and '4 Дня с Test IT' (Days with Test IT). The main content area is titled 'УРОВЕНЬ ПРОФИЛЯ' (Profile Level) and features a yellow duck icon. The user's name is 'Любопытный цыплёнок' (Curious chick) and they are at 'Уровень 1' (Level 1). A progress bar shows '540 баллов. Ещё 210 баллов до следующего уровня!' (540 points. 210 points to the next level!). To the right, there is a 'Прогресс уровня' (Level progress) section with a toggle for 'Показать в баллах' (Show in points) and statistics: '9 Создано тестов' (9 tests created), '23 Успешно пройдено тест-поинтов' (23 test points passed), and '2 Провалено тест-поинтов' (2 test points failed). Below this is a 'ДОСТИЖЕНИЯ' (Achievements) section with two rows of cards. The first row is 'Создание тестов (0/4)' (Test creation) with 9 tests for all time, and the second row is 'Редактирование тестов (0/4)' (Test editing) with 6 tests for all time. Each card shows a 'Закрытое достижение' (Locked achievement) and a progress bar for the current goal.

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.

The screenshot displays a user profile for 'System Administrator admin' at 'Уровень 1' (Level 1). The profile has 540 points, with a progress bar indicating 540/750 points towards the next level. The user has 9 tests created, 23 tests passed, and 2 tests failed. The achievements section shows progress for 'Создание тестов' (0/4) and 'Редактирование тестов' (0/4).

УРОВЕНЬ ПРОФИЛЯ	ПРОГРЕСС УРОВНЯ
Любопытный цыплёнок Уровень 1 540 баллов. Ещё 210 баллов до следующего уровня!	9 Создано тестов 23 Успешно пройдено тест-поинтов 2 Провалено тест-поинтов

ДОСТИЖЕНИЯ	ПРОГРЕСС
Создание тестов (0/4)	9 тестов за все время
Закрытое достижение Создано 10 тестов	
Закрытое достижение Создано 100 тестов	
Закрытое достижение Создано 500 тестов	
Закрытое достижение Создано 1000 тестов	
Редактирование тестов (0/4)	6 тестов за все время

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).

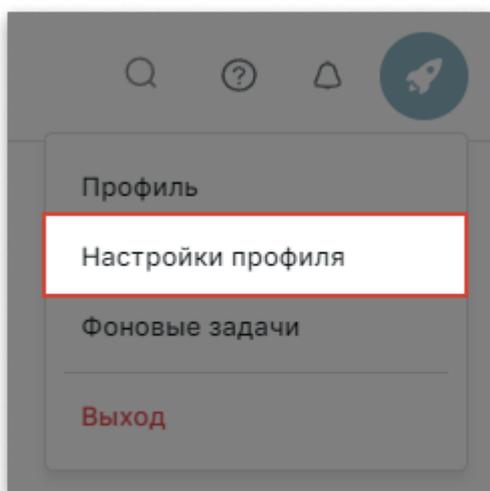
Обновлено: 10.09.2024, 16:22:05

# 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.

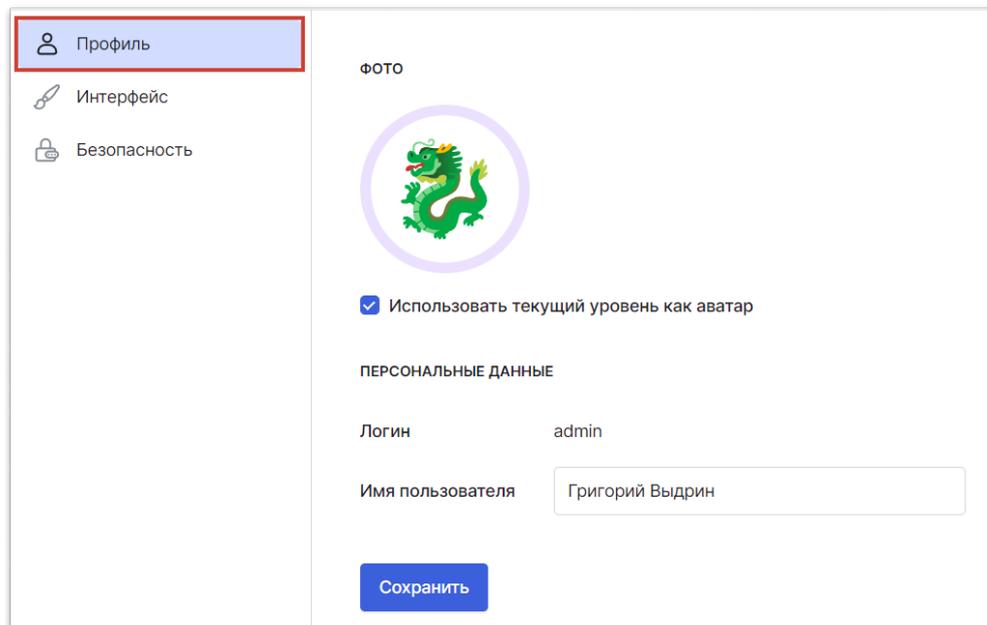
Чтобы перейти к настройкам профиля:

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

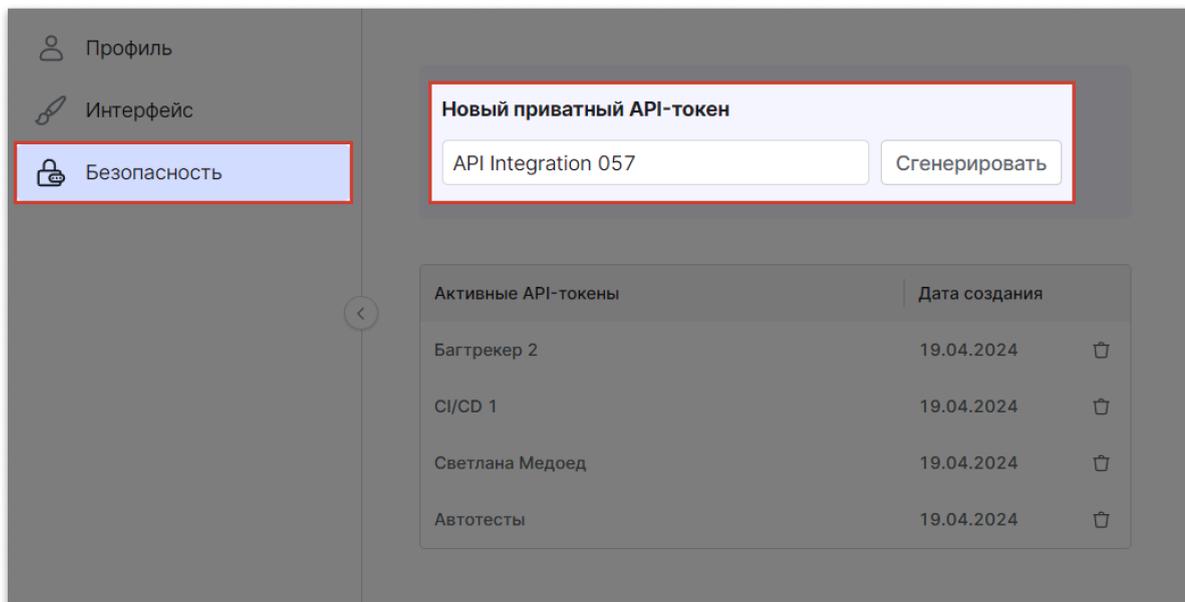
---

### 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.

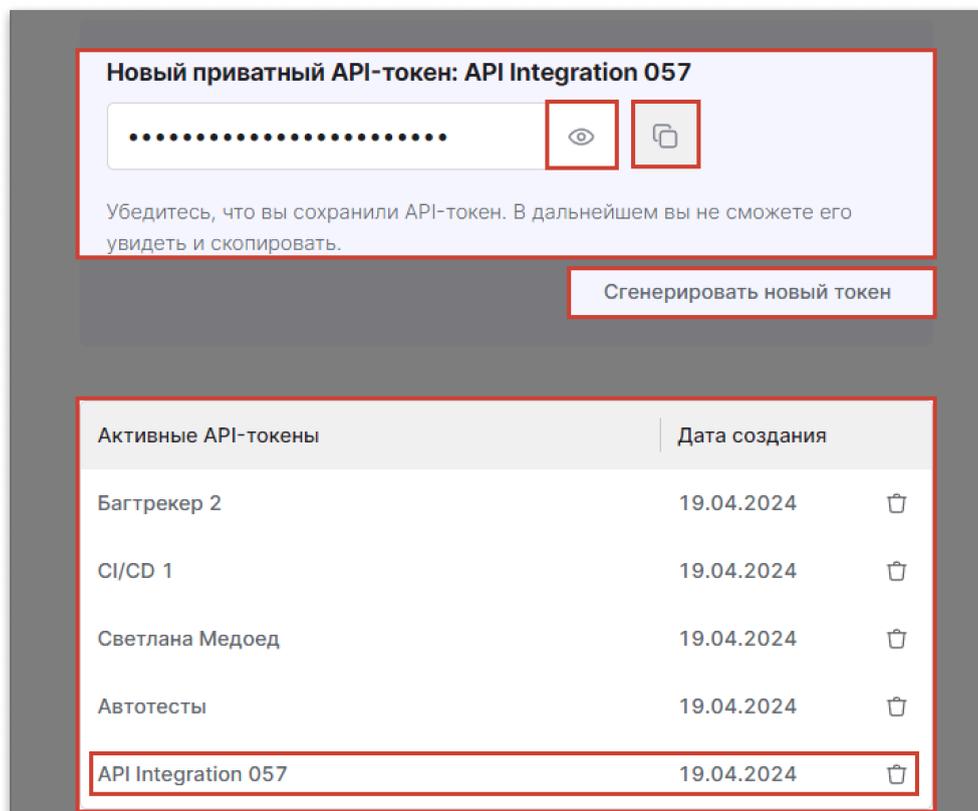
Система сгенерирует новый токен.

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.

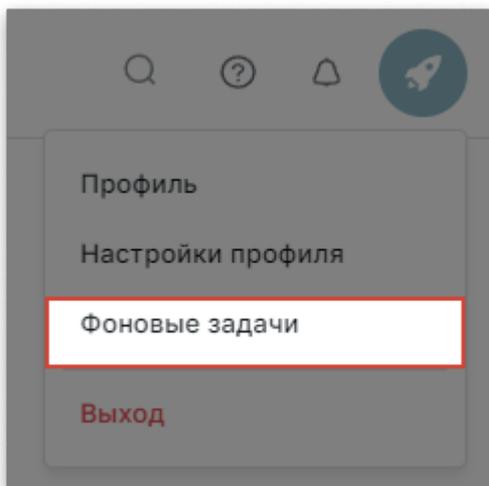
Обновлено: 07.11.2024, 12:35:52

# 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.

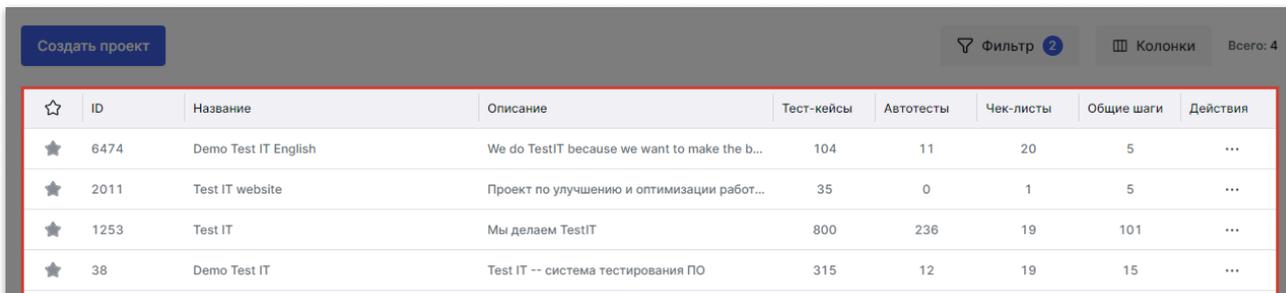
Вам доступны следующие действия:

- Чтобы отменить фоновую задачу в статусе **In Progress**, 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.

# Структура и компоненты проекта

## 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.

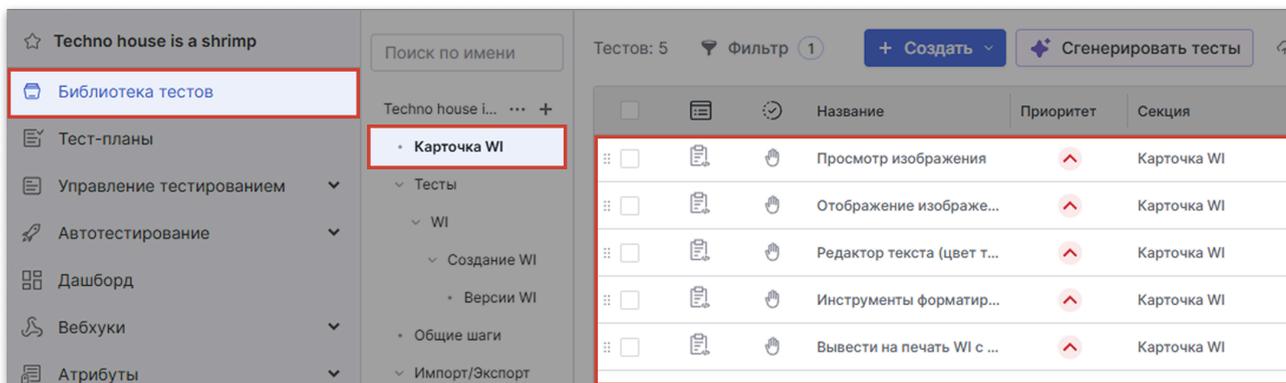


The screenshot shows a table with the following columns: ID, Название, Описание, Тест-кейсы, Автотесты, Чек-листы, Общие шаги, and Действия. There are four rows of project data.

ID	Название	Описание	Тест-кейсы	Автотесты	Чек-листы	Общие шаги	Действия
6474	Demo Test IT English	We do TestIT because we want to make the b...	104	11	20	5	...
2011	Test IT website	Проект по улучшению и оптимизации работ...	35	0	1	5	...
1253	Test IT	Мы делаем TestIT	800	236	19	101	...
38	Demo Test IT	Test IT -- система тестирования ПО	315	12	19	15	...

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

**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**.



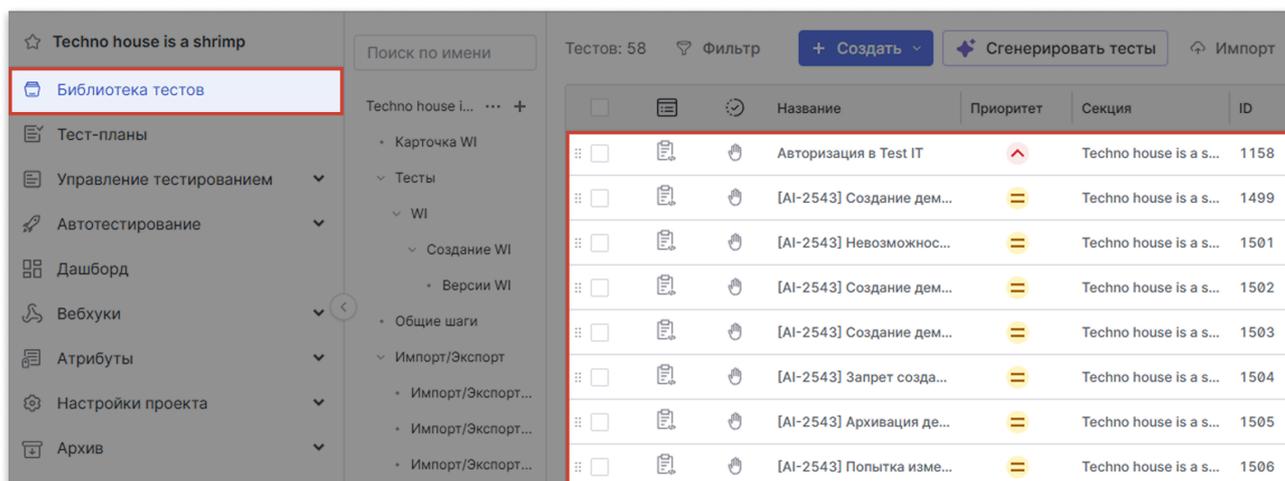
The screenshot shows the 'Test Library' interface. On the left is a sidebar with a menu. The main area displays a list of test cases grouped into sections. The 'Карточка WI' section is highlighted.

Название	Приоритет	Секция
Просмотр изображения	↑	Карточка WI
Отображение изображе...	↑	Карточка WI
Редактор текста (цвет т...	↑	Карточка WI
Инструменты форматир...	↑	Карточка WI
Вывести на печать WI с ...	↑	Карточка WI

## Ручное тестирование

**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-

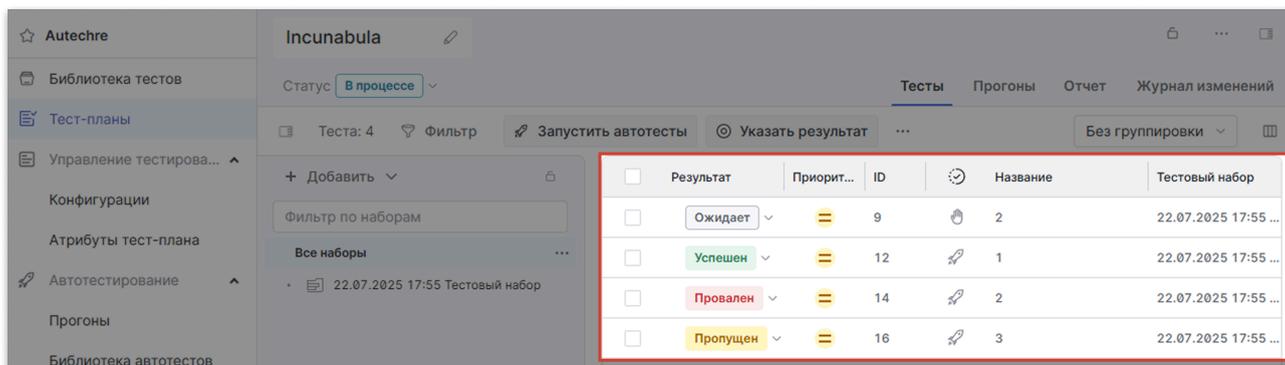
кейсы могут связываться с автоматизированными тестами. Тест-кейсы, входящие в проект, доступны на странице проекта во вкладке **Библиотека тестов**.



**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**.

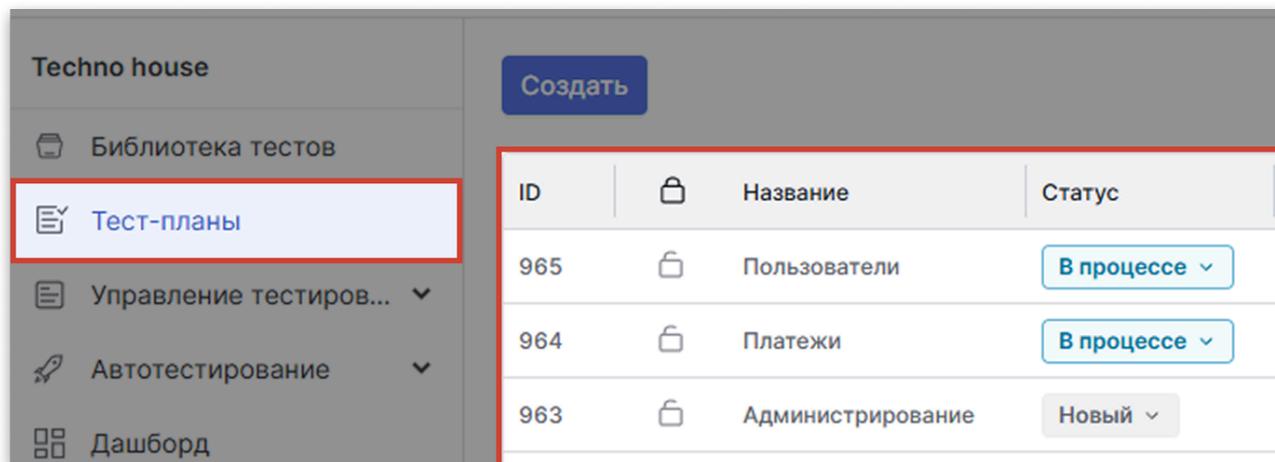
**Общий шаг** — шаг, который можно использовать в нескольких тест-кейсах, что позволяет сэкономить время на создание тестовых сценариев с типичными шагами. Общие шаги, входящие в проект, доступны на странице проекта во вкладке **Библиотека тестов**.

**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.

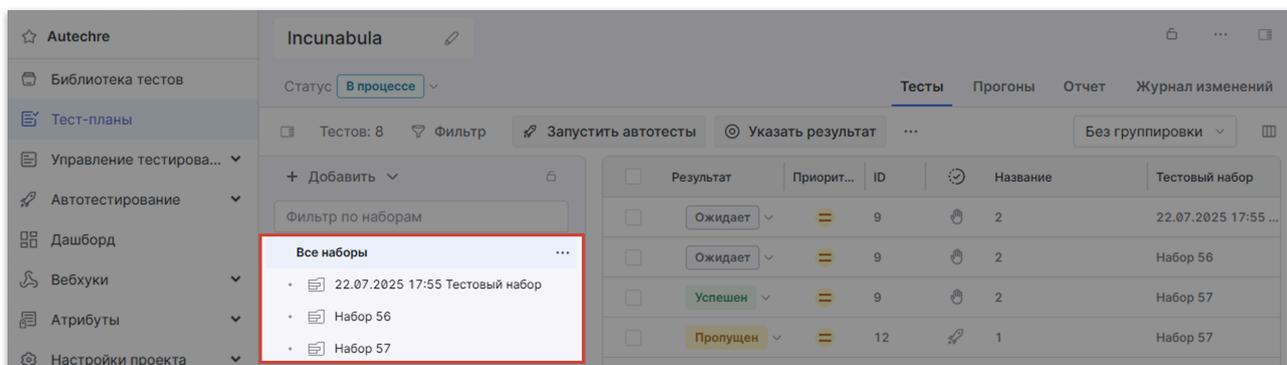


Тест-поинты, совершенные в рамках одного тест-кейса, доступны на странице тест-кейса во вкладке **История результатов**.

**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.

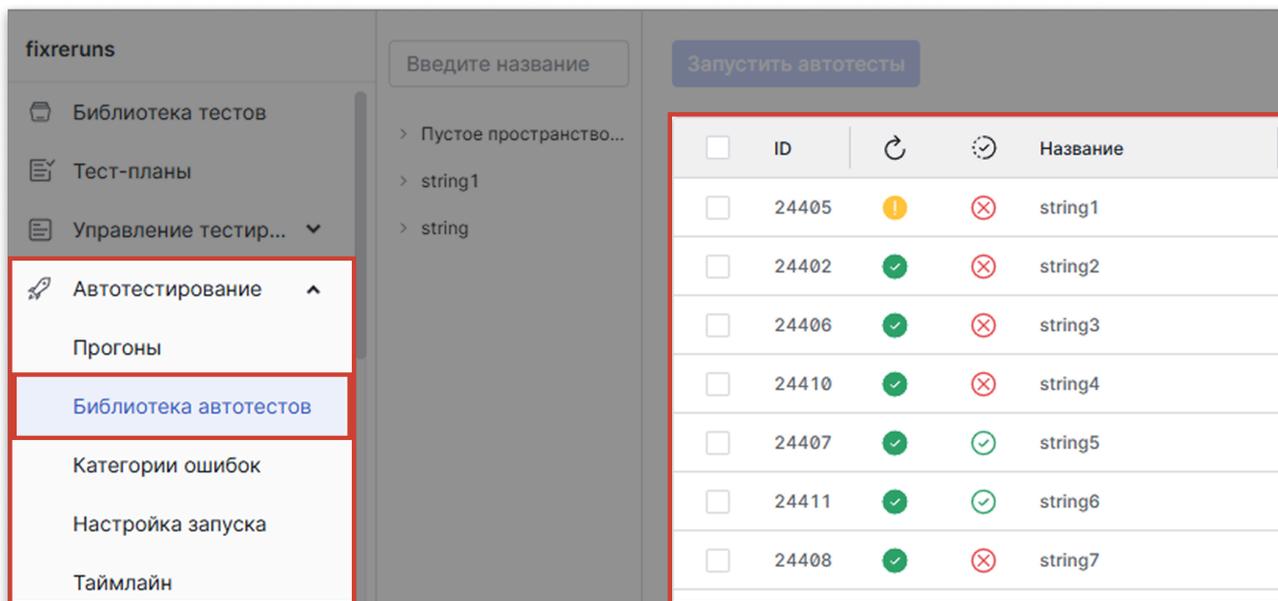


### Практический пример

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".

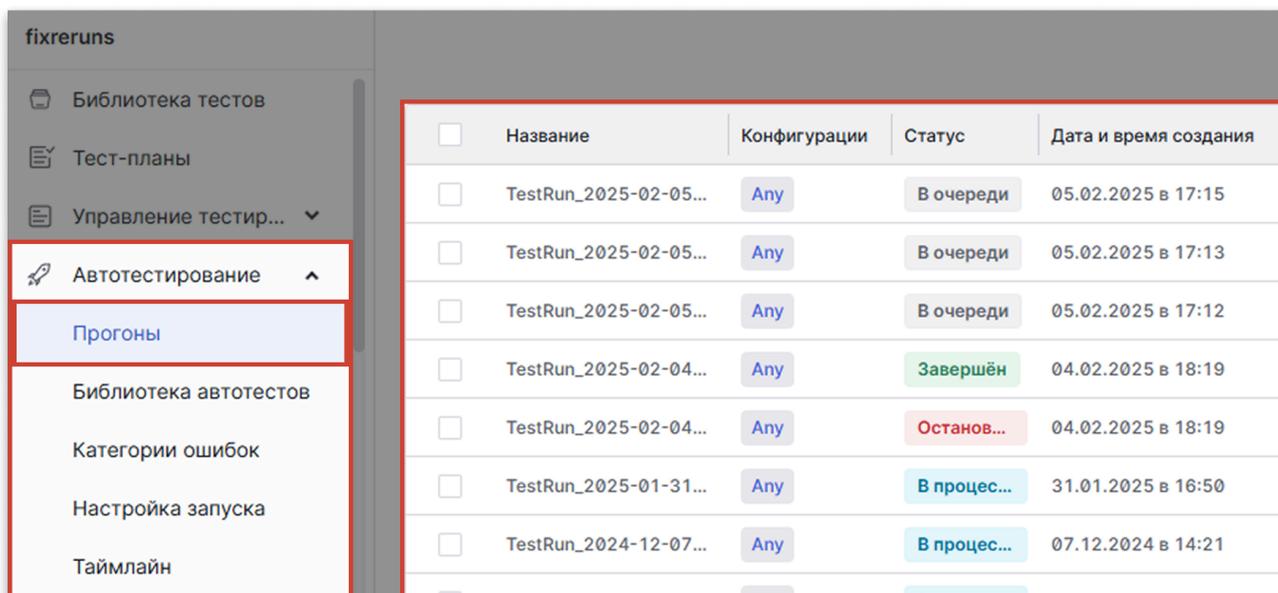
## Автоматизированные тесты

**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, пометать 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 **Прогон**ы. Результаты отдельного прогона доступны на странице прогона во вкладке **Результаты тестов**. Результаты прогонов отдельного автотеста доступны на странице автотеста во вкладке **История результатов**.

Прогон

ы, связанные с отдельным тест-поинтом, доступны на странице тест-поинта в разделе **Тест-планы**. Прогон

ы 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.

# Создание и редактирование проекта

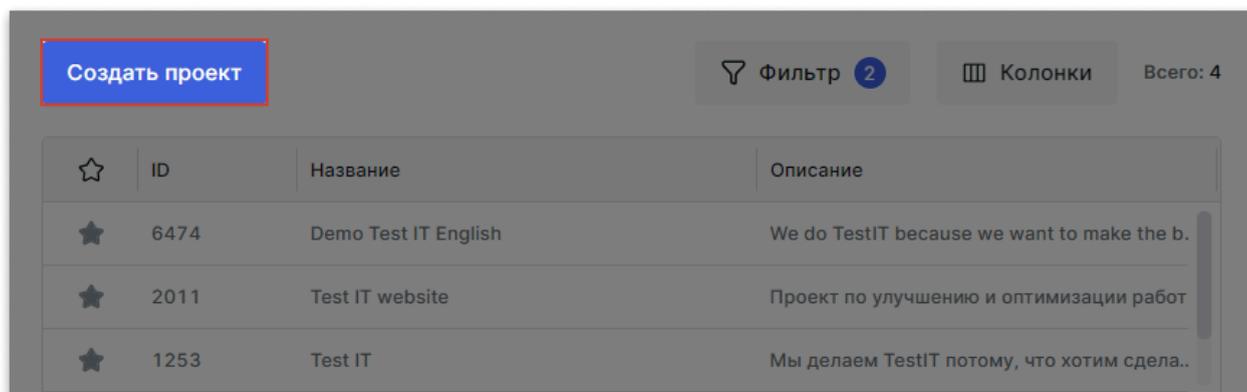
## 🔗 Тарифные ограничения для Lite

В тарифе **Lite** 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 Test IT Lite & Standard

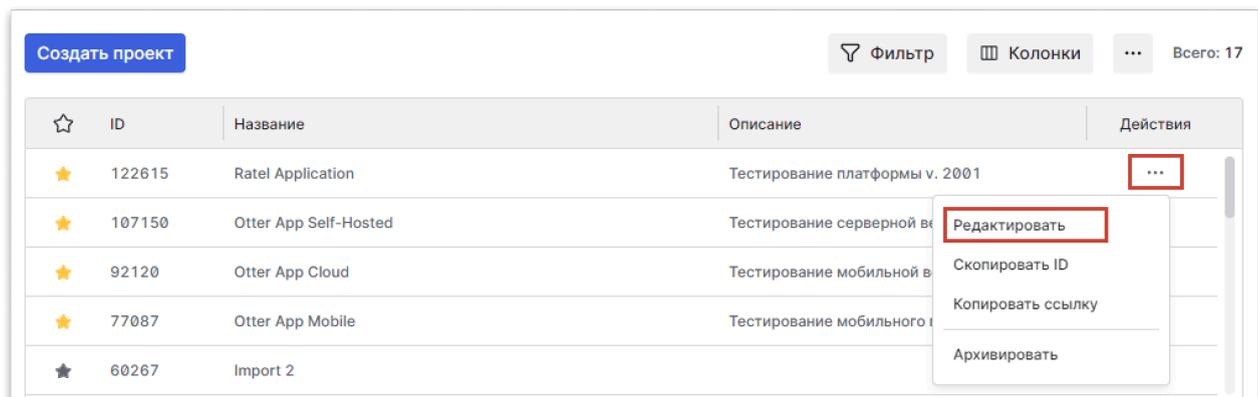
Если вы начинающий пользователь и работаете в **Lite** или **Standard**, вы можете создать демо-проект с готовыми примерами тестов, тест-планов и дашбордов, чтобы быстро познакомиться с возможностями системы. Для этого:

- 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:

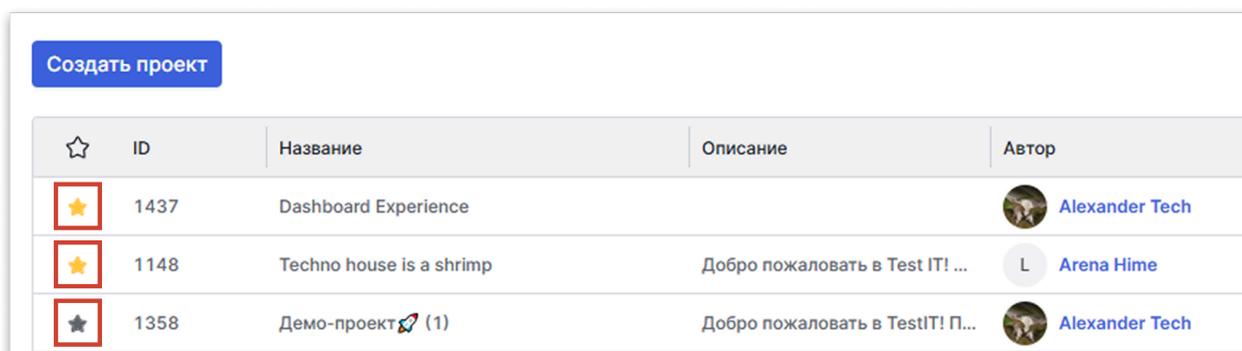
- Отредактировать проект
- Скопировать ссылку на проект
- Архивировать проект

## 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**.

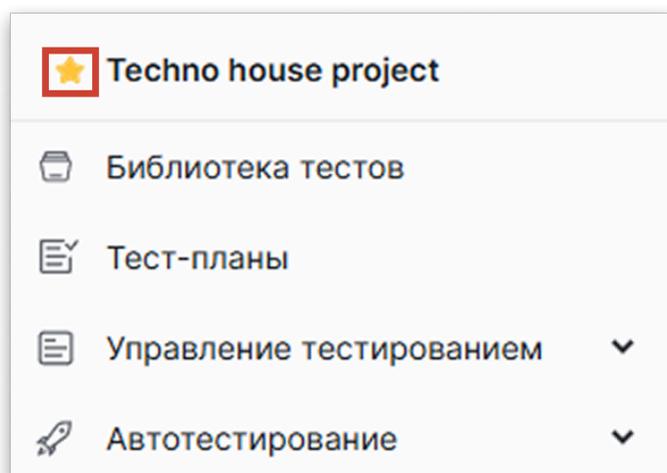
Отметить проекты как избранные или снять отметку избранности можно:

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



☆	ID	Название	Описание	Автор
★	1437	Dashboard Experience		 Alexander Tech
★	1148	Techno house is a shrimp	Добро пожаловать в Test IT! ...	 Arena Hime
★	1358	Демо-проект  (1)	Добро пожаловать в TestIT! П...	 Alexander Tech

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

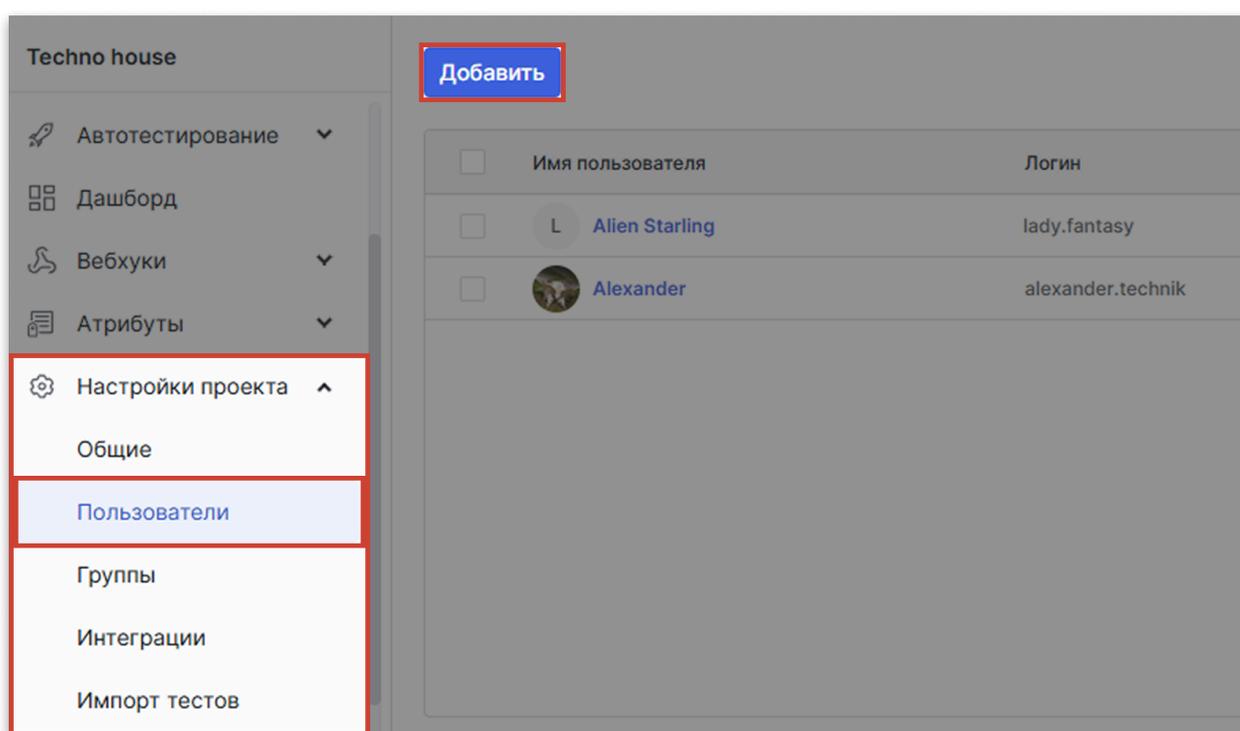


Также вы можете перейти к другим проектам, наведя курсор на название открытого проекта.

# Добавление пользователей и групп в проект

## Добавление пользователей

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



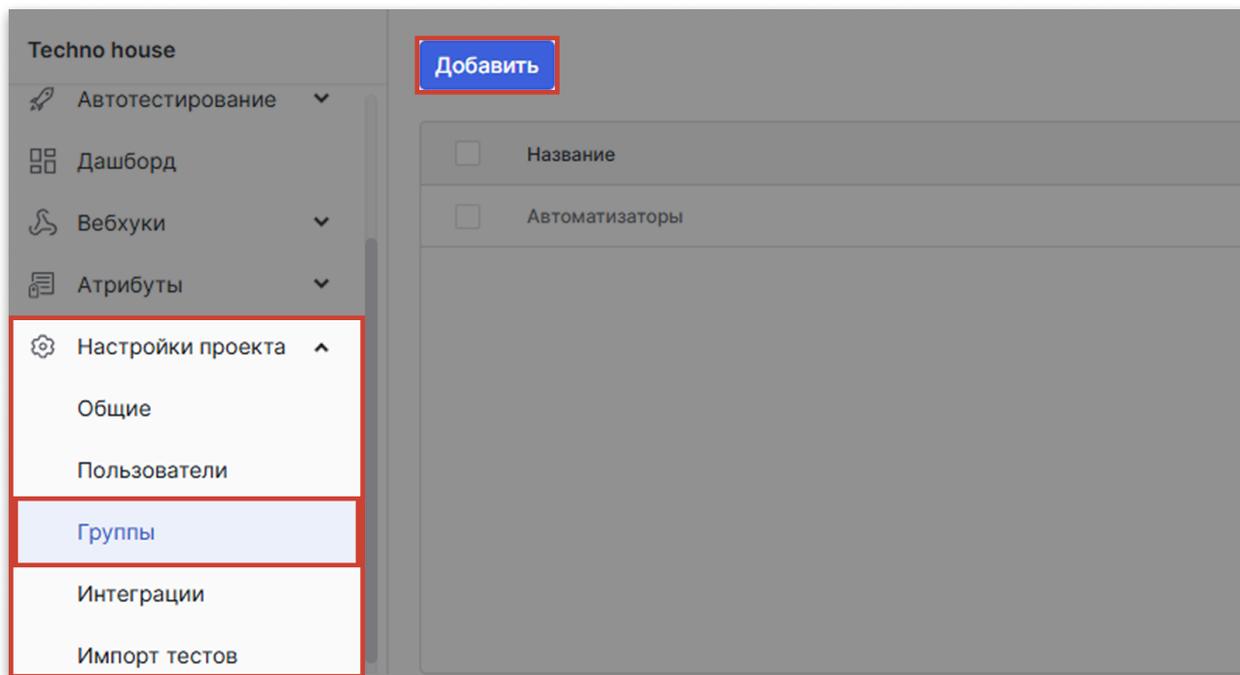
4. Введите имя пользователя, которого хотите добавить, или выберите его из выпадающего списка.
5. Выберите **проектные роли** для нового пользователя.
6. Нажмите **Сохранить**.

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**.



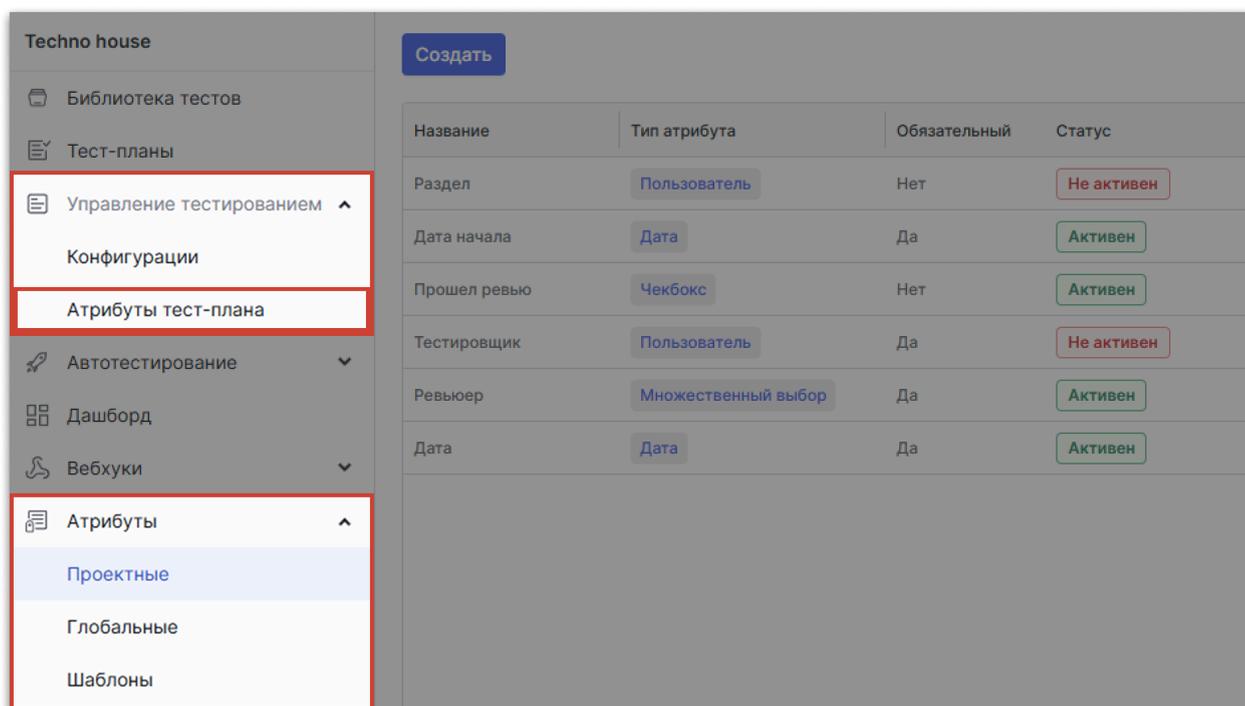
4. Введите название группы пользователей, которую хотите добавить, или выберите ее из выпадающего списка.
5. Выберите **проектную роль**. Проектная роль будет применена ко всем пользователям из группы.
6. Нажмите **Сохранить**.

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

# 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.



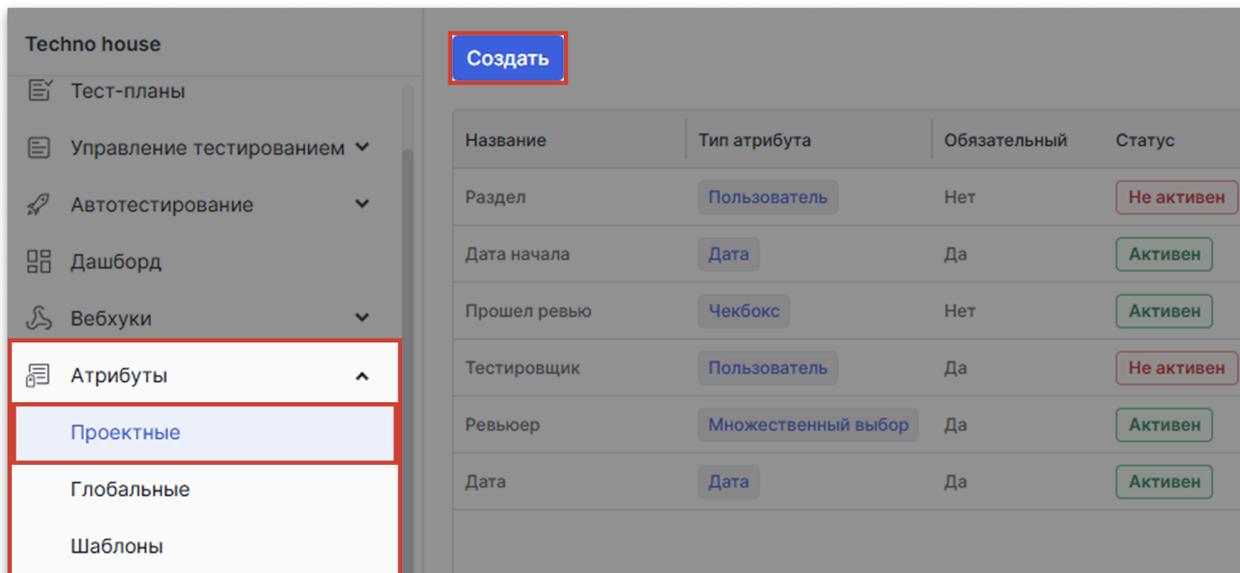
The screenshot shows the 'Techno house' interface. On the left is a sidebar menu with items like 'Библиотека тестов', 'Тест-планы', 'Управление тестированием', 'Конфигурации', 'Атрибуты тест-плана', 'Автотестирование', 'Дашборд', 'Вебхуки', 'Атрибуты', 'Проектные', 'Глобальные', and 'Шаблоны'. The 'Атрибуты тест-плана' and 'Атрибуты' items are highlighted with red boxes. On the right, there is a 'Создать' button and a table of attributes.

Название	Тип атрибута	Обязательный	Статус
Раздел	Пользователь	Нет	Не активен
Дата начала	Дата	Да	Активен
Прошел ревью	Чекбокс	Нет	Активен
Тестировщик	Пользователь	Да	Не активен
Ревьюер	Множественный выбор	Да	Активен
Дата	Дата	Да	Активен

## 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**.

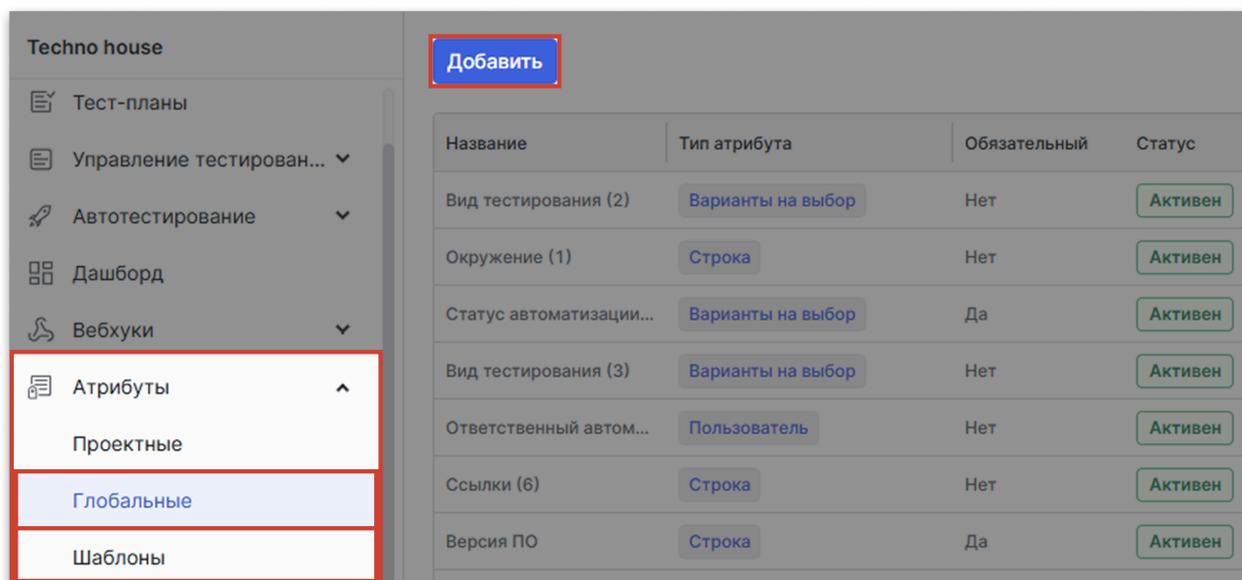
9. **Опционально:** Чтобы отредактировать или удалить атрибут, используйте меню **Действия** справа от строки атрибута.

## 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](#).

# Настройка вебхуков

## ₽ Тарифные ограничения для Lite

В тарифе **Lite** 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.

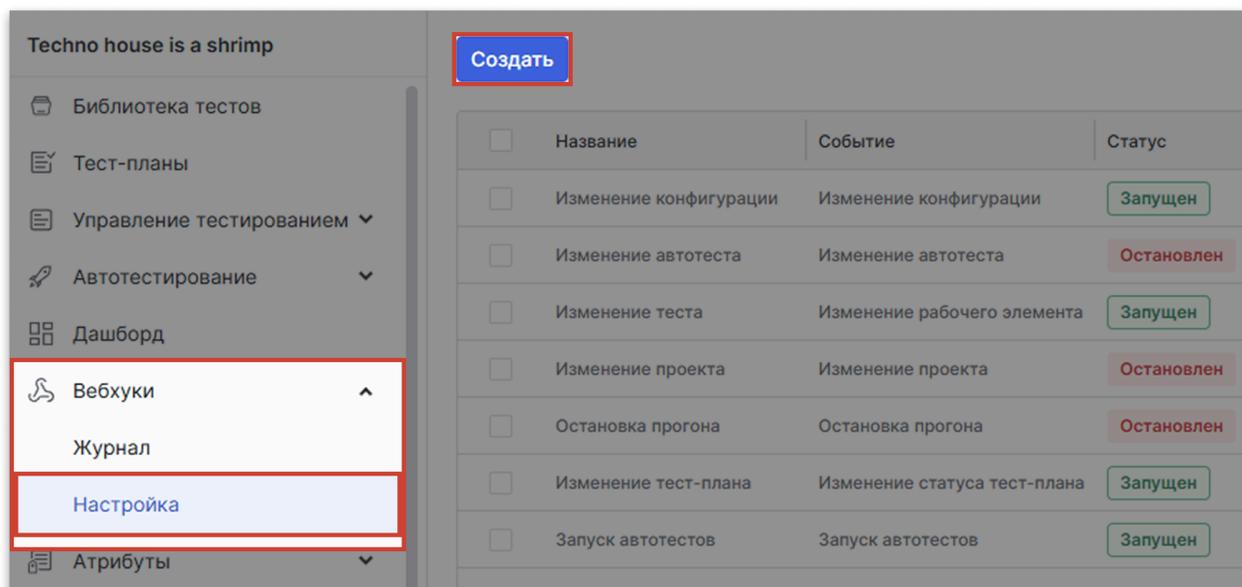
## Поддерживаются только HTTP- и HTTPS-запросы

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`.

## Новый вебхук ×

ШАГ 5

Введите контекст, который хотите отправлять в теле HTTP-запроса.

Тип контекста

Отправлять пользовательский контекст ▼

Заменить системные параметры

Экранировать параметры

Пользовательский контекст

`$AUTOTEST_IDS`

[Назад](#) [Сохранить](#)

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

## Отслеживание логов вебхуков

Вы можете посмотреть логи всех запущенных системой вебхуков за последние 7 дней. Для этого:

- Перейдите в раздел **Вебхуки** → **Журнал**.

## 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**.

Создать

Фильтр Всего: 2

	Статус	Действия
статуса тест-плана	Запущен	⋮
тестов	За	Редактировать Удалить

5. Сохраните изменения.

Обновлено: 11.11.2025, 21:27:49

# Вебхук для изменения проекта

HTTP-запрос передается при [создании](#), [редактировании](#) или [архивации](#) проекта.

- ▶ [Доступные специальные переменные](#)

- ▶ [Пример стандартного тела HTTP](#)

Обновлено: 15.05.2024, 13:57:56

# 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
- ▶ Пример стандартного тела HTTP

## Вебхук для изменения атрибута рабочего элемента

---

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

- ▶ Доступные специальные переменные
- ▶ Пример стандартного тела HTTP

# Вебхуки для работы с тест-планами

## Вебхук для изменения тест-плана

---

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

- ▶ Доступные специальные переменные
- ▶ Пример стандартного тела HTTP

## Вебхук для изменения статуса тест-плана

---

HTTP-запрос передается при изменении статуса тест-плана.

- ▶ Доступные специальные переменные
- ▶ Пример стандартного тела HTTP

## 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:

```
1   {
2   "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
3   }
```

json

В теле вебхука отобразится:

```
1   {
2   "OperatingSystem": "Windows"
3   }
```

json

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

► Доступные специальные переменные

► Пример стандартного тела HTTP

# 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:

```
1   {
2   "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
3   }
```

json

В теле вебхука отобразится:

```
1   {
2   "OperatingSystem": "Windows"
3   }
```

json

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

► Доступные специальные переменные

► Пример стандартного тела HTTP

# 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

► Пример стандартного тела HTTP

## 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:

```
1  {
2  "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
3  }
                                     json
```

В теле вебхука отобразится:

```
1  {
2  "OperatingSystem": "Windows"
3  }
                                     json
```



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

▶ Доступные специальные переменные

▶ Пример стандартного тела HTTP

Обновлено: 15.05.2024, 13:57:56

# 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:

```
1 | { | json
2 |   "OperatingSystem": "$CONFIGURATIONS_PARAMETERS[OS]"
3 | }
```

В теле вебхука отобразится:

```
1 | { | json
2 |   "OperatingSystem": "Windows"
3 | }
```

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

► Пример стандартного тела HTTP

## Вебхук для создания, редактирования и удаления карточек автотестов

---

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

► Доступные специальные переменные

► Пример стандартного тела HTTP

## 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

- ▶ Пример стандартного тела HTTP

## Вебхук для остановки прогона

---

HTTP-запрос передается при остановке прогона с помощью API-метода

`POST/api/v2/testRuns/{id}/stop` .

- ▶ Доступные специальные переменные

- ▶ Пример стандартного тела HTTP

## Вебхук для прохождения автотеста

---

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

- ▶ Доступные специальные переменные

- ▶ Пример стандартного тела HTTP

## Изменение связи между автотестом и рабочим элементом

---

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

▶ Доступные специальные переменные

▶ Пример стандартного тела HTTP

Обновлено: 20.11.2025, 13:35:38

# 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
- ▶ Пример стандартного тела HTTP

## Вебхук для упоминания пользователя

---

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

- ▶ Доступные специальные переменные
- ▶ Пример стандартного тела HTTP

# 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 `$`.

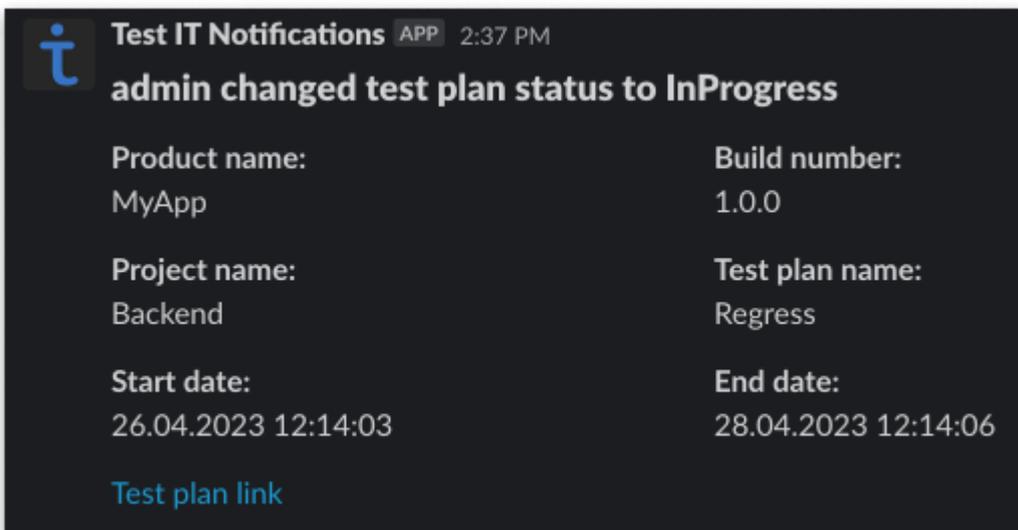
► [Пример стандартного тела HTTP \(развернуть\)](#)

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.



# 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 **ОК**.
4. Получите идентификатор группы или личной беседы, в которую хотите получать уведомление с помощью команды `/getchatid`. Отобразится сообщение 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

---

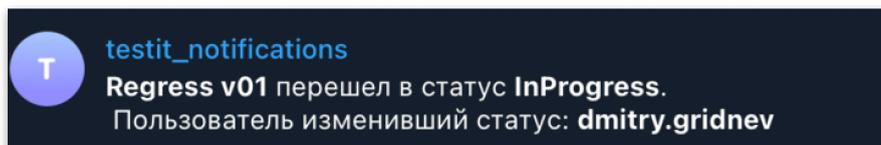
Для работы с Telegram вам потребуется создать новый вебхук в Test IT или отредактировать уже существующий с учетом следующих особенностей:

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)

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: 29.12.2025, 14:03:54

# 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 [дополнительные ресурсы](#).

## 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:

```
1 | pytest
2 | testit-adapter-pytest
```

text

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.

```
1 | [testit]
2 | url={URL}
3 | privatetoken={PRIVATE_TOKEN}
4 | testrunId={TEST_RUN_ID}
5 | projectId={TMS_PROJECT_ID}
6 | configurationId={TMS_CONFIGURATION_ID}
```

text

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` :

```
1  stages:
2  - test
3
4  run_tests:
5  image: python:3.9
6  stage: test
7
8  before_script:
9  --pip install -r requirements.txt
10 script:
11 - pytest --testit
```

yaml

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

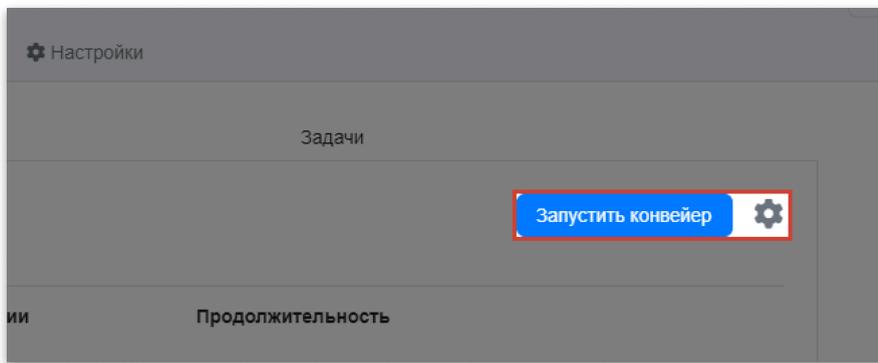
```
1  stages:
2  - test
3
4  run_tests:
5  image: python:3.9
6  stage: test
7
8  before_script:
9  --pip install -r requirements.txt
10 script:
11 - pytest --testit --testrunid=${TEST_RUN_ID} --
  testit_url=${URL} --privatetoken=${PRIVATE_TOKEN}
```

yaml

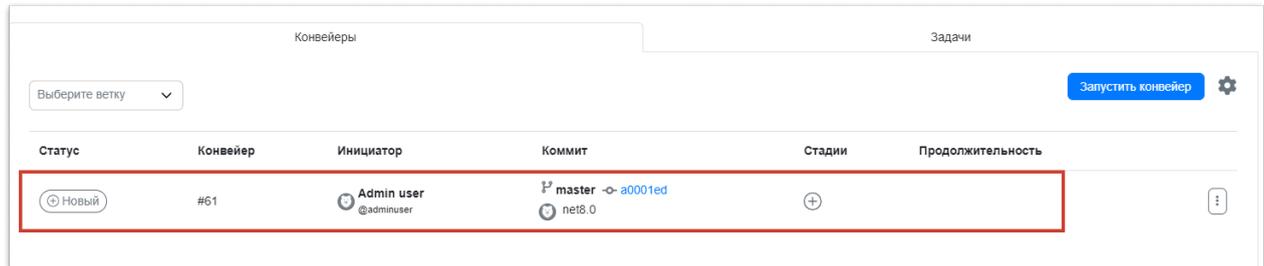
2. Выполните коммит изменений.

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:

```
1  {
2  "accessToken":
3  "868f3765-aaaa-aaaa-aaaa-13faw8a3ca68",
4  "refreshToken":
5  "cb4917e7-aaaa-aaaa-aaaa-9bb604b42e0c",
6  "expires": "2023-01-12T15:45:01.961001"
7  }
```

.json

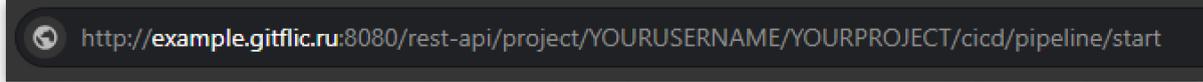
2. Скопируйте значение `accessToken` из полученного объекта `.json`.

## 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.

# 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:

```
1 | pytest
2 | testit-adapter-pytest
```

text

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.

```
1 [testit]
2 url={URL}
3 privatetoken={PRIVATE_TOKEN}
4 testrunID={TEST_RUN_ID}
```

text

## 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`.

```
1 stages:
2   - test
3
4 run_tests:
5   image: python:3.9
6   stage: test
7
8   before_script:
9     --pip install -r requirements.txt
10  script:
11    - pytest --testit
```

text

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

```
1 stages:
2   - test
3
4 run_tests:
5   image: python:3.9
6   stage: test
7
8 before_script:
9   --pip install -r requirements.txt
10 script:
11   - 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: 12.10.2025, 20:50:48

# 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:

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

Ответ:

```
1 StatusCode: 201 text
2 ReasonPhrase: 'Created'
3 Version: 1.1
4 Content:
5 System.Net.Http.HttpConnection+HttpConnectionResponseContent
6 Headers:
7 {
8 Date: Fri, 26 Jul 2019 16:32:34 GMT
9 X-Content-Type-Options: nosniff
10 Location: http://jenkins.mycompanyname.ru/queue/item/1777974/
11 Server: Jetty(9.4.z-SNAPSHOT)
12 Content-Length: 0
   }
```

В Jenkins запустилась сборка проекта с полученными параметрами.

# Импорт и экспорт проектов

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 Test IT Lite или Standard в Test IT Enterprise. Объем импортируемых и экспортируемых файлов — до 1 ГБ.

## Экспортируемые / импортируемые данные

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

- Проект и его описание
- Библиотека тестов и их версии
- Тест-планы
- Автотесты
- Прогноны 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

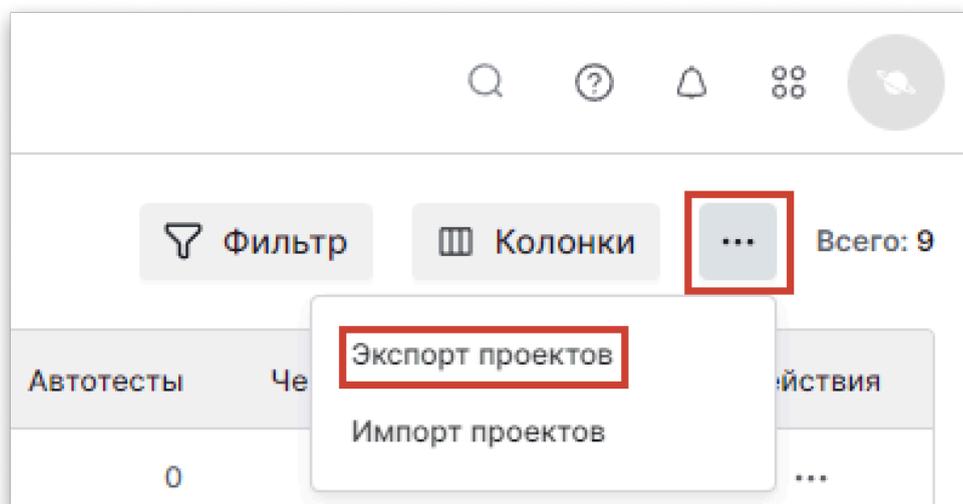
## Экспорт проектов

---

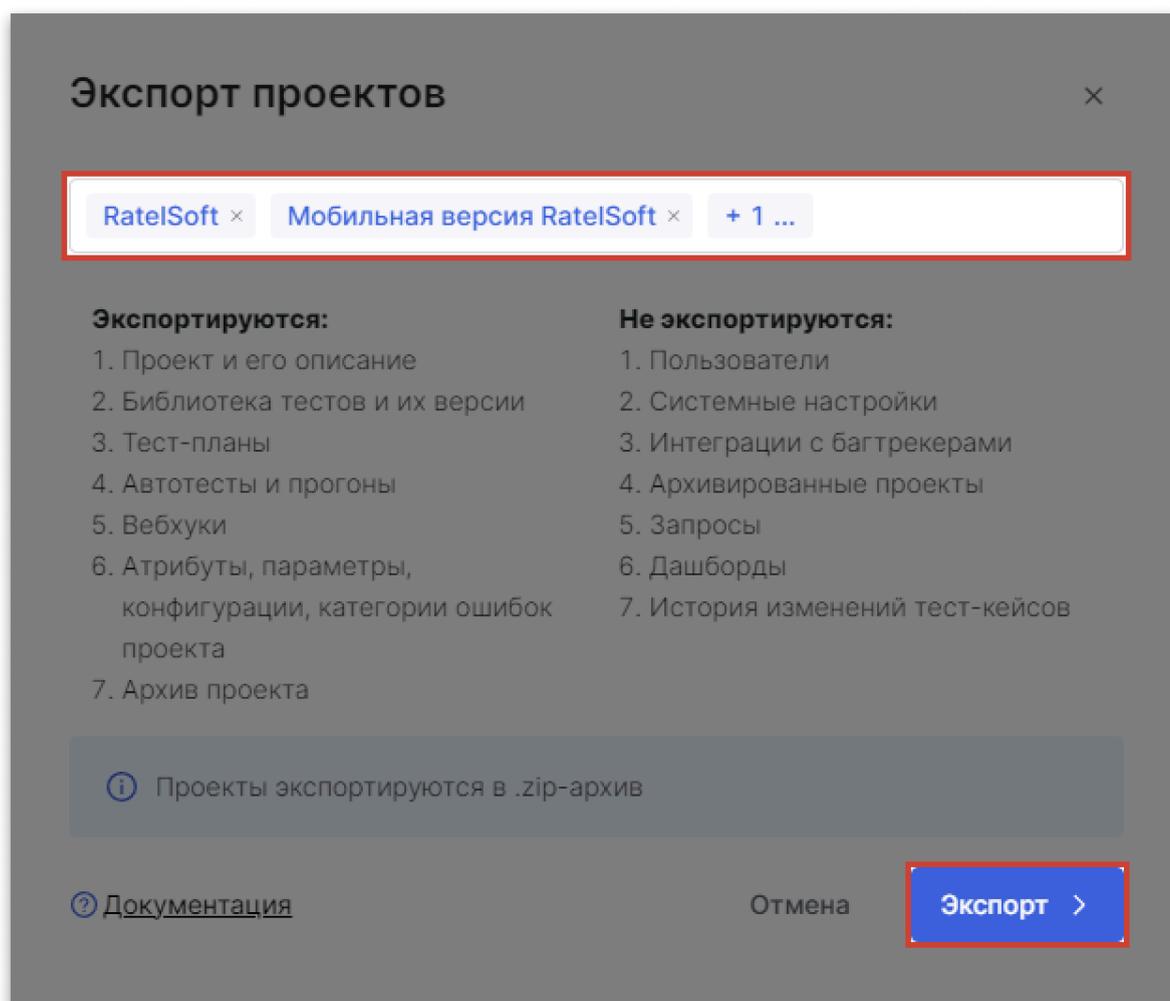
Вы можете экспортировать проекты из Test IT в формате `.zip`. Все экспортируемые проекты сохраняются в одном `.zip`-архиве. Массовый экспорт проектов доступен только при наличии действующей коммерческой лицензии.

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**.

- По окончании процесса скачайте `.zip` -архив с проектами на странице **Фоновых задач**.

## Импорт проектов

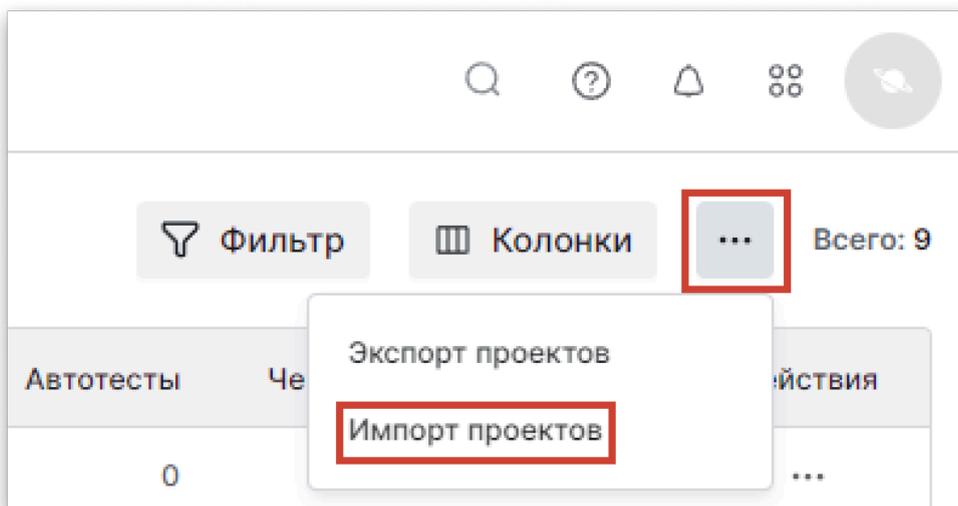
Вы можете импортировать проекты, архивированные в один `.zip` -файл. За один раз можно импортировать только один архив.

### Внимание!

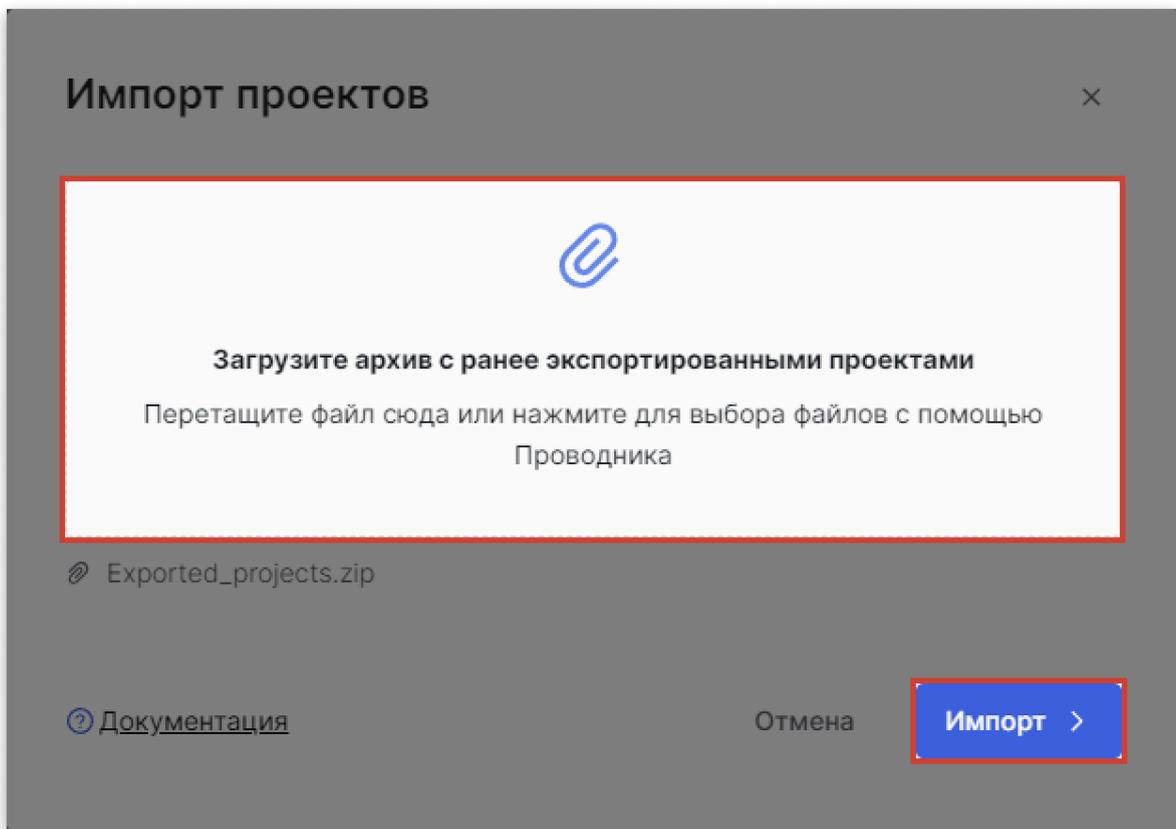
При повторном импорте проекты не будут заменены: в Test IT будут созданы новые идентичные проекты.

Чтобы импортировать проекты:

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



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

# Подключение task-трекера к проекту

## Требуется настроенная интеграция

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

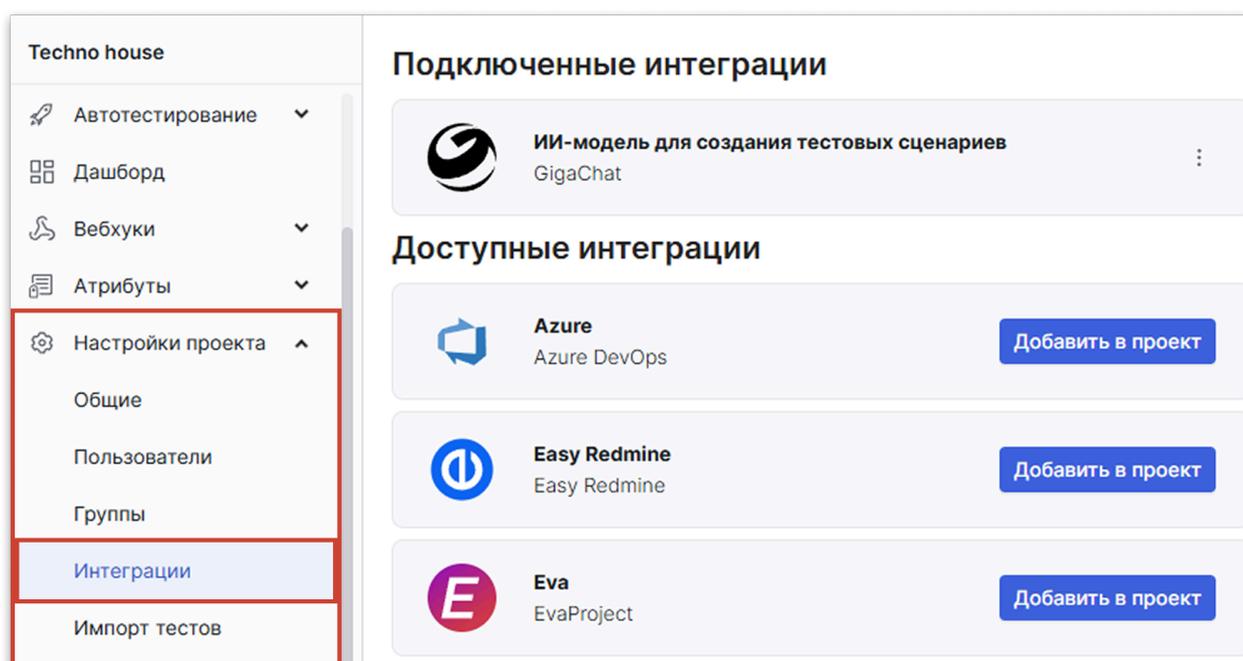
Test IT можно подключить к task-трекеру и создавать задачи из интерфейса Test IT.

### ► Список поддерживаемых task-трекеров

Чтобы подключить task-трекер к проекту Test IT:

1. Откройте проект.
2. Перейдите в раздел **Настройки проекта** → **Интеграции**.

Откроется список доступных интеграций, созданных ранее в разделе **Администрирование**.



The screenshot displays the 'Techno house' project settings page. On the left is a navigation sidebar with the following items: 'Автотестирование', 'Дашборд', 'Вебхуки', 'Атрибуты', 'Настройки проекта' (highlighted with a red box), 'Общие', 'Пользователи', 'Группы', 'Интеграции' (highlighted with a red box), and 'Импорт тестов'. The main content area is titled 'Подключенные интеграции' and shows one active integration: 'ИИ-модель для создания тестовых сценариев' (GigaChat). Below this is the 'Доступные интеграции' section, which lists three available integrations: 'Azure' (Azure DevOps), 'Easy Redmine', and 'Eva' (EvaProject). Each of these three integrations has a blue button labeled 'Добавить в проект'.

3. В списке **Доступные интеграции** выберите созданную ранее интеграцию с task-трекером и нажмите **Добавить в проект**.
4. Укажите проект в task-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: 22.10.2025, 19:28:57

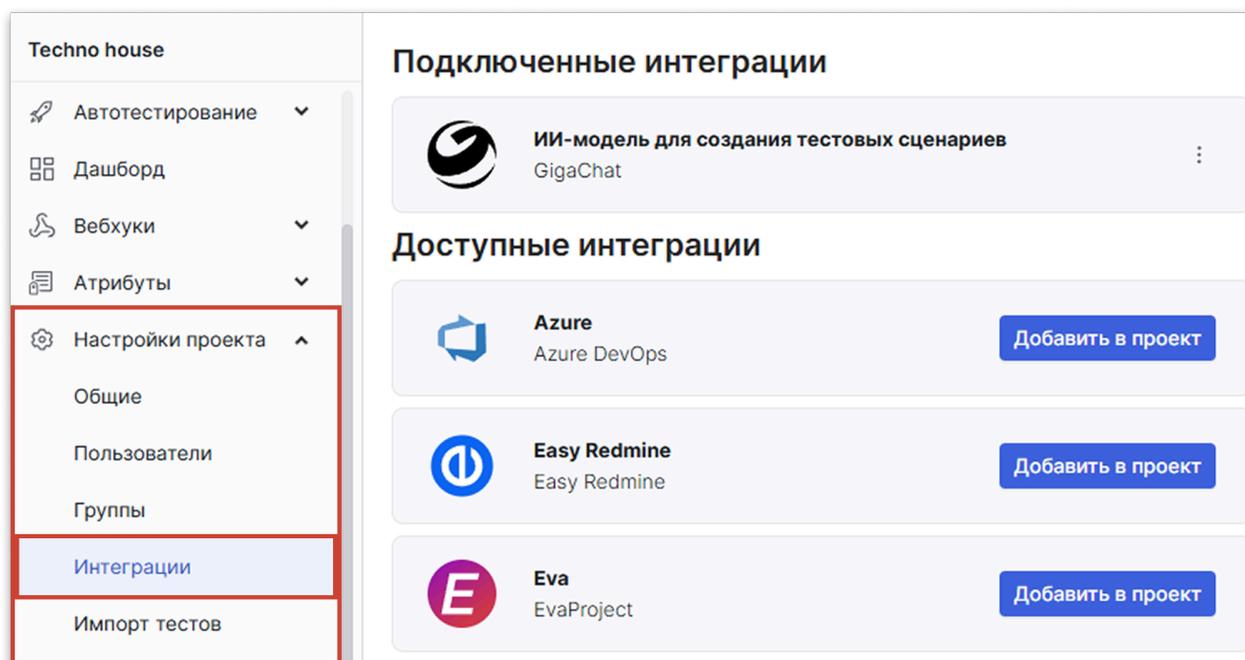
# Добавление ИИ-модели в проект

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 и 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**.

**Подключенные интеграции**

-  **Open AI Compatible: генерация тестов**  
OpenAI compatible
-  **TeamStorm**  
TeamStorm

**Доступные интеграции**

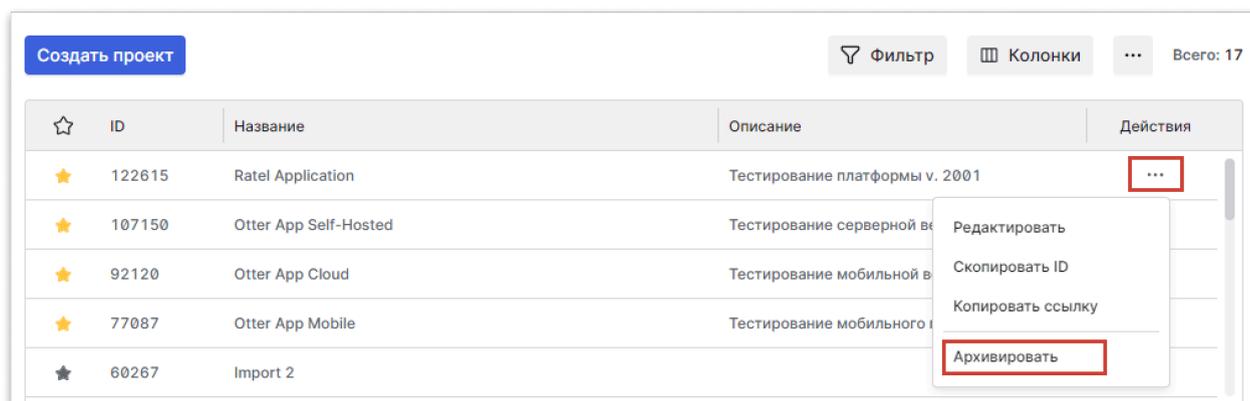
-  **Azure DevOps Issues**  
Azure DevOps [Добавить в проект](#)

# 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**.



The screenshot shows a web interface for project management. At the top left is a blue button labeled "Создать проект". To the right are buttons for "Фильтр", "Колонки", and a menu icon, followed by the text "Всего: 17". Below this is a table with the following columns: "ID", "Название", "Описание", and "Действия". The table contains five rows of project data. The first row is highlighted, and a context menu is open over the "Действия" cell of this row. The menu items are "Редактировать", "Скопировать ID", "Копировать ссылку", and "Архивировать". The "Архивировать" option is highlighted with a red box.

ID	Название	Описание	Действия
122615	Ratel Application	Тестирование платформы v. 2001	...
107150	Otter App Self-Hosted	Тестирование серверной в	Редактировать
92120	Otter App Cloud	Тестирование мобильной в	Скопировать ID
77087	Otter App Mobile	Тестирование мобильного	Копировать ссылку
60267	Import 2		Архивировать

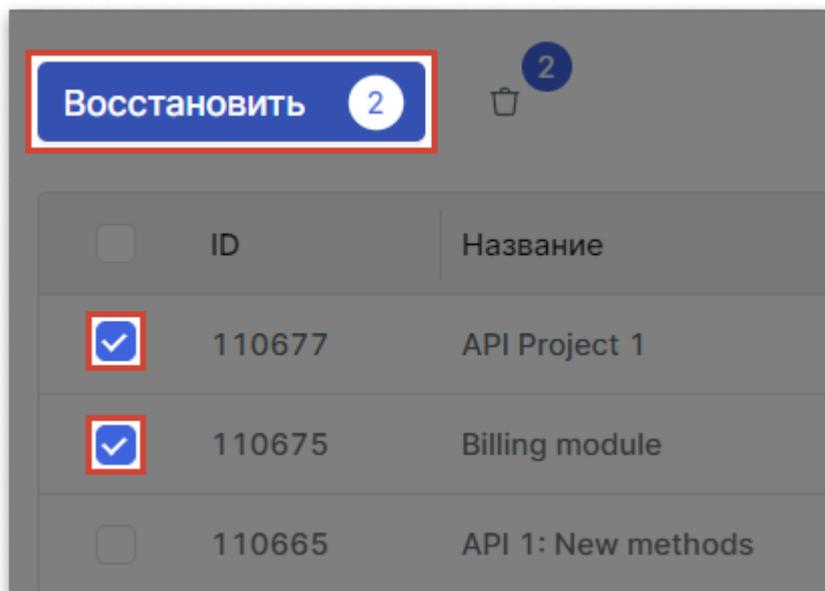
3. Подтвердите действие в диалоговом окне.

## Восстановление и удаление проектов

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

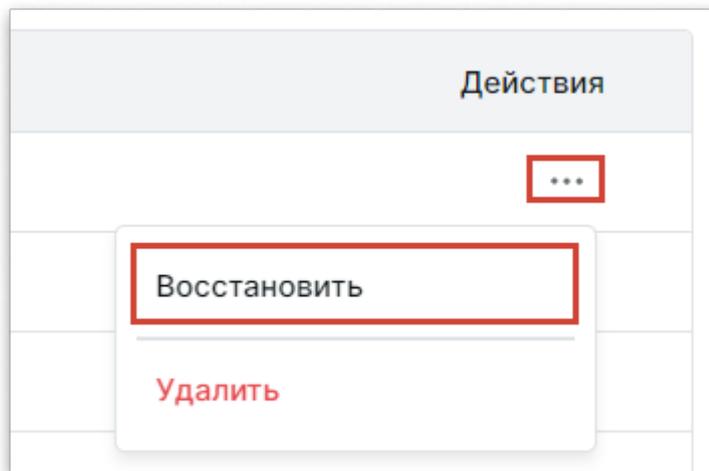
Чтобы восстановить проекты:

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.

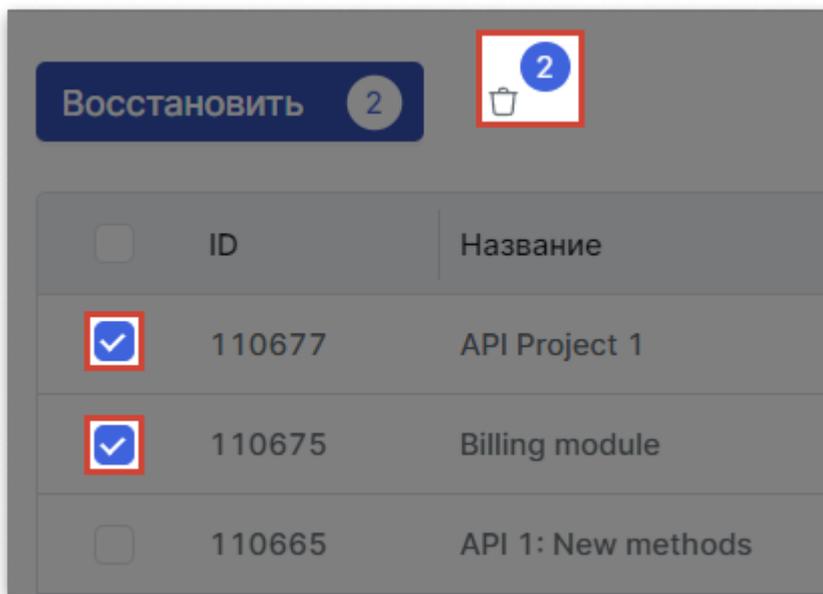
- **Опционально:** Чтобы восстановить отдельный проект, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.



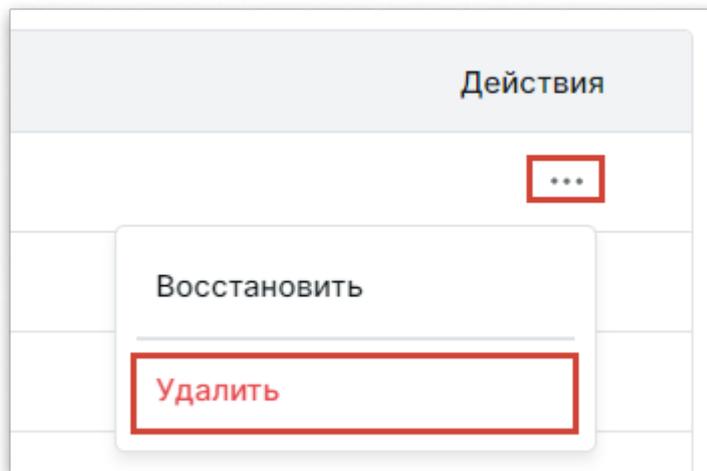
3. Подтвердите действие в диалоговом окне. Проект снова станет активным и появится в списке проектов.

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.

# 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.

- Для этого раздела мы подготовили видео: [Рабочие элементы в Test IT](#) .

## Используйте горячие клавиши

Для действий с рабочими элементами вы можете воспользоваться горячими клавишами.

## Типы рабочих элементов Test IT

---

В Test IT есть три вида 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](#).

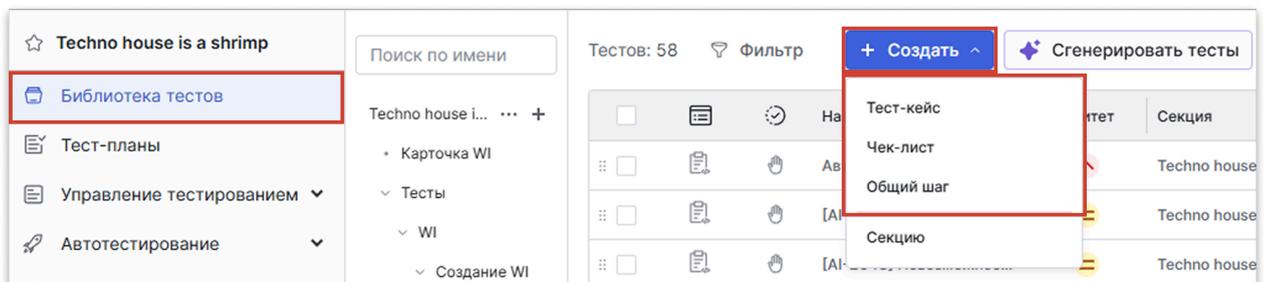
## Создание тест-кейсов, чек-листов и общих шагов

---

- 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**). В этой инструкции выбран тип **Тест-кейс**.



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

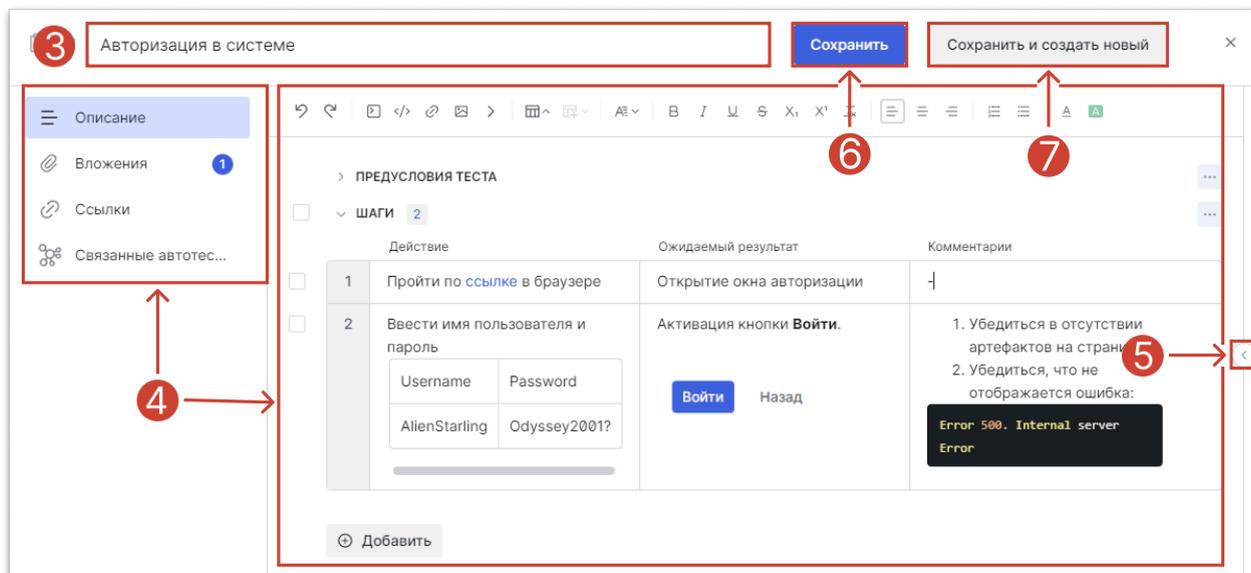
можно копировать таблицы из шагов, предусловий и постусловий в Test IT и вставлять их в файлы Excel и Word.

- Вы можете воспользоваться редактором для изменения уже существующих рабочих элементов.

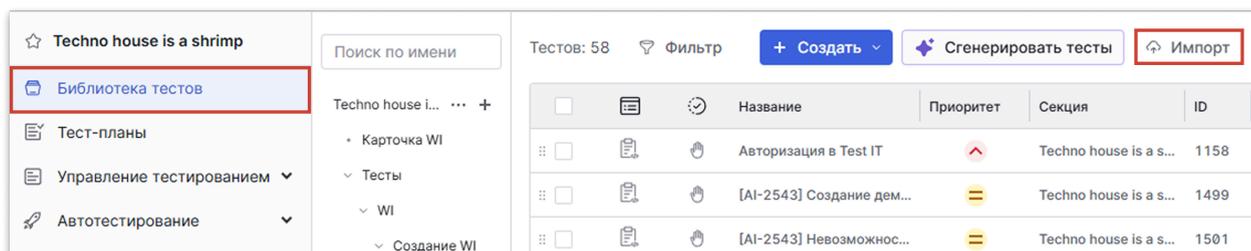
5. В правой части окна задайте время его прохождения, приоритет, статус готовности и значения атрибутов, добавленных в проект.

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 **Создать**.



## Генерация тестов с помощью ИИ-моделей

Требуется настроенная интеграция в проекте

Для ИИ-генерации тестов в проект должны быть добавлены интеграции с ИИ-моделью и 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 — до 10 (это значение предустановлено и не может быть изменено). ИИ-генерация платная, оплата производится провайдеру ИИ-модели.

## Рекомендации по формулированию задач

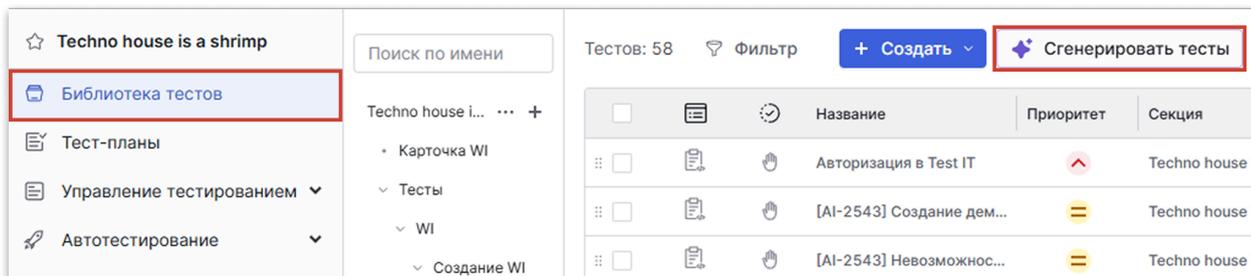
Чтобы создание тестов с помощью ИИ-модели было продуктивным и безопасным, перед началом работы убедитесь, что:

- 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](#) .

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. В поле **Номер** введите номер задачи из таск-трекера (например, *TS-2567*).
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.

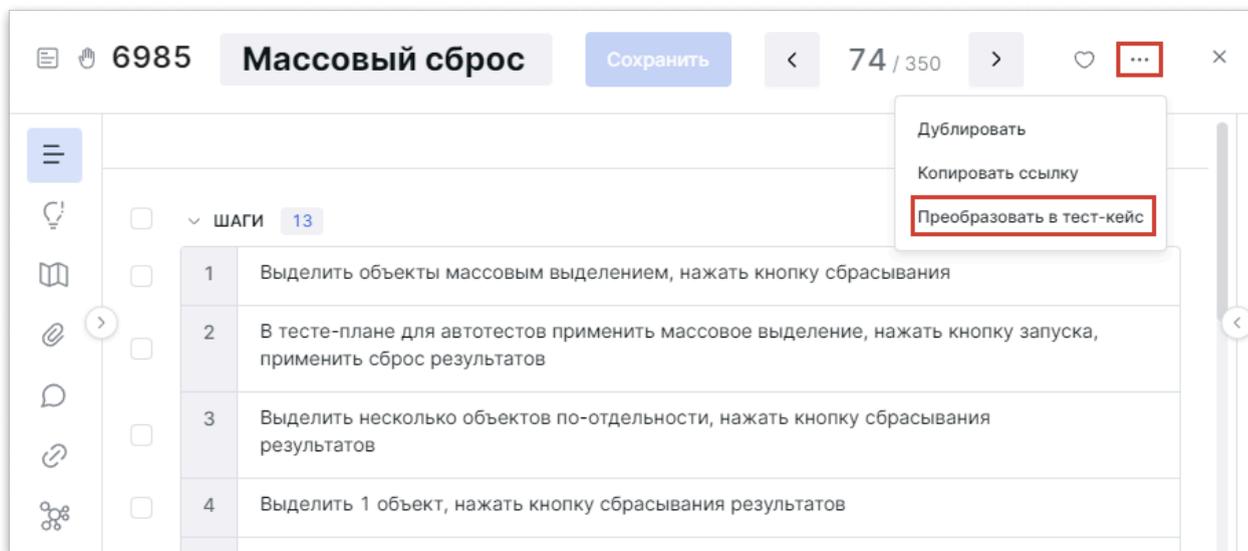
## Создание тест-кейсов из чек-листов

### 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: 26.01.2026, 12:58:29

# 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 <b>Тестовые данные</b> и <b>Комментарии</b>	Да	Нет	Да
Отслеживание истории результатов	Да	Да	Нет
Добавление входных параметров в шаги	Да	Да	Нет
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

# 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:

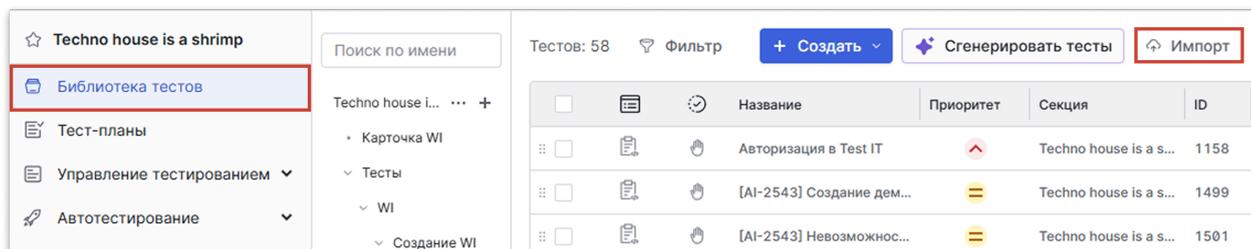
-  В Test IT Lite & Standard — 50 МБ
-  В Test IT Enterprise — 1 ГБ

## 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 текстНажмите **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: 19.01.2026, 16:47:34

# Версионирование тестов

## ₽ Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тариф **Lite** она не входит.

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.

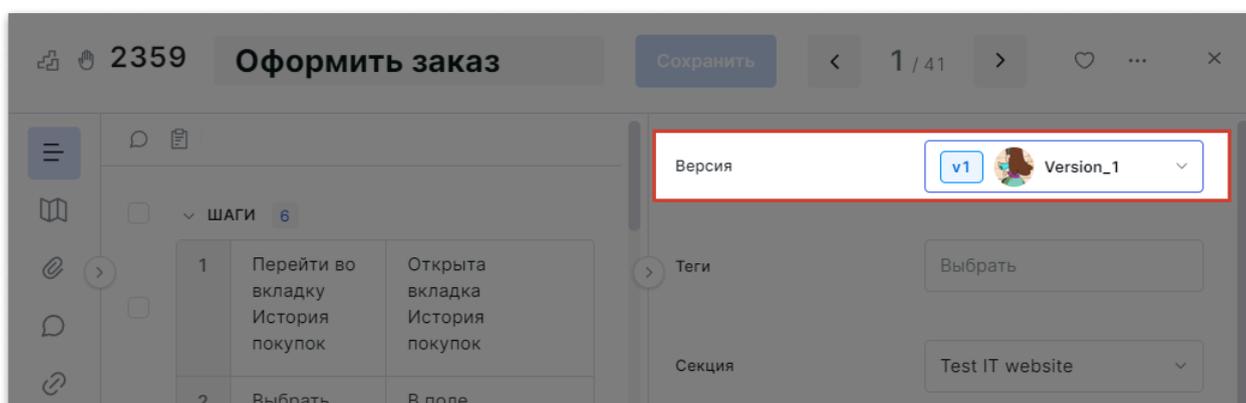
## Изменения тест-поинтов, связанные с версионированием тест-кейсов

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.

## 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.

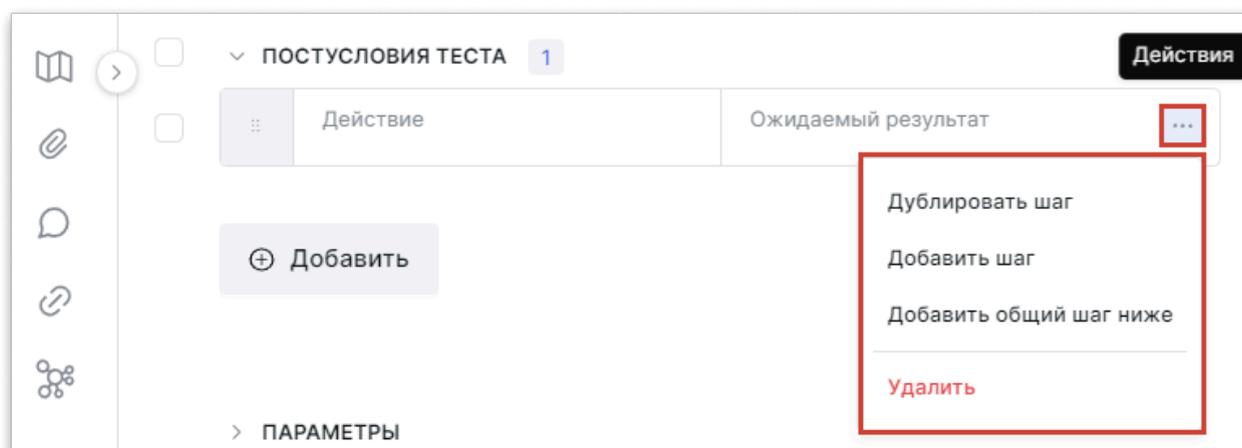
В журнале отображается, когда и кем было внесено изменение, а также значения до и после изменений. Удаления символов в текстовых полях выделяются красным цветом с перечеркиванием, а добавления символов — зеленым.

Обновлено: 17.10.2025, 13:38:51

# 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. Нажмите **Добавить общий шаг** в выпадающем меню в правой части строки шага.



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.

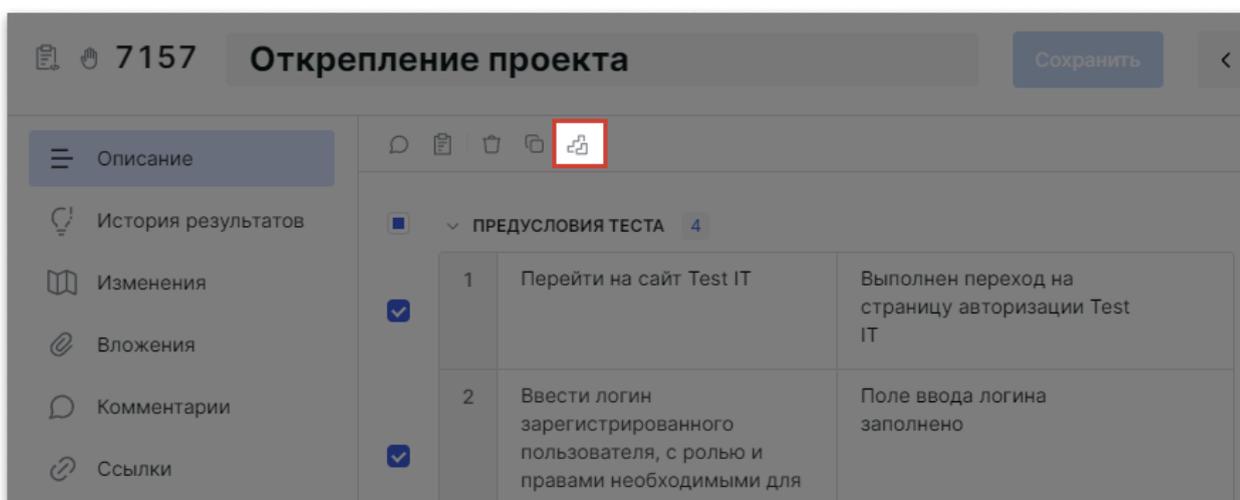
Обновлено: 11.10.2024, 18:42:03

# 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: 11.11.2025, 21:27:49

# Работа с параметрами тестов

Добавляйте параметры в тест-кейсы и чек-листы, которые хотите протестировать с разными наборами входных данных, не дублируя рабочие элементы. Например, для тестов по авторизации в сервис можно создать параметры `login` и `password` с несколькими возможными значениями для каждого из них.

## Параметры ассоциируются с проектами Test IT

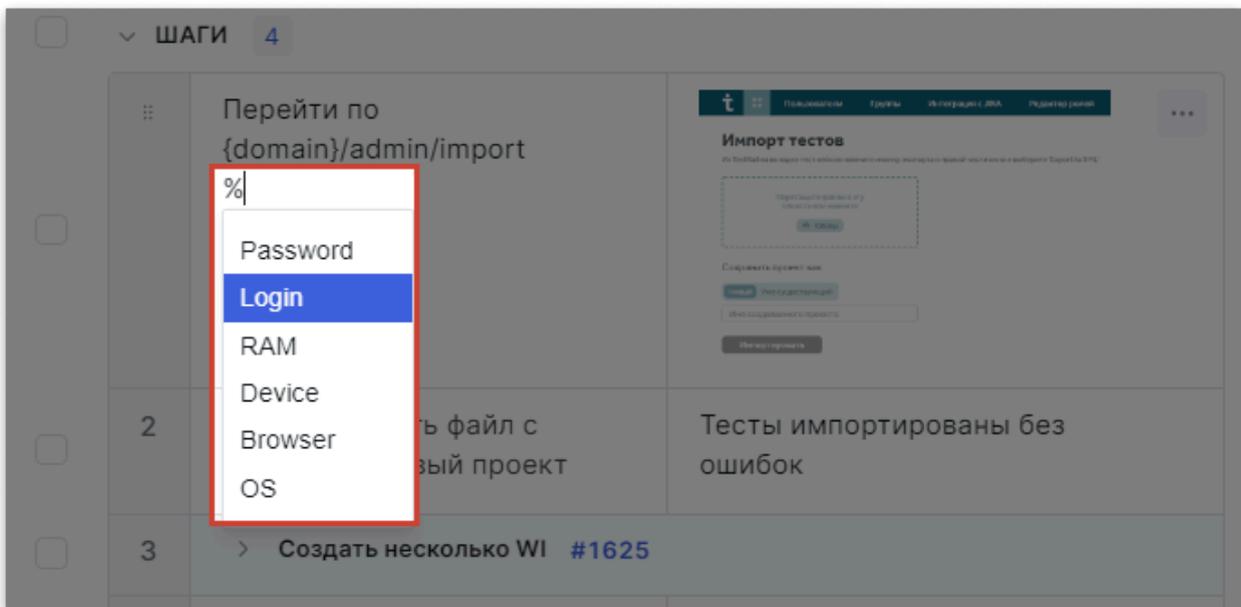
При создании и редактировании параметров указываются проекты, в которых параметр может быть использован. Если вы создали параметр и не обнаружили его в проекте, требуется **отредактировать параметр** в настройках проектов, добавив нужный проект.

## Добавление параметров теста

---

Чтобы добавить параметр в тест-кейс или чек-лист:

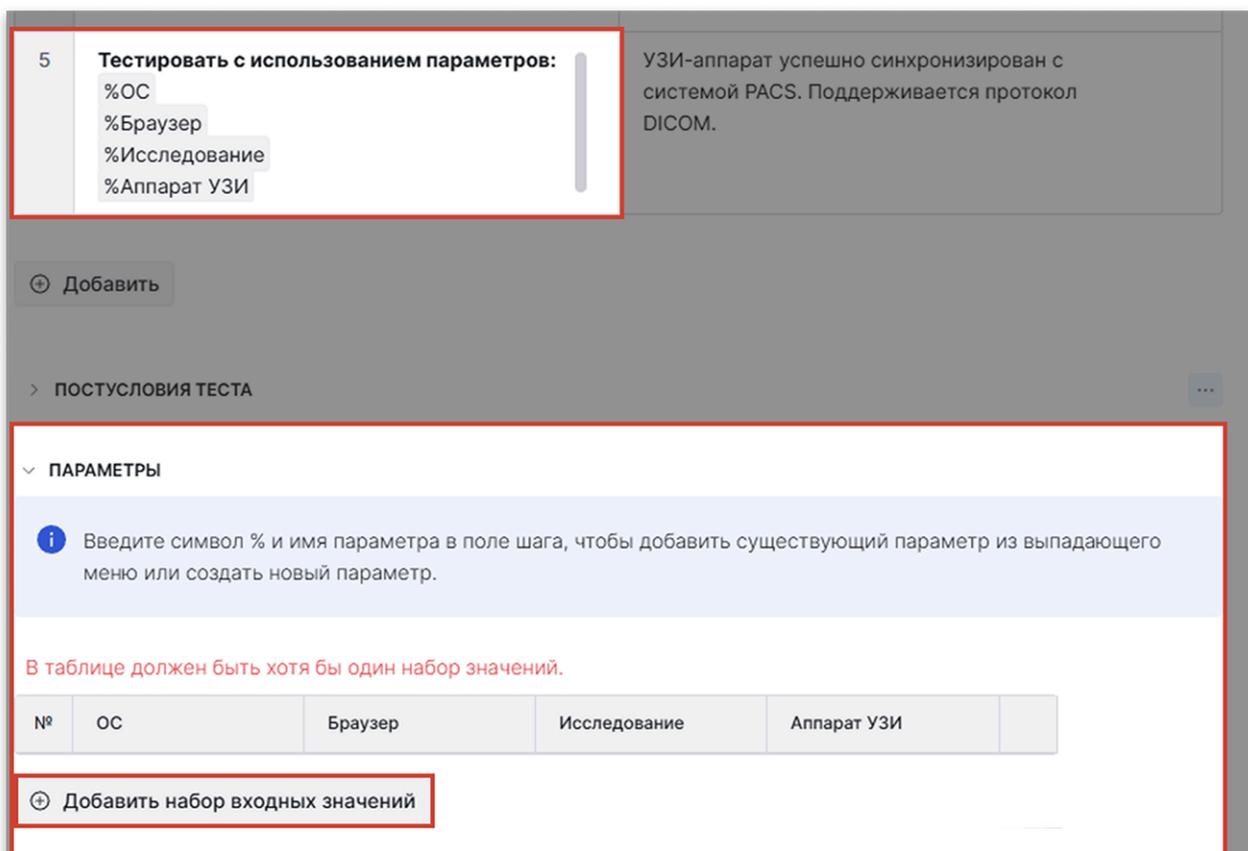
1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Создайте или откройте рабочий элемент, в который хотите добавить параметр.
3. Введите `%` в поле редактирования шага, чтобы указать параметр. Например, для теста по авторизации вы можете добавить параметры в 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**, расположенный в нижней части окна теста, добавьте наборы значений параметров, с которыми тест будет пройден. Для нажмите **Добавить набор входных значений** и добавьте нужные значения.



5. Сохраните изменения в тесте.

При использовании нескольких значений параметра, тест будет проходиться столько раз, сколько значений параметров задействовано в тесте. Например, при наличии двух значений параметра в тест-плане один тест будет отображаться как два тест-поинта с разными значениями параметра.

## Смотрите также

---

- [Создание тест-плана](#)
- [Выполнение тест-плана](#)

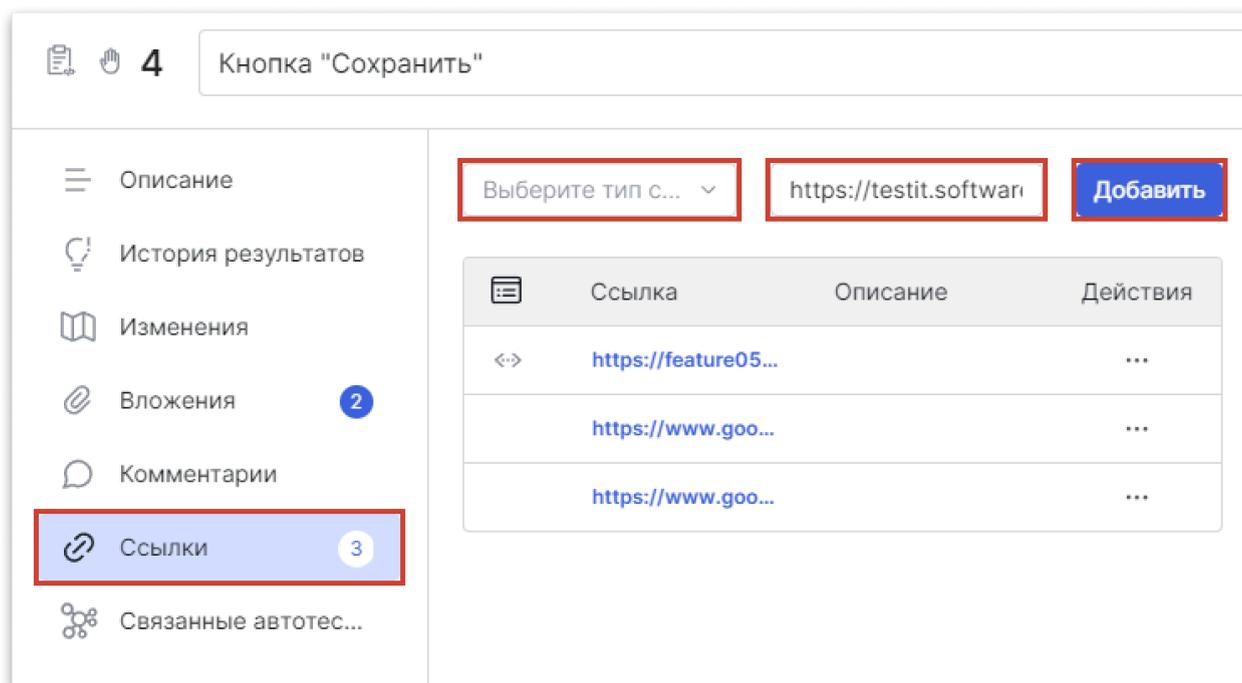
Обновлено: 23.12.2025, 13:01:31

# 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

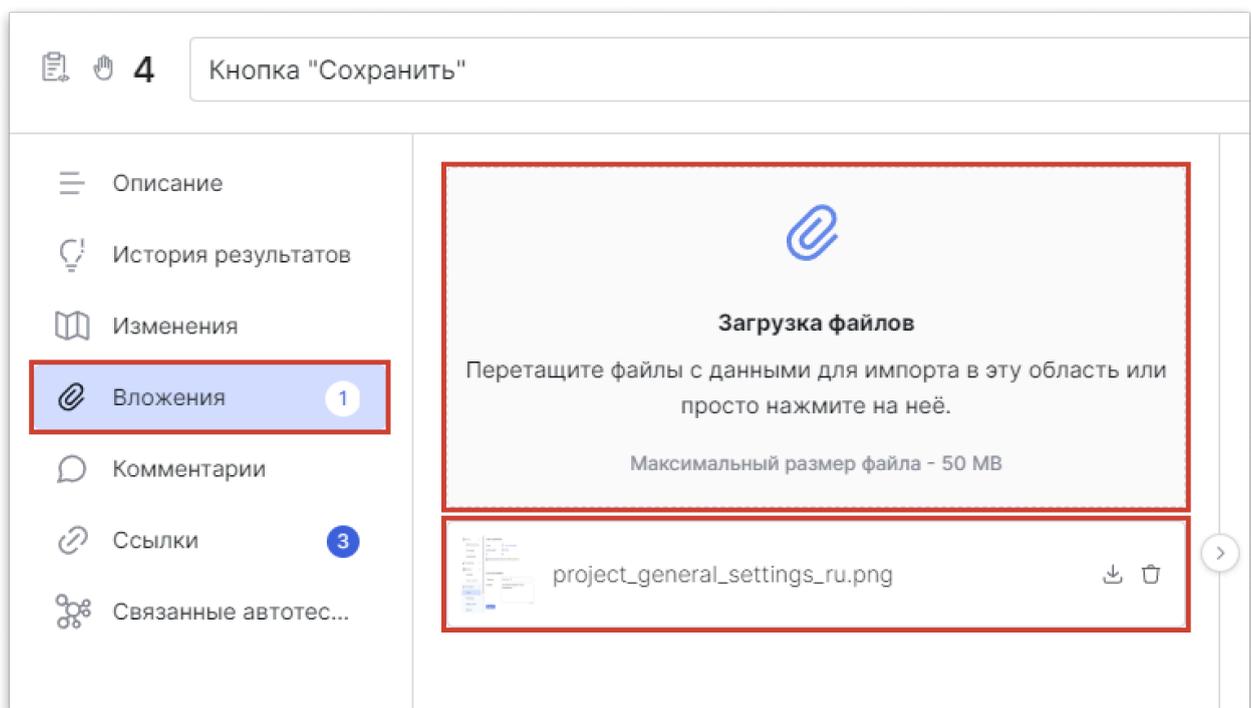
Обновлено: 24.09.2025, 20:40:39

# 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

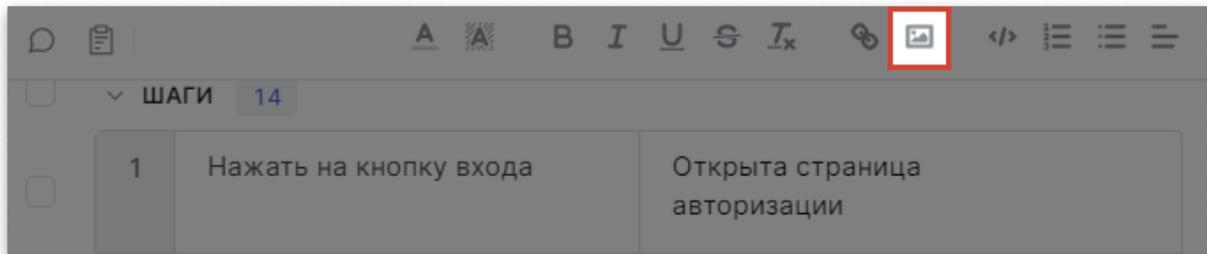
1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте рабочий элемент, в который хотите добавить вложение.
3. В окне редактирования рабочего элемента перейдите в раздел **Вложения**.



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.

# Использование тегов

## 🔖 Тарифные ограничения для Lite

В тарифе **Lite** вы можете создавать теги без возможности редактирования и удаления. Библиотека тегов в **Lite** недоступна.

Подробнее: [о тарифах Test IT](#).

Теги — это слова-метки, сообщающие дополнительную информацию о рабочем элементе или тест-плане. 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 к рабочему элементу:

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте рабочий элемент, в который хотите добавить тег.
3. Введите название тега в поле **Теги**. Если тег с таким названием уже существует, он отобразится в выпадающем списке. Если тег новый, нажмите **Enter**, чтобы добавить его к тест-кейсу и сохранить в системе.

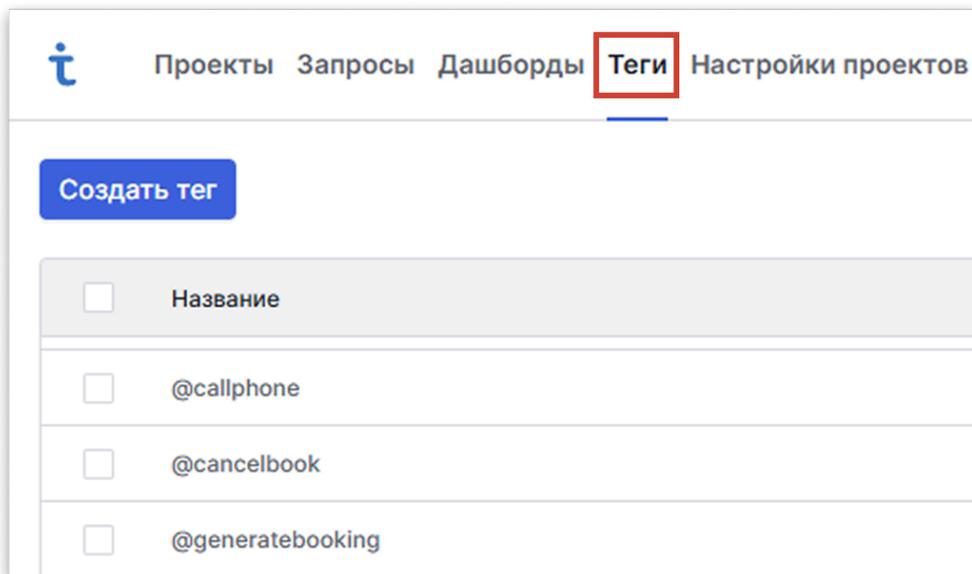


## Работа с библиотекой тегов

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

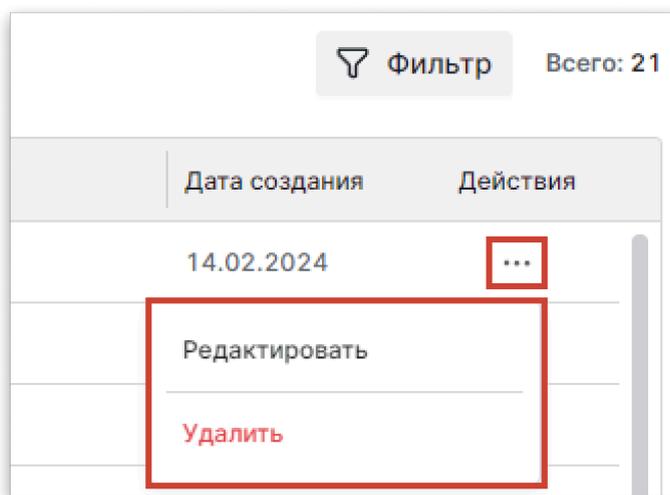
Чтобы открыть библиотеку тегов:

- 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.

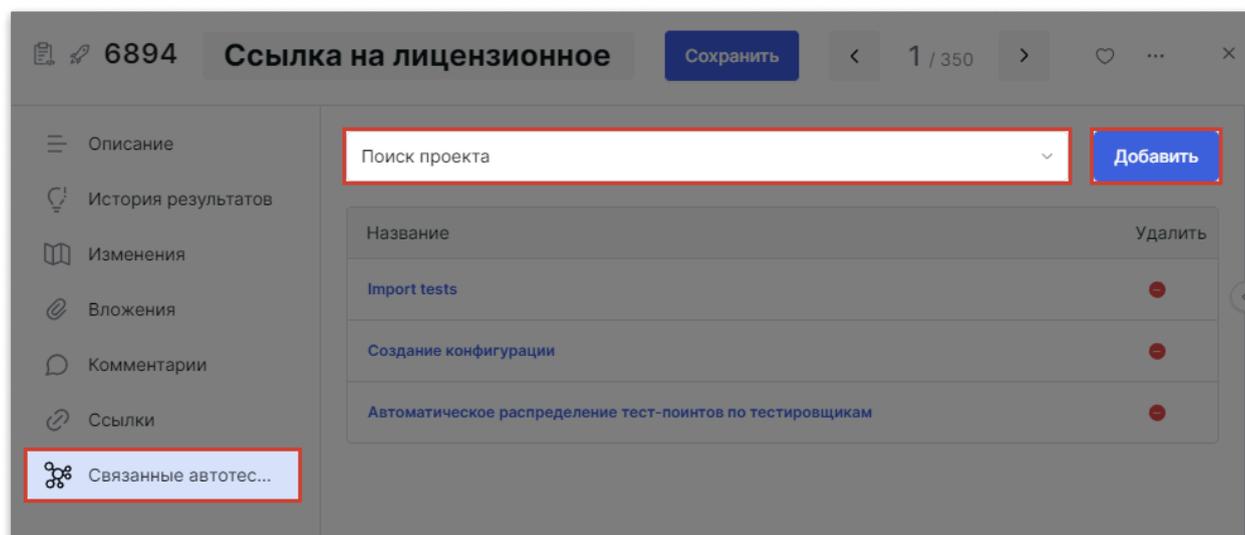
Обновлено: 11.11.2025, 21:27:49

# Автоматизация тест-кейсов

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

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.

Обновлено: 20.08.2025, 11:25:09

# 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

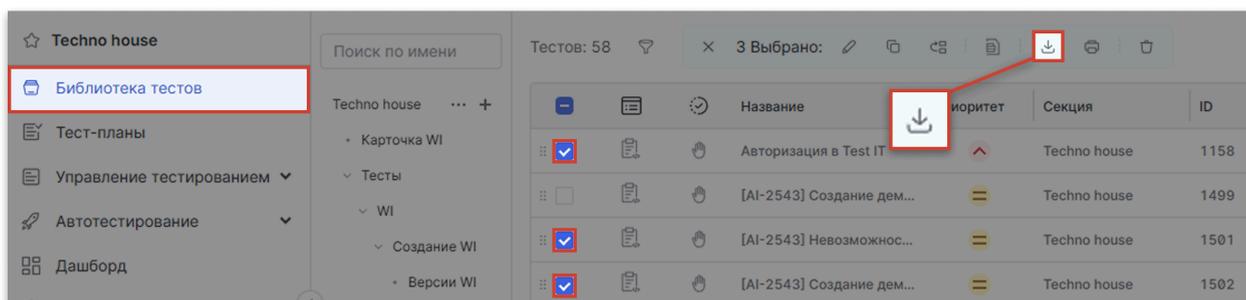
Вы можете экспортировать созданную вами в Test IT тестовую документацию в `.xlsx`.

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.



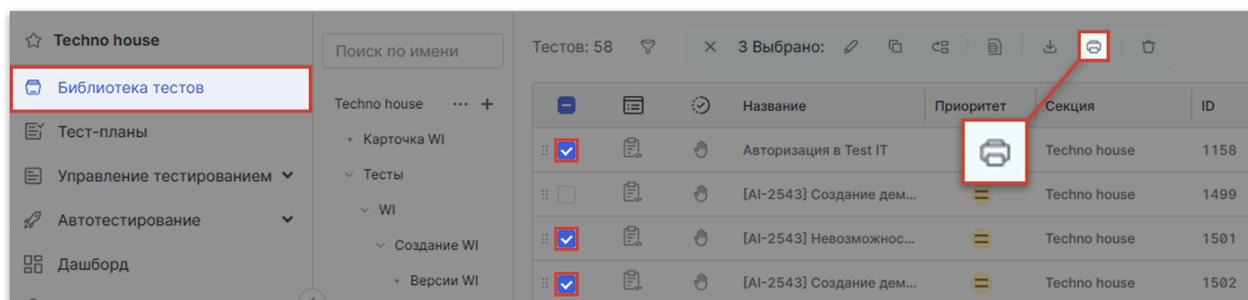
Файл с экспортированными тестами загрузится на ваше устройство. Чтобы узнать больше о том, как импортировать тестовую документацию из Excel-файлов в Test IT, смотрите [Импорт тестовой документации из XLS и XLSX](#).

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.



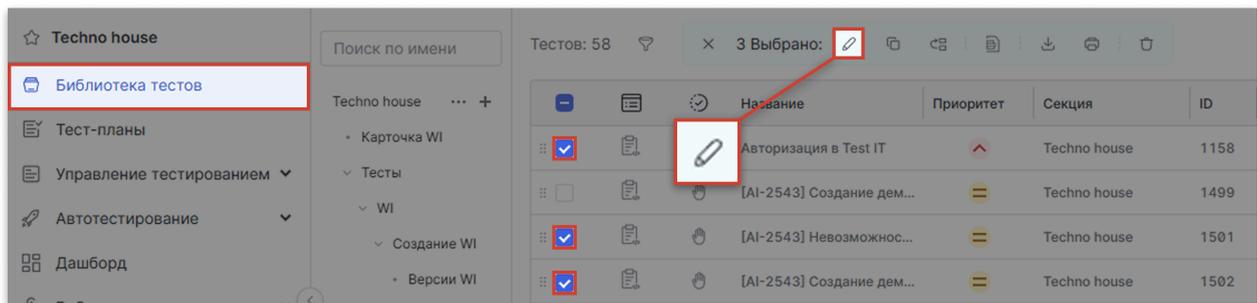
В новой вкладке откроется документ с рабочими элементами, конвертированными в версию для печати.

Обновлено: 11.11.2025, 21:27:49

# 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.

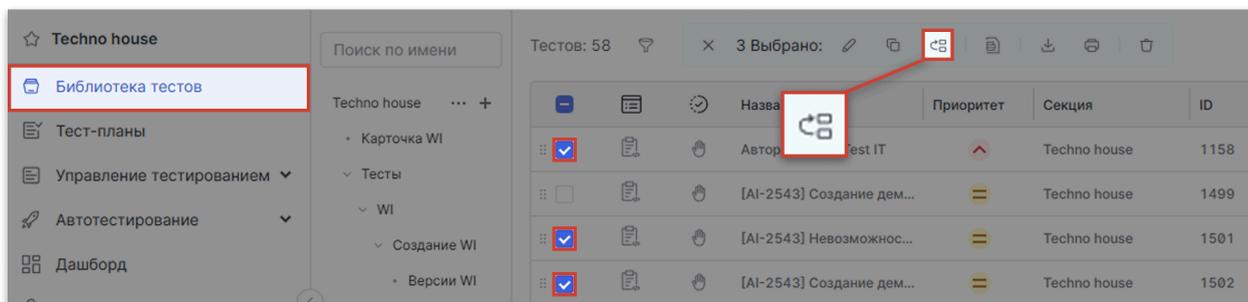
# Перемещение и копирование тестов в библиотеке

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

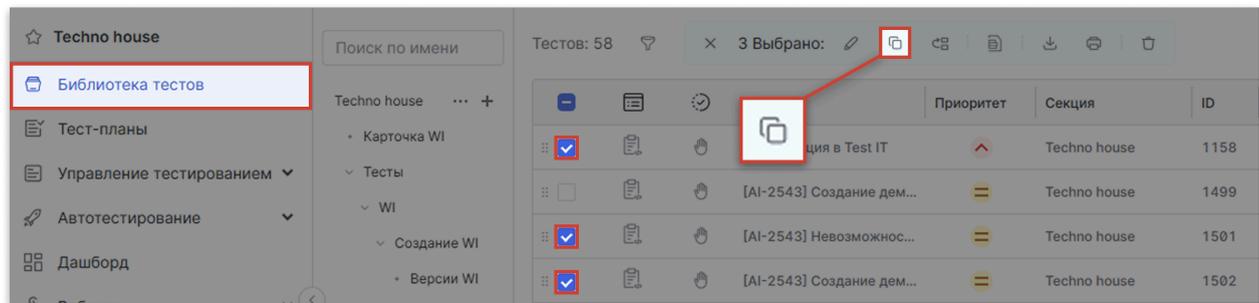
1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте рабочий элемент, который хотите переместить.
3. В поле **Секция** укажите секцию, в которую хотите переместить элемент.
4. Нажмите **Сохранить**.

## 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**.

# 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.

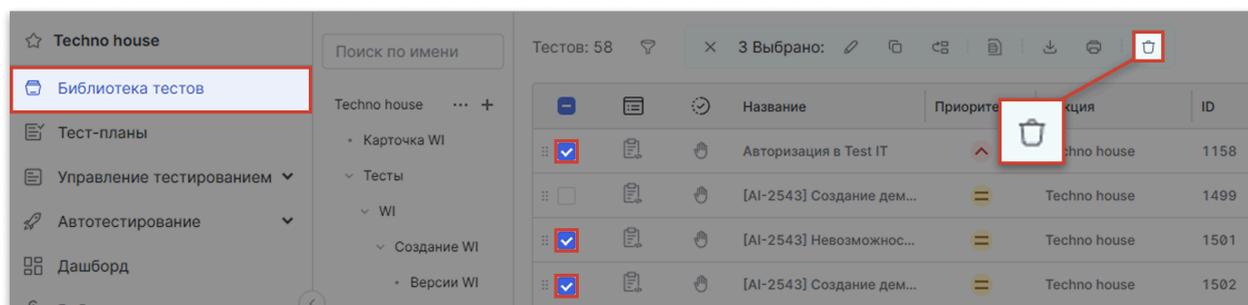
Вы можете **восстановить** рабочие элементы из архива в течение периода, заданного администратором системы в **системных настройках**. По окончании этого периода архив автоматически очищается.

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.



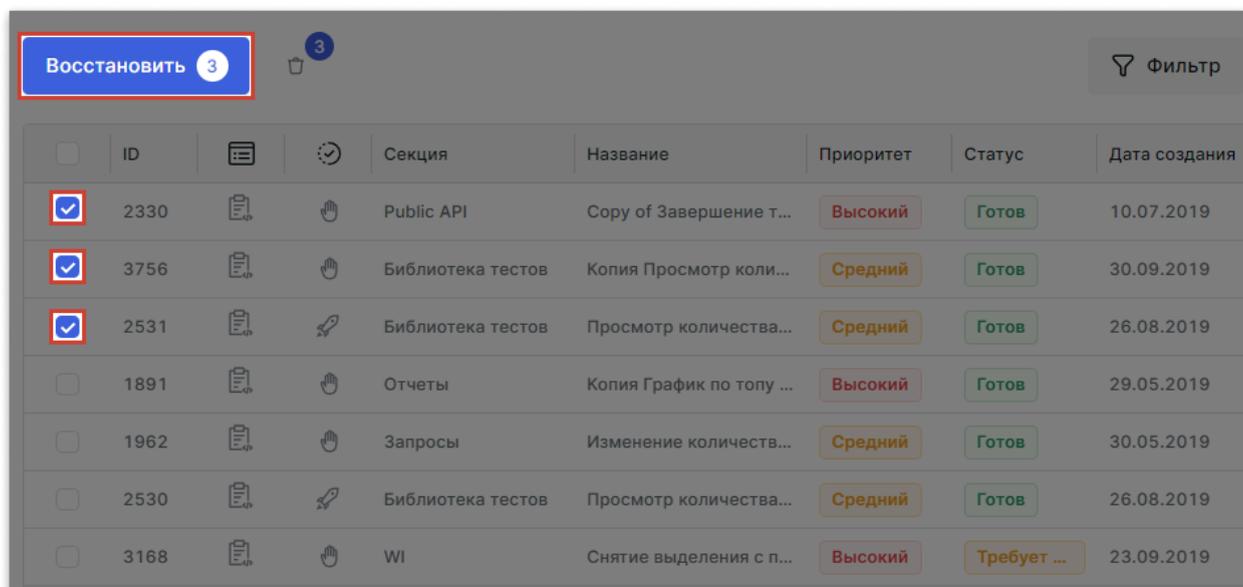
4. Подтвердите действие в диалоговом окне.  
Выбранные тесты перемещены в архив.

## 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:

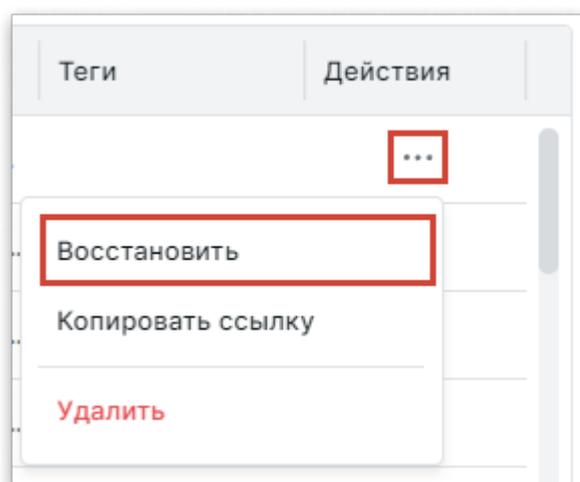
1. Откройте проект.
2. Перейдите в раздел **Архив** и выберите **Библиотека тестов**.
3. В списке ручных тестов выберите тесты, которые хотите восстановить, затем нажмите **Восстановить**.



The screenshot shows a table of test items in an archive. At the top left, there is a blue button labeled 'Восстановить' with a notification badge '3'. To its right is a trash icon with a badge '3'. On the far right, there is a 'Фильтр' button. The table has columns for selection, ID, actions, section, name, priority, status, and creation date. Three items are selected with checkboxes.

<input type="checkbox"/>	ID			Секция	Название	Приоритет	Статус	Дата создания
<input checked="" type="checkbox"/>	2330			Public API	Сору of Завершение т...	Высокий	Готов	10.07.2019
<input checked="" type="checkbox"/>	3756			Библиотека тестов	Копия Просмотр коли...	Средний	Готов	30.09.2019
<input checked="" type="checkbox"/>	2531			Библиотека тестов	Просмотр количества...	Средний	Готов	26.08.2019
<input type="checkbox"/>	1891			Отчеты	Копия График по топу ...	Высокий	Готов	29.05.2019
<input type="checkbox"/>	1962			Запросы	Изменение количеств...	Средний	Готов	30.05.2019
<input type="checkbox"/>	2530			Библиотека тестов	Просмотр количества...	Средний	Готов	26.08.2019
<input type="checkbox"/>	3168			WI	Снятие выделения с п...	Высокий	Требуется ...	23.09.2019

4. **Опционально:** Чтобы восстановить отдельные выбранные рабочие элементы, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.



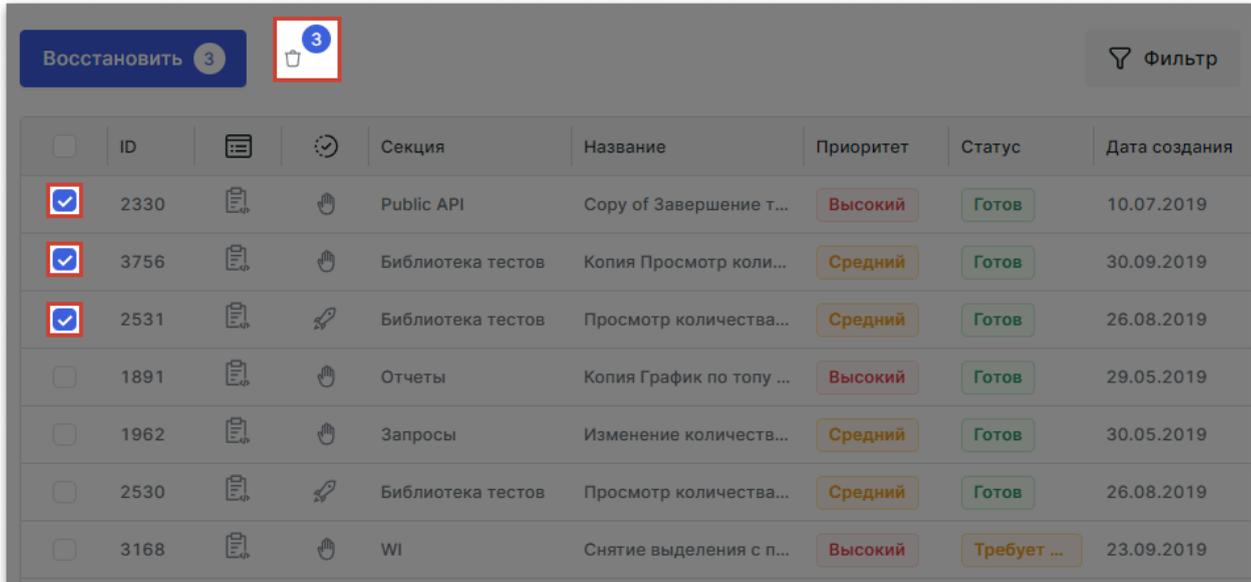
5. Подтвердите действие в диалоговом окне.  
Восстановленные рабочие элементы снова будут отображаться в разделе **Библиотека тестов**.

## Удаление тестов

Вы можете безвозвратно удалить рабочие элементы из архива.

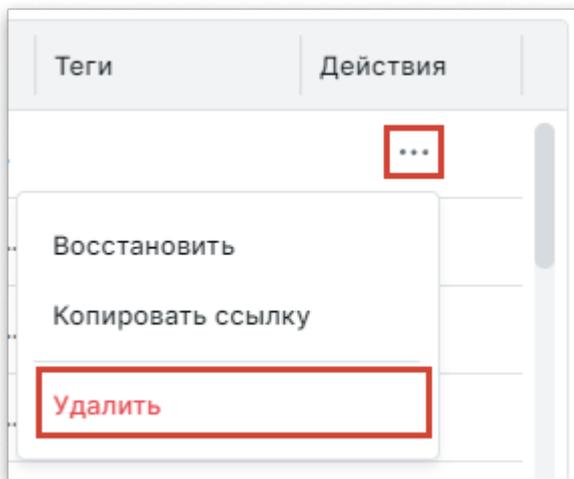
Чтобы удалить рабочие элементы:

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**.



<input type="checkbox"/>	ID			Секция	Название	Приоритет	Статус	Дата создания
<input checked="" type="checkbox"/>	2330			Public API	Сору of Завершение т...	Высокий	Готов	10.07.2019
<input checked="" type="checkbox"/>	3756			Библиотека тестов	Копия Просмотр коли...	Средний	Готов	30.09.2019
<input checked="" type="checkbox"/>	2531			Библиотека тестов	Просмотр количества...	Средний	Готов	26.08.2019
<input type="checkbox"/>	1891			Отчеты	Копия График по топу ...	Высокий	Готов	29.05.2019
<input type="checkbox"/>	1962			Запросы	Изменение количеств...	Средний	Готов	30.05.2019
<input type="checkbox"/>	2530			Библиотека тестов	Просмотр количества...	Средний	Готов	26.08.2019
<input type="checkbox"/>	3168			WI	Снятие выделения с п...	Высокий	Требуется ...	23.09.2019

4. **Опционально:** Чтобы удалить отдельные выбранные рабочие элементы, откройте выпадающее меню в колонке **Действия** и выберите **Удалить**.



5. Подтвердите действие в диалоговом окне.

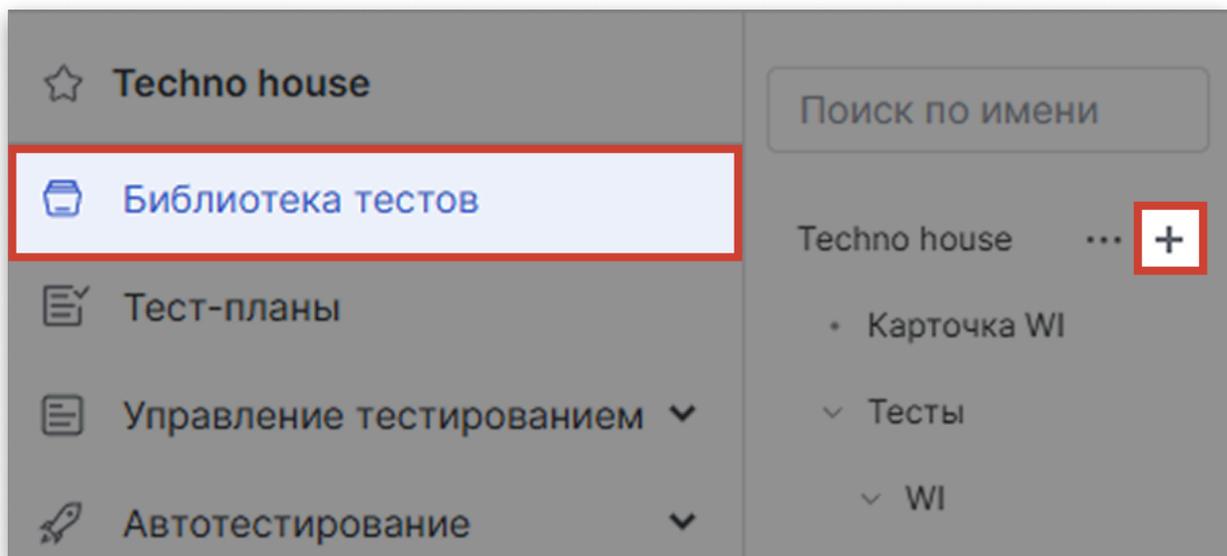
# Работа с секциями

- 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. В Test IT можно создать секцию и указывать пред- и постусловия для входящих в нее тестов.

## Создание секций

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. Откройте меню действий в панели библиотеки тестов и нажмите **Добавить**.



3. Введите название новой секции.
4. Нажмите **Сохранить**.

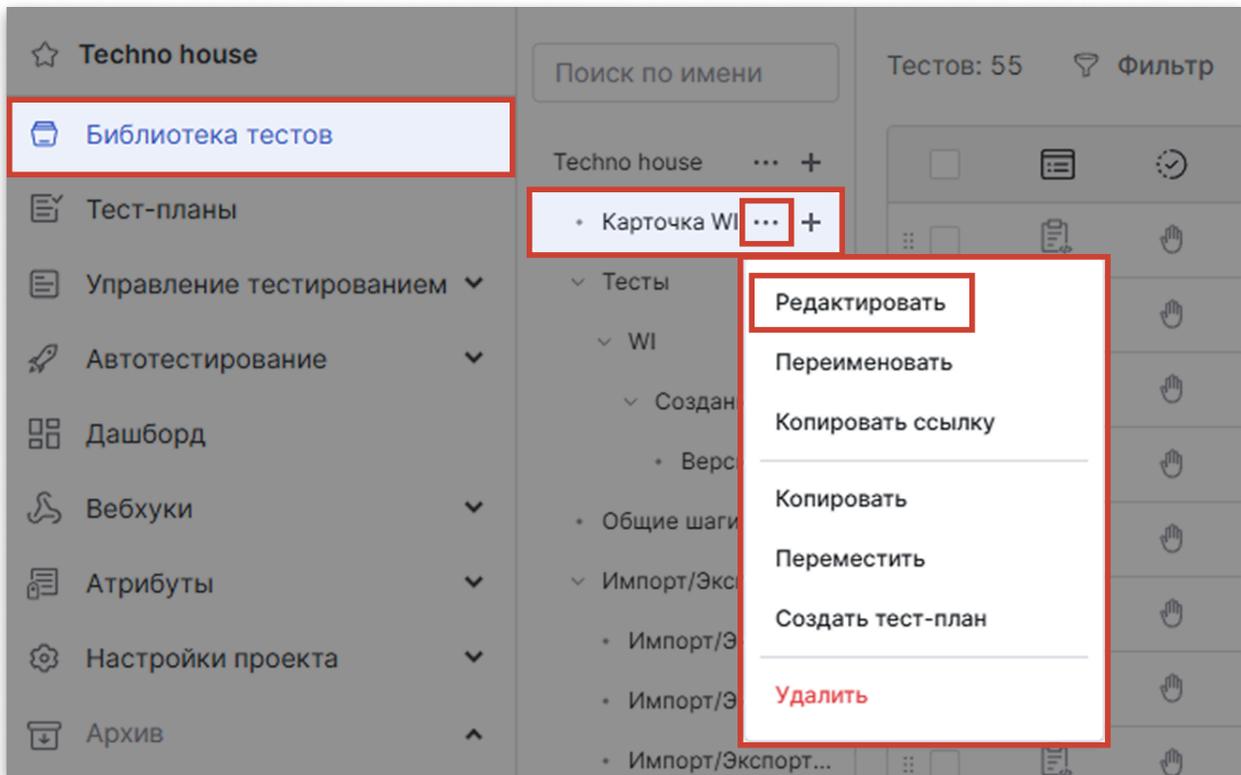
Созданная секция становится вложенной в ту секцию, из которой она была создана. Test IT поддерживает неограниченное количество уровней вложенности секций.

## Указание предусловий и постусловий секции

При создании секции вы можете задать общие предусловия и постусловия для всех тестов, входящих в секцию. Они применяются только к рабочим элементам,

вложенным в эту секцию, к дочерним секциям они не применяются. Чтобы задать общие условия:

1. Создайте новую секцию или откройте для редактирования уже существующие. Для этого откройте меню действий около секции и выберите **Редактировать**.



2. В окне редактирования секций откройте поля предусловий и постусловий.
3. Введите действия и ожидаемые результаты предусловий и постусловий тестов в секции. Вы можете воспользоваться встроенным редактором для форматирования текста. Если вам необходимо вставить в предусловия и постусловия общий шаг, откройте меню действий в правой части поля описания шага и нажмите **Добавить общий шаг ниже**.
4. Нажмите **Сохранить**.

После того как предусловия и постусловия секции заданы, они будут применяться ко всем рабочим элементам в этой секции. Предусловия и постусловия секции невозможно удалить или редактировать из рабочего элемента.

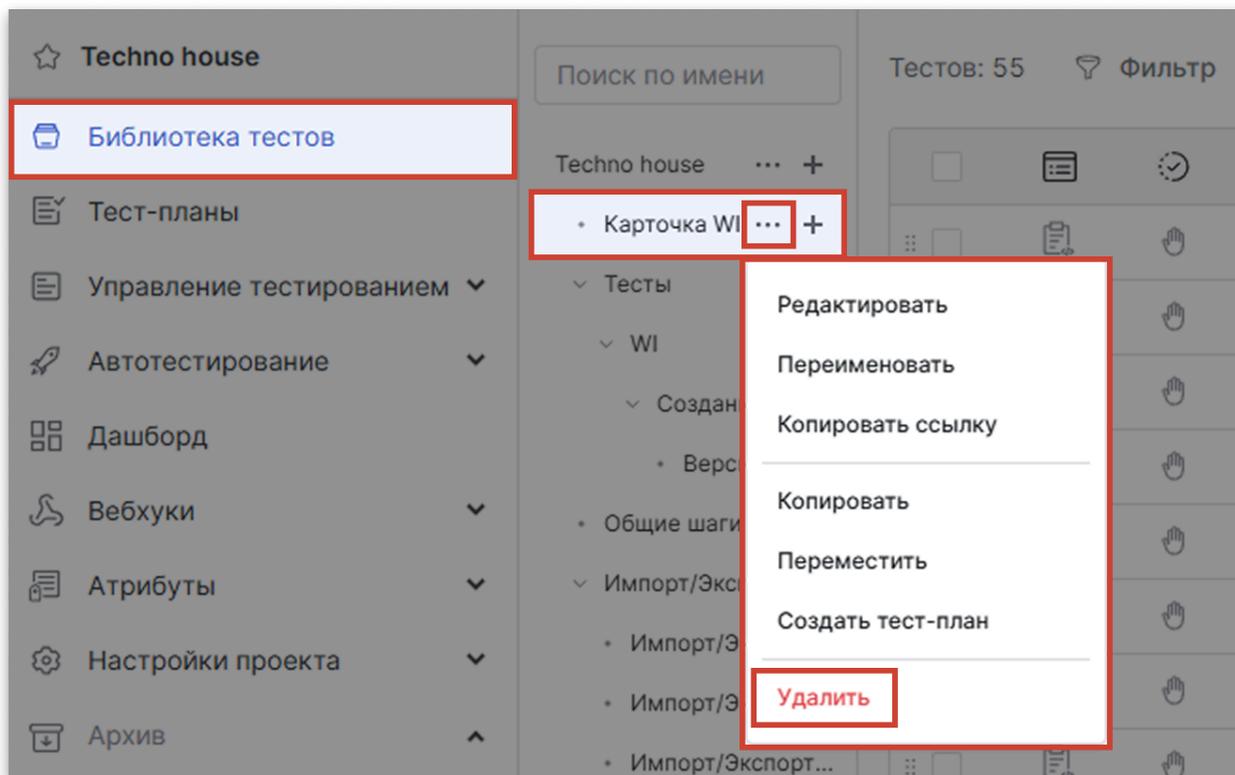
## Удаление секций

**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**

1. Откройте проект. При открытии автоматически отображается вкладка **Библиотека тестов**.
2. В панели библиотеки тестов справа от выбранной секции откройте меню действий и выберите **Удалить**.



3. Подтвердите действие в диалоговом окне.

# Работа с параметрами и конфигурациями

Параметр — это техническая характеристика окружения, на котором проводится тестирование ПО (например операционная система определенной версии, браузер, и т. д.). Они содержат значения и помогают избежать дублирования **тест-кейсов**.

## Способ использования параметров

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 платформы, на которой он запущен. Вы можете создавать пользовательские конфигурации.

## 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, которые будут лежать в основе конфигураций
- Создать конфигурацию, используя эти параметры

## Настройте уведомления об изменениях конфигураций

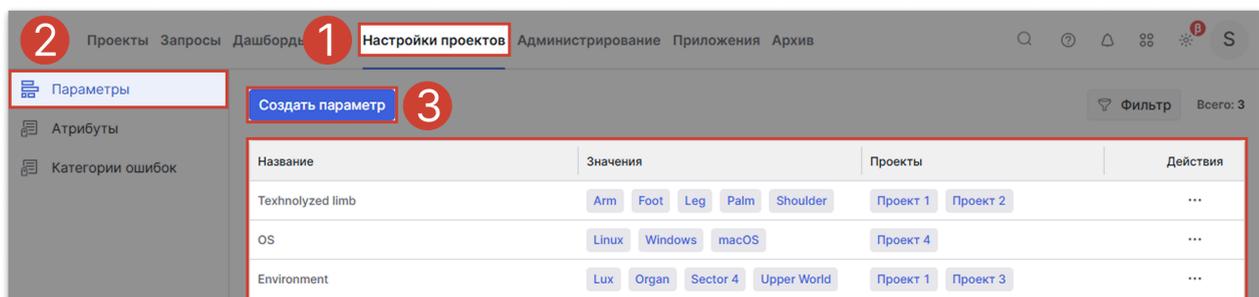
С Test IT вы можете получать в мессенджер уведомления о создании, редактировании или архивировании конфигурации. Для этого нужно **настроить вебхук**, указав необходимые переменные.

# Создание параметров

Вы можете задавать параметры, которые требуется проверить в тест-кейсах. Например, если ваш тестируемый продукт поддерживает несколько операционных систем, вы можете создать параметр `ОС`. Созданные параметры могут быть использованы в различных проектах и тест-кейсах.

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 [Работа с параметрами тестов](#).

# Создание конфигураций

1. Откройте проект.
2. Перейдите в раздел **Управление тестированием** → **Конфигурации**.
3. Нажмите **Создать конфигурацию**.

Weather Experience

Библиотека тестов

Тест-планы

Управление тестированием

Конфигурации

Атрибуты тест-плана

Конфигурации: 4

Фильтр

+ Создать конфигурацию

ID	Название	Описание	По умолчанию
6	High-performance hardw...	Modern & powerful hardware	Да
5	Low-performance hardware	Outdated & weak hardware	Нет
4	Win + Chrome	Operating system & web bro...	Да
2	AnyConfig	Default Any Configuration	Нет

Откроется окно создания конфигурации.

### Новая конфигурация

\* Название конфигурации

Win 7 / Edge

Описание

Некоторые элементы не отображаются в браузере Edge. Протестировать на Win 7.

76/255

Назначить по умолчанию для всех тест-поинтов

Параметры конфигурации

Выберите ключ и значение для каждого созданного параметра.

Ключ	Значение
ОС	Windows 7
Browser	MS Edge

+ Добавить параметр

Отмена

Сохранить

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 конфигурацию параметры 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 и значения. Например, если помимо параметра `ОС` вы создали параметр `Browser` с 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.

б. Нажмите **Сохранить**.

Созданная конфигурация отображается в таблице в разделе **Конфигурации**.

## Действия с конфигурациями

- Для совершения действия воспользуйтесь меню в колонке **Действия**.

ID	Название	Описание	По умолчанию	Дата создания	Действия
1519	Win + Chrome	Windows + Google Chrome	Нет	11.08.2025	...
960	Any	Default Any Configuration	Да	13.09.2	Редактировать
959	Eva-01	Evangelion 01: железо + ПО	Нет	13.09.2	Скопировать GUID
958	Nerv	Оборудование + ПО Nerv	Нет	13.09.2	Назначить по умолчанию Дублировать
					Архивировать

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**.

The screenshot shows the 'Mass Production Evangelions' project interface. On the left is a sidebar with a menu where 'Архив' (Archives) is selected, and 'Конфигурации' (Configurations) is highlighted. The main area displays a table of configurations with 5 total configurations, 3 of which are selected. A red box highlights the '3 Выбрано' (3 Selected) status bar, and another red box highlights the 'Восстановить' (Restore) icon in the top right of the table. A third red box highlights the 'ID' column header.

ID	Название ↑	Описание
<input type="checkbox"/>	960 Any	Default Any Configuration
<input checked="" type="checkbox"/>	1705 Eva-00	Evangelion 00: железо + ПО
<input checked="" type="checkbox"/>	959 Eva-01	Evangelion 01: железо + ПО
<input checked="" type="checkbox"/>	1704 Eva-02	Evangelion 02: железо + ПО
<input type="checkbox"/>	958 Nerv	Оборудование + ПО Nerv

4. **Опционально:** Чтобы восстановить отдельную конфигурацию, откройте выпадающее меню в колонке **Действия** и выберите **Восстановить**.

The screenshot shows a dropdown menu titled 'Действия' (Actions). It contains two options: 'Восстановить' (Restore) and 'Удалить' (Delete). The 'Восстановить' option is highlighted with a red box, and the dropdown menu itself is also outlined in red.

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**.

The screenshot shows the 'Mass Production Evangelions' project interface. On the left is a sidebar with a menu. The 'Archives' section is expanded, and 'Configurations' is highlighted. The main area displays a table of configurations with 5 items. Three items are selected, and a '3 Selected' status bar is visible at the top right. A trash icon is highlighted in the top right corner of the table, and another trash icon is highlighted in the 'Actions' column of the first row.

ID	Название ↑	Описание
<input type="checkbox"/>	Any	Default Any Configuration
<input checked="" type="checkbox"/>	Eva-00	Evangelion 00: железо + ПО
<input checked="" type="checkbox"/>	Eva-01	Evangelion 01: железо + ПО
<input checked="" type="checkbox"/>	Eva-02	Evangelion 02: железо + ПО
<input type="checkbox"/>	Nerv	Оборудование + ПО Nerv

4. **Опционально:** Чтобы удалить отдельную конфигурацию, откройте выпадающее меню в колонке **Действия** и выберите **Удалить**.

The screenshot shows a dropdown menu titled 'Действия' (Actions). The menu is open, showing two options: 'Восстановить' (Restore) and 'Удалить' (Delete). The 'Удалить' option is highlighted with a red box.

5. Подтвердите действие в диалоговом окне.

# Создание тест-плана

## Назначьте конфигурацию

Перед созданием тест-плана убедитесь, что в проекте **назначена конфигурация** по умолчанию. Вы также можете назначить нужные вам конфигурации.

## Интерфейс тест-плана обновлен

В новом интерфейсе тест-планов (в Test IT Lite & Standard и Test IT Enterprise 5.4, а также в более поздних версиях) объединены вкладки **Планирование** и **Выполнение**. Описание тест-плана расположено в правой навигационной панели на странице тест-плана.

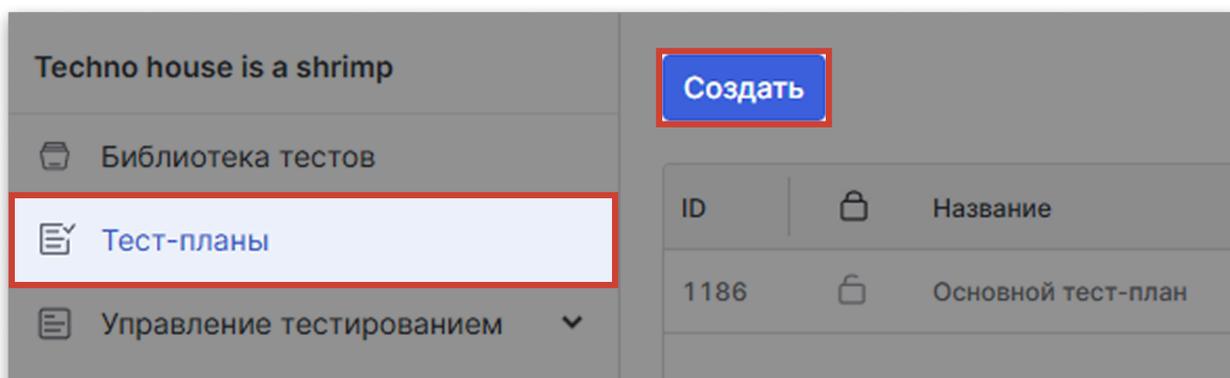
Вы можете создать тест-план:

- 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**

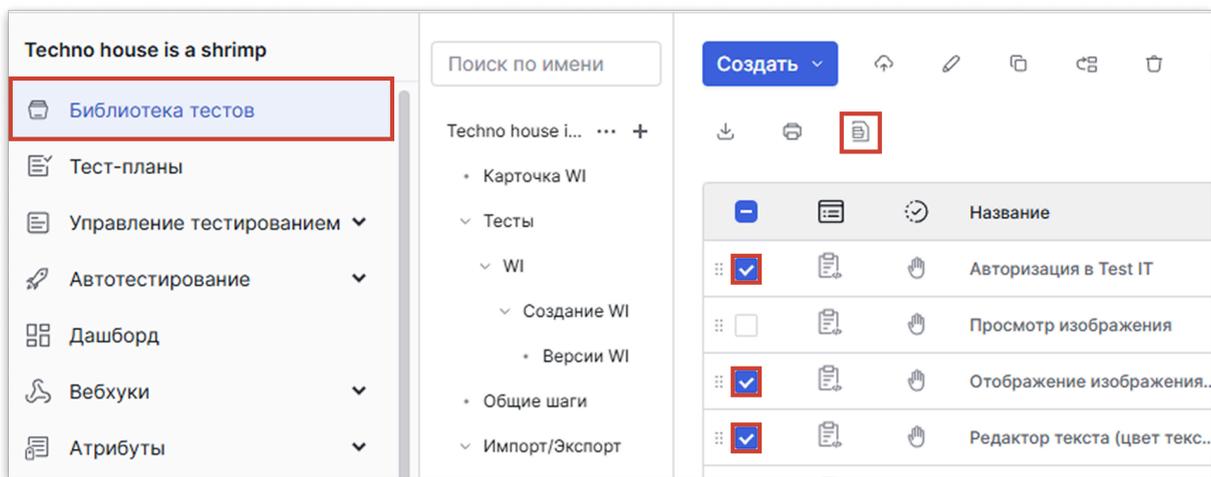
При создании тест-плана наборы тестов создаются автоматически.

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:

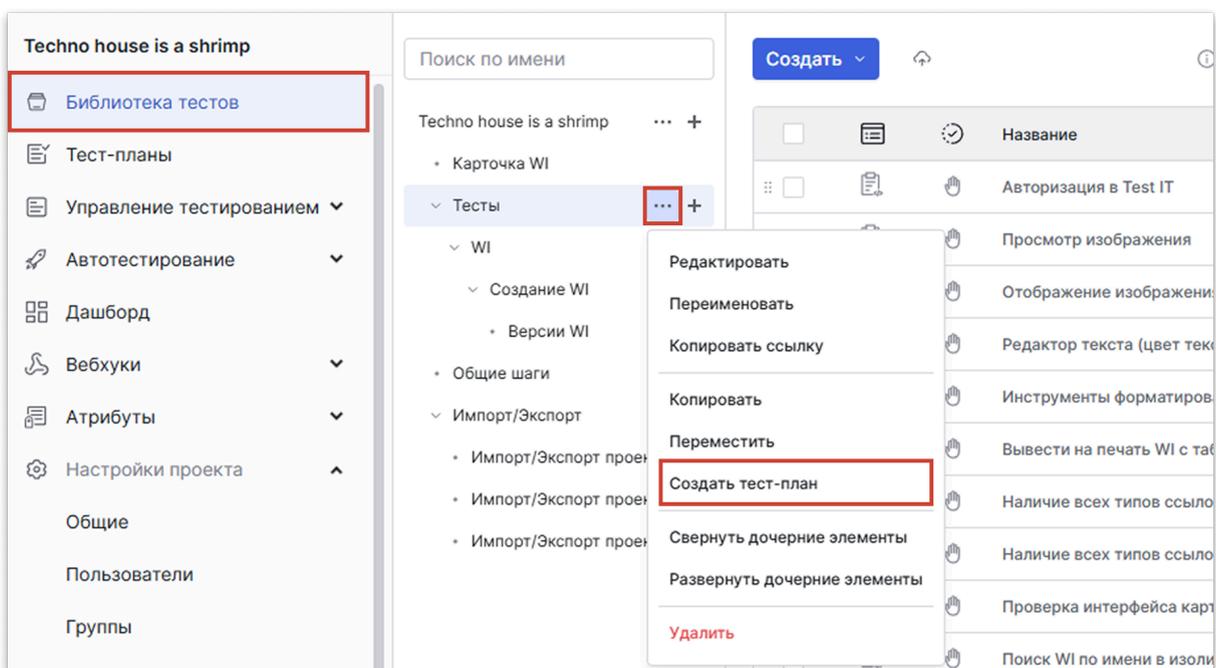
### Из списка тестов

- 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**.

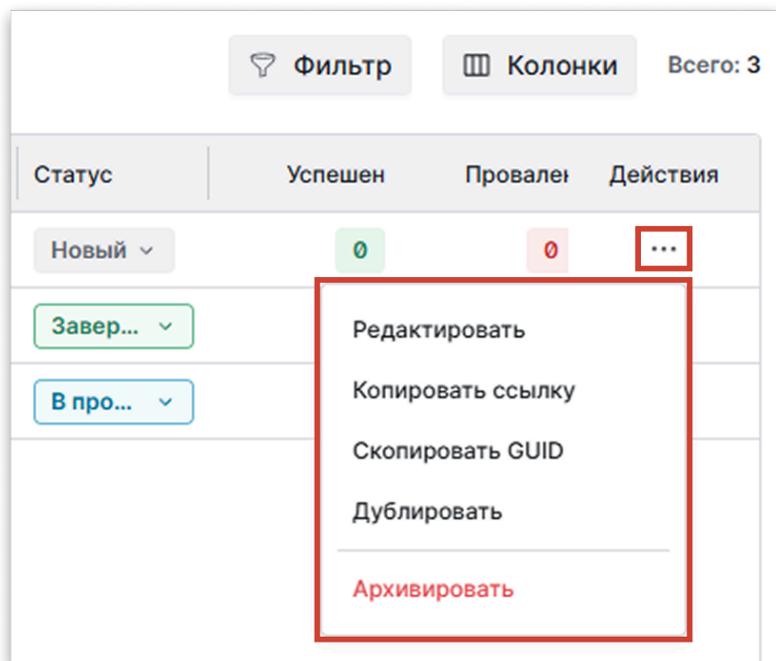
## Действия с тест-планами и просмотр тест-планов

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 от 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.

# 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.

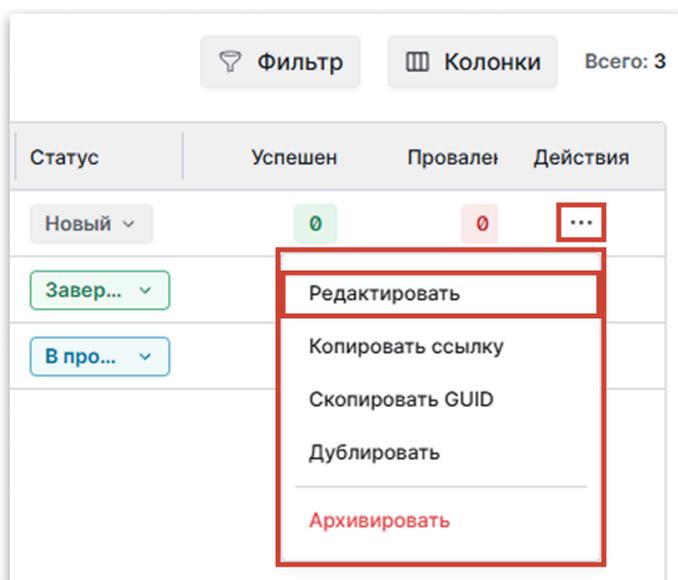
## Интерфейс тест-плана обновлен

В новом интерфейсе тест-планов (в Test IT Lite & Standard и Test IT Enterprise 5.4, а также в более поздних версиях) объединены вкладки **Планирование** и **Выполнение**. Описание тест-плана расположено в правой навигационной панели на странице тест-плана.

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**.

**Редактировать** Сохранить Отмена

\* **Название**  
Удаление пользователя из системы

Продукт: RateApp  
Версия: 5.0.0.1  
Даты: 27.05.2025 — 31.05.2025  
Прохождение: Автоматическое

**Описание**

**Информация для входа в систему:**  
**Домен:** myfairally  
**Логин:** TekkenFighter  
**Пароль:** Mata\_ne\_Subeta\_Senpai!\_28/03/2025

Теги: testit × new × 2 / 10

## 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: 16.10.2025, 18:35:17

# Блокировка тест-плана и запрос на его разблокировку

## ₽ Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тариф **Lite** она не входит.

Подробнее: [о тарифах Test IT](#) .

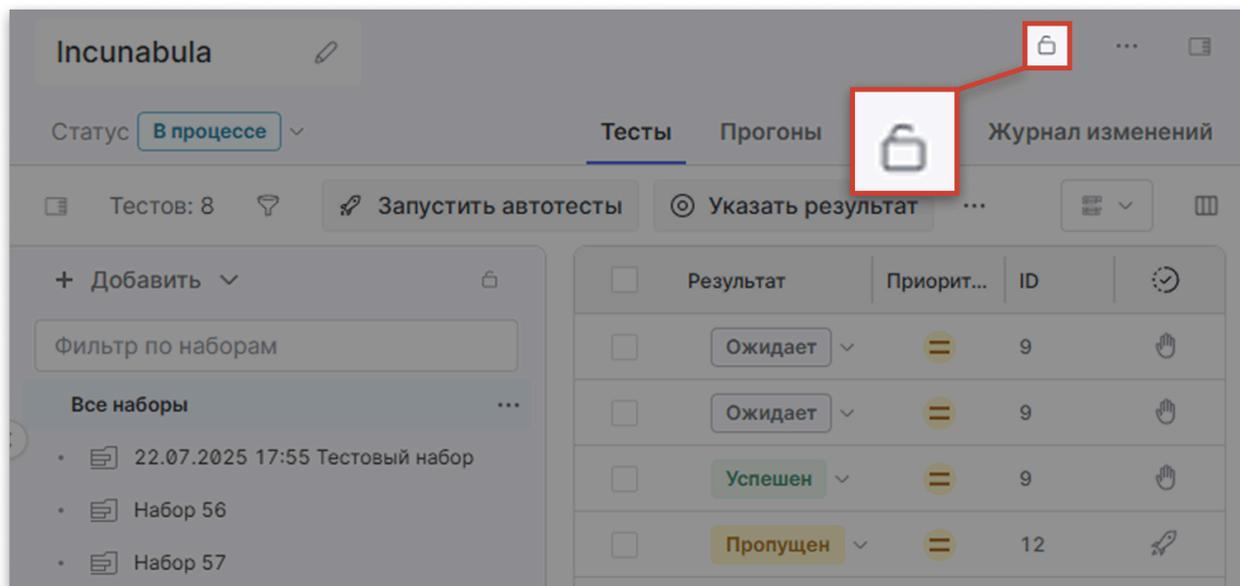
Test IT позволяет блокировать тест-планы, чтобы фиксировать версии тестов и не допускать изменений в тестовой документации. В заблокированном тест-плане невозможно создавать новые тестовые наборы, а также редактировать уже созданные. При блокировке тест-плана все пользователи, имеющие к нему доступ, 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.



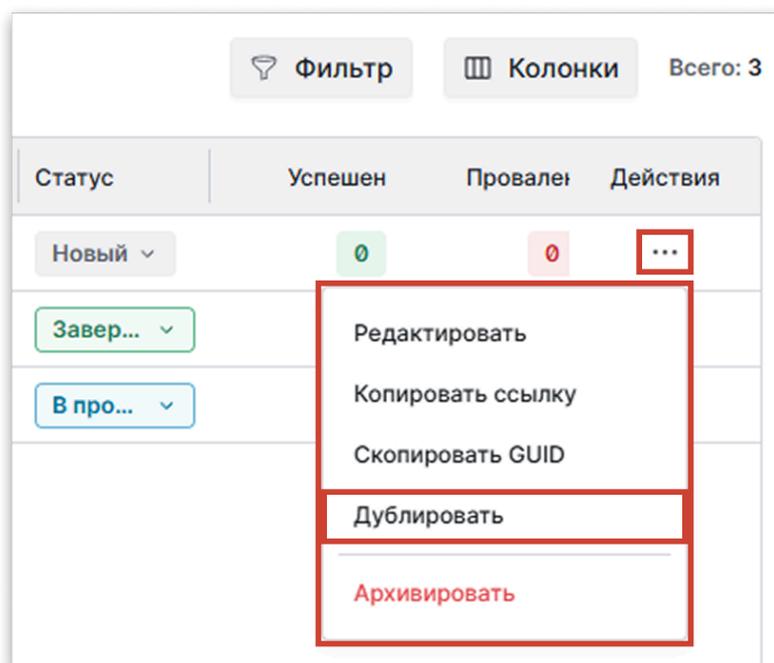
После отправления запроса пользователю, заблокировавшему тест-план, отправляется **уведомление** с запросом на разблокировку.

Обновлено: 17.10.2025, 13:38:51

# 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**.



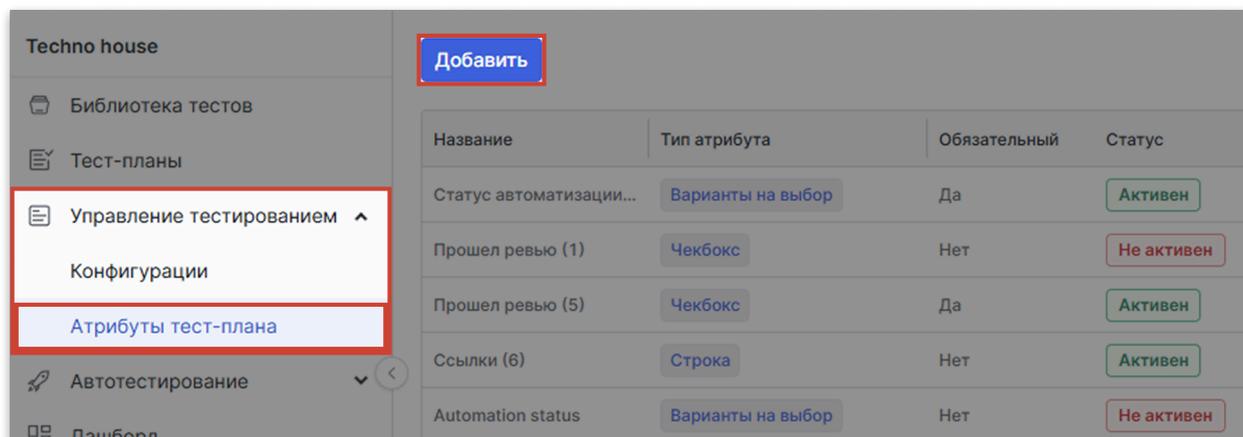
4. Подтвердите действие в диалоговом окне.

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.

# 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](#)

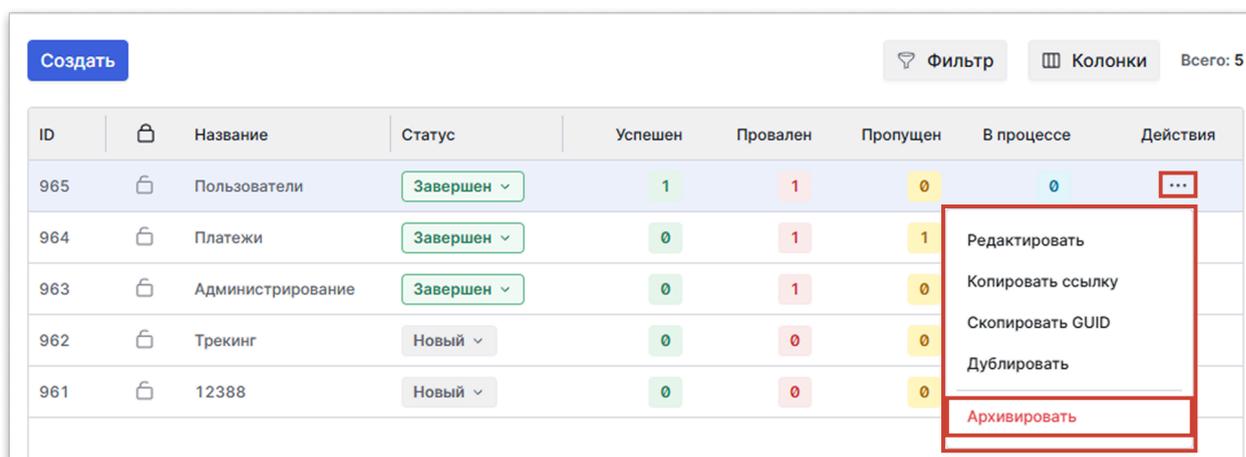
# Архивация, восстановление и удаление тест-плана

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**.



ID	Название	Статус	Успешен	Провален	Пропущен	В процессе	Действия
965	Пользователи	Завершен	1	1	0	0	...
964	Платежи	Завершен	0	1	1		Редактировать Копировать ссылку Скопировать GUID Дублировать Архивировать
963	Администрирование	Завершен	0	1	0		
962	Трекинг	Новый	0	0	0		
961	12388	Новый	0	0	0		

4. Подтвердите действие в диалоговом окне.

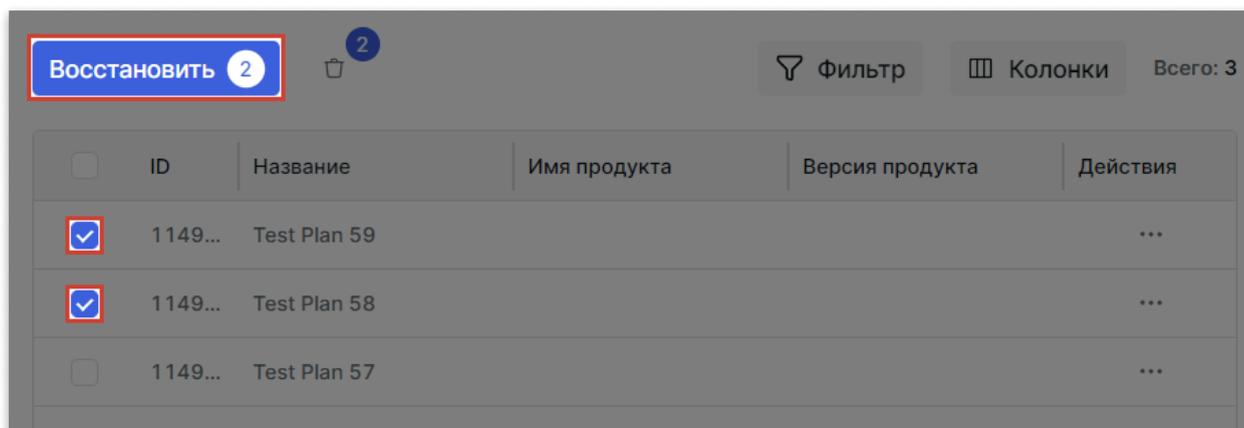
## Восстановление и удаление тест-планов

Вы можете восстанавливать или удалять архивные тест-планы. Тест-планы, удаленные из архива, не могут быть восстановлены.

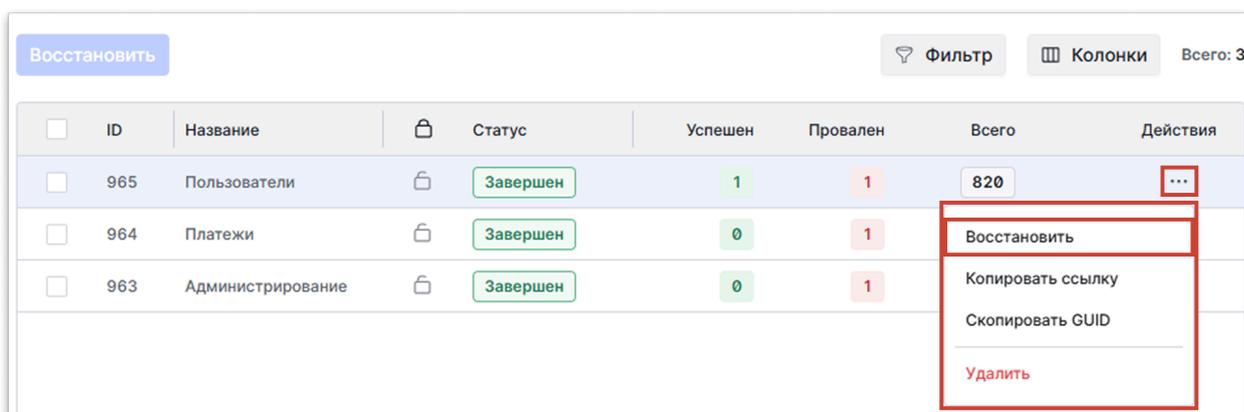
Чтобы восстановить тест-планы:

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**.

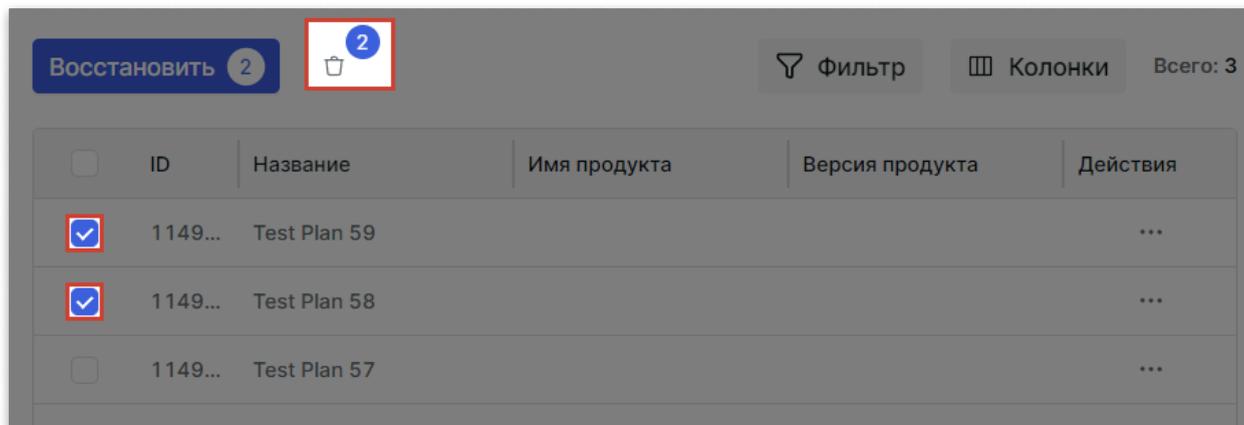


5. Подтвердите действие в диалоговом окне.

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**.

Восстановить

Фильтр Колонки Всего: 3

<input type="checkbox"/>	ID	Название		Статус	Успешен	Провален	Всего	Действия
<input type="checkbox"/>	965	Пользователи		Завершен	1	1	820	...
<input type="checkbox"/>	964	Платежи		Завершен	0	1		
<input type="checkbox"/>	963	Администрирование		Завершен	0	1		

Восстановить

Копировать ссылку

Скопировать GUID

Удалить

5. Подтвердите действие в диалоговом окне.

Обновлено: 11.11.2025, 21:27:49

# Формирование тестового набора

## 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

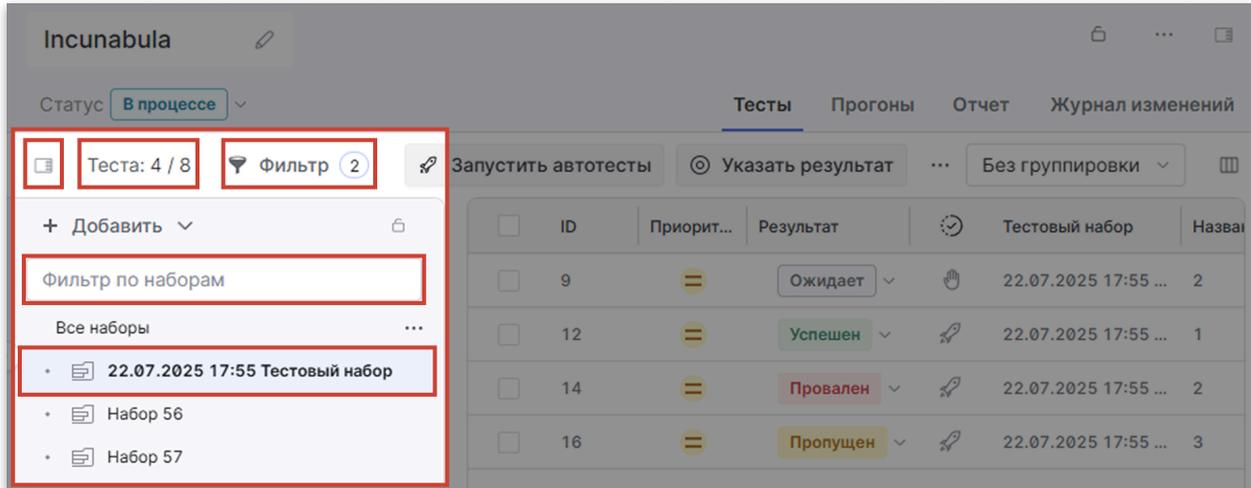
---

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

### ₽ Опция Standard и Enterprise: динамические наборы

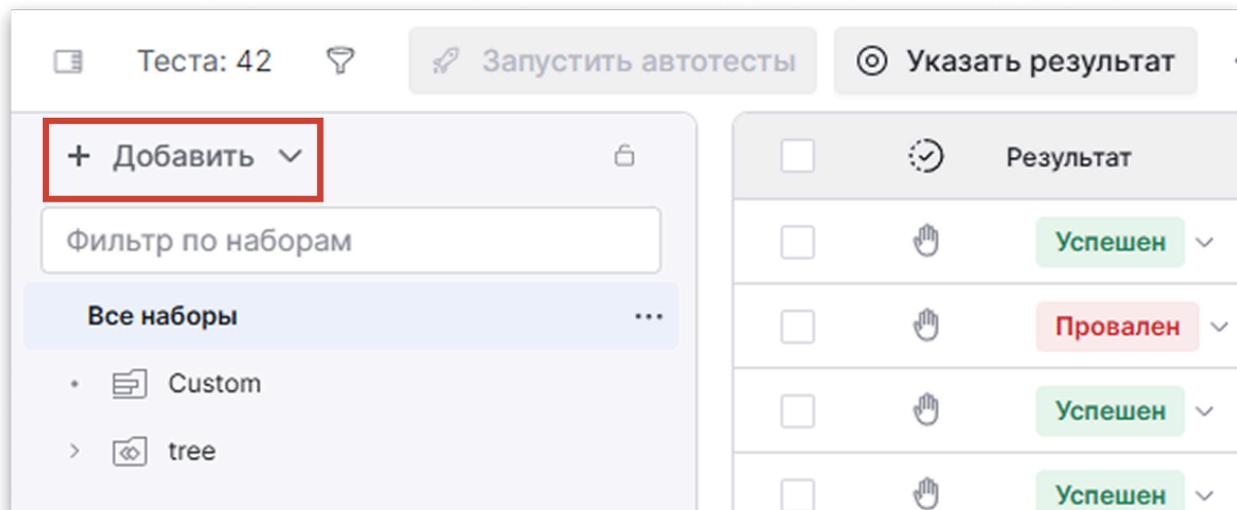
В тарифе **Lite** динамические тестовые наборы недоступны. Если вы создали динамический набор в тарифе **Standard**, а затем перешли на тариф **Lite**, автообновление будет недоступно.

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.

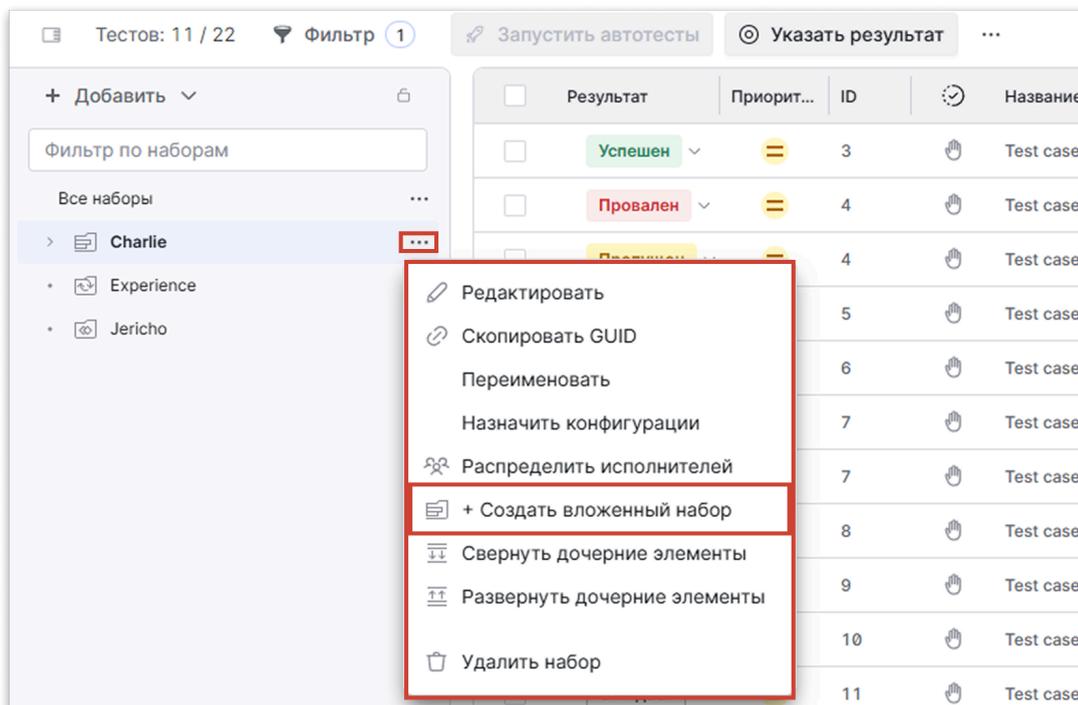
7. Нажмите **Сохранить**.

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**.

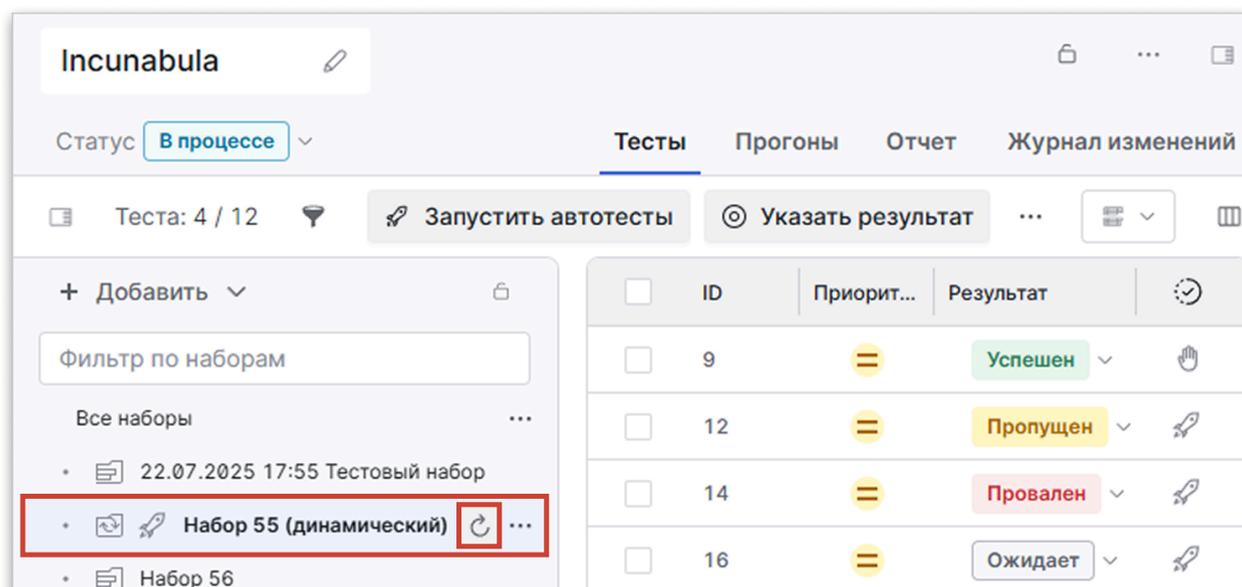
Вложенный тестовый набор появится в панели тестовых наборов.

## 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 динамический 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

---

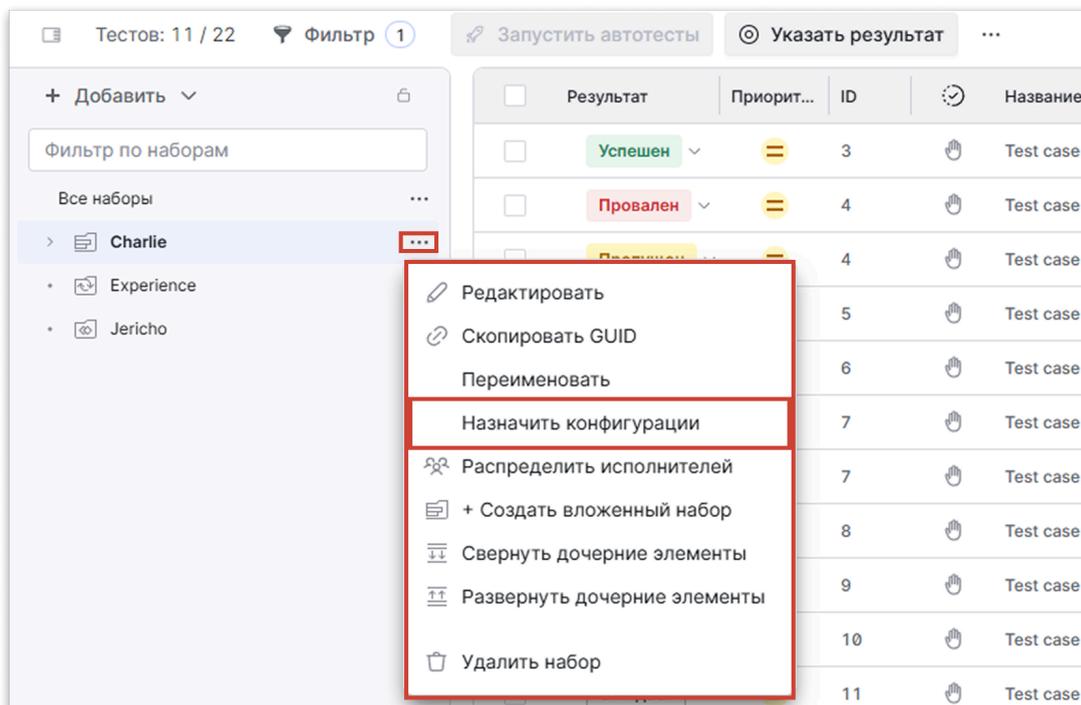
Вы можете добавлять тесты в созданный тестовый набор. Для этого:

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 от 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.

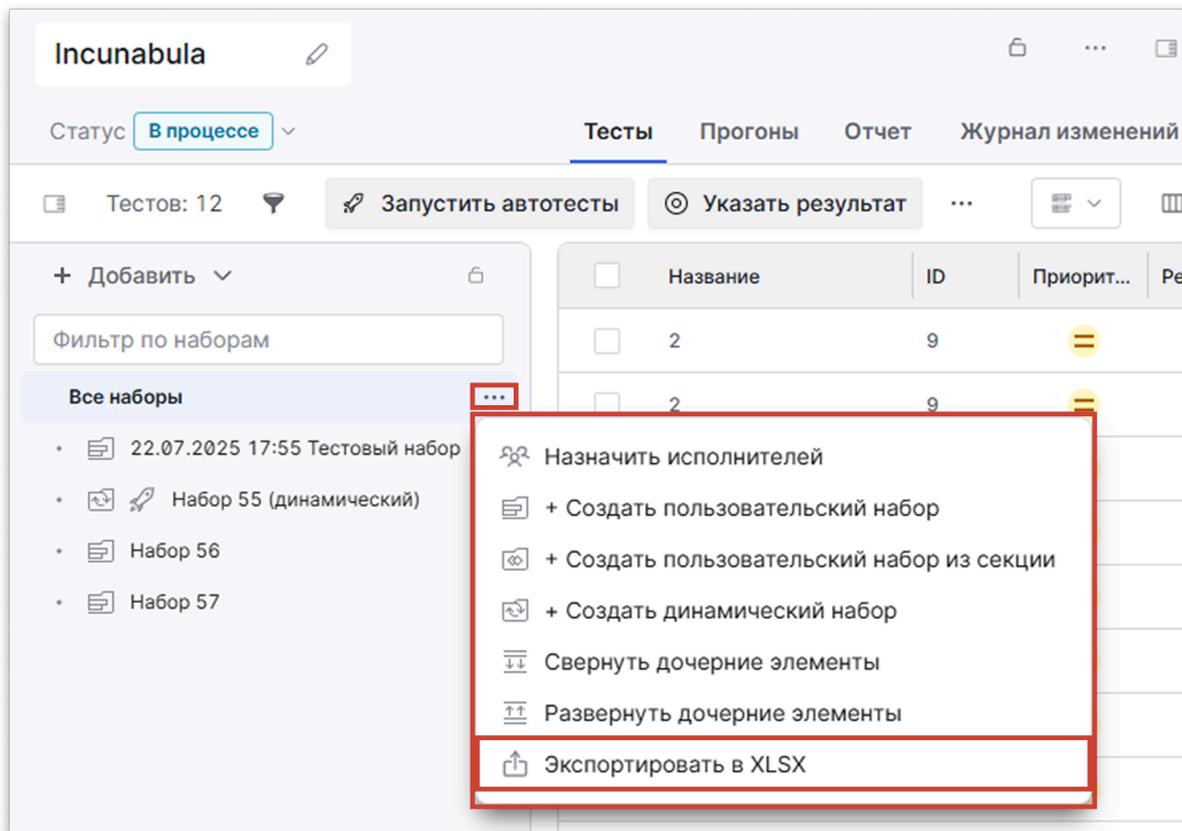
## Экспорт тестовых наборов

### 🔑 Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тарифе **Lite** она не входит. 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. Слева от списка **Все наборы** нажмите **Действия**, затем из выпадающего меню выберите **Экспортировать в XLSX**.



Откроется окно **Экспорт в XLSX**.

5. 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**.

# 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.

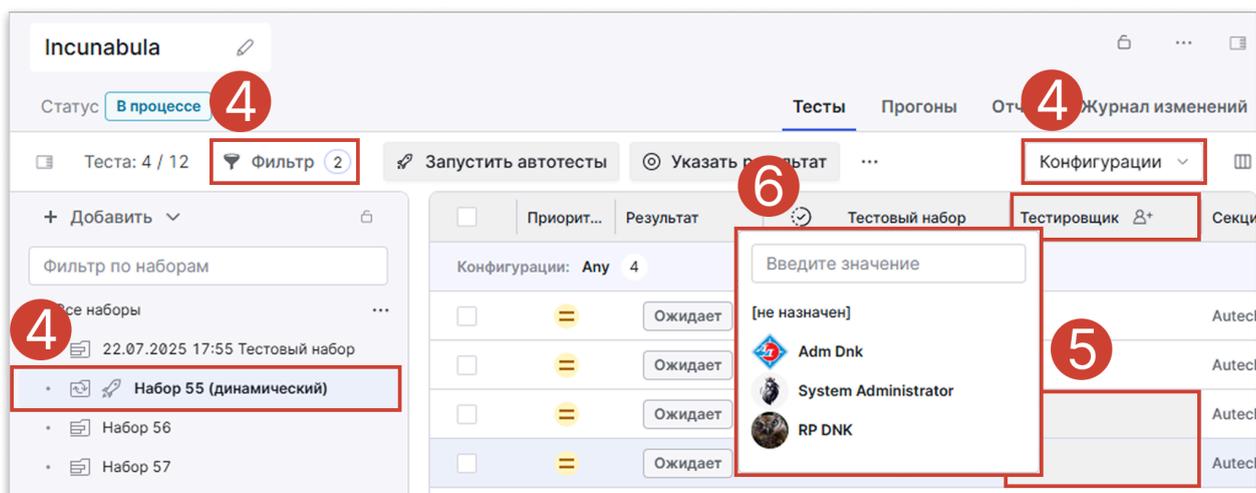
1. Откройте проект.
2. Перейдите в раздел **Тест-планы**.
3. Откройте тест-план.
4. **Опционально:** Отсортируйте тесты так, чтобы назначать тестировщика было удобнее. Для этого:
  - Выберите нужный тестовый набор.
  - Воспользуйтесь фильтром.
  - С помощью меню группировки упорядочьте тесты по нужному 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

### ₽ Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тарифе **Lite** она не входит.

Подробнее: [о тарифах Test IT](#)

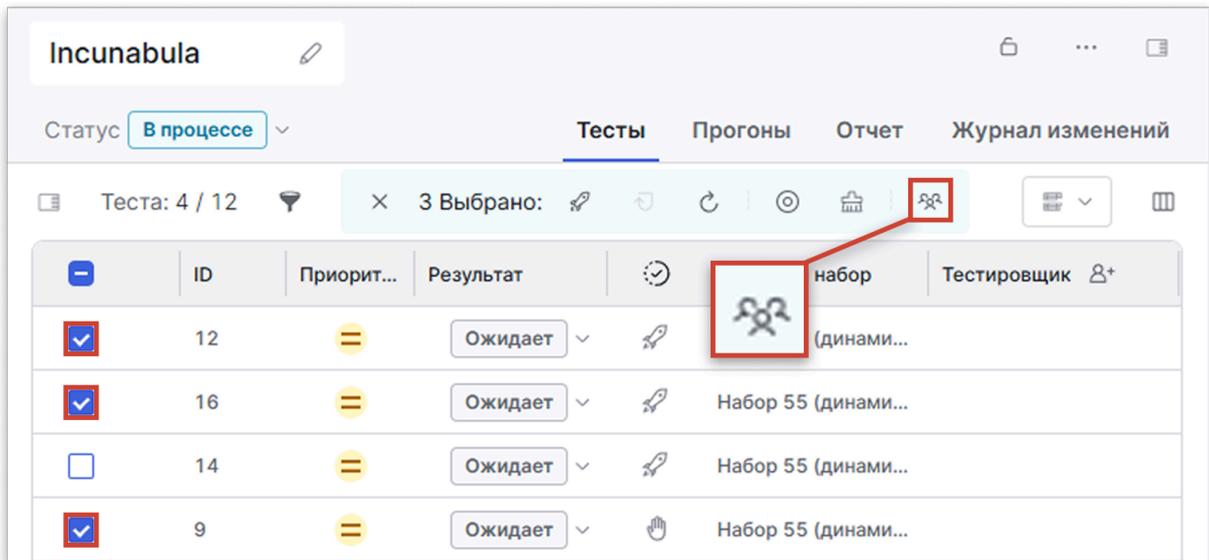
1. В проекте откройте тест-план.

Откроется список тест-поинтов.

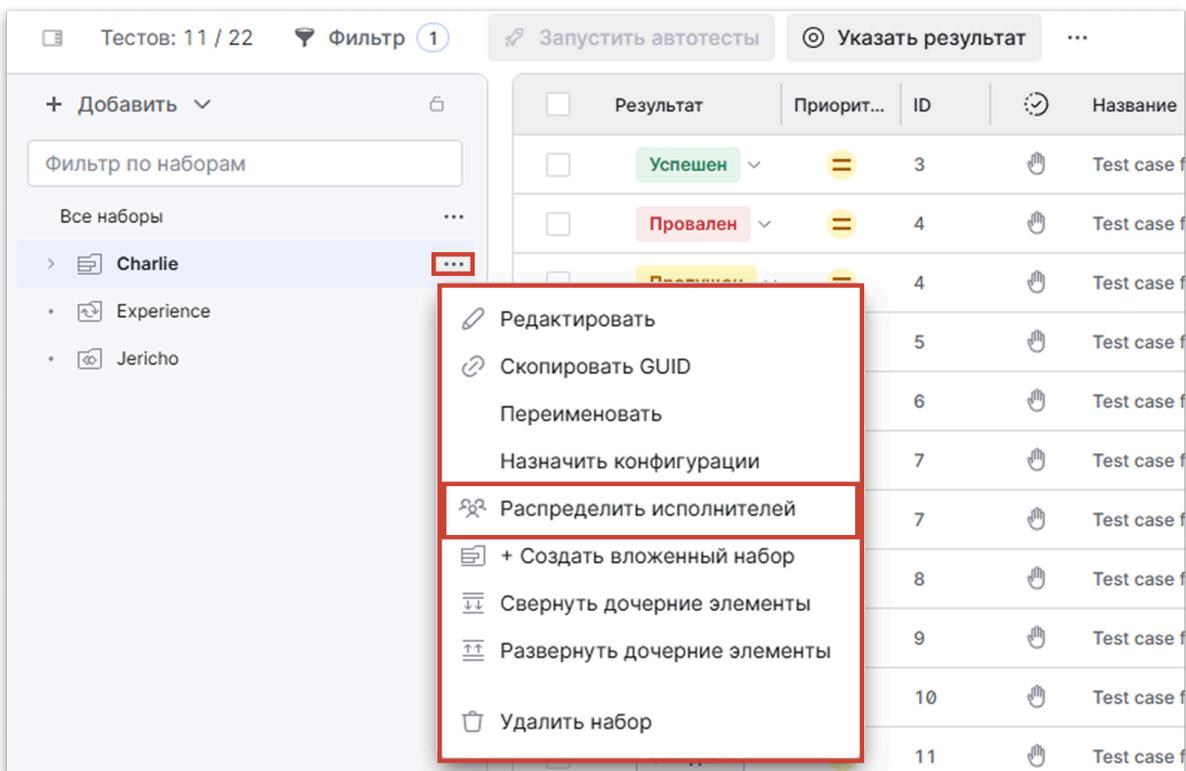
2. Выберите тесты, на которые хотите назначить исполнителей одним из способов:

- 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**.

Тест-поинты распределяются в равной пропорции между пользователями. При автоматическом назначении исполнителей учитываются два параметра: количество тест-поинтов и время их выполнения, указанное в тест-кейсах. 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

тестирующего Б — 1 тест на 10 минут. Если тестирующих 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.

Обновлено: 17.10.2025, 13:38:51

# 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

---

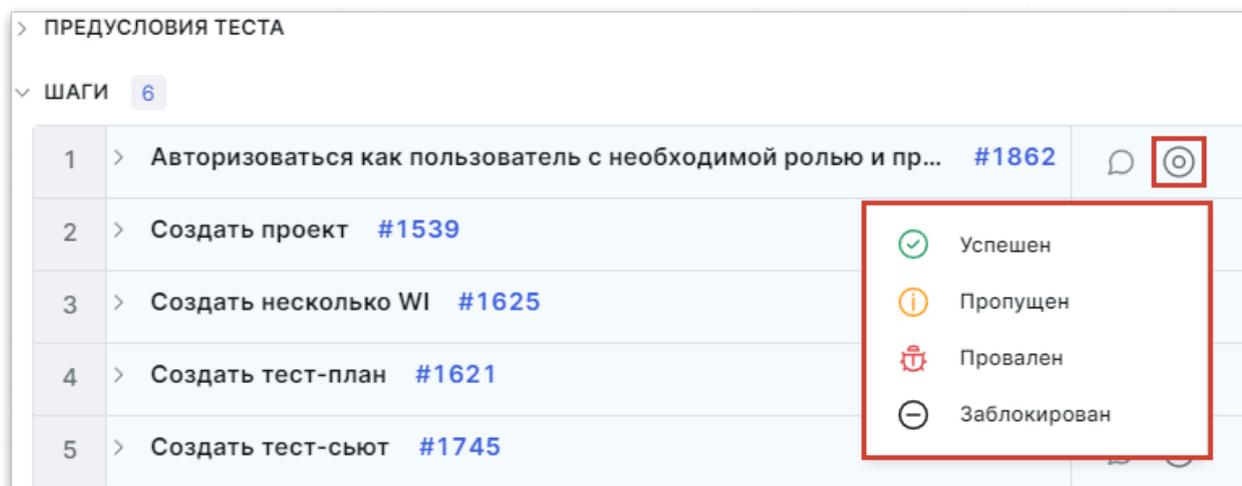
Вы можете найти информацию по запуску автотестов из тест-планов в соответствующем разделе.

## 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

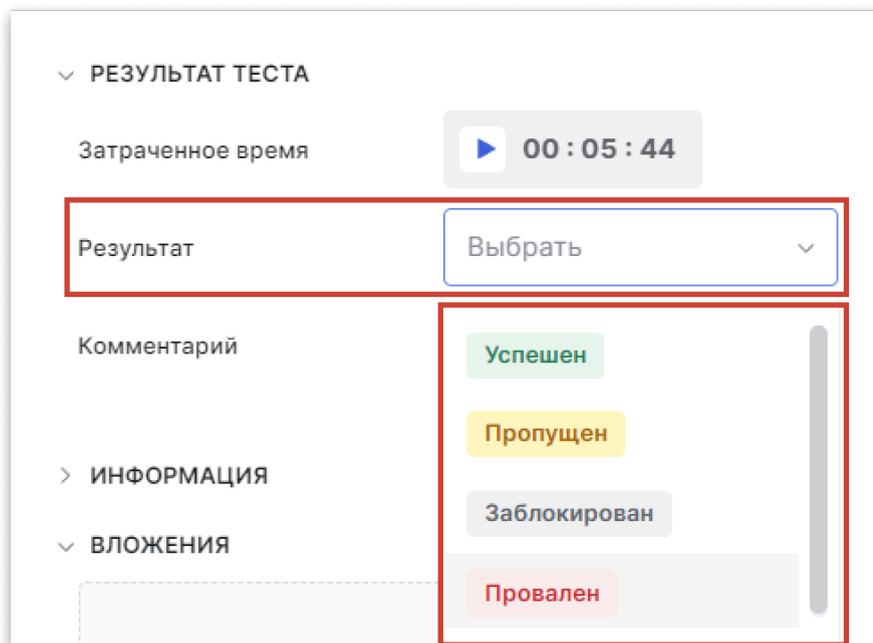
Вы можете указать результат одного теста:

- 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*.



Скриншот интерфейса для отметки результата теста. В верхней части отображается таймер 'Затраченное время' со значением 00:05:44. Ниже находится поле 'Результат' с выпадающим меню 'Выбрать'. Меню открыто, и в нем перечислены варианты: 'Успешен' (зеленый), 'Пропущен' (желтый), 'Заблокирован' (серый) и 'Провален' (красный). Также видны элементы 'Комментарий', 'ИНФОРМАЦИЯ' и 'ВЛОЖЕНИЯ'.

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

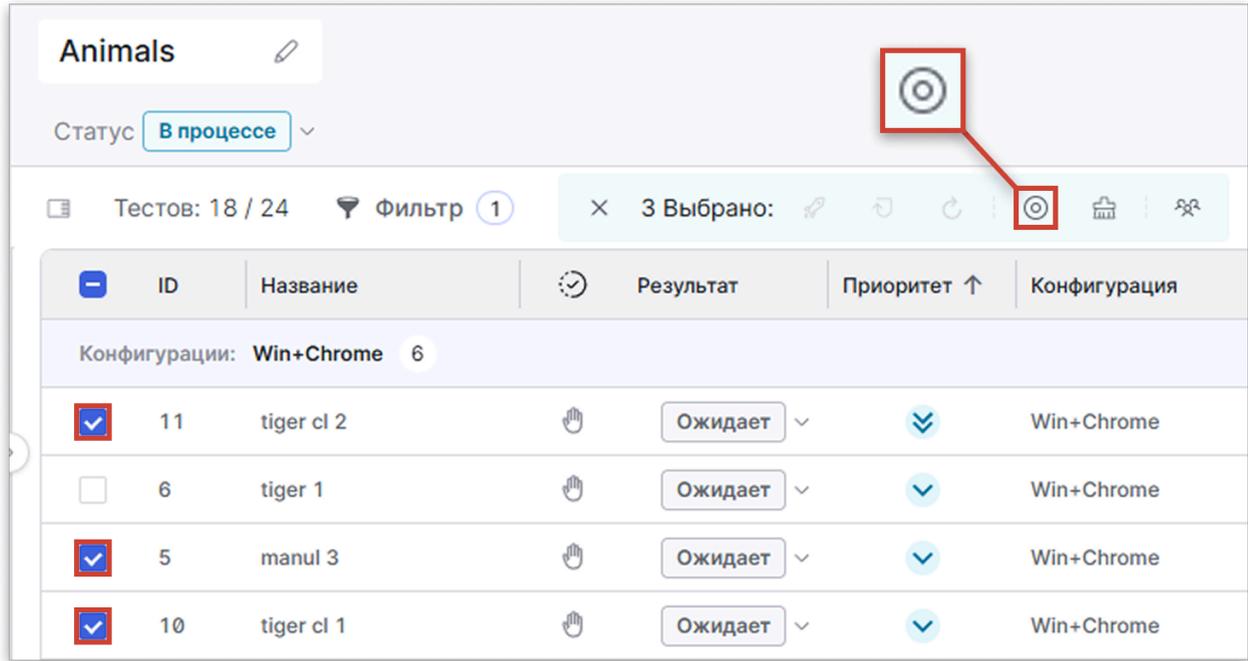
1. В проекте откройте тест-план.
2. В колонке **Результат** выберите *Успешен, Пропущен, Провален* или *Заблокирован*.

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.

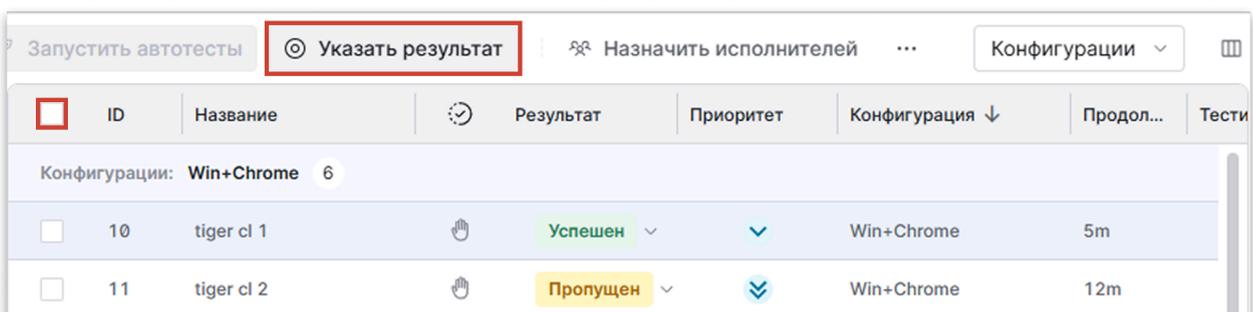


4. Во вкладке **Статус** укажите результат, который хотите указать для выбранных тестов: *Успешен*, *Пропущен*, *Провален* или *Заблокирован*.
5. При необходимости вы можете добавить в результат ссылки, комментарии и вложения в соответствующих вкладках.
6. Нажмите **Сохранить**.

## 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**. :::

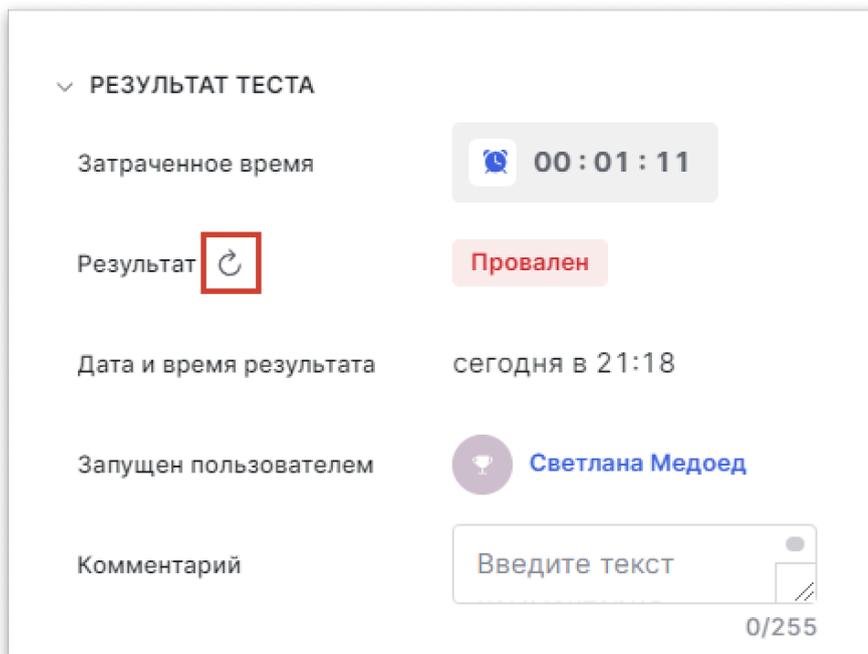
Test IT сохранит результат прохождения отмеченных флажками тестов и обновит их статус в тест-плане в зависимости от присвоенного результата.

## 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.

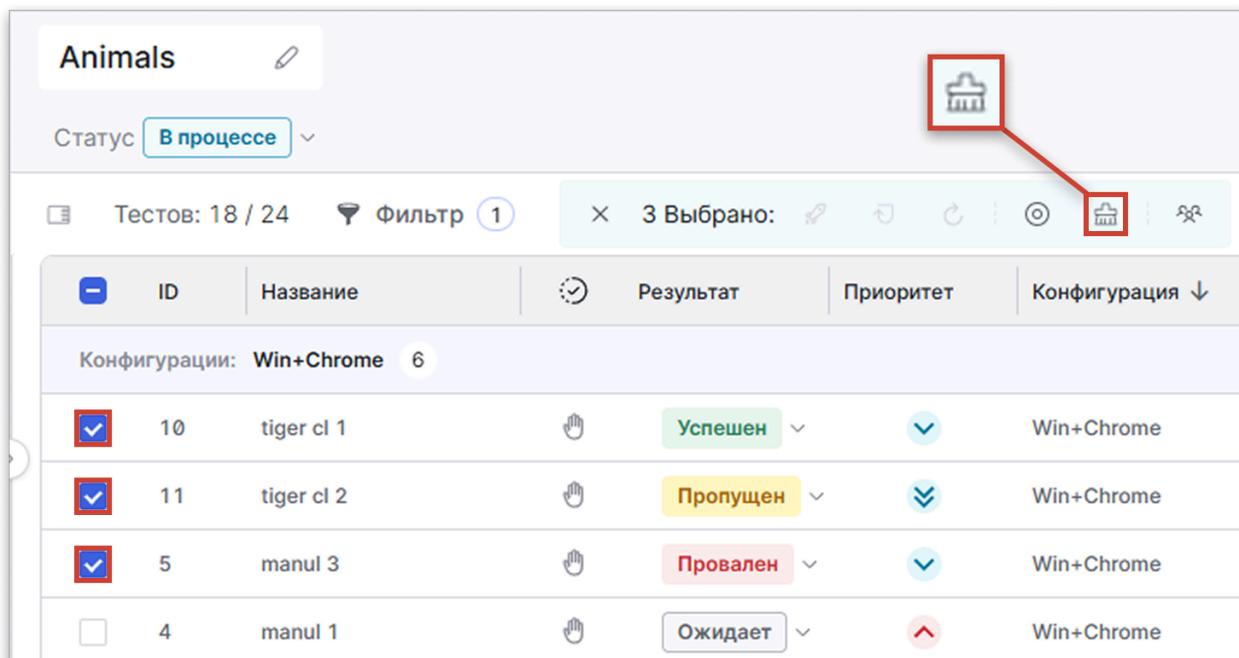
- Для сброса результата конкретного теста откройте его и нажмите на значок сброса.



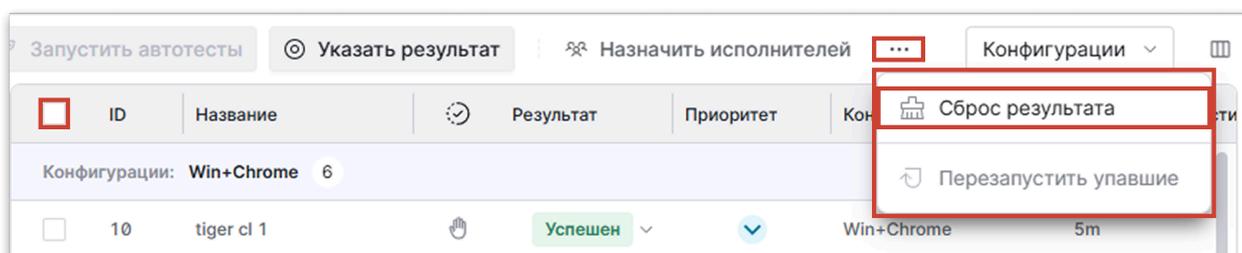
Скриншот интерфейса отображения результата теста. Вкладка **РЕЗУЛЬТАТ ТЕСТА** раскрыта. Показаны следующие данные:

- Затраченное время: 00:01:11
- Результат: **Провален**. Рядом с полем 'Результат' находится значок сброса (круг со стрелкой), выделенный красным квадратом.
- Дата и время результата: сегодня в 21:18
- Запущен пользователем: Светлана Медоед
- Комментарий: Введите текст (0/255)

- 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**.



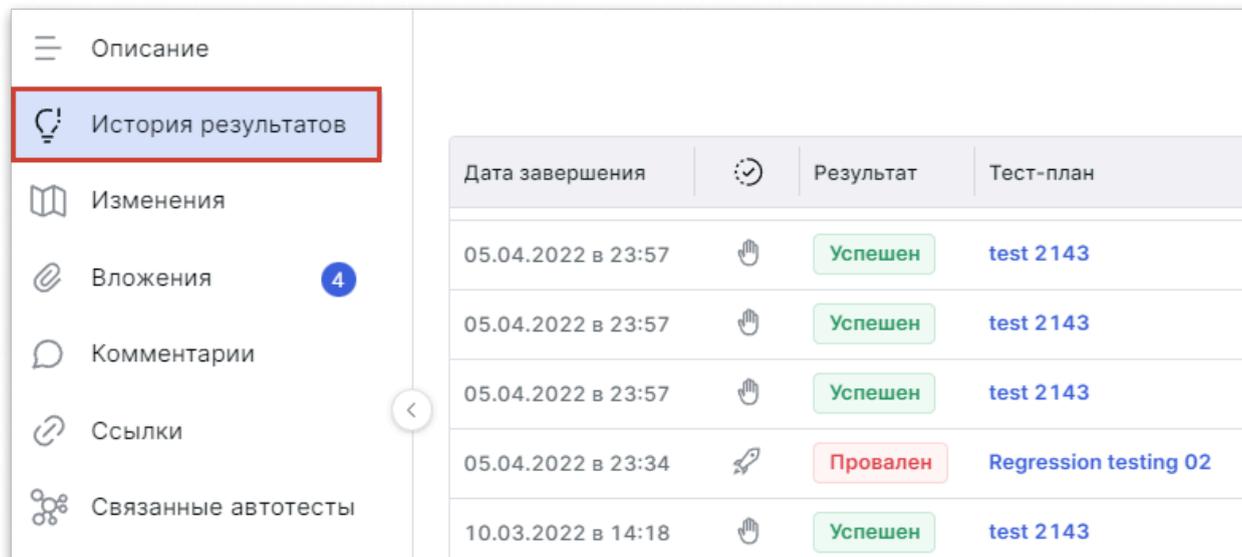
После сброса результатов тестов вы можете пройти их повторно. Test IT сохраняет все результаты, указанные для тест-поинта.

## 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.



Дата завершения		Результат	Тест-план
05.04.2022 в 23:57		Успешен	test 2143
05.04.2022 в 23:57		Успешен	test 2143
05.04.2022 в 23:57		Успешен	test 2143
05.04.2022 в 23:34		Провален	Regression testing 02
10.03.2022 в 14:18		Успешен	test 2143

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: 11.11.2025, 21:27:49

# 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.

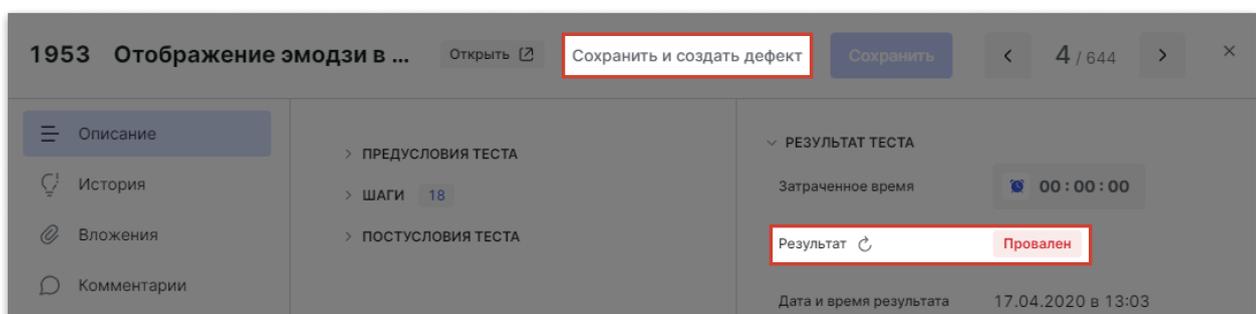
Если при прохождении тест-плана вы обнаружили дефект, можно создать задачу в таск-трекере из интерфейса Test IT. При создании дефекта задача в таск-трекере связывается с тестом, в котором обнаружен дефект, и с результатом прогона теста, на котором был обнаружен дефект. Таким образом, тестировщики могут 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 создайте дефект.

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.

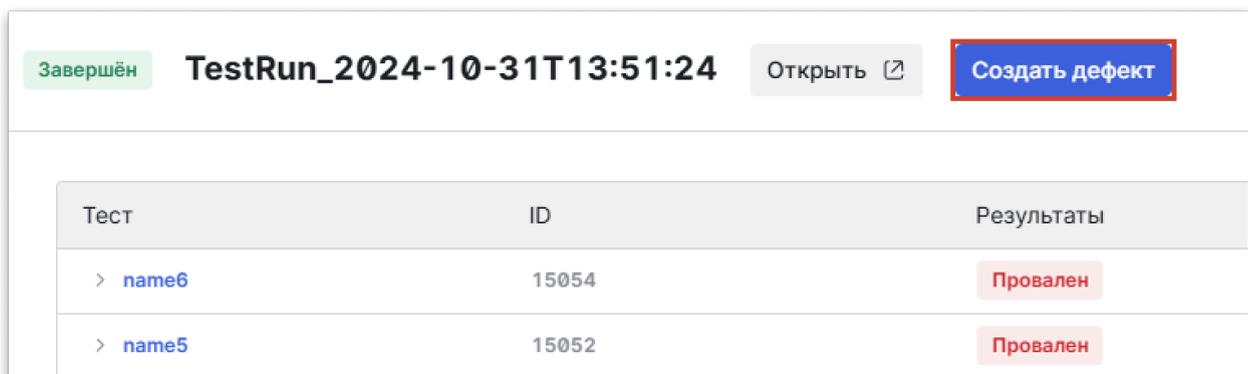
Вы можете отслеживать статус дефекта, его приоритет и исполнителя прямо в Test IT, наведя на него курсор.

## 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 создайте дефект. В системе будет создан дефект для всех автотестов в статусе *Провален*.

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: 01.10.2025, 17:08:09

# Отчет по тест-плану

## 🔒 Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тариф **Lite** она не входит.

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.

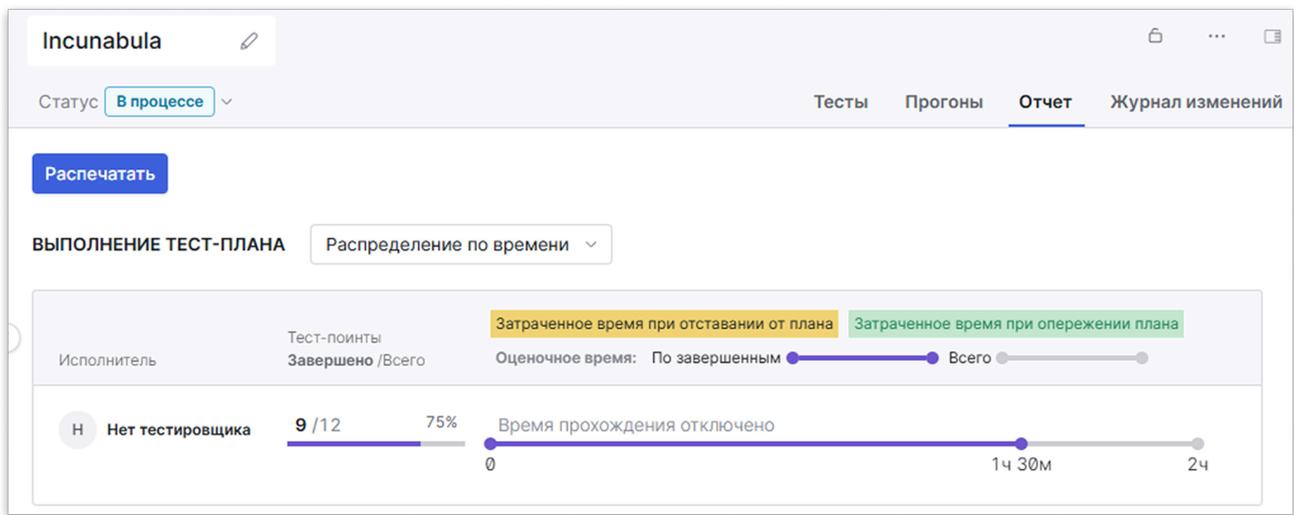
Вы можете:

- 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 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:

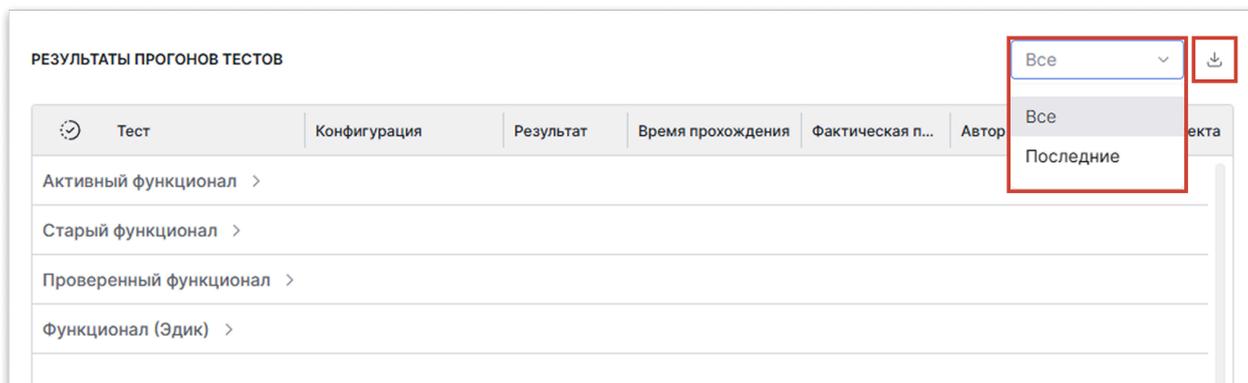
- Чтобы переключать отображение тест-поинтов или время их выполнения на оси Y, выберите **Время** или **Тест-поинты** из выпадающего меню **График сгорания задач по тест-плану**.
- Чтобы увеличить или уменьшить детализацию оси X, используйте шкалу в нижней части диаграммы или колесико мыши.

## 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.



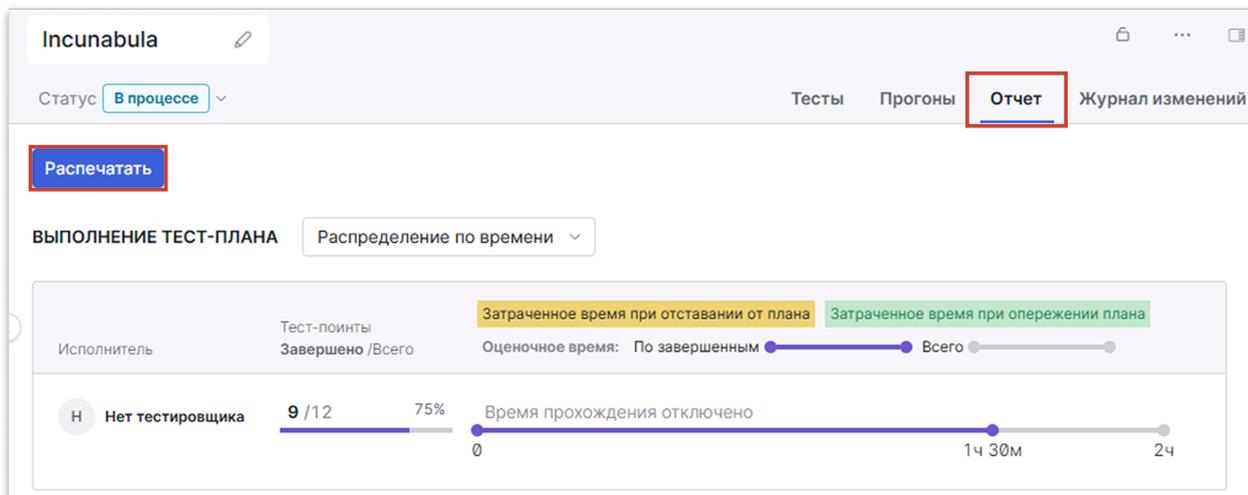
Откроется окно экспорта отчета.

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: 11.11.2025, 21:27:49

# 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.

## 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**
- **Работа с нестабильными автотестами**
- **Связывание автотестов с тест-кейсами**
- **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 ошибок автотестов**
- **Таймлайн**

Обновлено: 23.03.2025, 14:05:34

# Добавление автотестов в Test IT из внешнего репозитория

- 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

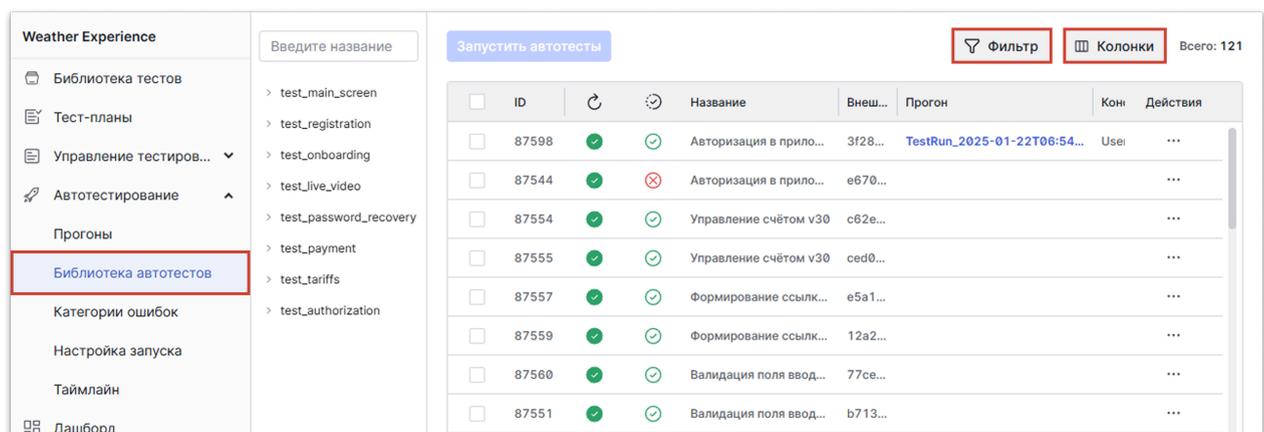
1. Авторизуйтесь в Swagger.
2. В секции **AutoTests** раскройте метод `POST /api/v2/autoTests`.
3. Нажмите **Try it out**. Тело запроса откроется для редактирования.
4. В запросе укажите: external ID, ссылку на автотест, ID проекта, предусловия тестов, постусловия тестов и шаги.
5. Нажмите **Execute**. 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.



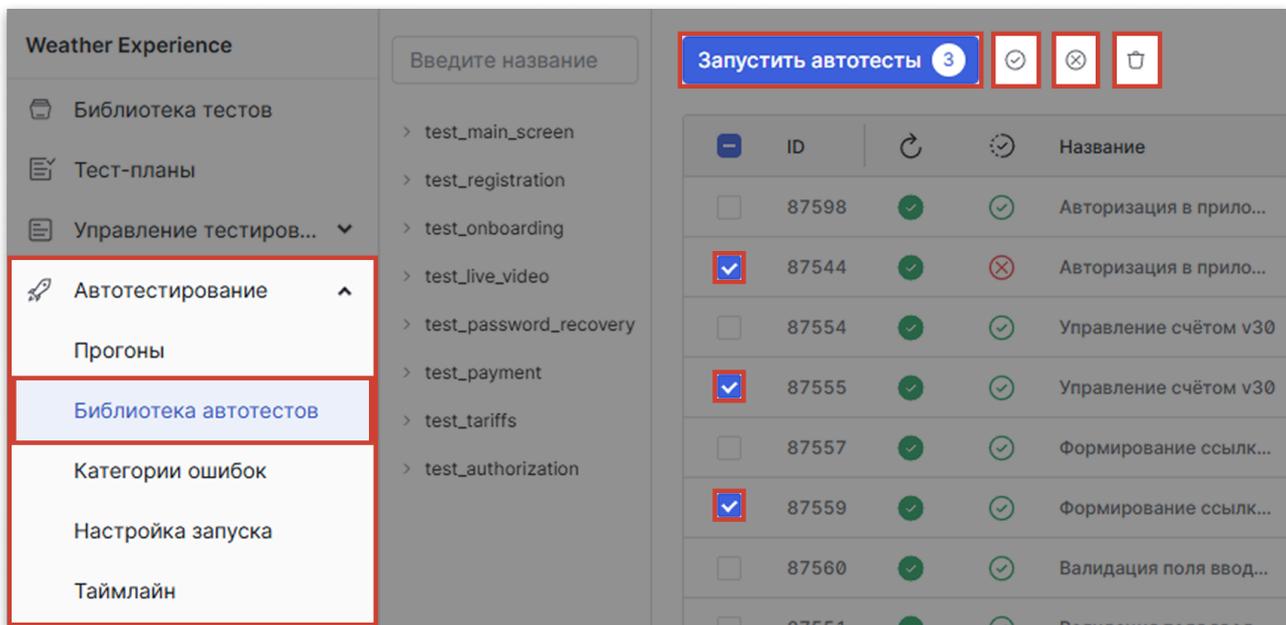
The screenshot shows the 'Библиотека автотестов' (Test Library) interface. On the left is a sidebar with navigation options: Библиотека тестов, Тест-планы, Управление тестиров..., Автотестирование (expanded), Прогоны, Библиотека автотестов (highlighted), Категории ошибок, Настройка запуска, Таймлайн, and Дашборд. The main area has a search bar 'Введите название', a 'Запустить автотесты' button, and controls for 'Фильтр' and 'Колонки'. Below is a table of tests with columns: ID, status icons, Name, External ID, Run ID, User, and Actions.

ID	Статус	Название	Внеш...	Прогон	Кон	Действия
87598	✓	Авторизация в прило...	3f28...	TestRun_2025-01-22T06:54...	User	...
87544	✓	Авторизация в прило...	e670...			...
87554	✓	Управление счётом v30	c62e...			...
87555	✓	Управление счётом v30	ced0...			...
87557	✓	Формирование ссылк...	e5a1...			...
87559	✓	Формирование ссылк...	12a2...			...
87560	✓	Валидация поля ввод...	77ce...			...
87551	✓	Валидация поля ввод...	b713...			...

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.

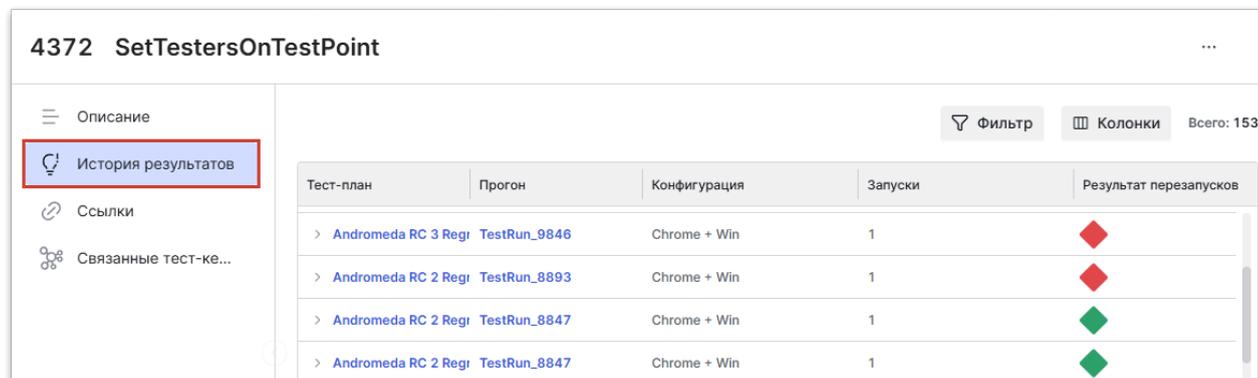
Чтобы просмотреть историю прохождения автотеста:

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.



Тест-план	Прогон	Конфигурация	Запуски	Результат перезапусков
> Andromeda RC 3 Regr TestRun_9846		Chrome + Win	1	❖
> Andromeda RC 2 Regr TestRun_8893		Chrome + Win	1	❖
> Andromeda RC 2 Regr TestRun_8847		Chrome + Win	1	❖
> Andromeda RC 2 Regr TestRun_8847		Chrome + Win	1	❖

5. **Опционально:** Чтобы посмотреть детализированную информацию исторического результата, нажмите на строку результата в списке.

# Работа с нестабильными автотестами

Вы можете пометить автотесты в системе как **нестабильные**, чтобы анализировать стабильность ваших автотестов. Отметить автотест как нестабильный (flaky) можно:

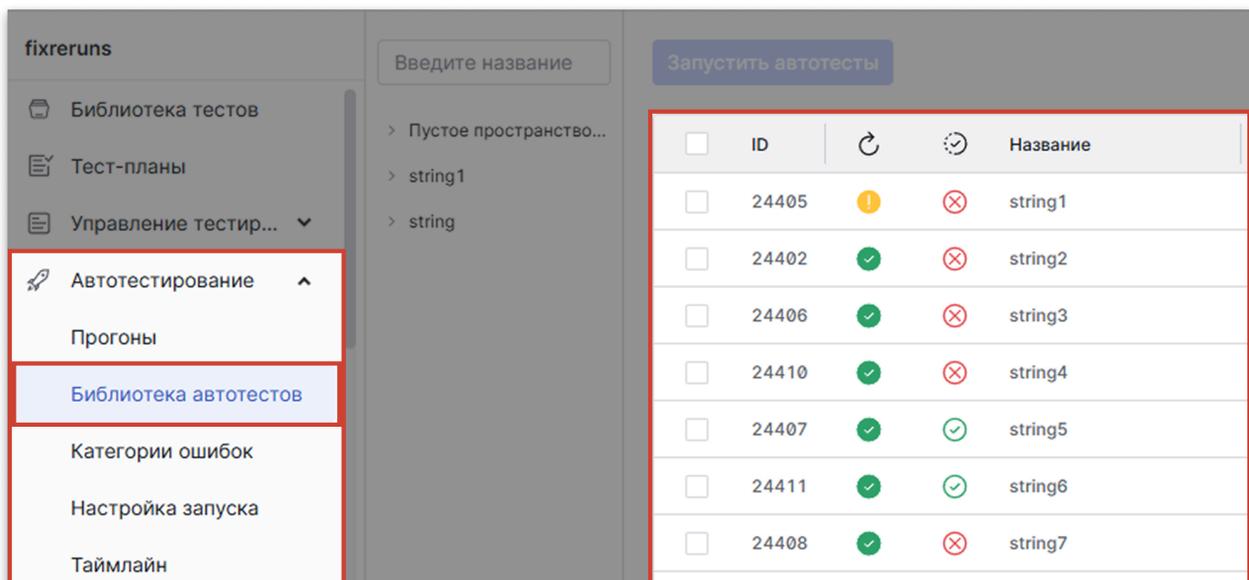
- В пользовательском интерфейсе Test IT
- Автоматически (необходимо произвести настройки)
- С помощью API

## 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**.



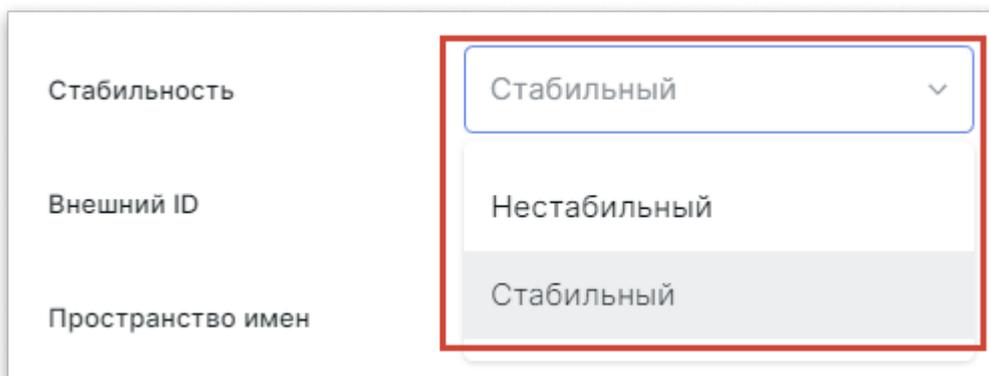
3. Откройте карточку автотеста.
4. Проверьте поле **Стабильность прохождения** в описании автотеста. Если стабильность теста менее 100%, вы можете пометить тест как нестабильный.

Вы можете проанализировать ошибки нестабильных автотестов и создать категории ошибок, чтобы система находила их автоматически.

## Проставление отметки о нестабильности через UI

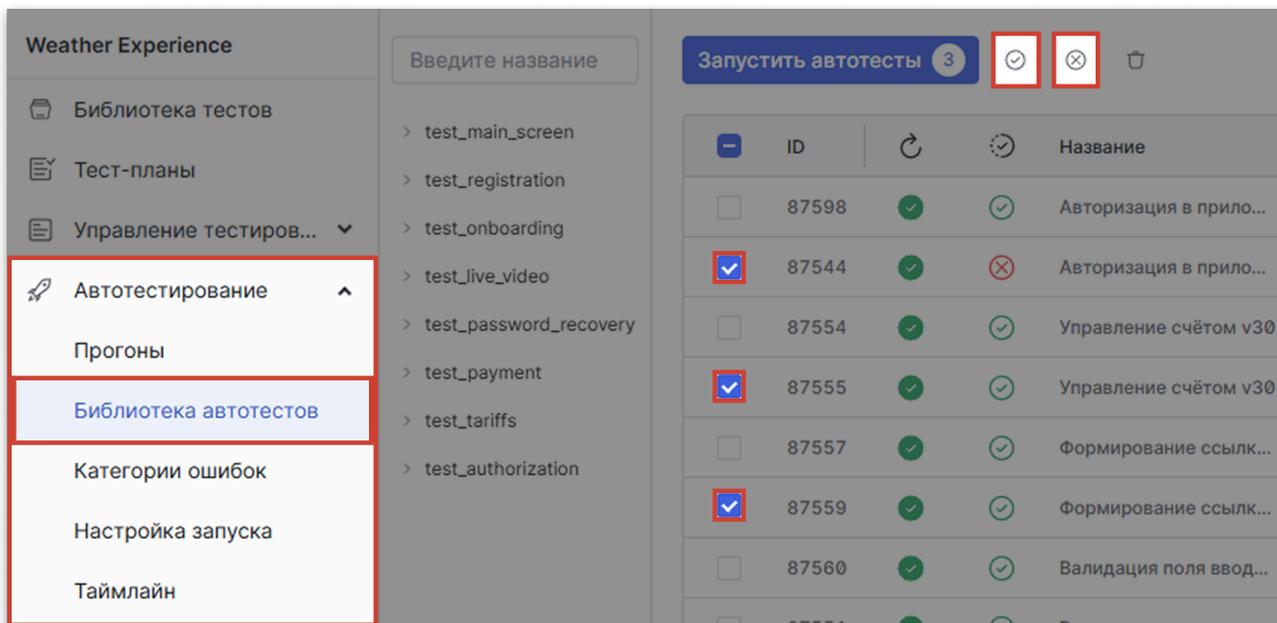
### Проставление отметки о нестабильности для одного автотеста

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

### 🔑 Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тарифе **Lite** она не входит.

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**.

**Weather Experience**

- Библиотека тестов
- Тест-планы
- Управление тестированием
- Автотестирование**
- Прогоны
- Библиотека автотестов
- Категории ошибок
- Настройка запуска**
- Таймлайн

### Настройки запуска автотестов

**Перезапуск автотестов**

Автоматически перезапускать проваленные автотесты

Лимит перезапусков:  раз

**Нестабильные автотесты**

Автоматически помечать нестабильные автотесты

Количество последних прогонов:

Процент стабильности теста менее:  %

**Сохранить**

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/falky/bulk` укажите значение параметра `isFlaky` как `true`.

Автотест, созданный в системе, будет автоматически отмечен как нестабильный.

# 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 и чек-листами, чтобы сделать их автоматизированными и запускать автотесты из тест-планов.

1. Откройте проект.
2. Перейдите в раздел **Автотестирование** → **Библиотека автотестов**.
3. Откройте карточку автотеста.
4. Перейдите в раздел **Связанные тест-кейсы**.

**1720 ChangeViewOptionForGlobalAttributes**

Описание  
История результатов  
Ссылки  
**Связанные тест-кейсы**

Фильтр Колонки Всего: 2

Выбрать Добавить

Приорит...	ID			Название	Действия
=	1624		✓	#1624 Log in	...
=	1728		✓	#1728 Чек-л	...

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**.

ID			Название	Секция	Приоритет	Статус	Автор	Действия
6896			#6896 Генерация ключей	Demo Test IT	Высокий	Готов	QA Anton	
760			#760 Проверка работоспособности	Библиотека тестов	Самый высокий	Требуется		Утвердить
6904			#6904 Отправка Web-тестов	Дерево секций	Средний	Готов		Отвязать
6905			#6905 Добавление и обновление тестов	Дерево секций	Высокий	Готов	QA Anton	...

Вы также можете добавлять связанные автотесты из рабочих элементов. Для этого смотрите раздел **Автоматизация рабочих элементов**.

# 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

### 🔗 Опция Standard и Enterprise: параметризованный запуск

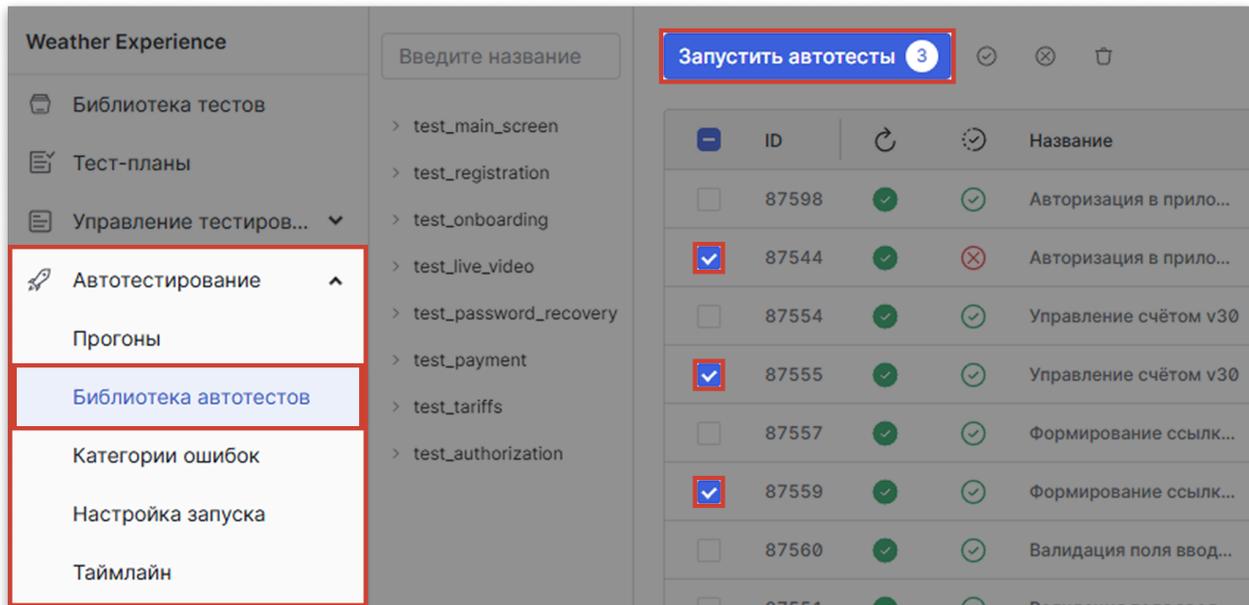
**Параметризованный запуск автотестов** (запуск с использованием дополнительных параметров) доступен в тарифах **Standard** и **Enterprise**. В тариф **Lite** он не входит.

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**.

#### 6. Нажмите **Создать прогон и запустить автотесты**.

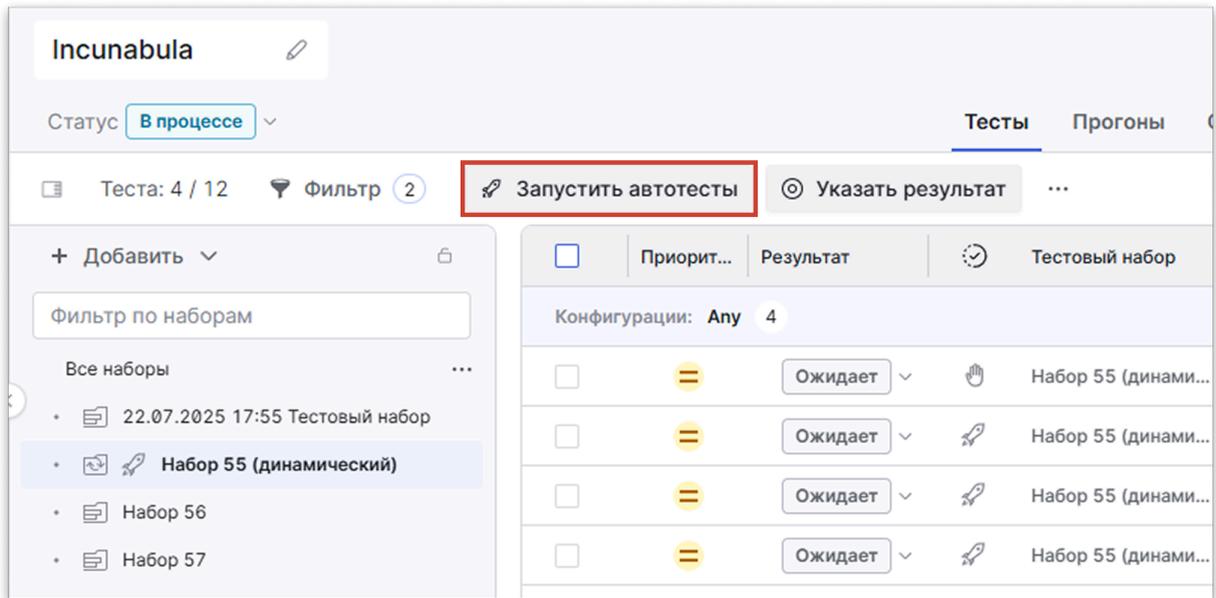
После запуска вам придет уведомление о нем, позволяющее перейти к прогону, где вы можете **проанализировать причины падения автотестов**.

## Запуск автотестов из тест-плана

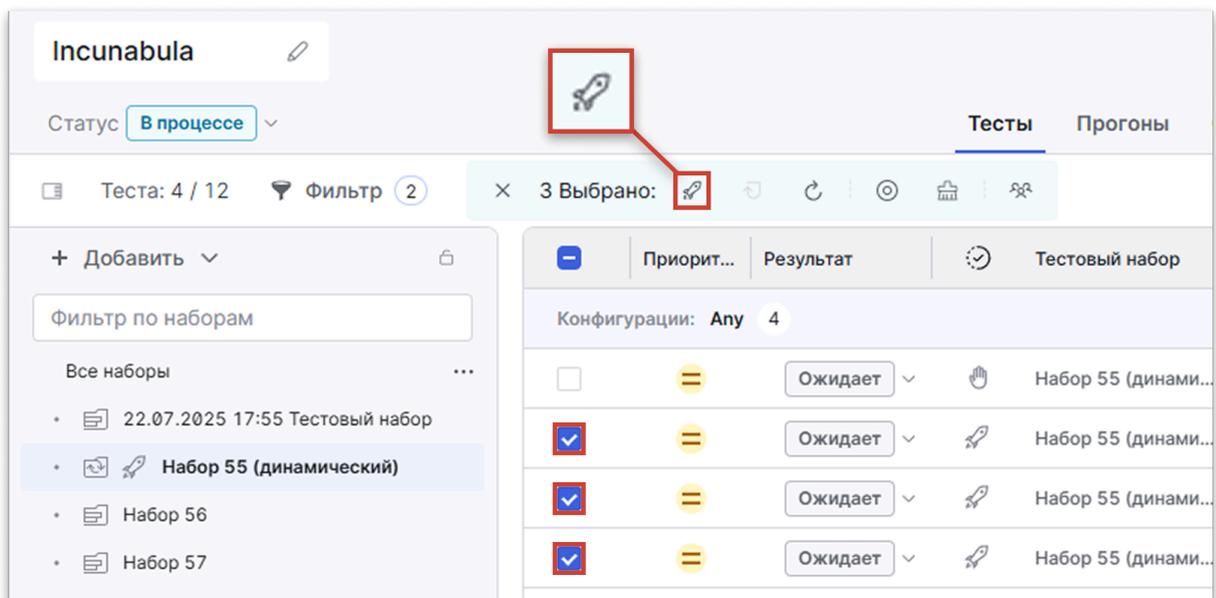
После того, как вы добавили автотесты в систему Test IT и **привязали их к тест-кейсам**, вы можете запускать автотесты прямо из системы управления тестированием. Чтобы запустить автотесты:

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.

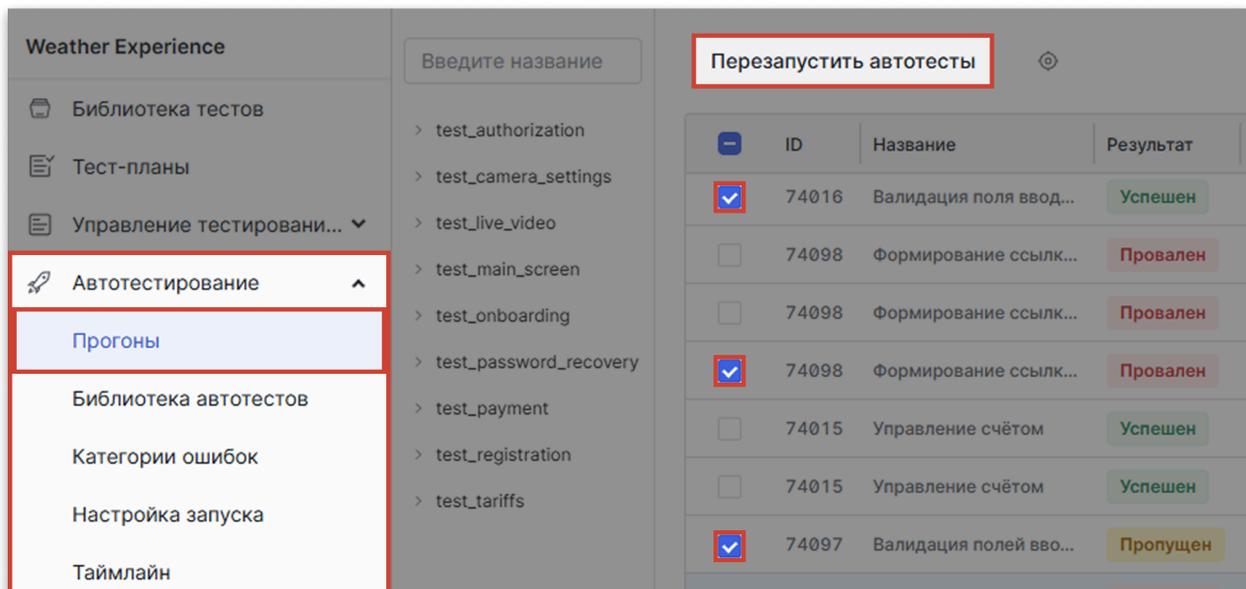
6. **Опционально:** Выберите **вебхуки** для запуска автотестов.
7. Нажмите **Создать прогон** и **запустить автотесты**.
8. **Опционально:** Чтобы просмотреть прогоны автотестов из тест-плана, откройте вкладку **Прогоны**.

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** → **Прогон**.
3. Откройте прогон.
4. Выберите автотесты и нажмите **Перезапустить автотесты**.



5. **Опционально:** Выберите **вехуки** для запуска автотестов.
6. Нажмите **Перезапустить**. Отобразится сообщение о запуске автотестов.

## Автоматический перезапуск автотестов с ошибкой

### 🔗 Опция Standard и Enterprise

Эта опция доступна в тарифах **Standard** и **Enterprise**. В тарифе **Lite** она не входит. 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**.

The screenshot shows the 'Weather Experience' application interface. On the left is a sidebar menu with the following items: 'Библиотека тестов', 'Тест-планы', 'Управление тестированием', 'Автотестирование' (highlighted with a red box), 'Прогоны', 'Библиотека автотестов', 'Категории ошибок', 'Настройка запуска' (highlighted with a red box), and 'Таймлайн'. The main content area is titled 'Настройки запуска автотестов' and contains two sections: 'Перезапуск автотестов' and 'Нестабильные автотесты'. In the 'Перезапуск автотестов' section, the 'Автоматически перезапускать проваленные автотесты' toggle is turned on, and the 'Лимит перезапусков' is set to 1. In the 'Нестабильные автотесты' section, the 'Автоматически помечать нестабильные автотесты' toggle is turned on, the 'Количество последних прогонов' is set to 100, and the 'Процент стабильности теста менее' is set to 100%. A blue 'Сохранить' button is located at the bottom of the settings area.

3. Включите опцию **Перезапускать автотесты**.
4. Установите ограничение на количество перезапусков (лимит перезапуска).
5. Нажмите **Сохранить**.

Обновлено: 17.10.2025, 13:38:51

# 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: 01.10.2025, 17:08:09

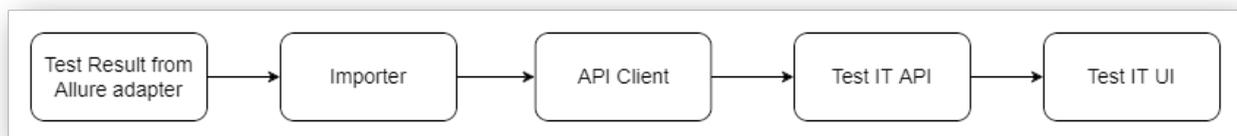
# Импорт результатов из Allure Adapters

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 TMS 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](#).



# 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.

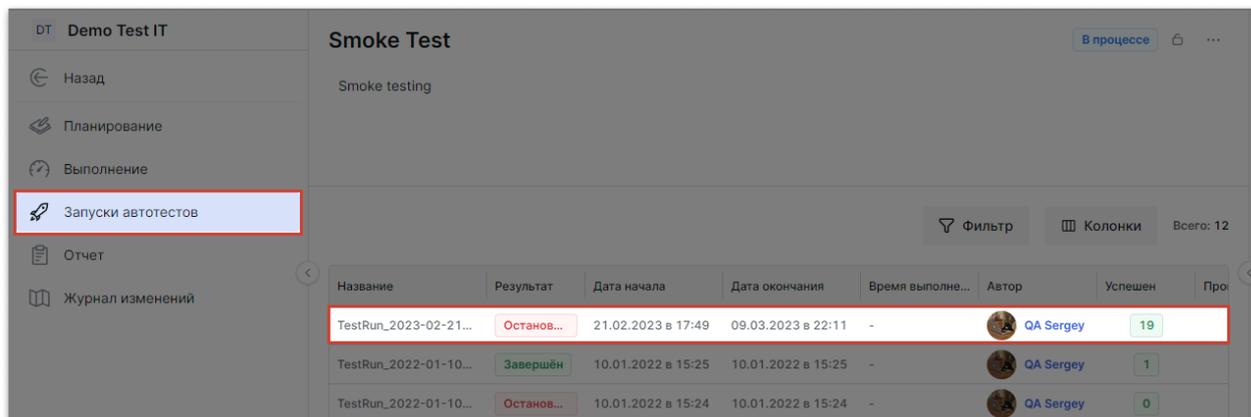
Прогоны отображаются в таблице. В ней вы можете отслеживать их названия, статус, дату начала, инициатора запуска, количество запущенных автотестов, результаты автотестов и категории ошибок. При необходимости вы можете **отфильтровать** прогоны.

## 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.



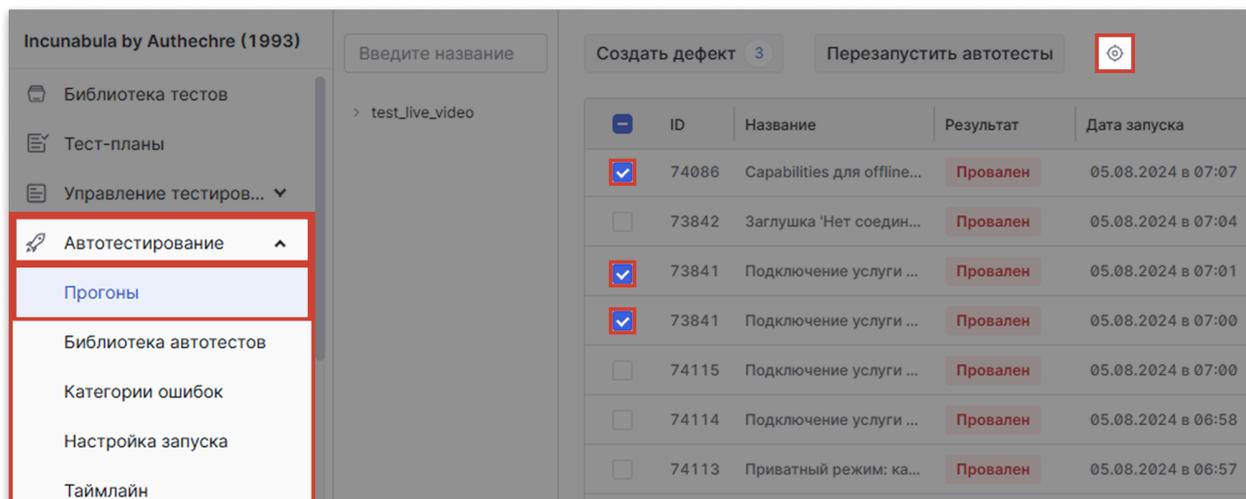
В окне прогона открывается таблица с данными о запущенных автотестах. Воспользуйтесь **фильтром**, чтобы отсортировать содержимое таблицы. Чтобы открыть результат конкретного теста, нажмите на строку данного теста в таблице.

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** → **Прогон**.
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, Инфраструктура** и/или **Без дефектов**). При необходимости добавьте

комментарий, вложение или ссылку.

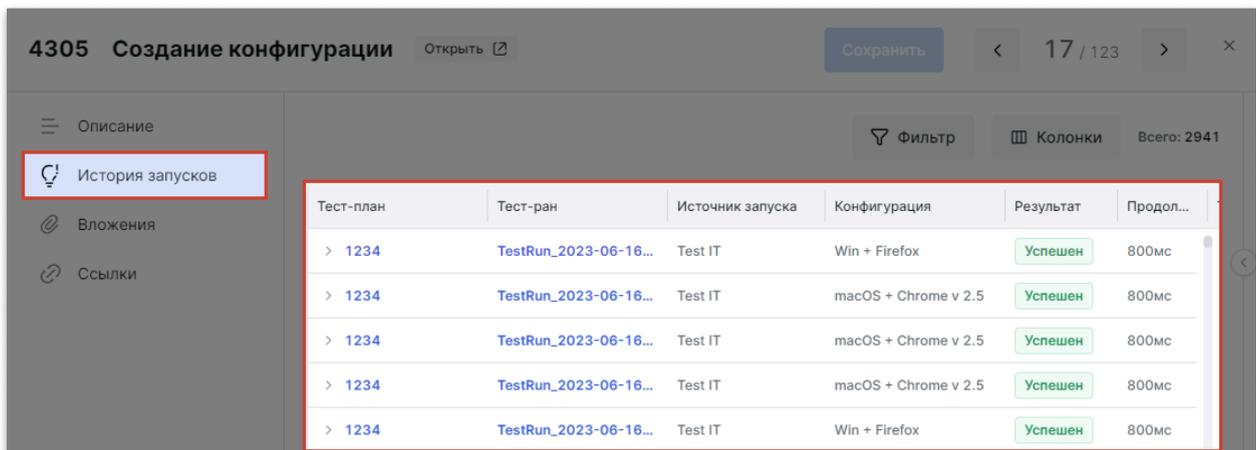
## 7. Нажмите **Сохранить**.

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**.



Тест-план	Тест-ран	Источник запуска	Конфигурация	Результат	Продол...
> 1234	TestRun_2023-06-16...	Test IT	Win + Firefox	Успешен	800мс
> 1234	TestRun_2023-06-16...	Test IT	macOS + Chrome v 2.5	Успешен	800мс
> 1234	TestRun_2023-06-16...	Test IT	macOS + Chrome v 2.5	Успешен	800мс
> 1234	TestRun_2023-06-16...	Test IT	macOS + Chrome v 2.5	Успешен	800мс
> 1234	TestRun_2023-06-16...	Test IT	Win + Firefox	Успешен	800мс

## 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: 01.10.2025, 17:08:09

# Создание дефектов в task-трекерах из прогонов автотестов

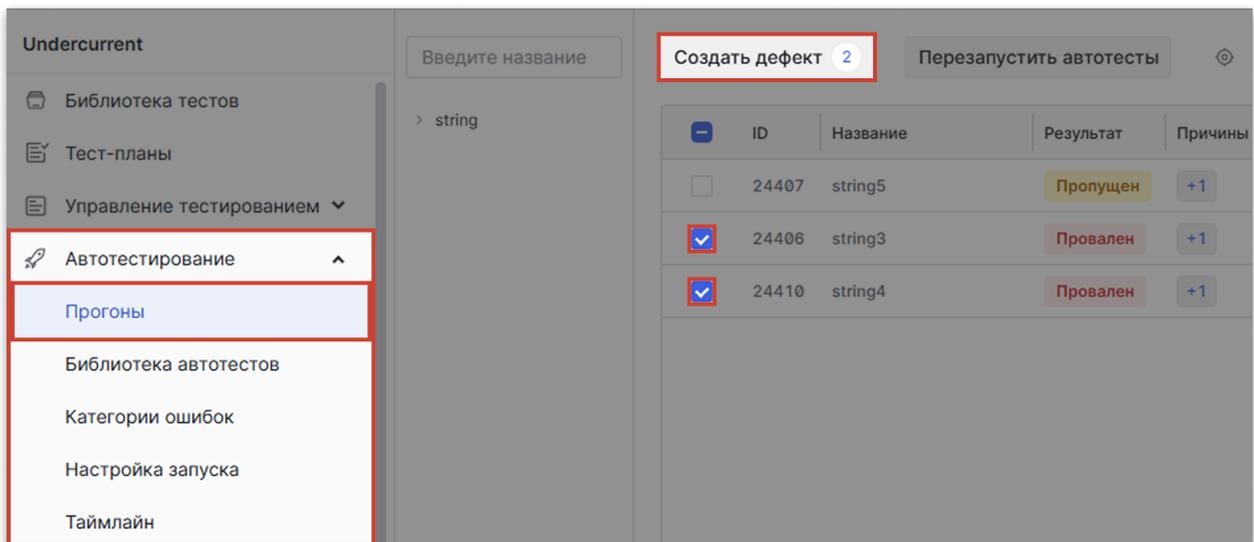
Требуется настроенная интеграция и подключение проекта!

Чтобы создавать тест-кейсы по задачам в вашем task-трекере, нужно настроить и протестировать интеграцию с task-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 task-трекере, выберите проект, в котором хотите создать дефект.

6. В открывшемся окне task-трекера создайте дефект.

После создания дефекта в task-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: 27.08.2025, 20:37:19

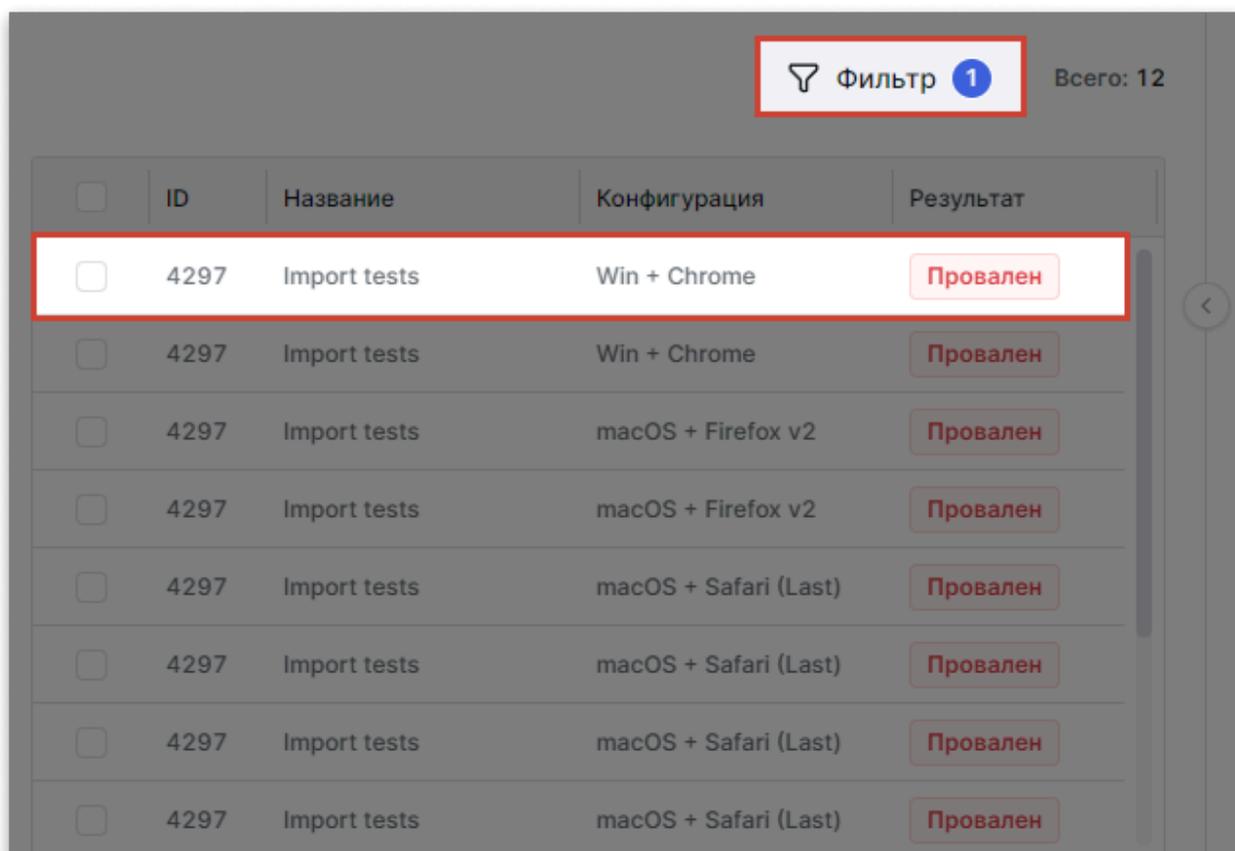
# Анализ результатов автотестов

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

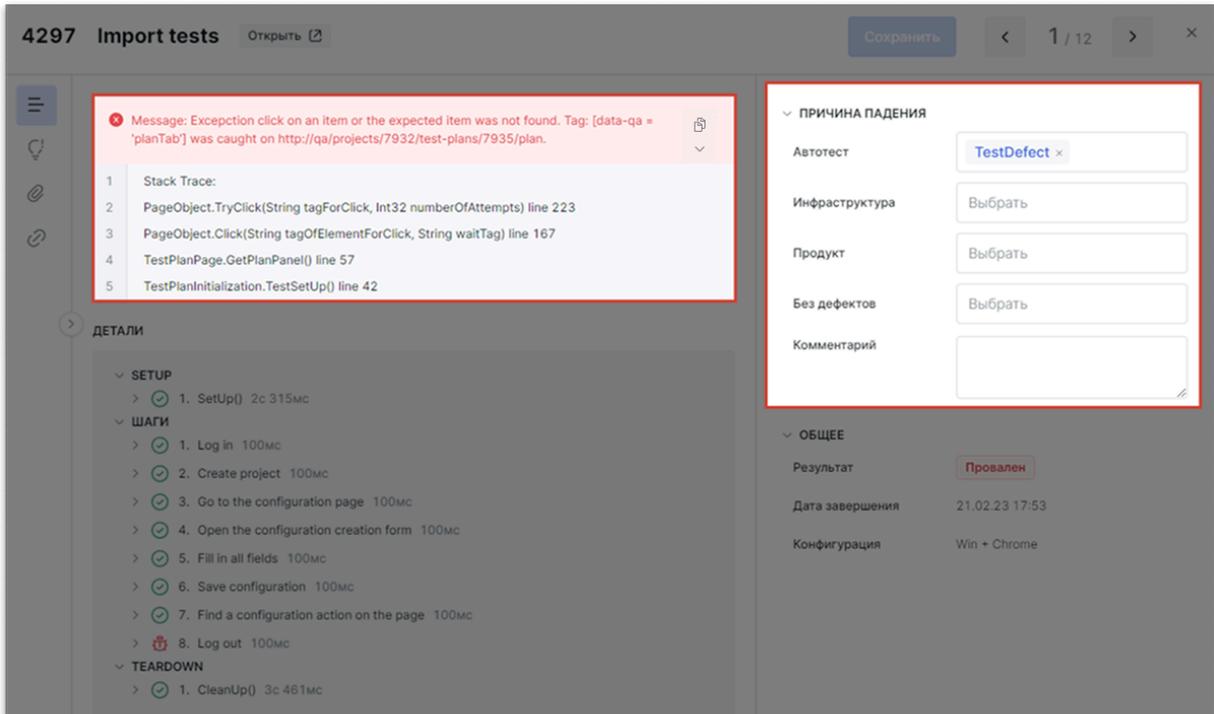
1. Откройте проект.
2. Перейдите в раздел **Прогоны**.
3. Откройте прогон.
4. Отфильтруйте проваленные автотесты. Для этого нажмите **Фильтр** над таблицей результатов тестов и в поле **Статус** выберите **Провален**.



The screenshot shows a table of test results. At the top right, there is a filter icon labeled 'Фильтр' with a blue circle containing the number '1', and the text 'Всего: 12'. The table has five columns: 'ID', 'Название', 'Конфигурация', and 'Результат'. The first row is highlighted with a red border. In this row, the 'ID' is 4297, the 'Название' is 'Import tests', the 'Конфигурация' is 'Win + Chrome', and the 'Результат' is 'Провален'. The other rows in the table also show 'Провален' as the result.

<input type="checkbox"/>	ID	Название	Конфигурация	Результат
<input type="checkbox"/>	4297	Import tests	Win + Chrome	Провален
<input type="checkbox"/>	4297	Import tests	Win + Chrome	Провален
<input type="checkbox"/>	4297	Import tests	macOS + Firefox v2	Провален
<input type="checkbox"/>	4297	Import tests	macOS + Firefox v2	Провален
<input type="checkbox"/>	4297	Import tests	macOS + Safari (Last)	Провален
<input type="checkbox"/>	4297	Import tests	macOS + Safari (Last)	Провален
<input type="checkbox"/>	4297	Import tests	macOS + Safari (Last)	Провален
<input type="checkbox"/>	4297	Import tests	macOS + Safari (Last)	Провален

- Откройте проваленный автотест. В верхней части описания результата вы увидите панель **Stack Trace**, отображающую строки кода, на которых тест упал.
- 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



- Нажмите **Сохранить**.

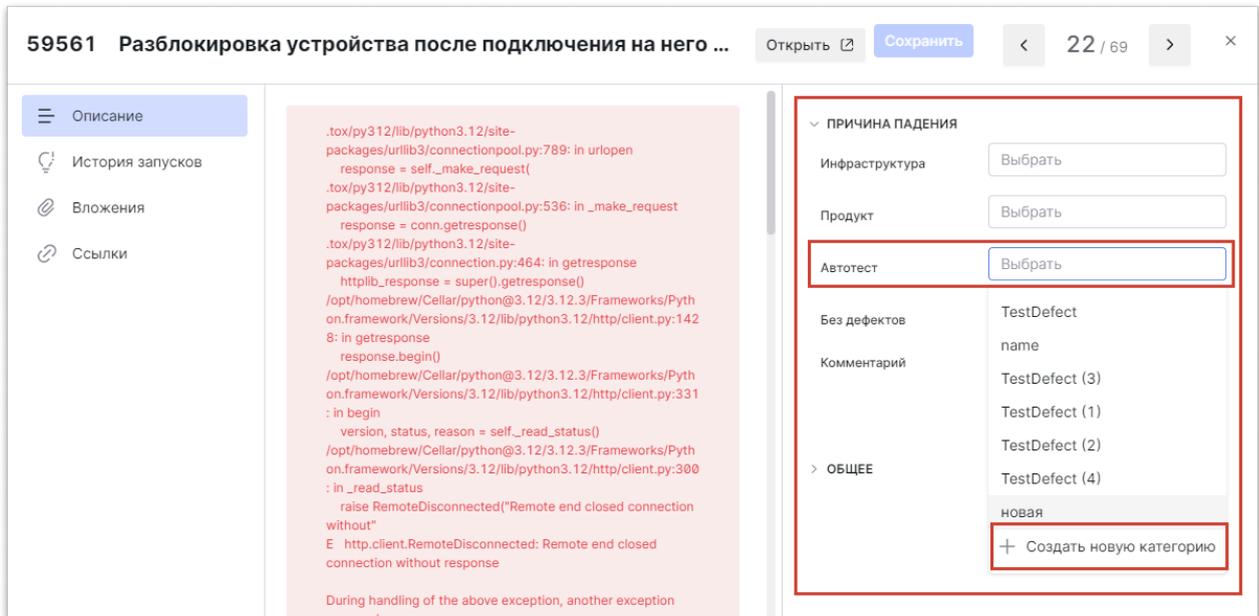
## Создание категорий ошибок

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:

- Откройте проект.
- Откройте раздел **Автотестирование** → **Прогон**.
- Откройте прогон.
- В прогоне откройте автотест.
- В окне автотеста в правой панели разверните секцию **Причина падения**.

- Нажмите левой кнопкой мыши в поле **Автотест**. Откроется выпадающее меню категорий ошибок.
- В выпадающем меню нажмите **Создать новую категорию**. Откроется окно создания категории ошибок.



- Заполните данные новой категории ошибок:

### Новая категория ошибок

\* Название категории

\* Причина падения

Регулярные выражения

```
def test_without_annotations_failed():> assert FalseE assert
Falsestests/test_annotations.py:11: AssertionError
```

+ Добавить еще выражение

Test IT автоматически скопирует первые 1054 символа кода из панели Stack Trace. Используйте этот код как регулярное выражение при создании категории ошибок. Система будет определять категорию ошибок для тестов с этими данными.

Отмена

- 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:

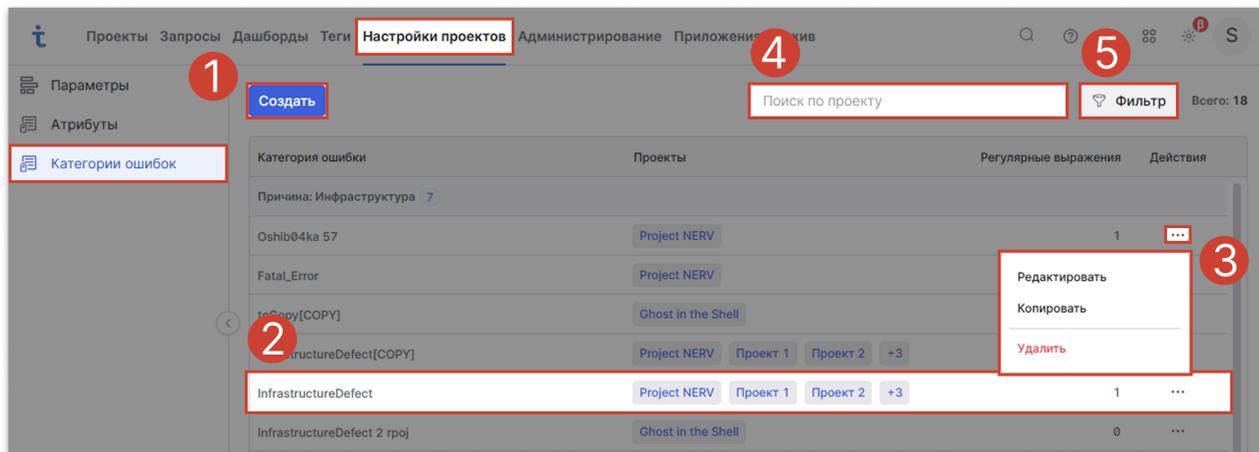
- **Глобальное управление** — на уровне всей системы Test IT в разделе **Настройки проектов**. Здесь можно назначать категории ошибок на различные проекты.
- **Управление в проекте** — на уровне отдельного проекта в разделе **Автотестирование**

### 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).

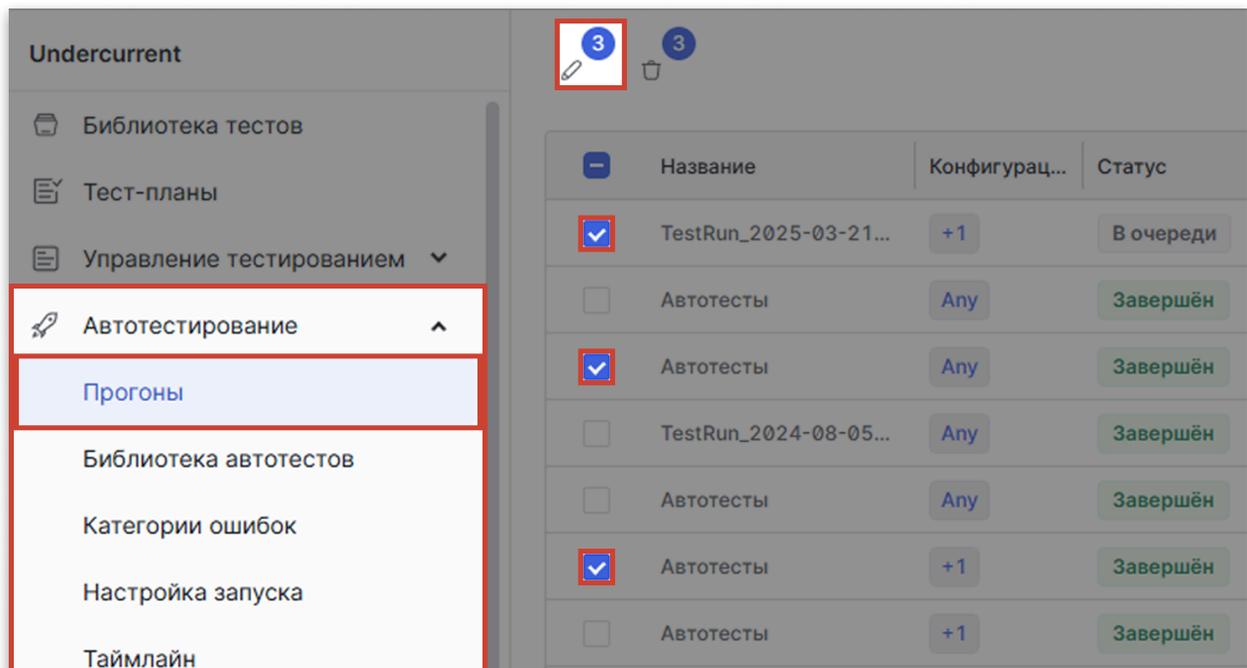
## Редактирование прогонов

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:

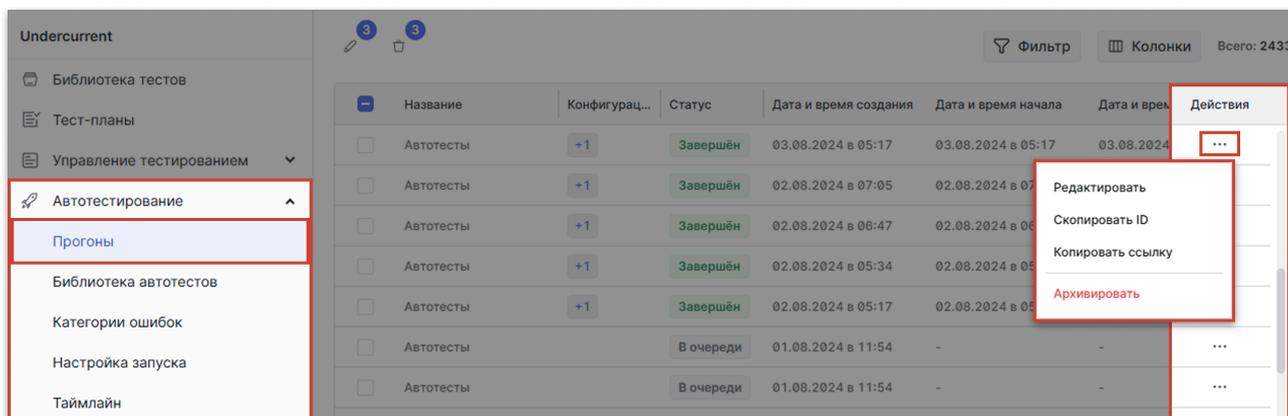
1. Откройте проект.
2. Перейдите в раздел **Автотестирование** → **Прогоны**.
3. Выберите нужные прогоны. Вы можете выбрать один или более прогонов. Чтобы выбрать все прогоны, поставьте флажок в ячейке **Название**.
4. Нажмите значок **Редактировать**.



- **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**.

# Действия с прогонами

Меню **Действия** расположено в правой части списка прогонов.



С помощью меню действий вы можете:

- Отредактировать отдельный прогон
- Скопировать ID прогона
- Скопировать ссылку на прогон
- Архивировать прогон

Чтобы открыть меню действий:

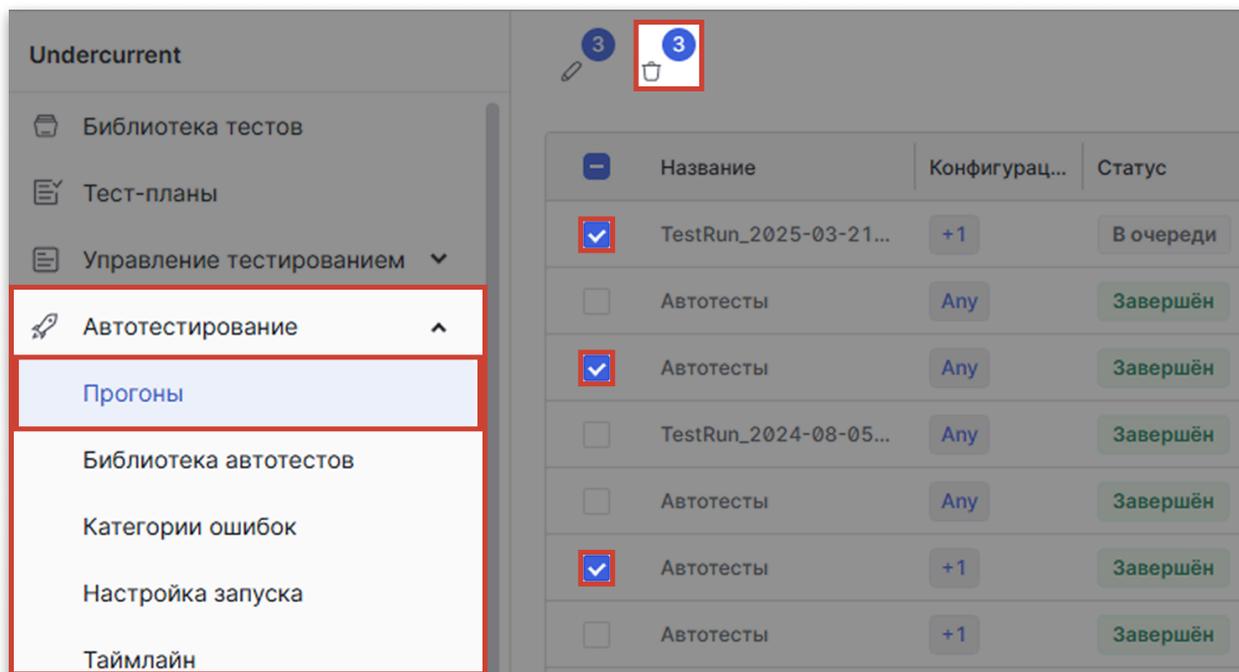
- Нажмите значок **Действия** справа в списке прогонов.

## Удаление 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.

5. **Опционально:** Для восстановления удаленных прогонов перейдите в **API-документацию** и используйте методы:

- Для восстановления одного прогона: `/api/v2/testRuns/{id}/restore`
- Для массового восстановления прогонов: `/api/v2/testRuns/{id}/restore`

## Отметка успешных автотестов

1. Откройте проект.

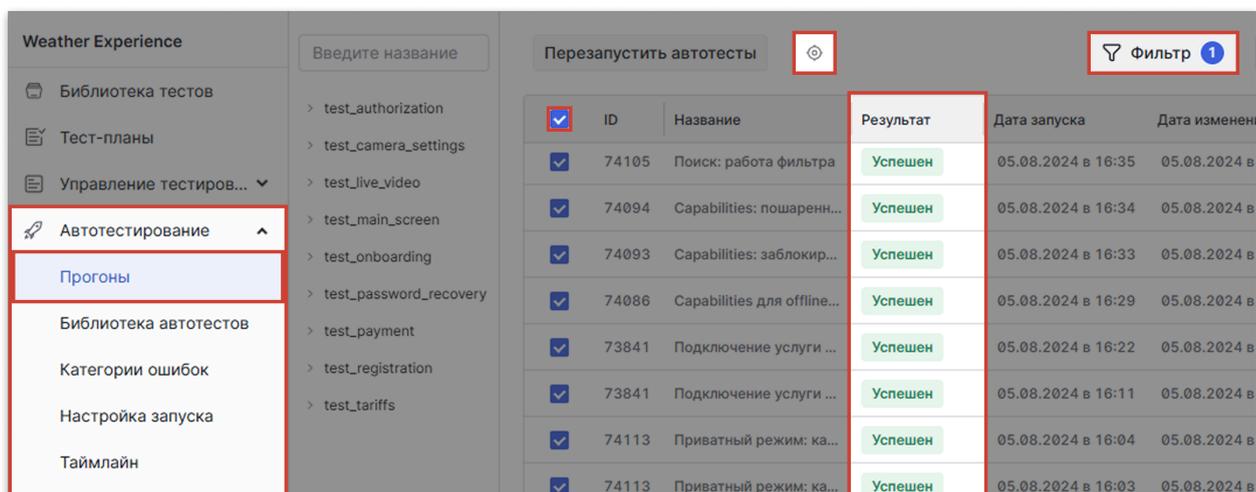
2. Перейдите в раздел **Автотестирование** → **Прогоны**.

3. Откройте прогон.

4. Отфильтруйте успешные автотесты. Для этого нажмите **Фильтр** над таблицей результатов тестов и в поле **Статус** выберите **Успешен**.

5. Выделите все успешные автотесты, поставив флажок в колонке **ID**.

6. Нажмите **Указать причину**.



7. В поле **Без дефектов** укажите значение `NoDefect` .

8. Нажмите **Сохранить**.

Обновлено: 11.11.2025, 21:27:49

# 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: 27.08.2025, 20:37:19

# 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:

```
1 $ pip install testit-cli
2 $ testit --help
3
4 usage: testit [options]
5 [...]
```

sh

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:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "TestNG test run" \
7 --results results/report.xml
```

sh

- <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

## Параметры запуска Test IT CLI

---

- Для получения полной справки о доступных параметрах используйте команду `testit results import --help` :

```
1 $ testit results import --help
2 usage: testit [options]
3
4 This tool is the command line wrapper of Test IT allowing you
5 to upload the test results in real time to Test IT
6
7 optional arguments:
8 -h, --help show this help message and exit
9 -u https://demo.testit.software, --url
10 https://demo.testit.software
11 Set url address of the Test IT instance
12 -t T2lKd2pLZGI4WHRhaVZUejNl, --token T2lKd2pLZGI4WHRhaVZUejNl
13 Set API token
14 -pi 5236eb3f-7c05-46f9-a609-dc0278896464, --project-id
15 5236eb3f-7c05-46f9-a609-dc0278896464
16 Set project id
17 -ci 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb, --configuration-id
18 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb
19 Set configuration id
20 -ti 3802f329-190c-4617-8bb0-2c3696abeb8f, --testrun-id
21 3802f329-190c-4617-8bb0-2c3696abeb8f
22 Set test run id
23 -tn TestRun01, --testrun-name TestRun01
24 Set test run name
25 -s ., --separator .
26 Separate the classname value in the results into namespace and
27 classname
28 -ns NameSpace01, --namespace NameSpace01
29 Set namespace
30 -cn ClassName01, --classname ClassName01
31 Set classname
32 -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.



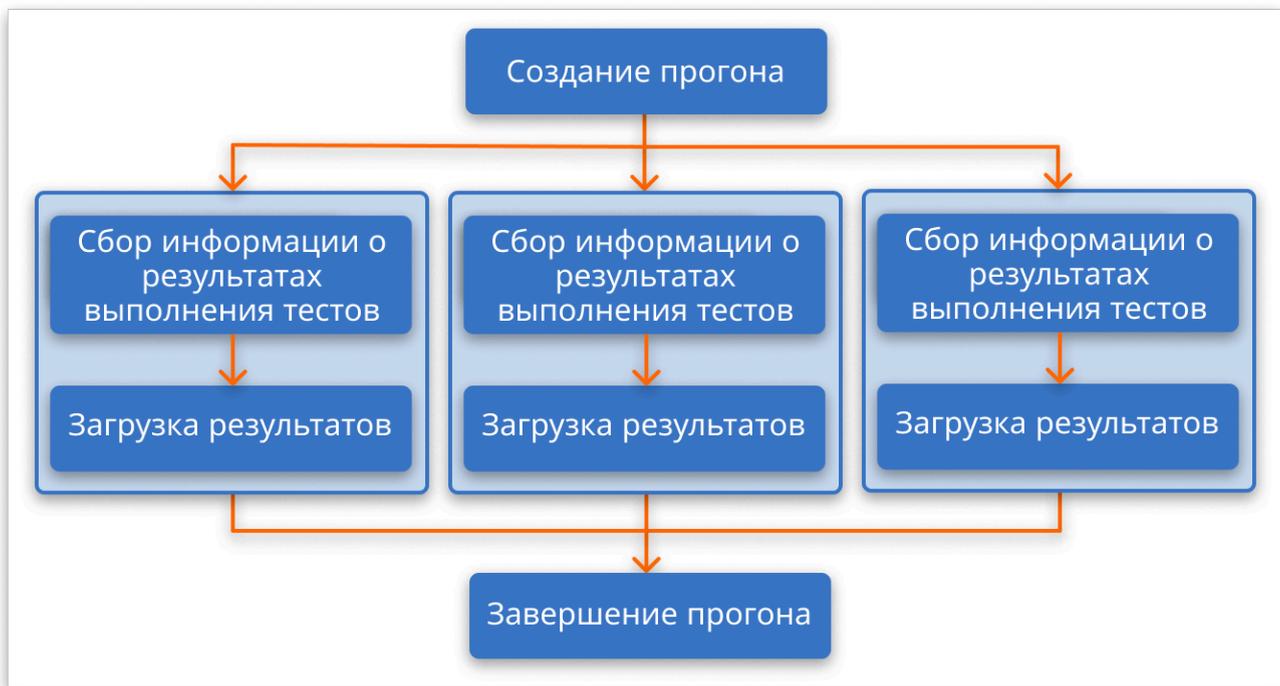
Процесс состоит из следующих этапов:

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

## Параллельный запуск

Режим параллельного запуска позволяет загружать результаты тестов при параллельном запуске тестов.



Процесс параллельного запуска включает в себя:

1. ► **Создание нового прогона**
2. ► **Загрузка результатов из разных потоков**
3. ► **Завершение прогона**

## 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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

# 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

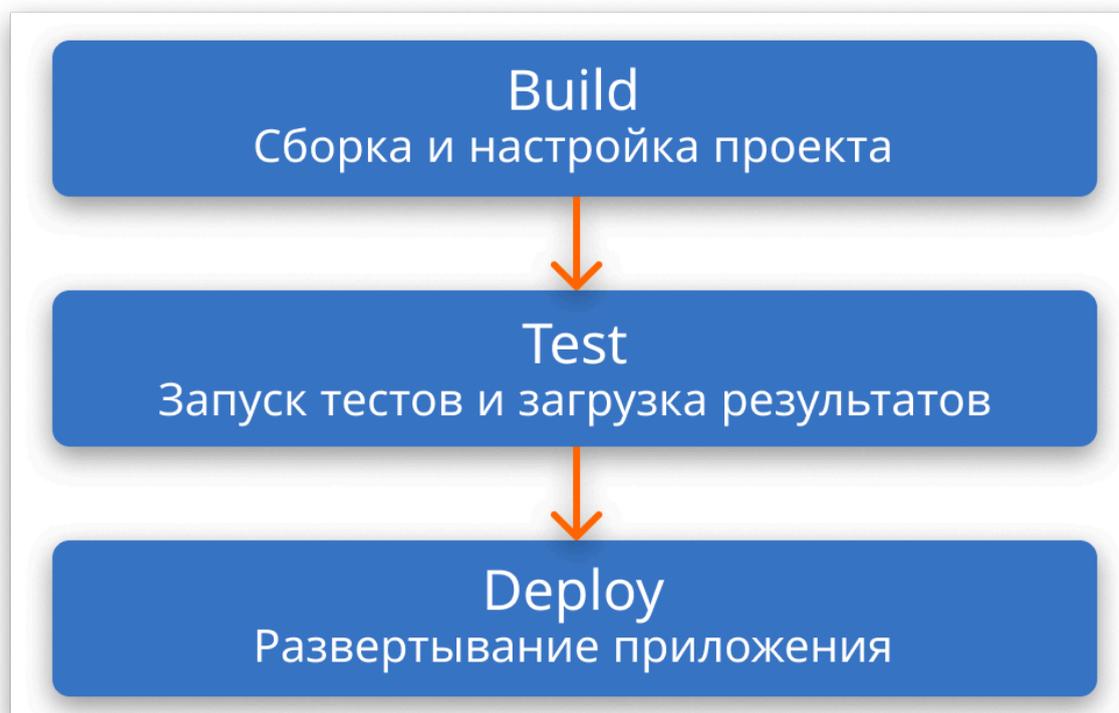
---

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**.

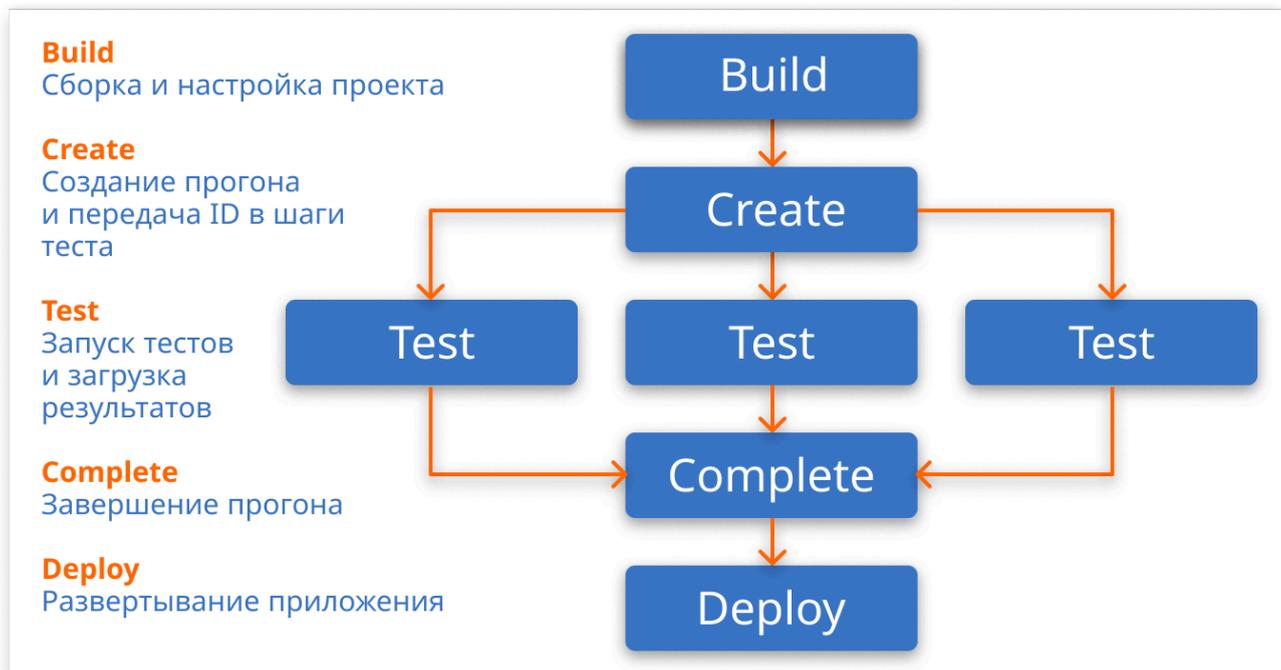
### Стандартный запуск

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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):

```
1 # azure-pipelines.yml
2 trigger:
3 - main
4
5 pool:
6 vmImage: ubuntu-latest
7 strategy:
8 matrix:
9 Python37:
10 python.version: '3.7'
11
12 steps:
13 # Install project dependencies
14 - script: |
15 python -m pip install --upgrade pip
16 pip install -r requirements.txt
17 displayName: 'Install dependencies'
18
19 # Execute test and generate report
20 - script: |
21 pytest --junitxml=results/test-results.xml src/tests.py
22 displayName: 'Run tests'
23
24 # Install Test IT CLI and submit test results to TMS
25 - script: |
26 pip install testit-cli
27 testit results import \
28 --url <YOUR_INSTANCE_URL> \
29 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
30 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
31 --testrun-name "Azure DevOps test run" \
32 --results results/test-results.xml
33 displayName: 'Submit results'
34 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):

```
1 # azure-pipelines.yml
2 trigger:
3 - main
4
5 pool:
6 vmImage: ubuntu-latest
7
8 jobs:
9 - job: Create
10 steps:
11 # Install Test IT CLI, create testrun and save testrun id to
12 variable
13 - script: |
14 pip install testit-cli
15 testit testrun create \
16 --url <YOUR_INSTANCE_URL> \
17 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
18 --token <TOKEN_VARIABLE> \
19 --testrun-name "Azure DevOps test run" \
20 --output output.txt
21 echo "##vso[task.setvariable
22 variable=testRunId;isoutput=true]$(cat output.txt)"
23 name: 'create'
24
25 - job: ParallelTesting
26 dependsOn: Create
27 variables:
28 testRunId: $[ dependencies.Create.outputs['create.testRunId'] ]
29 strategy:
30 parallel: 2
31 steps:
32 # Install project dependencies
33 - script: |
34 python -m pip install --upgrade pip
35 pip install -r requirements.txt
36 displayName: 'Install dependencies'
37
38 # Execute test and generate report
39 - script: |
40 pytest --junitxml=results/test-results.xml src/tests.py
41 displayName: 'Run tests'
42
43 # Install Test IT CLI and submit test results to TMS
44 - script: |
```

```
45 | pip install testit-cli
46 | testit results upload \
47 | --url <YOUR_INSTANCE_URL> \
48 | --testrun-id $(testRunId) \
49 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
50 | --results results/test-results.xml
51 | displayName: 'Submit results'
52 | condition: always()
53 |
54 | - job: Complete
55 | condition: always()
56 | dependsOn: [ParallelTesting, Create]
57 | variables:
58 | testRunId: $[ dependencies.Create.outputs['create.testRunId'] ]
59 | steps:
60 | # Install Test IT CLI and complete testrun
61 | - script: |
62 | pip install testit-cli
63 | testit testrun complete \
64 | --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 ознакомиться в [официальной документации](#) .



## 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):

```
1   # bamboo-specs/bamboo.yaml
2   plan:
3   project-key: <PROJECT_KEY>
4   key: BI
5   name: Bamboo Integration
6
7   stages:
8   - Test:
9   - Tests
10
11  Tests:
12  tasks:
13  - script:
14  # Install project dependencies
15  - pip install -r requirements.txt
16
17  # Execute test and generate report
18  - pytest --junitxml=results/test-results.xml src/tests.py
19  final-tasks:
20  - script:
21  # Install Test IT CLI
22  - pip install testit-cli
23
24  # Submit test results to TMS
25  - testit
26  --url <YOUR_INSTANCE_URL> \
27  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
28  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
29  --testrun-name "Bamboo test run" \
30  --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.yml` , который отвечает за работу пайплайнов в Bamboo (необходимо указать URL вашего ресурса и токен переменной):

```
1 # bamboo-specs/bamboo.yaml
2 version: 2
3 plan:
4 project-key: <PROJECT_KEY>
5 key: BI
6 name: Bamboo Integration
7
8 stages:
9 - Create:
10 - Create testrun
11 - Test:
12 jobs:
13 - Tests 1
14 - Tests 2
15 - Complete:
16 - Complete testrun
17
18 Create testrun:
19 tasks:
20 - script:
21 # Install Test IT CLI
22 - pip install testit-cli
23
24 # Create testrun
25 - testit testrun create \
26 --url <YOUR_INSTANCE_URL> \
27 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
28 --token <TOKEN_VARIABLE> \
29 --testrun-name "Bamboo test run" \
30 --output output.txt
31 artifacts:
32 -
33 name: Reports
34 location: .
35 pattern: 'output.txt'
36 shared: true
37
38 Tests 1:
39 tasks:
40 - script:
41 # Install project dependencies
42 - pip install -r requirements.txt
43
44 # Execute test and generate report
```

```
45 - pytest --junitxml=results/test-results.xml src/tests.py
46 final-tasks:
47 - script:
48 # Install Test IT CLI
49 - pip install testit-cli
50
51 # Submit test results to TMS
52 - testit results upload \
53 --url <YOUR_INSTANCE_URL> \
54 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
55 --testrun-id "$(cat output.txt)" \
56 --results results/test-results.xml
57
58 Tests 2:
59 tasks:
60 - script:
61 # Install project dependencies
62 - pip install -r requirements.txt
63
64 # Execute test and generate report
65 - pytest --junitxml=results/test-results.xml src/tests.py
66 final-tasks:
67 - script:
68 # Install Test IT CLI
69 - pip install testit-cli
70
71 # Submit test results to TMS
72 - testit results upload \
73 --url <YOUR_INSTANCE_URL> \
74 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
75 --testrun-id "$(cat output.txt)" \
76 --results results/test-results.xml
77
78 Complete testrun:
79 tasks:
80 - script:
81 # Install Test IT CLI
82 - pip install testit-cli
83
84 # Complete testrun
85 - testit testrun complete \
86 --url <YOUR_INSTANCE_URL> \
87 --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 ознакомиться в **официальной документации** .

Обновлено: 20.06.2025, 14:33:37



## 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):

```
1 # bitbucket-pipelines.yml
2 image: python:3.8
3
4 pipelines:
5   default:
6     - step:
7       name: Run tests and generate report
8       script:
9         # Install project dependencies
10        - pip install -r requirements.txt
11
12        # Execute test and generate report
13        - pytest --junitxml=results/test-results.xml src/tests.py
14      after-script:
15        # Install Test IT CLI
16        - pip install testit-cli
17
18        # Submit test results to TMS
19        - testit
20        --url <YOUR_INSTANCE_URL>
21        --project-id 5236eb3f-7c05-46f9-a609-dc0278896464
22        --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb
23        --testrun-name "Bitbucket test run"
24        --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):

```
1 # bitbucket-pipelines.yml
2 image: python:3.8
3
4 pipelines:
5   default:
6     - step:
7       name: Create testrun
8       script:
9         # Install Test IT CLI
10        - pip install testit-cli
11
12       # Create testrun
13       - testit testrun create \
14         --url <YOUR_INSTANCE_URL>
15         --project-id 5236eb3f-7c05-46f9-a609-dc0278896464
16         --token <TOKEN_VARIABLE> \
17         --testrun-name "Bitbucket test run"
18         --output output.txt
19
20       # Export TMS_TEST_RUN_ID env variable
21       - echo "export TMS_TEST_RUN_ID=$(cat output.txt)" >> tms_env.sh
22     artifacts:
23       - tms_env.sh
24     - parallel:
25       - step:
26         name: Test group 1
27         script:
28           # Install project dependencies
29           - pip install -r requirements.txt
30
31         # Execute test and generate report
32         - pytest --junitxml=results/test-results.xml src/tests.py
33     after-script:
34       # Use the artifact from the previous step
35       - cat tms_env.sh
36       - source tms_env.sh
37
38     # Install Test IT CLI
39     - pip install testit-cli
40
41     # Submit test results to TMS
42     - testit results upload \
43       --url <YOUR_INSTANCE_URL>
44     --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb
```

```
45 --testrun-name "Bitbucket test run"
46 --results results/test-results.xml
47 - step:
48 name: Test group 2
49 script:
50 # Install project dependencies
51 - pip install -r requirements.txt
52
53 # Execute test and generate report
54 - pytest --junitxml=results/test-results.xml src/tests.py
55 after-script:
56 # Use the artifact from the previous step
57 - cat tms_env.sh
58 - source tms_env.sh
59
60 # Install Test IT CLI
61 - pip install testit-cli
62
63 # Submit test results to TMS
64 - testit results upload \
65 --url <YOUR_INSTANCE_URL>
66 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb
67 --testrun-name "Bitbucket test run"
68 --results results/test-results.xml
69 - step:
70 name: Complete
71 script:
72 # Use the artifact from the previous step
73 - cat tms_env.sh
74 - source tms_env.sh
75
76 # Install Test IT CLI
77 - pip install testit-cli
78
79 # Complete testrun
80 - testit testrun complete \
81 --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 Vitbucket для хранения чувствительной информации. Подробнее с секретами вы можете ознакомиться в [официальной документации](#) .

Обновлено: 20.06.2025, 14:33:37



## 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):

```
1 # .circleci/config.yml
2 version: 2.1
3
4 jobs:
5 test:
6 docker:
7 - image: python:3.10-bullseye
8
9 steps:
10 - checkout
11
12 - attach_workspace:
13 at: ~/
14
15 # Install project dependencies
16 - run:
17 command: |
18 python -m pip install --upgrade pip
19 pip install -r requirements.txt
20
21 # Execute test and generate report
22 - run:
23 command: |
24 pytest --junitxml=results/test-results.xml src/tests.py
25
26 # Install Test IT CLI and submit test results to TMS
27 - run:
28 when: always
29 command: |
30 pip install testit-cli
31 testit results import \
32 --url <YOUR_INSTANCE_URL> \
33 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
34 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
35 --testrun-name "CircleCI test run" \
36 --results results/test-results.xml
37
38 - store_artifacts:
39 path: results
40 destination: test_results
41 - store_test_results:
42 path: results
43
44 workflows:
```

```
45 | version: 2
46 | full:
47 | jobs:
48 | - 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):

```
1 # .circleci/config.yml
2 version: 2.1
3
4 jobs:
5   create:
6     docker:
7       - image: python:3.10-bullseye
8
9     steps:
10      - attach_workspace:
11        at: ~/
12
13      # Install Test IT CLI and create testrun
14      - run:
15        command: |
16          pip install testit-cli
17          testit testrun create \
18            --url <YOUR_INSTANCE_URL> \
19            --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
20            --token <TOKEN_VARIABLE> \
21            --testrun-name "CircleCI test run" \
22            --output output.txt
23      - persist_to_workspace:
24        root: ~/
25      paths:
26        - project
27
28      test:
29        docker:
30          - image: python:3.10-bullseye
31
32      parallelism: 4
33
34      steps:
35        - checkout
36
37        - attach_workspace:
38          at: ~/
39
40      # Install project dependencies
41      - run:
42        command: |
43          python -m pip install --upgrade pip
44          pip install -r requirements.txt
```

```
45
46 # Execute test and generate report
47 - run:
48   command: |
49     pytest --junitxml=results/test-results.xml src/tests.py
50
51 # Install Test IT CLI and submit test results to TMS
52 - run:
53   when: always
54   command: |
55     pip install testit-cli
56     testit results upload \
57       --url <YOUR_INSTANCE_URL> \
58       --testrun-id "$(cat output.txt)" \
59       --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
60       --results results/test-results.xml
61
62 - store_artifacts:
63   path: results
64   destination: test_results
65 - store_test_results:
66   path: results
67
68 complete:
69 docker:
70 - image: python:3.10-bullseye
71
72 steps:
73 - attach_workspace:
74   at: ~/
75
76 # Install Test IT CLI and complete testrun
77 - run:
78   when: always
79   command: |
80     pip install testit-cli
81     testit testrun complete \
82       --url <YOUR_INSTANCE_URL> \
83       --testrun-id "$(cat output.txt)"
84 workflows:
85   version: 2
86   full:
87   jobs:
88     - create
89     - test:
```

```
90 | requires:
91 | - create
92 | - complete:
93 | requires:
94 | - 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 ознакомиться в [официальной документации](#) .



## 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):

```
1      # .gitflic-ci.yml
2      default:
3      image: python:3.10-bullseye
4
5      stages:
6      - test
7
8      test:
9      stage: test
10     script:
11     # Install project dependencies
12     - pip install -r requirements.txt
13
14     # Execute test and generate report
15     - pytest --junitxml=results/test-results.xml src/tests.py
16     after_script:
17     # Install Test IT CLI
18     - pip install testit-cli
19
20     # Submit test results to TMS
21     - testit results import \
22     --url <YOUR_INSTANCE_URL> \
23     --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
24     --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
25     --testrun-name "GitFlic test run" \
26     --results results/test-results.xml
27     artifacts:
28     when: always
29     reports:
30     junit:
31     - 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):

```
1 # .gitflic-ci.yml
2 default:
3 image: python:3.10-bullseye
4
5 stages:
6 - create
7 - test
8 - complete
9
10 create:
11 stage: create
12 script:
13 # Install Test IT CLI
14 - pip install testit-cli
15
16 # Create testrun
17 - testit testrun create \
18 --url <YOUR_INSTANCE_URL> \
19 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
20 --token <TOKEN_VARIABLE> \
21 --testrun-name "GitFlic test run" \
22 --output output.txt
23
24 # Export TMS_TEST_RUN_ID env variable
25 - echo "TMS_TEST_RUN_ID=$(cat output.txt)" > tms.env
26 artifacts:
27 reports:
28 dotenv: tms.env
29
30 test:
31 parallel: 4
32 stage: test
33 script:
34 # Install project dependencies
35 - pip install -r requirements.txt
36
37 # Execute test and generate report
38 - pytest --junitxml=results/test-results.xml src/tests.py
39 after_script:
40 # Install Test IT CLI
41 - pip install testit-cli
42
43 # Submit test results to TMS
44 - testit results upload \
```

```
45 --url <YOUR_INSTANCE_URL> \  
46 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
47 --results results/test-results.xml  
48 artifacts:  
49 when: always  
50 reports:  
51 junit:  
52 - results/test-results.xml  
53 dependencies:  
54 - create  
55  
56 complete:  
57 stage: complete  
58 script:  
59 # Install Test IT CLI  
60 - pip install testit-cli  
61  
62 # Complete testrun  
63 - testit testrun complete \  
64 --url <YOUR_INSTANCE_URL>  
65 dependencies:  
66 - test  
67 - create  
68 when: always  
69
```

## 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](#) .



## 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):

```
1 # .python-app.yml
2 name: Python application
3
4 on:
5 push:
6 branches: [ "main" ]
7
8 permissions:
9 contents: read
10
11 jobs:
12 build:
13
14 runs-on: ubuntu-latest
15
16 steps:
17 - uses: actions/checkout@v3
18 - name: Set up Python 3.10
19   uses: actions/setup-python@v3
20   with:
21     python-version: "3.10"
22
23 # Install project dependencies
24 - name: Install dependencies
25   run: |
26     python -m pip install --upgrade pip
27     pip install -r requirements.txt
28
29 # Execute test and generate report
30 - name: Test with pytest
31   run: |
32     pytest --junitxml=results/test-results.xml src/tests.py
33
34 # Install Test IT CLI and submit test results to TMS
35 - name: Submit results
36   if: always()
37   run: |
38     pip install testit-cli
39     testit results import \
40     --url <YOUR_INSTANCE_URL> \
41     --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
42     --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
43     --testrun-name "GitHub test run" \
44
```

```
--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):

```
1 # .python-app.yml
2 name: Python application
3
4 on:
5 push:
6 branches: [ "main" ]
7
8 permissions:
9 contents: read
10
11 jobs:
12 create:
13 runs-on: ubuntu-latest
14 outputs:
15 test-run-id: ${ steps.create-testrun.outputs.test-run-id }
16 steps:
17 # Install Test IT CLI, create testrun and save testrun id to
18 # variable
19 - name: Create testrun
20 id: create-testrun
21 run: |
22 pip install testit-cli
23 testit testrun create \
24 --url <YOUR_INSTANCE_URL> \
25 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
26 --token <TOKEN_VARIABLE> \
27 --testrun-name "GitHub test run" \
28 --output output.txt
29 ID=$(cat output.txt)
30 echo "test-run-id=$ID" >> $GITHUB_OUTPUT
31 test:
32 needs: create
33 runs-on: ubuntu-latest
34
35 strategy:
36 fail-fast: false
37 matrix:
38 ci_index: [0, 1, 2, 3]
39 ci_total: [4]
40
41 steps:
42 - uses: actions/checkout@v3
43 - name: Set up Python 3.10
44 uses: actions/setup-python@v3
```

```
45 with:
46 python-version: "3.10"
47
48 # Install project dependencies
49 - name: Install dependencies
50 run: |
51 python -m pip install --upgrade pip
52 pip install -r requirements.txt
53
54 # Execute test and generate report
55 - name: Test with pytest
56 run: |
57 pytest --junitxml=results/test-results.xml src/tests.py
58
59 # Install Test IT CLI and submit test results to TMS
60 - name: Submit results
61 if: always()
62 run: |
63 pip install testit-cli
64 testit results upload \
65 --url <YOUR_INSTANCE_URL> \
66 --testrun-id $(testRunId) \
67 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
68 --results results/test-results.xml
69 env:
70 CI_TOTAL: ${matrix.ci_total}
71 CI_INDEX: ${matrix.ci_index}
72
73 complete:
74 needs: [create, test]
75 if: always()
76 runs-on: ubuntu-latest
77 steps:
78 # Install Test IT CLI and complete testrun
79 - name: Complete testrun
80 if: always()
81 run: |
82 pip install testit-cli
83 testit testrun complete \
84 --url <YOUR_INSTANCE_URL> \
  --testrun-id $(testRunId)
```

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 ознакомиться в **официальной документации** .

Обновлено: 20.06.2025, 14:33:37



# GitLab

## 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):

```
1 # .gitlab-ci.yml
2 default:
3 image: python:3.10-bullseye
4
5 stages:
6 - test
7
8 test:
9 stage: test
10 script:
11 # Install project dependencies
12 - pip install -r requirements.txt
13
14 # Execute test and generate report
15 - pytest --junitxml=results/test-results.xml src/tests.py
16 after_script:
17 # Install Test IT CLI
18 - pip install testit-cli
19
20 # Submit test results to TMS
21 - testit results import \
22 --url <YOUR_INSTANCE_URL> \
23 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
24 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
25 --testrun-name "GitLab test run" \
26 --results results/test-results.xml
27 artifacts:
28 when: always
29 reports:
30 junit:
31 - 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):

```
1 # .gitlab-ci.yml
2 default:
3 image: python:3.10-bullseye
4
5 stages:
6 - create
7 - test
8 - complete
9
10 create:
11 stage: create
12 script:
13 # Install Test IT CLI
14 - pip install testit-cli
15
16 # Create testrun
17 - testit testrun create \
18 --url <YOUR_INSTANCE_URL> \
19 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
20 --token <TOKEN_VARIABLE> \
21 --testrun-name "GitLab test run" \
22 --output output.txt
23
24 # Export TMS_TEST_RUN_ID env variable
25 - echo "TMS_TEST_RUN_ID=$(cat output.txt)" > tms.env
26 artifacts:
27 reports:
28 dotenv: tms.env
29
30 test:
31 parallel: 4
32 stage: test
33 script:
34 # Install project dependencies
35 - pip install -r requirements.txt
36
37 # Execute test and generate report
38 - pytest --junitxml=results/test-results.xml src/tests.py
39 after_script:
40 # Install Test IT CLI
41 - pip install testit-cli
42
43 # Submit test results to TMS
44 - testit results upload \
```

```
45 --url <YOUR_INSTANCE_URL> \  
46 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
47 --results results/test-results.xml  
48 artifacts:  
49 when: always  
50 reports:  
51 junit:  
52 - results/test-results.xml  
53 dependencies:  
54 - create  
55  
56 complete:  
57 stage: complete  
58 script:  
59 # Install Test IT CLI  
60 - pip install testit-cli  
61  
62 # Complete testrun  
63 - testit testrun complete \  
64 --url <YOUR_INSTANCE_URL>  
65 dependencies:  
66 - test  
67 - create  
68 when: always  
69
```

## 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 ознакомиться в [официальной документации](#) .



# Jenkins

## 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):

```
1 # Jenkinsfile
2 pipeline {
3   agent {
4     docker { image 'python:3.10-bullseye' }
5   }
6
7   stages {
8     stage('Test') {
9       steps {
10        // Install project dependencies
11        sh '''
12        python -m pip install --upgrade pip
13        pip install -r requirements.txt
14        '''
15
16        // Execute test and generate report
17        sh 'pytest --junitxml=results/test-results.xml src/tests.py'
18
19        // Install Test IT CLI and submit test results to TMS
20        sh '''
21        pip install testit-cli
22        testit results import \
23        --url <YOUR_INSTANCE_URL> \
24        --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
25        --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
26        --testrun-name "Jenkins test run" \
27        --results results/test-results.xml
28        '''
29      }
30    }
31  }
32  post {
33    always {
34      junit 'results/*.xml'
35    }
36  }
37 }
```

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):

```
1 # Jenkinsfile
2 pipeline {
3   agent {
4     docker { image 'python:3.10-bullseye' }
5   }
6
7   stages {
8     stage('Create') {
9       steps {
10        // Install Test IT CLI and create testrun
11        sh '''
12        pip install testit-cli
13        testit testrun create \
14        --url <YOUR_INSTANCE_URL> \
15        --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
16        --token <TOKEN_VARIABLE> \
17        --testrun-name "Jenkins test run" \
18        --output output.txt
19        '''
20
21        // Install project dependencies
22        sh '''
23        python -m pip install --upgrade pip
24        pip install -r requirements.txt
25        '''
26      }
27    }
28
29    stage('Test') {
30      steps {
31        parallel(
32          thread_1: {
33            // Execute test and generate report
34            sh 'pytest --junitxml=results/test-results-1.xml src/tests.py'
35
36            // Submit test results to TMS
37            sh '''
38            testit results upload \
39            --url <YOUR_INSTANCE_URL> \
40            --testrun-id "$(cat output.txt)" \
41            --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
42            --results results/test-results-1.xml
43            '''
44          }
45        )
46      }
47    }
48  }
49 }
```

```
45 },
46 thread_2: {
47 // Execute test and generate report
48 sh 'pytest --junitxml=results/test-results-2.xml src/tests.py'
49
50 // Submit test results to TMS
51 sh '''
52 testit results upload \
53 --url <YOUR_INSTANCE_URL> \
54 --testrun-id "$(cat output.txt)" \
55 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
56 --results results/test-results-2.xml
57 '''
58 }
59 ...
60 thread_n: {
61 // Execute test and generate report
62 sh 'pytest --junitxml=results/test-results-n.xml src/tests.py'
63
64 // Submit test results to TMS
65 sh '''
66 testit results upload \
67 --url <YOUR_INSTANCE_URL> \
68 --testrun-id "$(cat output.txt)" \
69 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
70 --results results/test-results-n.xml
71 '''
72 }
73 )
74 }
75 }
76 }
77 post {
78 always {
79 junit 'results/*.xml'
80 // Complete testrun
81 sh '''
82 testit testrun complete \
83 --url <YOUR_INSTANCE_URL> \
84 --testrun-id "$(cat output.txt)"
85 '''
86 }
87 }
88 }
```

# 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 ознакомиться в **официальной документации** .

Обновлено: 20.06.2025, 14:33:37



## 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):

```
1 import jetbrains.buildServer.configs.kotlin.*
2 import
3 jetbrains.buildServer.configs.kotlin.buildFeatures.parallelTests
4 import jetbrains.buildServer.configs.kotlin.buildFeatures.perfmon
5 import jetbrains.buildServer.configs.kotlin.buildSteps.script
6 import jetbrains.buildServer.configs.kotlin.triggers.vcs
7
8
9 version = "2022.10"
10
11 project {
12
13     buildType(Build)
14 }
15
16 object Build : BuildType({
17     name = "Build"
18
19     vcs {
20         root(DslContext.settingsRoot)
21     }
22
23     steps {
24         script {
25             name = "Install dependencies"
26             scriptContent = """
27             python -m pip install --upgrade pip
28             pip install -r requirements.txt
29             """.trimIndent()
30         }
31
32         script {
33             name = "Run tests"
34             scriptContent = """ pytest --junitxml=results/test-results.xml
35             src/tests.py """
36         }
37         script {
38             executionMode = BuildStep.ExecutionMode.ALWAYS
39             name = "Submit results"
40             scriptContent =
41             """
42             pip install testit-cli
43             testit --url <YOUR_INSTANCE_URL> \
44             --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
```

```
45 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
46 --testrun-name "Teamcity test run" \  
47 --results results/test-results.xml  
48 """.trimIndent()  
49 }  
50 }  
51  
52 triggers {  
53 vcs {  
54 }  
55 }  
56  
57 features {  
58 perfmon {  
59 }  
60 }  
    })
```

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 ознакомиться в [официальной документации](#) .

# 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:

```
1 | /ваш-проект text
2 | |-- .gitlab-ci.yml
3 | |-- common.properties
4 | |-- /jmeter_tests
5 | | |-- Login_Test.jmx
6 | | `-- Checkout_Process.jmx
7 | |-- /jmeter_plugins
8 | | `-- jmeter-junit-reporter-3.1.jar
9 | |-- /test_data
10 | | `-- users.csv
11 | `-- /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:

```
1 | /JmeterRep | text
2 | |-- /config # Конфигурационные файлы JMeter
3 | | |-- jmeter.properties # Основной файл конфигурации JMeter
4 | | |-- common.properties # Пользовательские свойства для тестов
5 | | |-- log4j2.xml # Конфигурация логирования
6 | | |-- user.properties # Дополнительные пользовательские настройки
7 | | `-- system.properties # Системные свойства JMeter
8 | |-- /extPlugins # Внешние плагины JMeter
9 | | |-- jmeter-plugins-*.jar # JAR-файлы плагинов
10 | | `-- ApacheJMeter_*.jar # Компоненты JMeter
11 | `-- /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:

```
1 | jmeter -p /path/to/JmeterRep/config/jmeter.properties | sh
```

2. Укажите путь к дополнительным плагинам через системное свойство:

```
1 | -Djmeter.ext.dirs=/path/to/JmeterRep/extPlugins | sh
```

3. **Опционально:** Вместо указания пути к дополнительным плагинам настройте переменные окружения:

```
1 | export | sh
2 | JMETER_PROPERTIES=/path/to/JmeterRep/config/jmeter.properties
   | 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:

```
1 | mklink /D "%JMeter_HOME%\lib\ext" " | cmd  
  | <YOUR_PROJECT_PATH>\extPlugins"
```

- Для конфигурационных файлов используются ссылки:

```
1 | mklink "%JMeter_HOME%\bin\jmeter.properties" " | cmd  
2 | <YOUR_PROJECT_PATH>\config\jmeter.properties"  
3 | 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 репозиториум

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 параметризовать 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:

```
1 variables: yml
2 HOST_VM_PERF: "your.performance.server.com" # Пример: IP-адрес
3 или домен сервера для performance тестирования
  JMETER_application: "cloud" # По умолчанию используется cloud
  версия
```

- Переменные для интеграции с Test IT (в шаблоне `.common_job`):

```
1 variables: yml
2 TESTIT_UPLOAD_URL: your_test_it_url # Пример: замените на ваш
3 адрес Test IT
4 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):

```
1 stages: yml
2 - prepare # Подготовительная стадия - загрузка данных,
3 настройка окружения
4 - perf_tests # Основные performance тесты (полные сценарии
5 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:

```
1 perf-tms-cloud-job:
2 extends:
3   - .before
4   - .common_job
5 stage: perf_tests
6 variables:
7   TEST_PLAN_FILE: "jmx/perfTests/tms.jmx"
8 rules:
9   - if: '$CI_PIPELINE_SOURCE == "trigger" && $JMETER_testName
10     == "tms" && $JMETER_application == "cloud"'
11   when: on_success
12     - when: manual
13   allow_failure: true
14 variables:
15   RUN_TYPE: "tst"
```

## 2. Сервисный тест:

```
1 perf-auth-cloud-job:
2 extends:
3   - .before
4   - .common_job
5 stage: service
6 variables:
7   TEST_PLAN_FILE: "jmx/service/auth.jmx"
8 rules:
9   - if: '$CI_PIPELINE_SOURCE == "trigger" && $JMETER_testName
10     == "auth" && $JMETER_application == "cloud"'
11   when: on_success
12     - when: manual
13   allow_failure: true
```

## 3. Enterprise-версия теста:

```

1  perf-tms-ent-job:
2  extends:
3  - .before
4  - .common_job
5  stage: perf_tests
6  variables:
7  TEST_PLAN_FILE: "jmx/perfTests/tms.jmx"
8  RUN_TYPE: "tst"
9  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
<code>TESTIT_PRIVATE_TOKEN</code>	Приватный токен для доступа к Test IT API	<code>your-secret-token</code>	Masked
<code>SSH_PRIVATE_KEY</code>	SSH ключ для доступа к performance серверу	<code>-----BEGIN OPENSSSH PRIVATE KEY--- --...</code>	Masked

- Переменные для настройки окружения:

Переменная	Описание	Значение по умолчанию
HOST_VM_PERF	IP-адрес performance сервера	your.performance.server.cor
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	00000000-0000-0000-0000-000000000000
TESTIT_CONFIGURATION_ID	ID конфигурации в Test IT	11111111-1111-1111-1111-111111111111
PUBLIC_REPORTS_URL	Базовый URL для HTML отчетов	http://your-reports-server.com/reports/
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:

```
1 | user.properties = ./config/common.properties | 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` :

```

1                                     properties
2  =====
3  # ОСНОВНЫЕ ПАРАМЕТРЫ ТЕСТА
4  #
5  =====
6  # Количество виртуальных пользователей (потоков)
7  threads=10
8  # Время разгона (в секундах)
9  rampup=5
10 # Продолжительность теста (в секундах)
11 duration=300
12
13 #
14 =====
15 # ПАРАМЕТРЫ ОКРУЖЕНИЯ
16 #
17 =====
   # 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`** : а. Скрипт определяет план тестирования для запуска на основе имени задания ( `$CI_JOB_NAME` ). б. Скрипт запускает JMeter с указанным планом тестирования и файлом свойств. в. JMeter выполняет тест и с помощью плагина JUnit Reporter создает файл `junit.xml` . д. 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:

```
1 # -n : запуск в неинтерактивном режиме (без GUI)
2 # -t : путь к файлу тестового плана (.jmx файл)
3 # -l : файл для сохранения результатов в формате JTL
4 # -e : генерация HTML отчета после выполнения
5 # -o : директория для HTML отчета
6 # -q : файл с дополнительными свойствами (наша конфигурация)
7
8 # Задержка только для ночных прогонов (nightly run)
9 if [ "$FINAL_RUN_TYPE" = "nr" ]; then
10 echo "Nightly run detected - sleeping for 900 seconds to avoid
11 peak hours"
12 sleep 900
13 else
14 echo "Non-nightly run - starting immediately (FINAL_RUN_TYPE:
15 $FINAL_RUN_TYPE)"
16 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:

```
1 | # Перейдите в директорию с Dockerfile | sh
2 | cd JMeter-TestIT-GitLab-Template
3 |
4 | # Соберите образ
5 | docker build -t your-registry/jmeter-testit:latest .
```

## 2. Загрузите Docker-образ в реестр:

```
1 | # Загрузите образ в ваш Docker реестр | sh
2 | docker push your-registry/jmeter-testit:latest
```

## 3. Обновите файл .gitlab-ci.yml:

```
1 | # Укажите ваш образ в секции image | yml
2 | .common_джоб:
3 |   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 \(развернуть / свернуть\)](#)

## Быстрый старт

1. Клонировать 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:

```
1 | # Ваши текущие переменные → Переменные шаблона                                yml
2 | HOST_VM_PERF → HOST_VM_PERF (без изменений)
3 | JMETER_application → JMETER_application (без изменений)
```

### Шаг 2: Адаптация задач (jobs)

1. Замените `extends` в ваших задачах:

```
1 # Было:
2 your-job:
3 extends: .your_template
4
5 # Стало:
6 your-job:
7 extends:
8 - .before
9 - .common_job
```

yml

## 2. Обновите переменные задач (jobs):

```
1 # Добавьте переменную TEST_PLAN_FILE если ее нет:
2 variables:
3 TEST_PLAN_FILE: "jmx/perfTests/your-test.jmx"
```

yml

## Шаг 3: Настройка интеграции с Test IT

### 1. Добавьте переменные Test IT в .common\_job:

```
1 variables:
2 TESTIT_UPLOAD_URL: your_test_it_url # Пример адреса Test IT
3 TESTIT_PROJECT_ID: your-project-id # Пример ID проекта
4 (получить можно через Chrome DevTools)
5 TESTIT_CONFIGURATION_ID: your-configuration-id # Пример ID
6 конфигурации (получить можно через Chrome DevTools)
```

yml

### 2. Настройте правила запуска вебхуков:

```
1 rules:
2 - if: '$CI_PIPELINE_SOURCE == "trigger" && $JMETER_testName ==
3 "your-test-name"'
4 when: on_success
5 - when: manual
6 allow_failure: true
```

yml

## 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:

```
1 | └─ jmx/
2 |   └─ perfTests/ # Основные performance-тесты
3 |   └─ service/ # Сервисные тесты
4 |   └─ custom/ # Кастомные тесты
5 | └─ testData/ # Тестовые данные
6 | └─ plugins/ # JMeter-плагины
```

text

## 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 <code>.jmx</code> -файла. Например, для файла <code>Login_Test.jmx</code> параметр должен быть <code>Login_Test</code> .	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:

```
1      {
2      "ref": "main",
3      "token": "YOUR_GITLAB_TOKEN",
4      "variables": {
5      "TESTIT_PROJECT_ID": "$PROJECT_ID",
6      "TESTIT_CONFIGURATION_ID": "$CONFIGURATION_IDS",
7      "TESTIT_TEST_RUN_ID": "$TEST_RUN_ID",
8      "JMETER_loopCount": "$CONFIGURATIONS_PARAMETERS[loopCount]",
9      "JMETER_usrNum": "$CONFIGURATIONS_PARAMETERS[usrNum]",
10     "JMETER_host": "$CONFIGURATIONS_PARAMETERS[host]",
11     "JMETER_application":
12     "$CONFIGURATIONS_PARAMETERS[application]",
13     "JMETER_testName": "$CONFIGURATIONS_PARAMETERS[testName]",
14     "runType": "$CONFIGURATIONS_PARAMETERS[runType]"
15     }
16     }
```

json

**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:** <https://docs.testit.software/user-guide/integrations/cli.html>
- **Документация Test IT:** <https://docs.testit.software/>

# 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

# 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

### 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` — уникальный ID автотеста в Test IT
- `title` — имя автотеста, указанное в его карточке. Если имя не указано, используется имя `displayName`.
- `description` — описание автотеста, указанное в карточке автотеста
- `labels` — 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
- `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

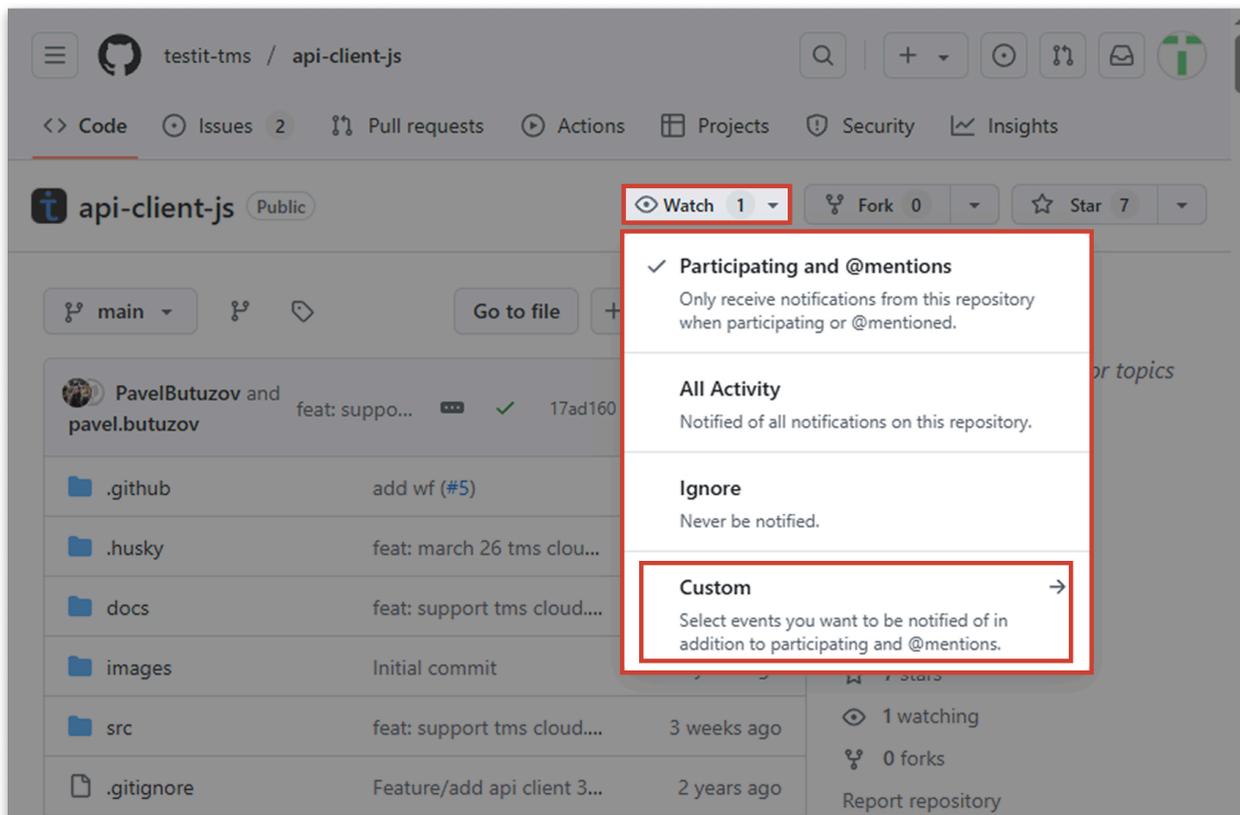
---

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 Management для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 28.06.2025, 21:17:11



# 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:

```
1 | $ java org.testng.TestNG testng1.xml | sh
```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Appium test run" \
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# Behave

Test IT позволяет загрузить 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:

```
1 | $ behave --junit
```

sh

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit -h
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit \
3 | --url <YOUR_INSTANCE_URL>\
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Behave test run" \
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ java org.testng.TestNG testng1.xml | sh
```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "BrowserStack test run" \
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



## 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:

```
1 | $ test --reporter junit --out test-results.xml
```

sh

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:

```
1 | $ pip install testit-cli
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN>
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Catch test run" \
7 | --results test-results.xml
```

sh

- `<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

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

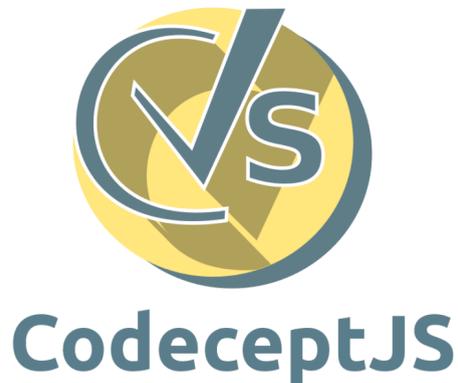
## Дополнительные ресурсы

---

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 26.06.2025, 21:57:55



# 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:

```
1 | $ npm install mocha-junit-reporter | sh
```

2. Для подключения пакета к проекту добавьте его в `codecept.conf.js` :

```
1 | "mocha": { | conf
2 | "reporterOptions": {
3 | "mochaFile": "output/result.xml"
4 | }
5 | },
```

3. Для генерации отчета воспользуйтесь командой:

```
1 | $ codeceptjs run --reporter mocha-junit-reporter | sh
```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения `TMS_TOKEN` и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "CodeceptJS test run" \
7 --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



## 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:

```
1 | $ cucumber --format junit --out results/test-results.xml sh
```

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:

```
1 | $ pip install testit-cli sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Cucumber test run" \
7 | --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

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**

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# Cypress

Test IT позволяет загружать результаты автоматизированных тестов из тестового фреймворка Cypress. 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. Cypress поддерживает генерацию JUnit XML отчета. Подробнее можно прочитать в [официальной документации](#) .

1. Для подключения функции необходимо указать ее в `cypress.config.js` :

```
1 reporter: 'junit',
2 reporterOptions: {
3 mochaFile: 'results/my-test-output-[hash].xml',
4 },
```

conf

2. Для генерации отчета воспользуйтесь командой:

```
1 $ cypress run
```

sh

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:

```
1 $ pip install testit-cli
2 $ testit --help
3
4 usage: testit [options]
5 [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "Cypress test run" \
7 --results results
```

sh

- `<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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



## 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:

```
1  ./emceeFree runTests \
2  --queue "ssh://emcee:qwerty@ios-build-machine77" \
3  --worker "ssh://emcee:qwerty@ios-build-machine77" \
4  --worker "ssh://emcee:qwerty@ios-build-machine78" \
5  --device "com.apple.CoreSimulator.SimDeviceType.iPhone-14" \
6  --runtime "com.apple.CoreSimulator.SimRuntime.iOS-16-4" \
7  --test-bundle derivedData/Build/Products/Debug-
8  iphonesimulator/EmceeSampleTestsWithoutHost.xctest \
   --junit test-output/tests_without_host_junit.xml
```

sh

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:

```
1  $ pip install testit-cli
2  $ testit --help
3
4  usage: testit [options]
5  [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "Emcee test run" \
7 --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



## 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](#)

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 Management для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 27.06.2025, 23:18:40



# 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:

```
1 | $ go install gotest.tools/gotestsum@latest | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ gotestsum --junitfile results/unit-tests.xml | sh
```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Golang test run" \
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



# 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:

```
1 | $ test --gtest_output="xml:results/test-results.xml" sh
```

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:

```
1 | $ pip install testit-cli sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "GoogleTest test run" \
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 26.06.2025, 21:57:55



# 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:

```
1 | $ npm i jasmine-reporters
```

sh

2. Для подключения пакета к проекту зарегистрируйте его в Jasmine:

```
1 | var path = require('path');
2 | var reporters = require('jasmine-reporters');
3 | var junitReporter = new reporters.JUnitXmlReporter({
4 |   savePath: path.join(__dirname, '/results'),
5 |   consolidateAll: false
6 | });
7 | jasmine.getEnv().addReporter(junitReporter)
```

js

3. Для генерации отчета воспользуйтесь командой:

```
1 | $ npx jasmine
```

sh

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:

```
1 | $ pip install testit-cli
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "Jasmine test run" \
7 --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 <dependency>
2 <groupId>ru.testit</groupId>
3 <artifactId>jbehave-reporter-junitxml</artifactId>
4 <version>LATEST_VERSION</version>
5 <scope>compile</scope>
6 </dependency>
```

xml

◦ Gradle:

```
1 implementation "ru.testit:jbehave-reporter-
  junitxml:LATEST_VERSION"
```

groovy

2. Добавьте его в конфигурацию JBehave:

```
1 import org.jbehave.core.ConfigurableEmbedder;
2 import org.jbehave.core.configuration.Configuration;
3 import org.jbehave.core.configuration.MostUsefulConfiguration;
4 import org.jbehave.core.embedder.Embedder;
5 import org.jbehave.core.embedder.EmbedderControls;
6 import org.jbehave.core.embedder.NullEmbedderMonitor;
7 import org.jbehave.core.io.LoadFromClasspath;
8 import org.jbehave.core.reporters.NullStoryReporter;
9 import org.jbehave.core.reporters.StoryReporterBuilder;
10 import org.jbehave.core.steps.InjectableStepsFactory;
11 import org.jbehave.core.steps.InstanceStepsFactory;
12 import org.junit.jupiter.api.Test;
13 import org.junit.jupiter.api.io.TempDir;
14 import ru.testit.reporter.JunitXmlReporter;
15
16 import java.io.File;
17 import java.nio.file.Path;
18 import java.util.ArrayList;
19 import java.util.List;
20
21 public class TestStoriesRunner extends ConfigurableEmbedder {
22     public Embedder embedder;
23     @TempDir
24     Path temp;
25
26     @Override
27     @Test
28     public void run() {
29         embedder = new Embedder();
30         embedder.useEmbedderMonitor(new NullEmbedderMonitor());
31         embedder.useEmbedderControls(new EmbedderControls()
32             .doGenerateViewAfterStories(false)
33             .doFailOnStoryTimeout(false)
34             .doBatch(false)
35             .doIgnoreFailureInStories(true)
36             .doIgnoreFailureInView(true)
37             .doVerboseFailures(false)
38             .doVerboseFiltering(false)
39         );
40         embedder.useConfiguration(configuration());
41         embedder.useCandidateSteps(stepsFactory().createCandidateSteps());
42
43         File dir = new File("./src/test/resources/stories");
44         List<String> stories = new ArrayList<>();
```

```

45
46 for (File file : dir.listFiles()){
47     if ( file.isFile() )
48         stories.add("stories/" + file.getName());
49     }
50
51     embedder.runStoriesAsPaths(stories);
52 }
53
54 public Configuration configuration() {
55     final JunitXmlReporter reporter = new JunitXmlReporter();
56
57     return new MostUsefulConfiguration()
58         .useStoryLoader(new LoadFromClasspath(this.getClass()))
59         .useStoryReporterBuilder(
60             new TestStoryReporterBuilder(temp.toFile())
61             .withReporters(reporter))
62         .useDefaultStoryReporter(new NullStoryReporter());
63 }
64
65 public InjectableStepsFactory stepsFactory() {
66     return new InstanceStepsFactory(embedder.configuration(), new
67         SampleSteps());
68 }
69
70 static class TestStoryReporterBuilder extends StoryReporterBuilder
71 {
72
73     private final File outputDirectory;
74
75     TestStoryReporterBuilder(final File outputDirectory) {
76         this.outputDirectory = outputDirectory;
77     }
78
79     @Override
80     public File outputDirectory() {
81         return outputDirectory;
82     }
83 }

```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL>\
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "JBehave test run" \
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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` :

```
1 module.exports = {
2   reporters: [
3     "default",
4     ["jest-junit", {
5       outputFile: "results/test-results.xml",
6       ancestorSeparator: " > " ,
7     }]
8   ],
9   };
```

js

2. Для генерации отчета воспользуйтесь командой:

```
1 $ jest
```

sh

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:

```
1 $ pip install testit-cli
2 $ testit --help
3
4 usage: testit [options]
5 [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "Jest test run" \
7 --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



## 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:

```
1  $ pip install testit-cli
2  $ testit --help
3
4  usage: testit [options]
5  [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1  $ export TMS_TOKEN=<YOUR_TOKEN>
2  $ testit results import \
3  --url <YOUR_INSTANCE_URL> \
4  --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5  --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6  --testrun-name "Junit test run" \
7  --results test-output/test/
```

sh

- **<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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



## 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](#)

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 Management для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 27.06.2025, 23:18:40



# 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:

```
1 | marathon-cloud run android | sh
2 | --application=app.apk
3 | --test-application=appTest.apk
4 | --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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Marathon Cloud test run" \
7 | --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

## 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 Management 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:

```
1 | $ gem install minitest-reporters | sh
```

2. Добавьте в `test_helper.rb` следующие строки:

```
1 | require "minitest/reporters" | rb  
2 | Minitest::Reporters.use! Minitest::Reporters::JUnitReporter.new
```

3. Для генерации отчета воспользуйтесь командой:

```
1 | $ rake test | sh
```

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:

```
1 | $ pip install testit-cli | sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "Minitest test run" \  
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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` :

```
1  "use strict";
2
3  module.exports = {
4    reporter: 'mocha-junit-reporter',
5    reporterOptions: [
6      "mochaFile=results/test-results.xml",
7      "jenkinsMode=1",
8      "testCaseSwitchClassnameAndName=1",
9      "outputs=1"
10   ]
11 }
```

conf

2. Для генерации отчета воспользуйтесь командой:

```
1  $ npx mocha
```

sh

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:

```
1  $ pip install testit-cli
2  $ testit --help
3
4  usage: testit [options]
5  [...]
```

sh

2. Добавьте переменную окружения `TMS_TOKEN` и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "Mocha test run" \
7 --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



# 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:

```
1 | $ dotnet add package JunitXml.TestLogger | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ dotnet test --logger:"junit;LogFilePath=results/test- | sh  
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:

```
1 | $ pip install testit-cli | sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "MSTest test run" \  
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40

# nose

## 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` :

```
1 [unittest] conf
2 plugins = nose2.plugins.junitxml
```

2. Для генерации отчета воспользуйтесь командой:

```
1 $ nose2 --junit-xml sh
```

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:

```
1 $ pip install testit-cli sh
2 $ testit --help
3
4 usage: testit [options]
5 [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "Nose2 test run" \
7 --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



## 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:

```
1 | $ dotnet add package JunitXml.TestLogger | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ dotnet test --logger:"junit;LogFilePath=results/test- | sh  
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:

```
1 | $ pip install testit-cli | sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "NUnit test run" \  
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



## 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:

```
1 | $ phpunit --log-junit results/test-results.xml
```

sh

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:

```
1 | $ pip install testit-cli
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN>
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "PHPunit test run" \
7 | --results results/test-results.xml
```

sh

- `<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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 26.06.2025, 21:57:55



# 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` :

```
1 | "reporter: [['junit', {                                     conf
2 |   outputFile: 'results/test-results.xml',
3 |   embedAnnotationsAsProperties: true,
4 | }]],
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ npx playwright test                                     sh
```

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:

```
1 $ pip install testit-cli
2 $ testit --help
3
4 usage: testit [options]
5 [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "Playwright test run" \
7 --results results
```

sh

- **<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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ newman run collection.json -r cli,junit --reporter-junit-export results/test-results.xml | sh
```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Postman test run" \
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ pytest --junit-xml=reports/results.xml sh
```

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:

```
1 | $ pip install testit-cli sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Pytest test run" \
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



## 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:

```
1 | $ pytest --junit-xml=reports/results.xml
```

sh

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:

```
1 | $ pip install testit-cli
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN>
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Pytest-BDD test run" \
7 | --results reports
```

sh

- `<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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 26.06.2025, 21:57:55



# 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:

```
1 | $ robot --xunit reports/test-results.xml examples sh
```

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:

```
1 | $ pip install testit-cli sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL>\
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Robot Framework test run" \
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ gem install rspec_junit_formatter sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ rspec --format RspecJunitFormatter --out results/test-  
results.xml sh
```

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:

```
1 | $ pip install testit-cli sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "Rspec test run" \  
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ npm install mocha-junit-reporter | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ npx mocha --reporter node_modules/mocha-junit-reporter -- | sh  
  | 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:

```
1 | $ pip install testit-cli | sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "Sauce Labs test run" \  
7 | --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

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:

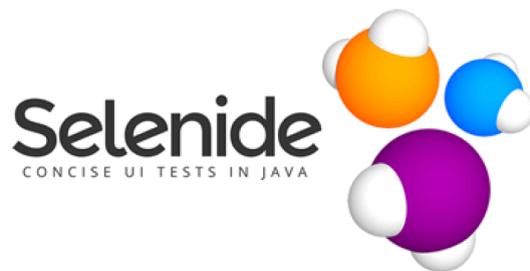
1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

## 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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ java org.testng.TestNG testng1.xml | sh
```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Selenide test run" \
7 | --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

## 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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ npm install mocha-junit-reporter | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ npx mocha --reporter node_modules/mocha-junit-reporter -- | sh  
  | 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:

```
1 | $ pip install testit-cli | sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "Selenium test run" \  
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# SoapUI

Test IT позволяет загружать результаты автоматизированных тестов с использованием SoapUI. Существуют следующие способы интеграции:

- CLI
- Public API

Смотрите также: [Дополнительные ресурсы](#).

## CLI

---

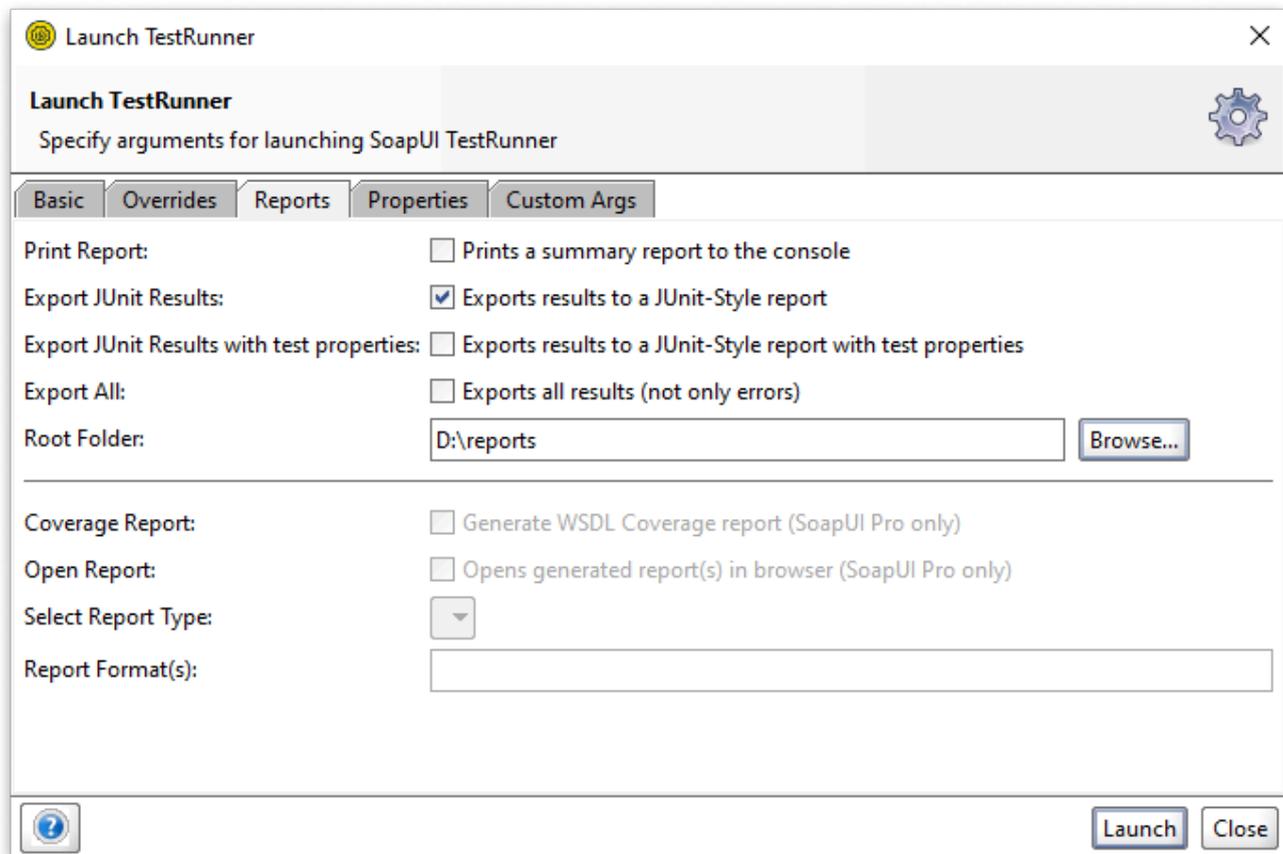
Test IT поддерживает загрузку тестов и результатов их выполнения в формате JUnit XML. Для этого необходимо:

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:

```
1 $ pip install testit-cli
2 $ testit --help
3
4 usage: testit [options]
5 [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL> \
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "SoapUI test run" \
7 --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ dotnet add package JunitXml.TestLogger | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ dotnet test --logger:"junit;LogFilePath=results/test- | sh  
  | 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:

```
1 | $ pip install testit-cli | sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "SpecFlow test run" \  
7 | --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

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

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 Management для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 27.06.2025, 23:18:40



Telerik

# Test Studio

## 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:

```
1 | $ Telerik.ApiTesting.Runner test <YOUR_PROJECT_PATH> -f junit sh
```

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:

```
1 | $ pip install testit-cli sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Telerik test run" \
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ npm install testcafe-reporter-junit sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ npx testcafe chrome tests --reporter junit:reports/report.xml sh
```

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:

```
1 | $ pip install testit-cli sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "TestCafe test run" \
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



# 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:

```
1 | $ TestComplete.exe <YOUR_PROJECT_PATH> /run /p: sh
   | <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:

```
1 | $ pip install testit-cli sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "TestComplete test run" \
7 | --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

Результаты загружены в систему Test IT.

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 26.06.2025, 21:57:55



## 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:

```
1 | $ testim --token <YOUR_ACCESS_TOKEN> --project                                sh
   | <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:

```
1 | $ pip install testit-cli                                                    sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN>                                           sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "Testim test run" \
7 | --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 26.06.2025, 21:57:55

# TestNG

## 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:

```
1 | $ java org.testng.TestNG testng1.xml
```

sh

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:

```
1 | $ pip install testit-cli
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

sh

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN>
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "TestNG test run" \
7 | --results test-output/test/
```

sh

- `<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

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

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

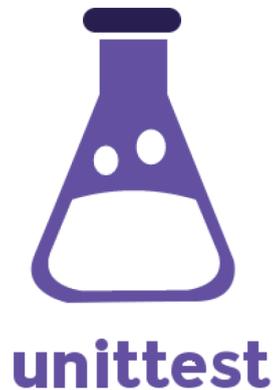
## Дополнительные ресурсы

---

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



# Unitest

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:

```
1 | $ pip install unittest-xml-reporting sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ python -m xmlrunner discover -t ~/mycode/tests -o /junit- sh  
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:

```
1 | $ pip install testit-cli sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "Unittest test run" \  
7 | --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

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:

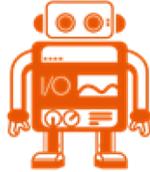
1. Самостоятельно взаимодействовать с API
2. Использовать готовый API-клиент

## 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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



WEBDRIVER 

# 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:

```
1 | $ npm install @wdio/junit-reporter --save-dev
```

sh

2. Для подключения пакета к проекту добавьте его в `wdio.conf.js` :

```
1 | module.exports = {
2 | // ...
3 | reporters: [
4 |   'dot',
5 |   ['junit', {
6 |     outputDir: './results/',
7 |     outputFileFormat: function(options) {
8 |       return 'test-results.xml';
9 |     }
10 |   }],
11 | ],
12 | // ...
13 | };
```

conf

3. Для генерации отчета воспользуйтесь командой:

```
1 | $ npx wdio run wdio.conf.js
```

sh

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:

```
1 | $ pip install testit-cli
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

sh

2. Добавьте переменную окружения `TMS_TOKEN` и выполните команду:

```
1 $ export TMS_TOKEN=<YOUR_TOKEN>
2 $ testit results import \
3 --url <YOUR_INSTANCE_URL>\
4 --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 --testrun-name "WebdriverIO test run" \
7 --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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)



# 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:

```
1 | $ brew install xcbeautify | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ set -o pipefail && xcodebuild [flags] test | xcbeautify -- report junit | sh
```

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:

```
1 | $ pip install testit-cli | sh
2 | $ testit --help
3 |
4 | usage: testit [options]
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh
2 | $ testit results import \
3 | --url <YOUR_INSTANCE_URL> \
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \
6 | --testrun-name "XCTest test run" \
7 | --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

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

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 Management для IntelliJ IDEA](#)
- [Взаимодействие с REST API](#)

Обновлено: 27.06.2025, 23:18:40



## 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:

```
1 | $ dotnet add package JunitXml.TestLogger | sh
```

2. Для генерации отчета воспользуйтесь командой:

```
1 | $ dotnet test --logger:"junit;LogFilePath=results/test- | sh  
  | 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:

```
1 | $ pip install testit-cli | sh  
2 | $ testit --help  
3 |  
4 | usage: testit [options]  
5 | [...]
```

2. Добавьте переменную окружения **TMS\_TOKEN** и выполните команду:

```
1 | $ export TMS_TOKEN=<YOUR_TOKEN> | sh  
2 | $ testit results import \  
3 | --url <YOUR_INSTANCE_URL> \  
4 | --project-id 5236eb3f-7c05-46f9-a609-dc0278896464 \  
5 | --configuration-id 15dbb164-c1aa-4cbf-830c-8c01ae14f4fb \  
6 | --testrun-name "xUnit test run" \  
7 | --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

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

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 Management для IntelliJ IDEA
- Взаимодействие с REST API

Обновлено: 27.06.2025, 23:18:40

# 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.

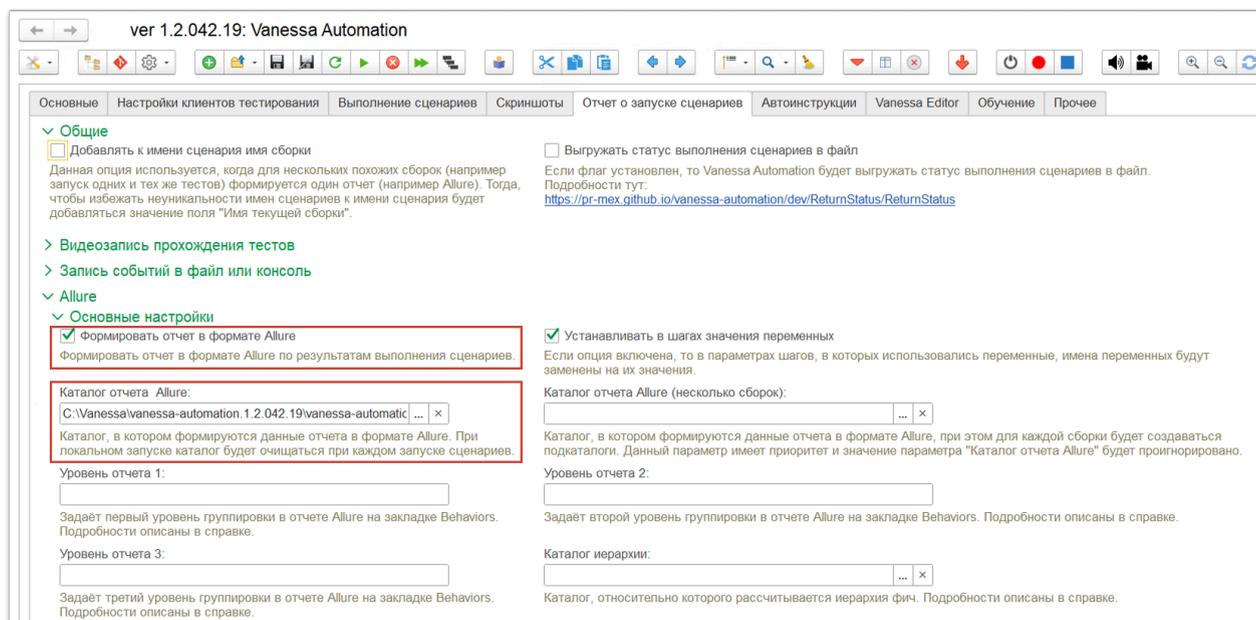
Настройка интеграции включает:

- Настройку создания отчетов Allure
- Установку и настройку Python
- Установку и настройку импортера Allure

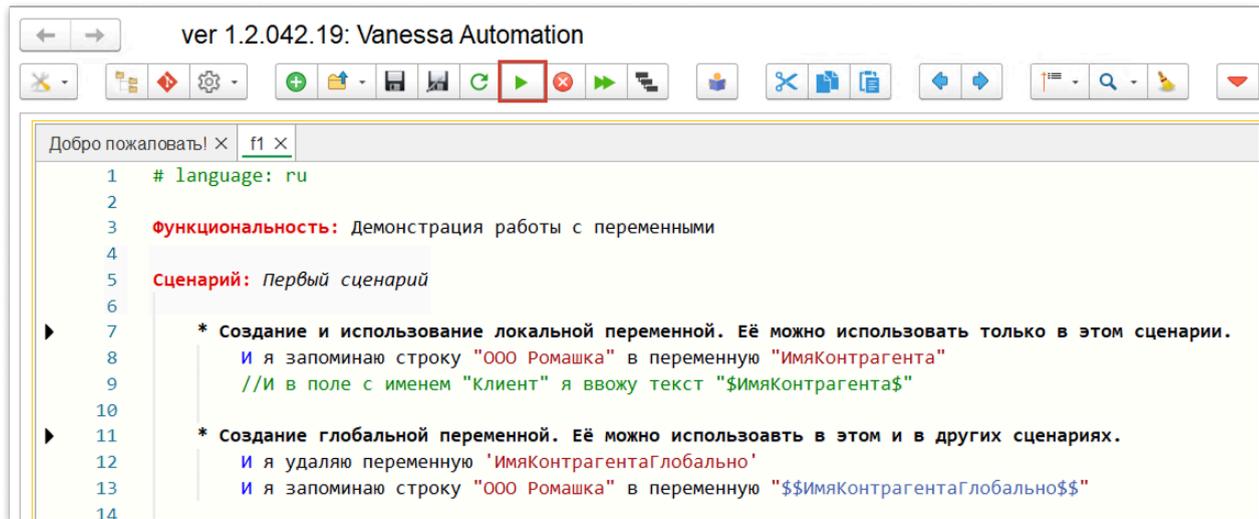
## Настройка создания отчетов Allure

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



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 совместимую версию импортера

Выберите версию импортера, совместимую с вашей версией Test IT. Информация о совместимости версий доступна на странице [GitHub](#) Test IT:

- 📦 Для **Test IT Enterprise** смотрите номер версии.
- ☁ Для **Test IT Lite** и **Standard** 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:

```
1 | pip install testit-importer-allure==1.14.4 | sh
```

3. Настройте конфигурацию для загрузки документов:

```
1 | testit --url <url> | sh
2 | testit --privatetoken <token>
3 | testit --projectid <id>
4 | testit --configurationid <id>
```

4. **Опционально:** Для доступа к полному списку параметров используйте команду:

```
1 | pip install testit-importer-allure==1.14.4 | sh
```

## 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:

```
1 | testit --resultsdir %Путь_к_папке_с_отчетами% | sh
```

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 **Прогон** появится новый прогон, содержащий тестовые сценарии (если тесты запускались без параметра `--testrunid` ).

# Устранение ошибки сертификата

---

При возникновении ошибки сертификата:

```
1 | certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1077)'')) sh
```

- Установите пакет `pip-system-certs`, используя команду:

```
1 | pip install pip-system-certs sh
```

## 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 Management для систем IDE](#)
- [Примеры интеграции с тестовыми фреймворками](#)
- [Взаимодействие с REST API](#)

# Что такое Test IT Management?

📄 **Пользователям Test IT Enterprise: требуется версия 5.0 или более поздняя**

Плагин **Test IT Management 1.0.9** совместим с версией **Test IT 5.0** и выше.

Актуальная версия Test IT Enterprise доступна на [официальном сайте](#) .

- Для этого раздела мы подготовили видео:

Плагин Test IT Management — это утилита для интеграции ручных тестов (**рабочих элементов Test IT**) в код среды разработки.

Плагин Test IT Management 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)

## 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: 25.11.2025, 14:40:11

# Test IT Management для JetBrains IDE

 **Пользователям Test IT Enterprise: требуется версия 5.0 или более поздняя**

Плагин **Test IT Management 1.0.9** совместим с версией **Test IT 5.0** и выше. Актуальная версия Test IT Enterprise доступна на [официальном сайте](#) .

Плагин Test IT Management 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

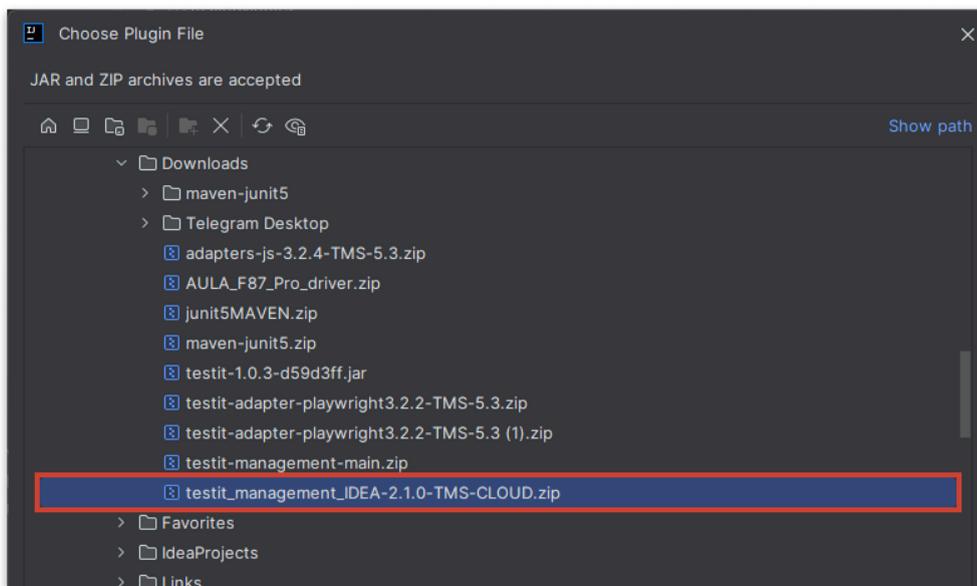
## Установка

Установка плагина для различных продуктов JetBrains 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.

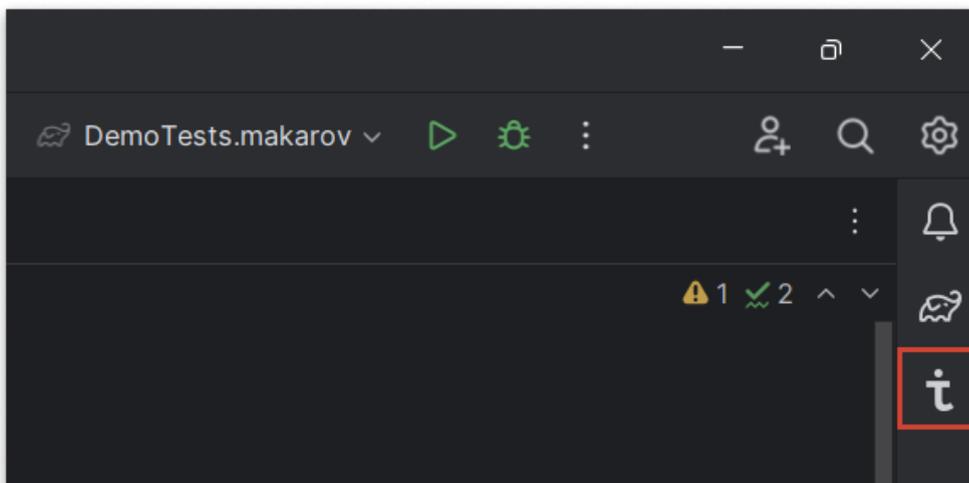
### IntelliJ IDEA и OpenIDE

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**.

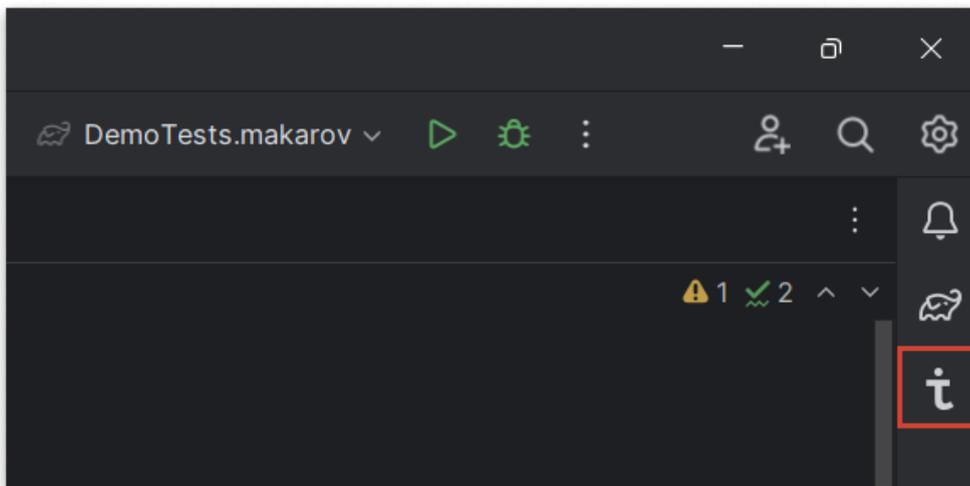


2. Установите плагин, следуя инструкции из [документации IntelliJ IDEA](#) . После установки значок плагина появится в навигационной панели вашей системы IDE.



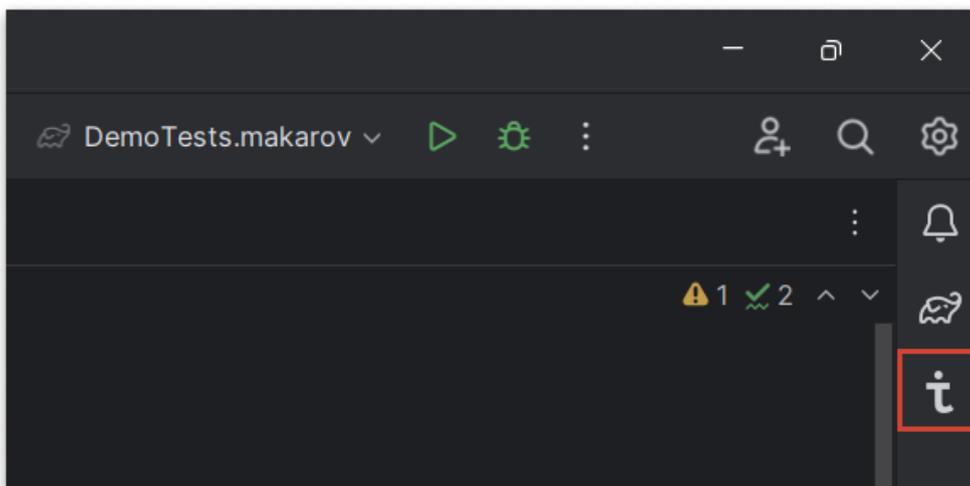
## PyCharm

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.



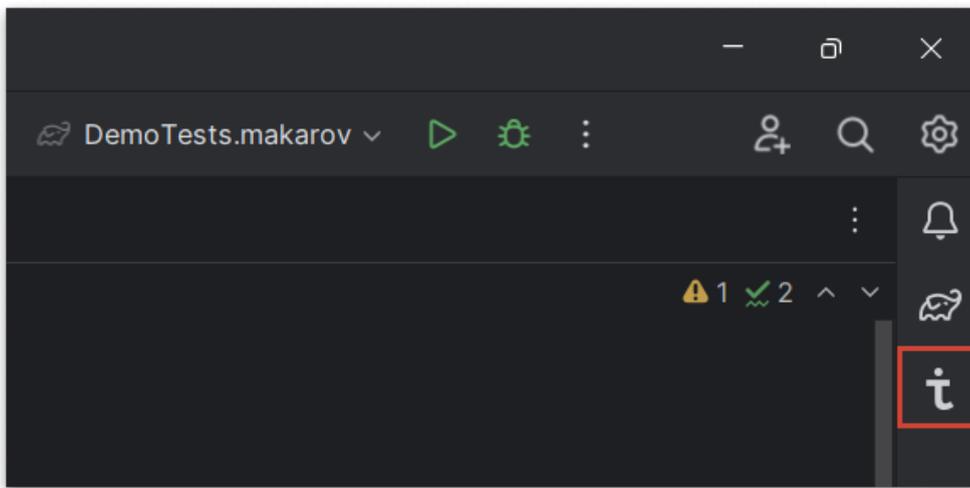
## Rider

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.



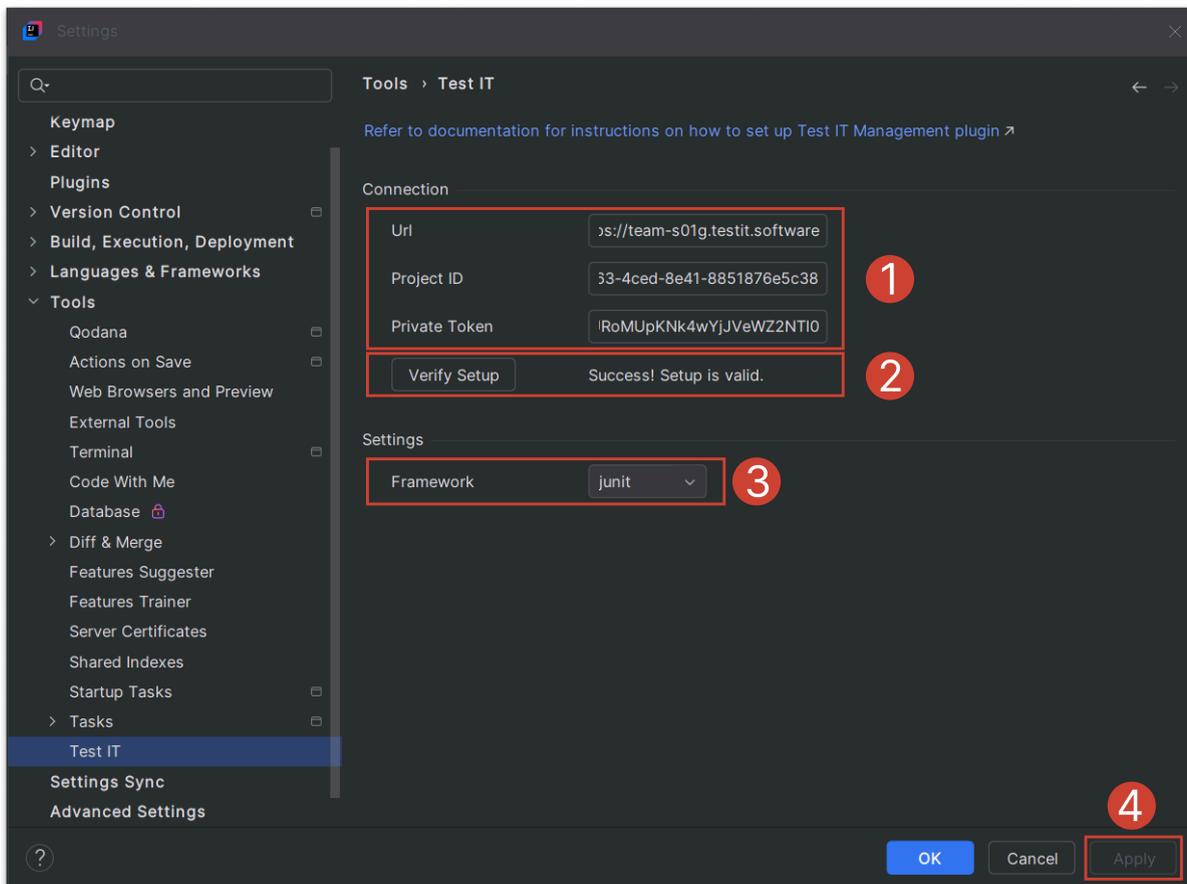
## WebStorm

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.



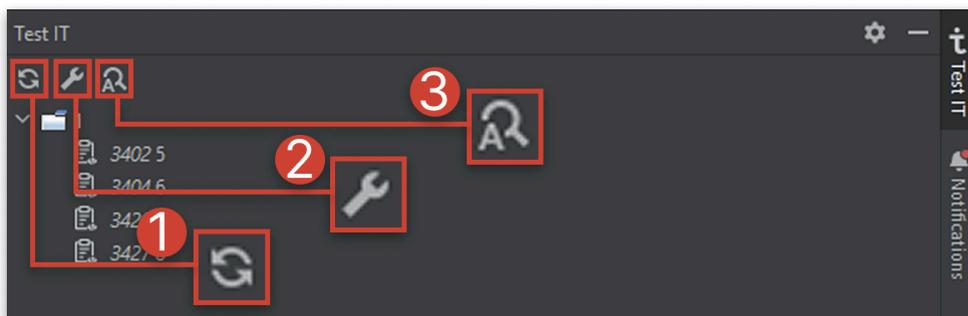
## Настройка

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

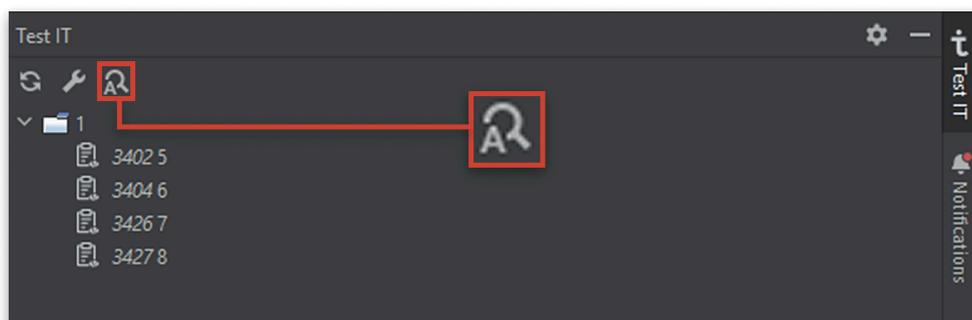
Один из способов адаптации ваших автотестов для работы с Test IT — замена декораторов Allure на декораторы Test IT. Опция доступна для работы с фреймворками:

- Behave
- JUnit
- Playwright
- Pytest
- Robot Framework

Подробную инструкцию смотрите на [GitHub Test IT](#).

Для замены декораторов:

1. В проекте нажмите значок **Search Allure Methods**.



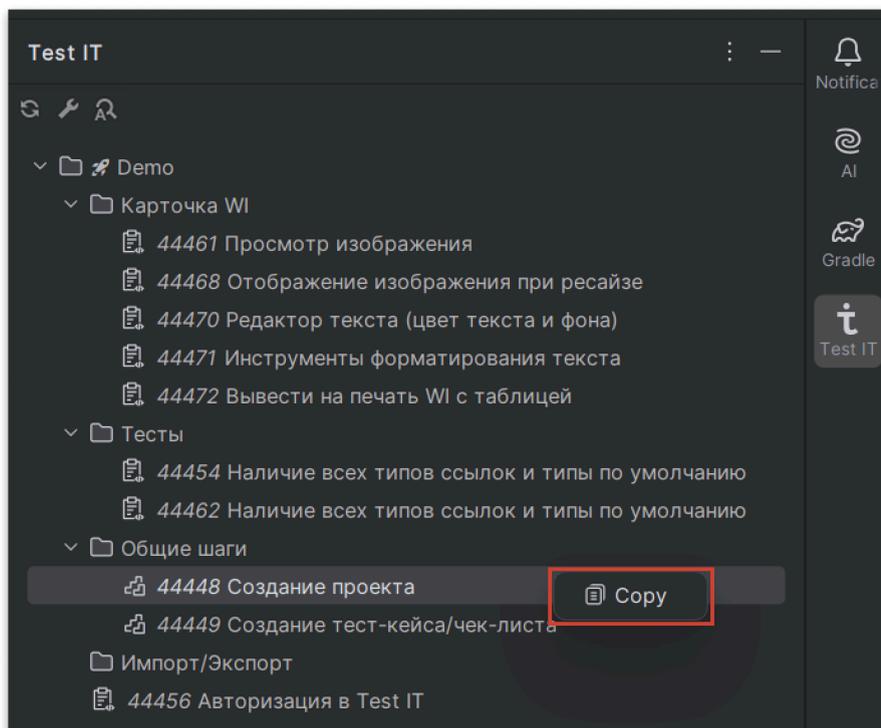
Откроется меню найденных декораторов.

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 фрагмент кода ручного теста, отображенный в системе IDE. Шаблоны используются для последующего написания автотестов. Чтобы получить фрагмент 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 snippets:*

```
3
4 > public class DemoTests {
5
6     @WorkItemIds("44456")
7     @Test
8 >     public void авторизацияВTestIT() {
9         // See work item [44456] for detailed steps description
10        // Pre:
11        //     Создание проекта
12        // Steps:
13        //     Перейти на страницу авторизации Test IT
14        //     Ввести %Login
15        //     Ввести %Password
16        //     Нажать кнопку Войти
17        // Post:
18        //     Удалить локального пользователя
19    }
20 }
21 |
```

## 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.

```
6 public class DemoTests {
7     @WorkItemIds("30271")
8     @Test
9     public void makarov() {
10         // See work item [30271] for detailed steps
11         // Pre:
12         // Steps:
13         // Post:
14     }
15 }
```

4110 Параметры тест  
4398 add text like attachment without file name  
30271 makarov  
4400 3 is 2  
4386 title annotation failed  
3615 without annotations failed  
4374 correctly sum 1,2 to 3  
3627 displayName Annotation success  
4378 add some paths attachments

## 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: 04.12.2025, 18:26:41

# Test IT Management для Visual Studio Code

 **Пользователям Test IT Enterprise: требуется версия 5.0 или более поздняя**

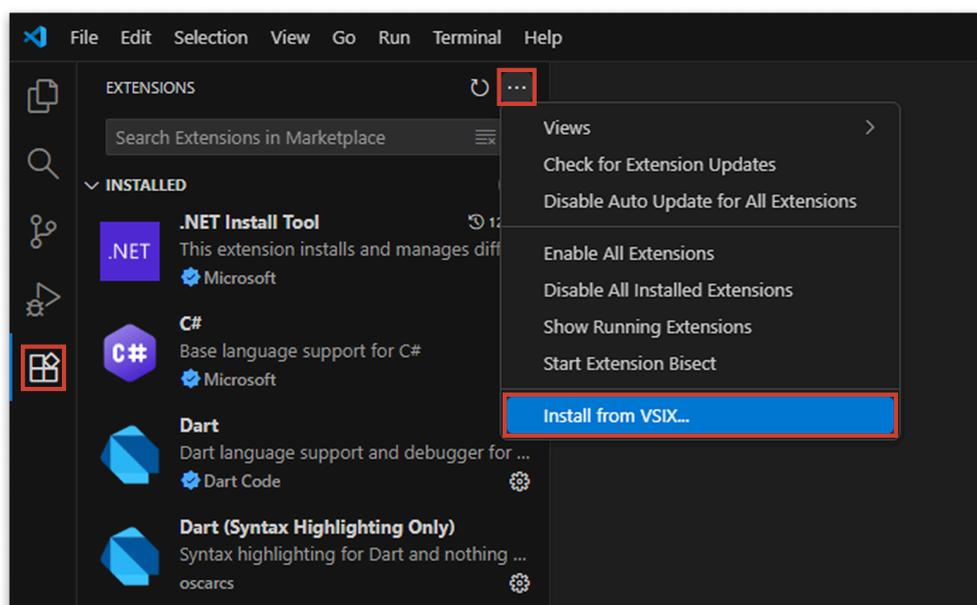
Плагин **Test IT Management 1.0.0** совместим с версией **Test IT 5.0** и выше.  
Актуальная версия Test IT Enterprise доступна на [официальном сайте](#) .

Плагин Test IT Management 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

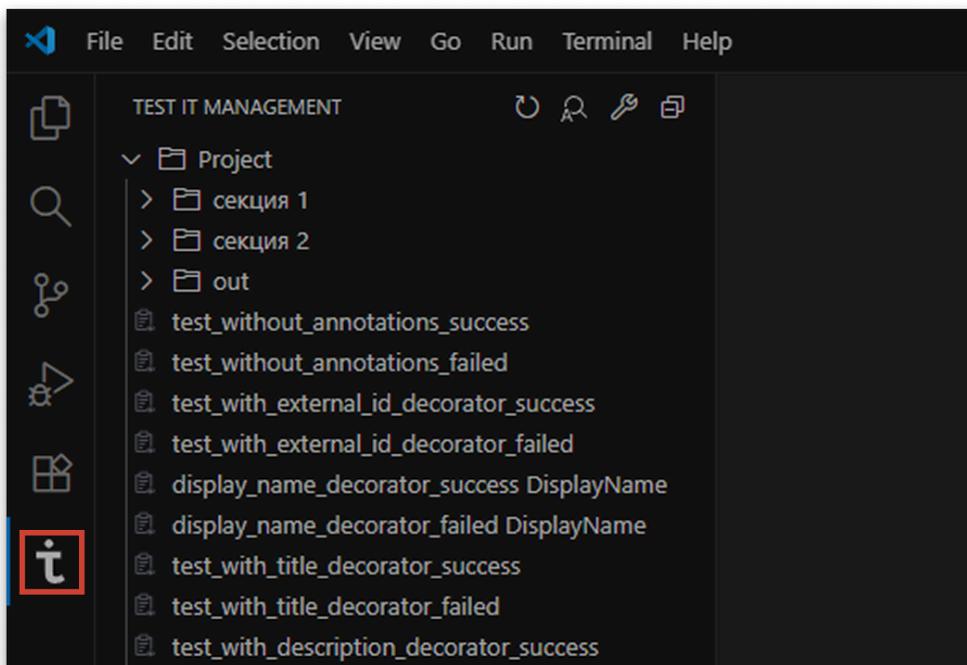
## Установка

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

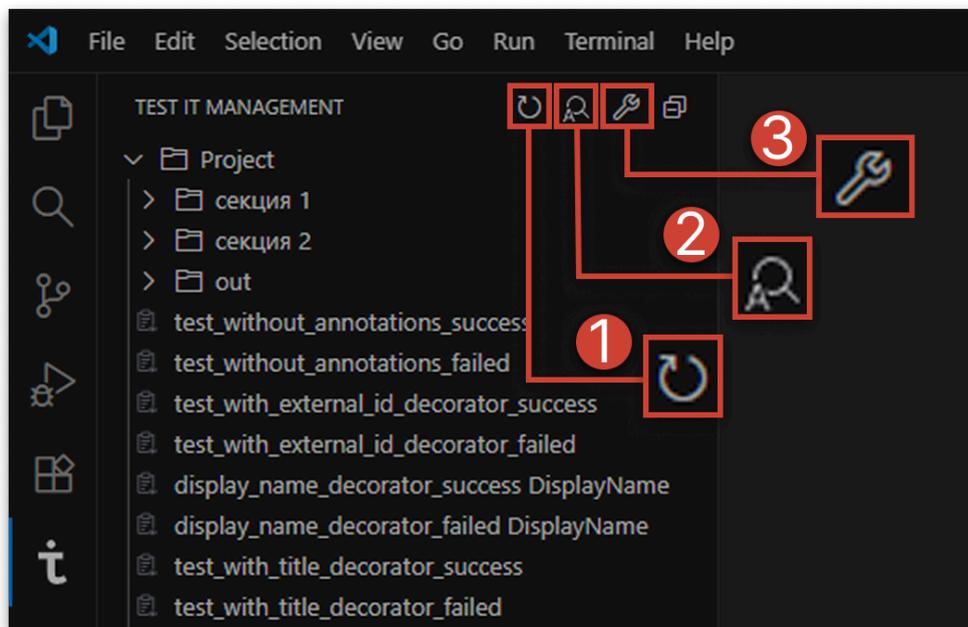
---

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.

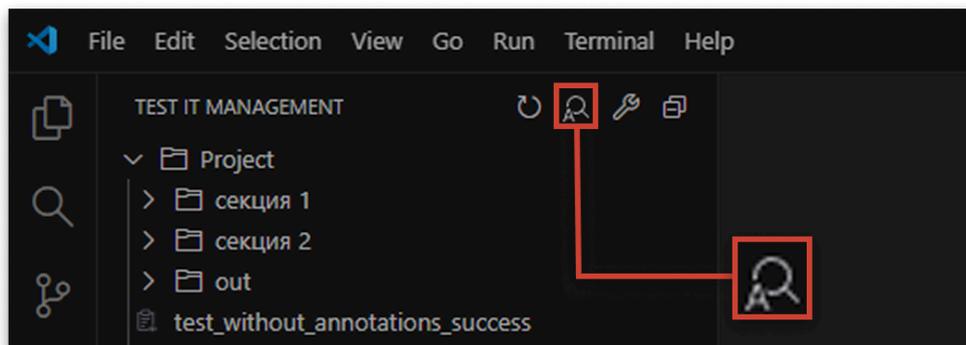


## Замена декораторов Allure

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](#).

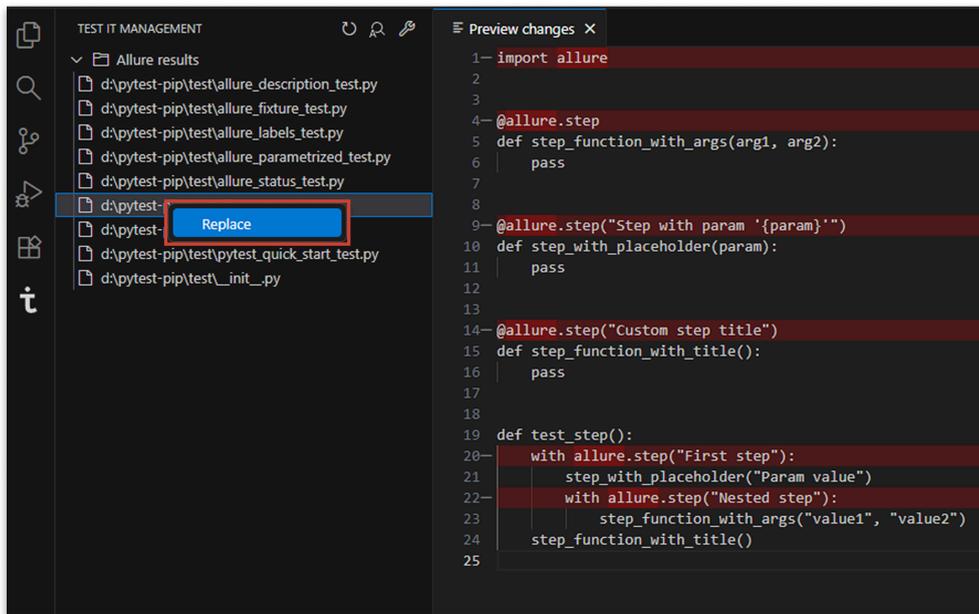
Для замены декораторов:

1. В проекте нажмите значок **Search Allure Methods**.



Откроется меню найденных декораторов и окно просмотра изменений **Preview Changes**.

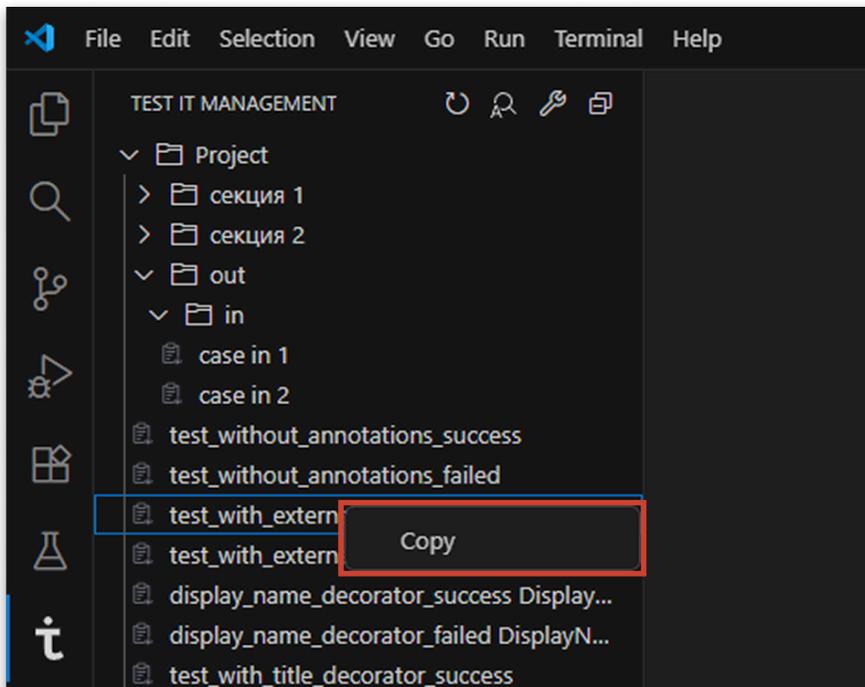
2. Правой кнопкой мыши нажмите на файл или его директорию.
3. Нажмите **Replace**, чтобы заменить декораторы в выбранном файле и **Replace all**, чтобы заменить декораторы во всех файлах.
4. В проекте нажмите значок **Search Allure Methods**.



## Создание шаблонов автотестов

Шаблон автотеста — это фрагмент кода ручного теста, отображенный в системе IDE. Шаблоны используются для последующего написания автотестов. Чтобы получить фрагмент 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:*

```
3
4 public class DemoTests {
5     @WorkItemIds("6802")
6     @Test
7     public void test_with_external_id_decorator_success() {
8         // See work item [globalId] for detailed steps description
9         // Pre:
10        //   preconditions
11        // Steps:
12        //   testSteps
13        // Post:
14        //   postconditions
15    }
16
17 }
```

## 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: 13.11.2025, 13:30:01

# 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

# 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
- HP ALM
- PractiTest
- Qase
- SpiraTest
- TestCollab

- TestLink
- Xray
- Zephyr Scale
- Zephyr Squad

## Процесс миграции

---

Перенос документации из сторонних TMS в Test IT осуществляется с помощью набора утилит (мигрантора), 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 мигрантором нужно выбрать импортер и экспортер для своей операционной системы (Windows, macOS или Linux).

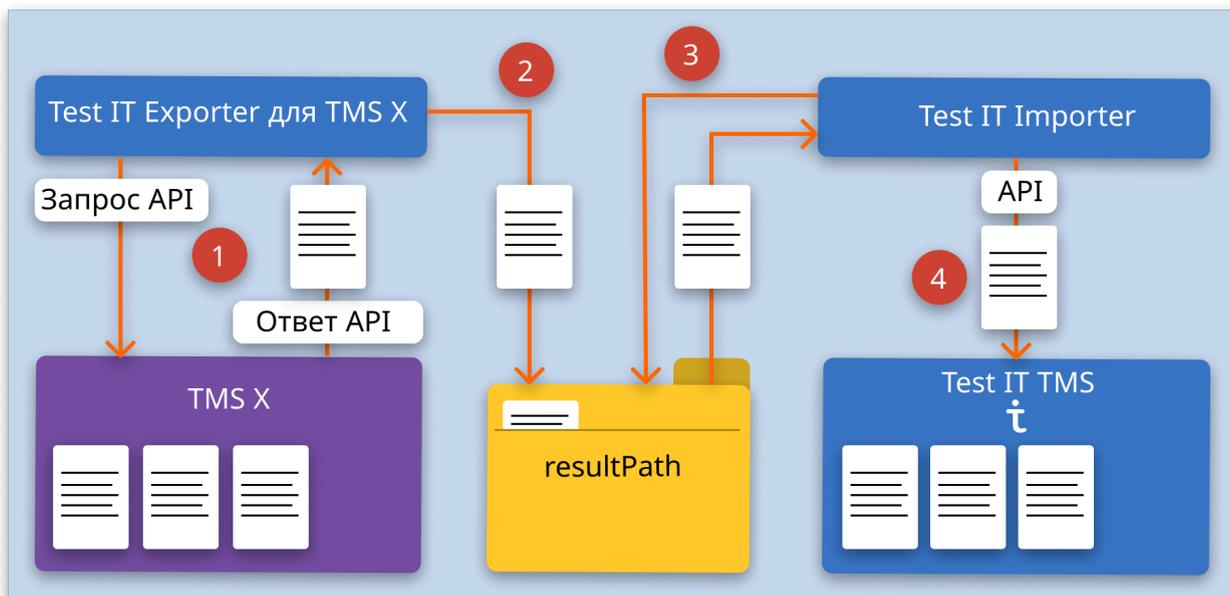
### Учитывайте версию TMS

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: 12.10.2025, 20:50:48

# 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

## 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:

```
1  {                                                                                               json
2  "resultPath": "/Users/user01/Documents/importer",
3  "allure": {
4  "url": "https://allure.ru/",
5  "apiToken": "49a1238b-b0a6-4ebb-a47c-acb2b7a9c4e9",
6  "projectName": "ProjectName",
7  "migrateAutotests": "true"
8  }
9  }
```

- Пример настройки соединения в конфигурационном файле с помощью токена на предъявителя:

```
1 | {                                                                                               json
2 |   "resultPath": "/Users/user01/Documents/importer",
3 |   "allure": {
4 |     "url": "https://allure.ru/",
5 |     "bearerToken":
6 |     "MmEwMjd4OWYtNGMxMC0zYjhkLWExMmUtNTZjYmY0OTE0MGExIDcxMjAyMCUzQ
7 |     "projectName": "ProjectName",
8 |     "migrateAutotests": "true"
9 |   }
  | }
```

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:

```
1 | sudo chmod +x ./AllureExporter{OS}-{version_number}    sh
2 | ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer",
3      "allure": {
4      "url": "https://allure.ru/",
5      "apiToken": "49a1238b-b0a6-4ebb-a47c-acb2b7a9c4e9",
6      "projectName": "ProjectName",
7      "migrateAutotests": "true"
8      }
9      }
```

- Пример настройки соединения в конфигурационном файле с помощью токена на предъявителя:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer",
3      "allure": {
4      "url": "https://allure.ru/",
5      "bearerToken":
6      "MmEwMjd4OWYtNGMxMC0zYjhkLWExMmUtNTZjYmY0OTE0MGExIDcxMjAyMCUzQ
7      "projectName": "ProjectName",
8      "migrateAutotests": "true"
9      }
10     }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

---

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

Обновлено: 16.09.2025, 12:15:30

# 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	AzureExporter-linux-x64-2.1.9
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	AzureExporter-linux-x64-2.1.9
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	AzureExporter-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	AzureExporter-2019-linux-x64-2.1.9

Сохраните экспортер в отдельной папке.

- 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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer",
3      "azure": {
4      "url": "https://dev.azure.com/testit",
5      "token":
6      "oveqpv4qsgi5fv7tb5eysqddodkux2cwp3faileved56qandckq",
7      "projectName": "ProjectName"
8      }
9      }
```

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:

```
1      sudo chmod +x ./AzureExporter{OS}-{version_number}                                     sh
2      ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer",
3      "azure": {
4      "url": "https://dev.azure.com/testit",
5      "token":
6      "oveqpv4qsgi5fv7tb5eysqddodkux2cwp3faileved56qandckq",
7      "projectName": "ProjectName"
8      }
9      }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

# 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:

```
1   {                                                                                               json
2   "resultPath": "/Users/user01/Documents/importer/hpalm",
3   "hpalm": {
4   "url": "[URL of your HP ALM Server]",
5   "clientId": "apikey-egaoktntfhidsjoddmcp",
6   "secret": "dpojbgifjifbjplf",
7   "domainName": "DEFAULT",
8   "projectName": "Migration"
9   }
10  }
```

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:

```
1   sudo chmod +x ./HPALMExporter{OS}-{version_number}    sh
2   ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1   {                                                                                               json
2   "resultPath": "/Users/user01/Documents/importer/hpalm",
3   "hpalm": {
4   "url": "[URL of your HP ALM Server]",
5   "clientId": "apikey-egaoktntfhidsjoddmcp",
6   "secret": "dpojbgifjifbjplf",
7   "domainName": "DEFAULT",
8   "projectName": "Migration"
9   }
10  }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

# Загрузка тестов в Test IT

---

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

Обновлено: 16.09.2025, 12:15:30

# 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:

```
1  {                                                                                               json
2  "resultPath": "/Users/user01/Documents/importer",
3  "practiTest": {
4  "url": "[URL of your PractiTest server]",
5  "token": "a7b9bea53f1e7a6ef7d3052969b55926680bc13b",
6  "projectId": "15467"
7  }
8  }
```

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:

```
1  sudo chmod +x ./PractiTestExporter{OS}-{version_number}  sh
2  ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1  {
2  "resultPath": "/Users/user01/Documents/importer",
3  "practiTest": {
4  "url": "[URL of your PractiTest server]",
5  "token": "a7b9bea53f1e7a6ef7d3052969b55926680bc13b",
6  "projectId": "15467"
7  }
8  }
```

json

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

Обновлено: 16.09.2025, 12:15:30

# 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@yoonion.ru](mailto:support@yoonion.ru).

- 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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer/qase",
3      "qase": {
4      "url": "http://api.qase.io/",
5      "token":
6      "e62b232d32e69d82h53e8f0211209ccf652efdccddf3d1fb11ef764ghi998a4+
7      "projectKey": "KEY"
8      }
9      }
```

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:

```
1      sudo chmod +x ./QaseExporter{OS}-{version_number}      sh
2      ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer/qase",
3      "qase": {
4      "url": "http://api.qase.io/",
5      "token":
6      "e62b232d32e69d82h53e8f0211209ccf652efdccddf3d1fb11ef764ghi998a4",
7      "projectKey": "KEY"
8      }
9    }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь инструкцией.

# 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:

```
1  {                                                                                               json
2  "resultPath": "/Users/user01/Documents/importer/spiratest",
3  "spiraTest": {
4  "url": "[SpiraTest server URL with company name]",
5  "username": "administrator",
6  "token": "{620F2362-41CB-4F87-ACDA-0CF223C72737}",
7  "projectName": "Library Information System (Sample)"
8  }
9  }
```

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:

```
1  sudo chmod +x ./SpiraTestExporter{OS}-{version_number}  sh
2  ./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@yoonion.ru](mailto:support@yoonion.ru)).

## Windows

1. Создайте конфигурационный файл `spiratest.config.json` и сохраните его в папке с экспортом. Настройте соединение в файле `spiratest.config.json` . Пример настройки соединения в конфигурационном файле:

```
1  {                                                                                               json
2  "resultPath": "/Users/user01/Documents/importer/spiratest",
3  "spiraTest": {
4  "url": "[SpiraTest server URL with company name]",
5  "username": "administrator",
6  "token": "{620F2362-41CB-4F87-ACDA-0CF223C72737}",
7  "projectName": "Library Information System (Sample)"
8  }
9  }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

Обновлено: 16.09.2025, 12:15:30

# 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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer/testcollab",
3      "testcollab": {
4      "url": "https://api.testcollab.io/",
5      "token": "bPM3ji6XEw64fUHq",
6      "projectName": "ProjectName"
7      }
8      }
```

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:

```
1      sudo chmod +x ./TestCollabExporter{OS}-{version_number}  sh
2      ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer/testcollab",
3      "testcollab": {
4      "url": "https://api.testcollab.io/",
5      "token": "bPM3ji6XEw64fUHq",
6      "projectName": "ProjectName"
7      }
8      }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

Обновлено: 16.09.2025, 12:15:30

# 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:

```
1  {                                                                                               json
2  "resultPath": "/Users/user01/Documents/importer/",
3  "testLink": {
4  "url": "[URL of your TestLink API]",
5  "token": "15fb632cdd8b606561a8b60d69a7149e",
6  "projectName": "ProjectName"
7  }
8  }
```

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:

```
1  sudo chmod +x ./TestLinkExporter{OS}-{version_number}  sh
2  ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1  {                                                                                               json
2  "resultPath": "/Users/user01/Documents/importer/",
3  "testLink": {
4  "url": "[URL of your TestLink API]",
5  "token": "15fb632cdd8b606561a8b60d69a7149e",
6  "projectName": "ProjectName"
7  }
8  }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

# 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. Скачайте экспортер TestRailExporter 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 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:

```
1   {                                                                                               json
2   "resultPath": "/Users/user01/Documents/importer",
3   "testrail": {
4   "url": "[your TestRail URL]",
5   "login": "user",
6   "password": "pass",
7   "projectName": "ProjectName"
8   }
9   }
```

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:

```
1   sudo chmod +x ./TestRailExporter{OS}-{version_number}  sh
2   ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1   {                                                                                               json
2   "resultPath": "/Users/user01/Documents/importer",
3   "testrail": {
4   "url": "[your TestRail URL]",
5   "login": "user",
6   "password": "pass",
7   "projectName": "ProjectName"
8   }
9   }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

Обновлено: 16.09.2025, 12:15:30

# 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:

```
1 | { | json
2 |   "resultPath": "/Users/user01/Documents/importer/xray",
3 |   "xray": {
4 |     "url": "[URL of your Jira server]",
5 |     "token": "MTg1Mzk4NDQxMTE30jy/3+hs+mkQ8cqhpKh0bMUPwU3e",
6 |     "projectKey": "XRAYT"
7 |   }
8 | }
```

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:

```
1 | sudo chmod +x ./XRayExporter{OS}-{version_number} | sh
2 | ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1   {                                                                                               json
2   "resultPath": "/Users/user01/Documents/importer/xray",
3   "xray": {
4   "url": "[URL of your Jira server]",
5   "token": "MTg1Mzk4NDQxMTE30jy/3+hs+mkQ8cqhpKh0bMUPwU3e",
6   "projectKey": "XRAYT"
7   }
8   }
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

# 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:

```
1 {
2   "resultPath": "/Users/user01/Documents/importer",
3   "zephyr": {
4     "url": "Link to your JIRA instance",
5     "exportArchived": true,
6     "confluence": "URL of your Confluence server",
7     "token": "MDc2MjIxNjVzNjg400kJCA43J4AfsIRBXomRs8bKw81+D",
8     "confluenceToken":
9     "MDc2MjIxNjVzNjg400kJCA43J4AfsIRBXomRs8bKw81+D",
10    "projectKey": "PK"
11  }
12 }
```

- Пример настройки соединения в конфигурационном файле с помощью логина и пароля:

```
1 {
2   "resultPath": "/Users/user01/Documents/importer",
3   "zephyr": {
4     "url": "Link to your JIRA instance",
5     "exportArchived": true,
6     "confluence": "https://confluence.instance.ru",
7     "login": "user",
8     "password": "pass",
9     "confluenceLogin": "User",
10    "confluencePassword": "pass",
11    "projectKey": "PK"
12  }
13 }
```

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:

```
1 | sudo chmod +x ./ZephyrScaleServerExporter{OS}-{version_number} | sh
2 | ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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:

```
1 {
2   "zephyr": {
3     "partial": true,
4     "partialFolderName": "batch",
5     "count": 1000
6   }
7 }
```

json

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:

```
1 {
2   "zephyr": {
3     "merge": true
4   }
5 }
```

json

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:

```
1  {                                                                                               json
2  "resultPath": "/Users/user01/Documents/importer",
3  "zephyr": {
4  "url": "https://api.zephyrscale.smartbear.com/v2/",
5  "token":
6  "oveqpv4qsgi5fv7tb5eysqddodkux2cwp3faileved56qandckq",
7  "projectName": "ProjectName"
8  }
}
```

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:

```
1  sudo chmod +x ./ZephyrScaleExporter{OS}-{version_number}  sh
2  ./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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

---

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

Обновлено: 27.11.2025, 15:18:34

# 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:

```
1   {
2   "resultPath": "/Users/user01/Documents/importer",
3   "zephyr": {
4   "url": "Link to your JIRA instance",
5   "token":
6   "MDc2MjIxNjVzNjg4O0kJCA43J4AfsIRBXomRs8bKw81+D",
7   "projectKey": "PK"
8   }
   }
```

json

- Пример настройки соединения в конфигурационном файле с помощью логина и пароля:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer",
3      "zephyr": {
4      "url": "Link to your JIRA instance",
5      "login": "user",
6      "password": "pass",
7      "projectKey": "PK"
8      }
9      }
```

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:

```
1      {                                                                                               json
2      "resultPath": "/Users/user01/Documents/importer",
3      "zephyr": {
4      "url": "https://prod-api.zephyr4jiracloud.com/",
5      "accessKey":
6      "MmEwMjd40WYtNGMxMC0zYjhkLWExMmUtNTZjYmY0OTE0MGExIDcxMjAyMCUzQ
7      "secretKey": "GkxCBle-7ib1mB0f90eiy8HZjJDw4ESnTZzYh3rz10g",
8      "accountId": "712021:3c9f032d-eeee-4dc0-8622-9b0f6843ec34",
9      "projectId": 10000,
10     "projectName": "ProjectName"
11     }
12     }
```

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:

```
1 | sudo chmod +x .\ZephyrSquadServerExporter{OS}-{version_number} | sh
2 | |
   | .\ZephyrSquadServerExporter{OS}-{version_number}
```

- Для **облачной** поставки:

```
1 | sudo chmod +x ./ZephyrSquadExporter{OS}-{version_number} | sh
2 | |
   | ./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@yoonion.ru](mailto:support@yoonion.ru)).

## 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

```
1 {
2   "resultPath": "/Users/user01/Documents/importer",
3   "zephyr": {
4     "url": "Link to your JIRA instance",
5     "token":
6     "MDc2MjIxNjVzNjg400kJCA43J4AfsIRBXomRs8bKw81+D",
7     "projectKey": "PK"
8   }
9 }
```

- Пример настройки соединения в конфигурационном файле с помощью логина и пароля:

json

```
1 {
2   "resultPath": "/Users/user01/Documents/importer",
3   "zephyr": {
4     "url": "Link to your JIRA instance",
5     "login": "user",
6     "password": "pass",
7     "projectKey": "PK"
8   }
9 }
```

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:

```
1 {
2   "resultPath": "/Users/user01/Documents/importer",
3   "zephyr": {
4     "url": "https://prod-api.zephyr4jiracloud.com/",
5     "accessKey":
6     "MmEwMjd40WYtNGMxMC0zYjhkLWExMmUtNTZjYmY00TE0MGExIDcxMjAyMCUzQ
7     "secretKey": "GkxCBle-7ib1mB0f90eiy8HZjJDw4ESnTZzYh3rz10g",
8     "accountId": "712021:3c9f032d-eeee-4dc0-8622-9b0f6843ec34",
9     "projectId": 10000,
10    "projectName": "ProjectName"
11  }
}
```

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@yoonion.ru](mailto:support@yoonion.ru)).

## Загрузка тестов в Test IT

- Чтобы загрузить тестовые данные в Test IT с помощью импортера, воспользуйтесь [инструкцией](#).

# 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 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 Enterprise 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:

```
1      {
2      "resultPath" : "/Users/user01/Documents/importer",
3      "tms" : {
4      "url" : "https://testit.software/",
5      "privateToken" : "cmZzWDkYTfBvNvVMcXhzN3Vy",
6      "certValidation" : true
7      }
8      }
```

json

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:

```
1      sudo chmod +x ./Importer{OS}-{version_number}
2      ./Importer{OS}-{version_number}
```

sh

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:

```
1  {
2  "resultPath" : "/Users/user01/Documents/importer",
3  "tms" : {
4  "url" : "https://testit.software/",
5  "privateToken" : "cmZzWDkYTfBvNvVMcXhzN3Vy",
6  "certValidation" : true
7  }
8  }
```

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.

# Работа с дашбордами

## ₽ Тарифные ограничения для Lite

В тарифе **Lite** доступен 1 дашборд с 3 виджетами. В тарифах **Standard** и **Enterprise** 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.

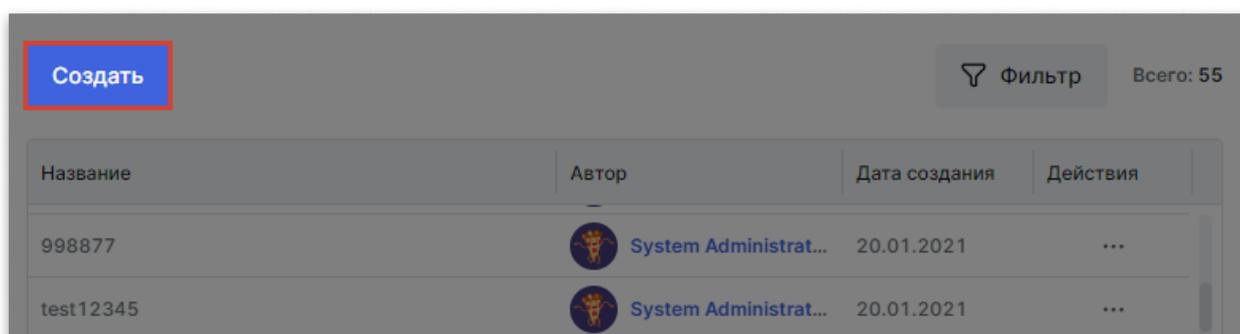
Дашборды позволяют визуализировать данные аналитики по проекту и настраивать формат их отображения с помощью виджетов. Вы можете настроить

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**.

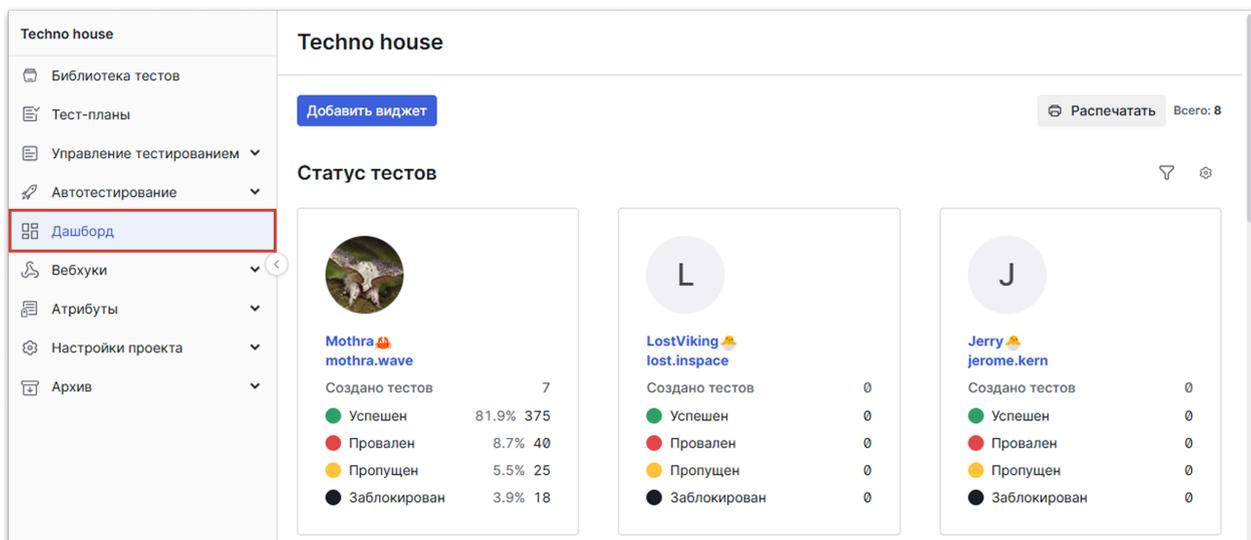
## Работа с проектными дашбордами

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:

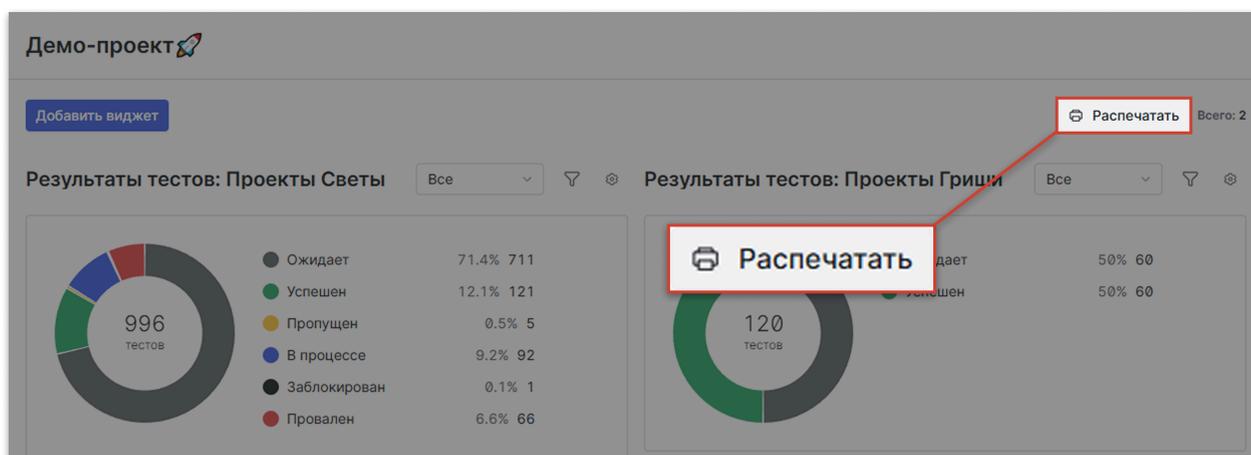
- Тренды результатов по прохождению тестов
- Распределение тестов по приоритетам
- Распределение тестов по статусу готовности
- Распределение тестов по типу тестирования
- Результаты тест-планов
- Команда
- Таблица результатов по секциям

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: 27.11.2025, 15:18:34

# 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.

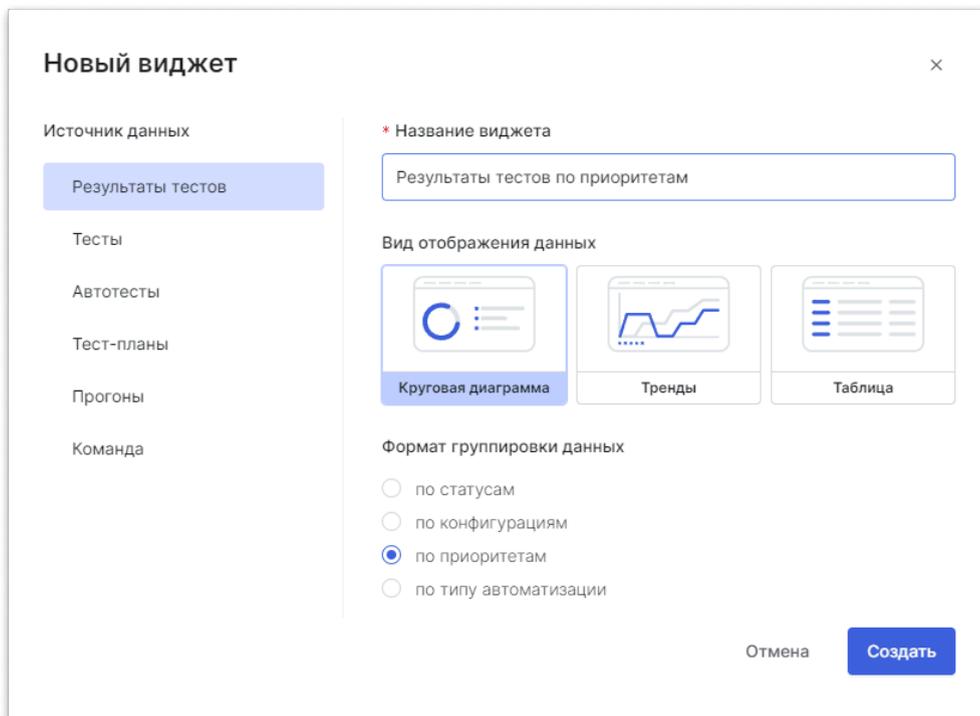
С помощью виджетов вы можете визуализировать следующие аналитические данные:

- 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**.

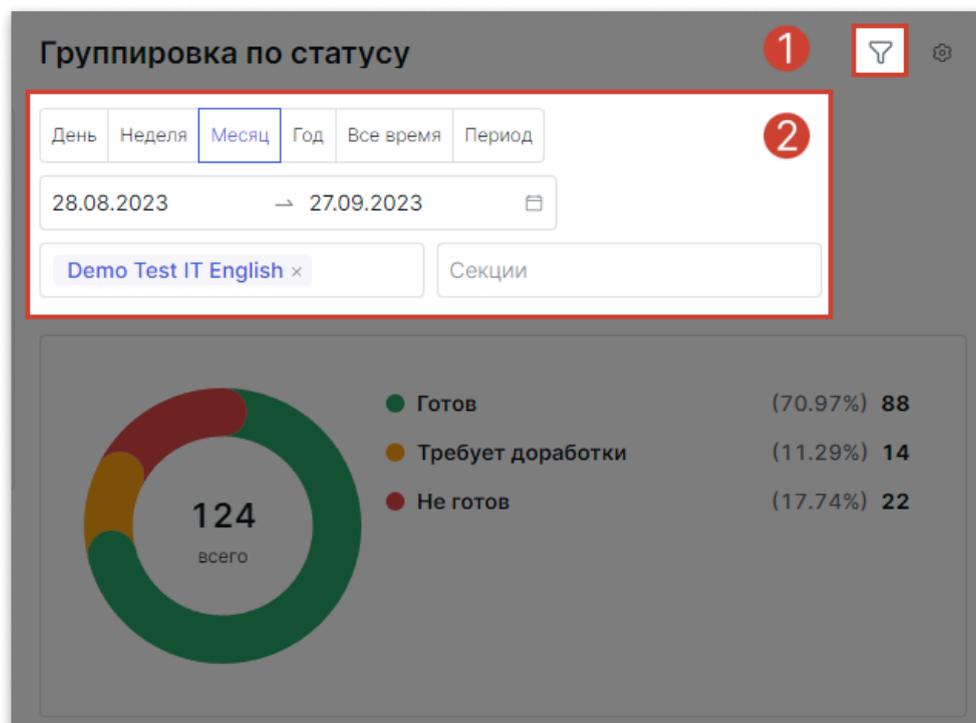


Новый виджет появится на вашем дашборде.

## 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:

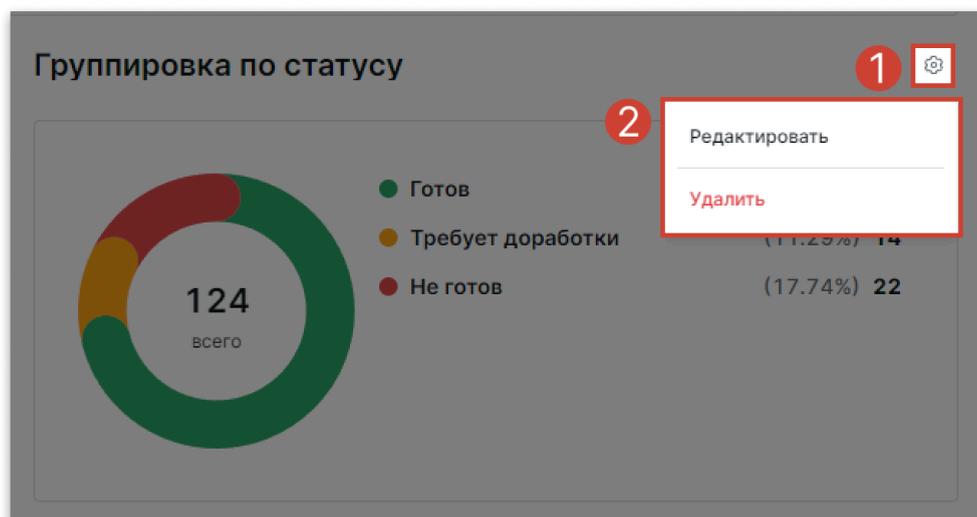
1. Нажмите значок **Фильтр** в правом верхнем углу виджета.
2. В открывшемся фильтре выберите требуемые значения.



# 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.

# 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 предоставленным результатом, распределяя их по конфигурациям, для которых были предоставлены 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 предоставленным 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 предоставленным 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 предоставленным результатом, распределяя их по специалистам, предоставившим результаты. Вы можете отобразить аналитику в формате **трендов**.

## Группировка по тест-поинтам

---

Вы можете визуализировать аналитические данные по тестам с предоставленным результатом, распределяя их по секциям библиотеки тестов. Вы можете отобразить аналитику в формате **таблицы**.

# 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.

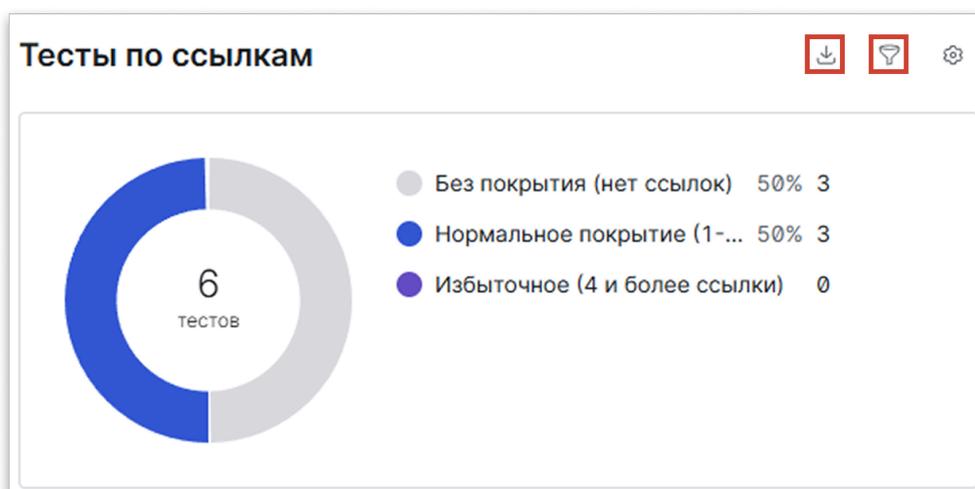
## Анализ покрытия тестов ссылками

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](/user-guide/background-tasks.md).

### 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 данныхДанные отобразятся на диаграмме и будут доступны для скачивания.

# 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**.

# 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: 17.03.2025, 20:09:16

# 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: 17.03.2025, 20:09:16

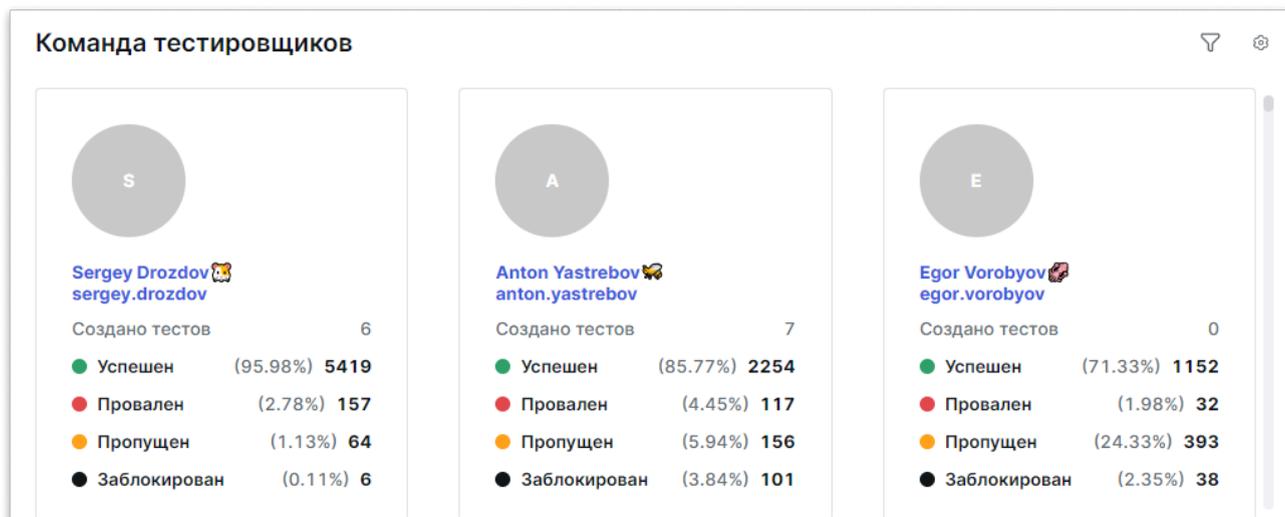
# Анализ активности команды

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**.

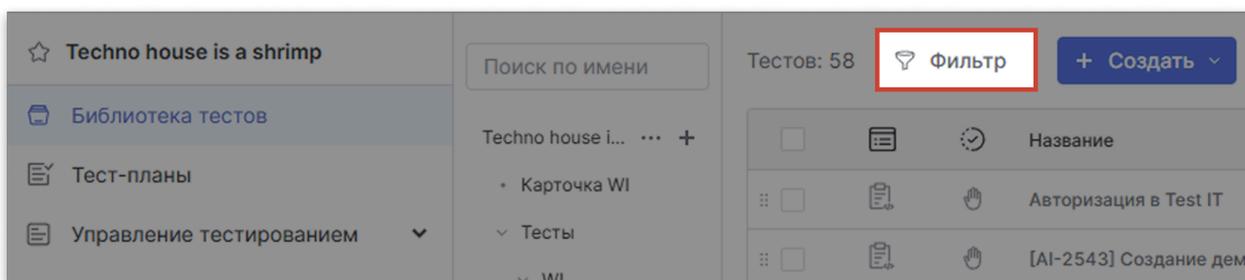
# Использование фильтров

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.

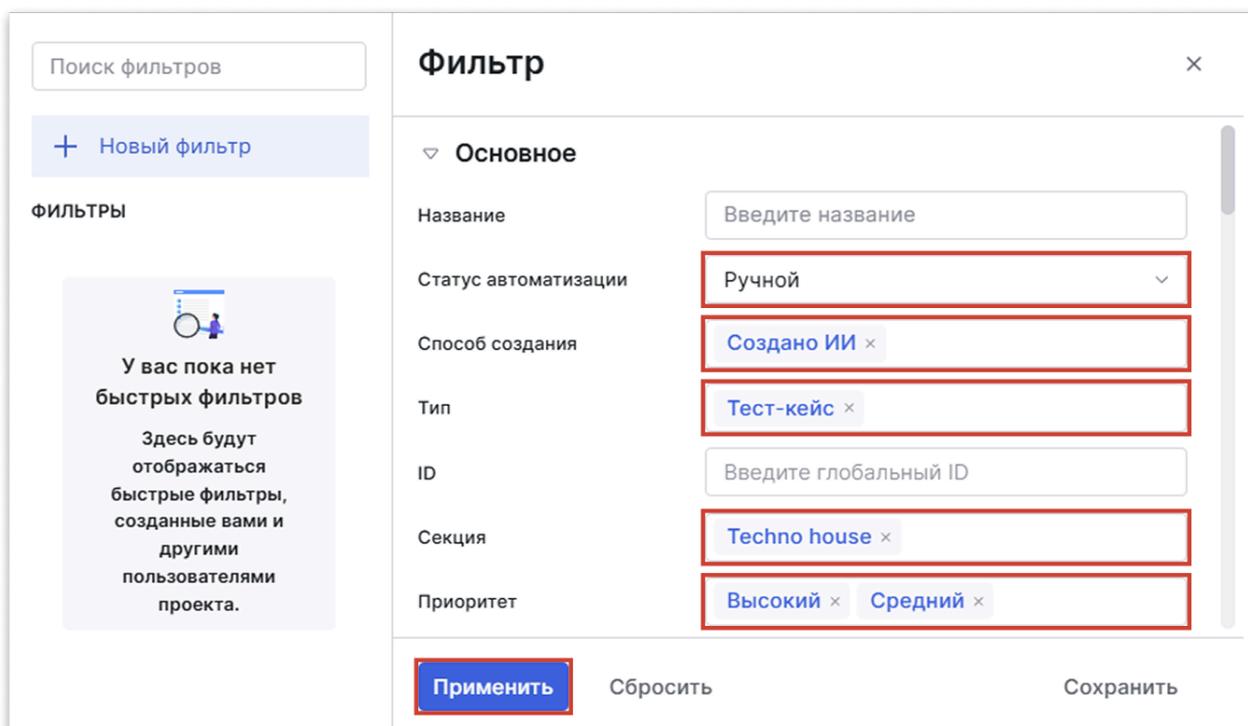
## Настройка фильтра данных

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**.

Поиск фильтров

**Фильтр** ×

+ Новый фильтр

**ФИЛЬТРЫ**

У вас пока нет быстрых фильтров

Здесь будут отображаться быстрые фильтры, созданные вами и другими пользователями проекта.

Основное

Название: Введите название

Статус автоматизации: Ручной

Способ создания: Создано ИИ ×

Тип: Тест-кейс ×

ID: Введите глобальный ID

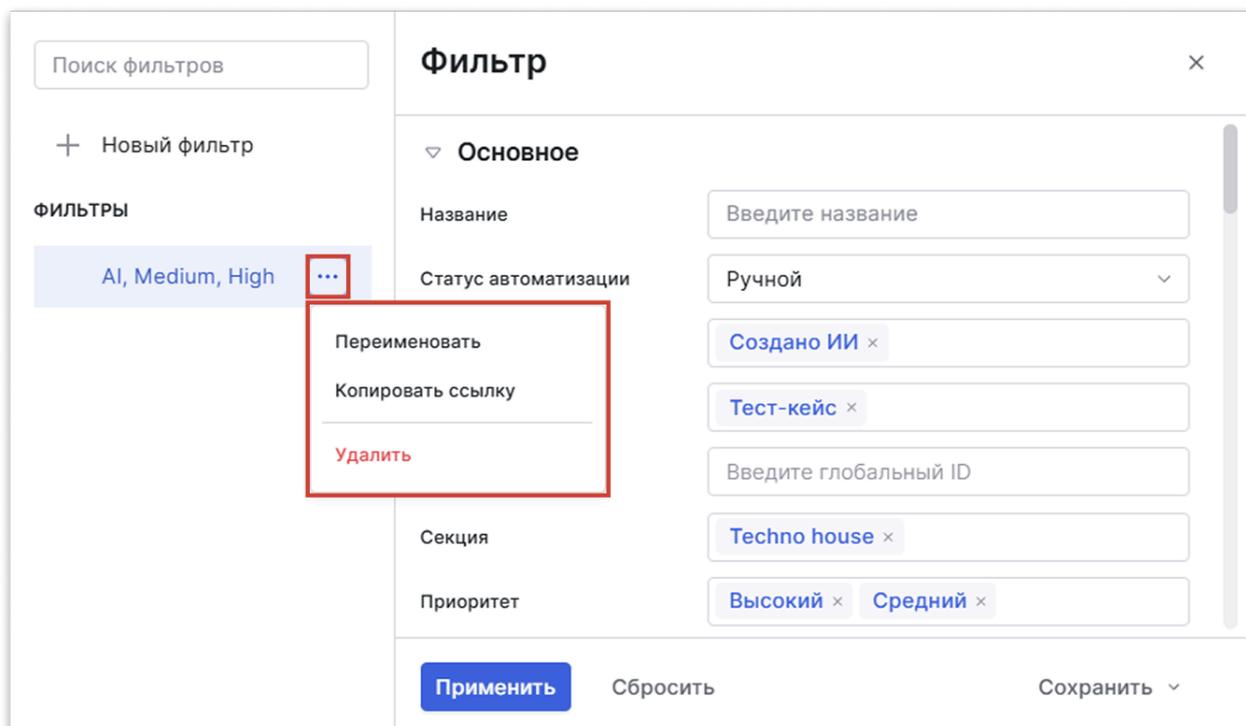
Секция: Techno house ×

Приоритет: Высокий × Средний ×

Применить Сбросить Сохранить

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: 01.08.2025, 13:03:57

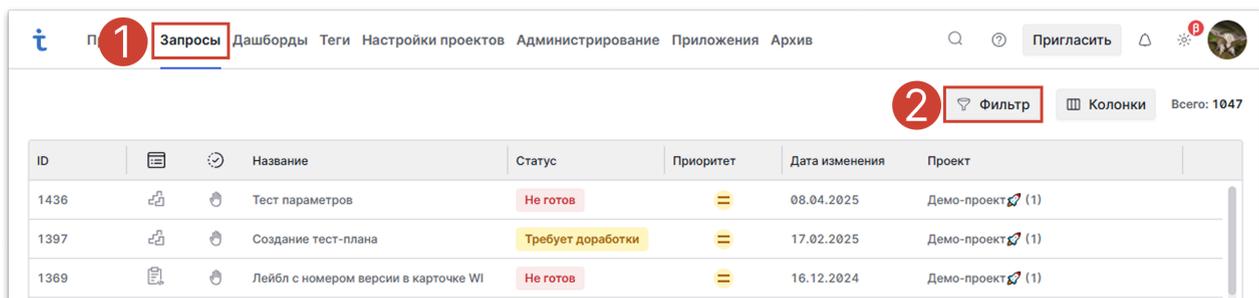
# Работа с запросами

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

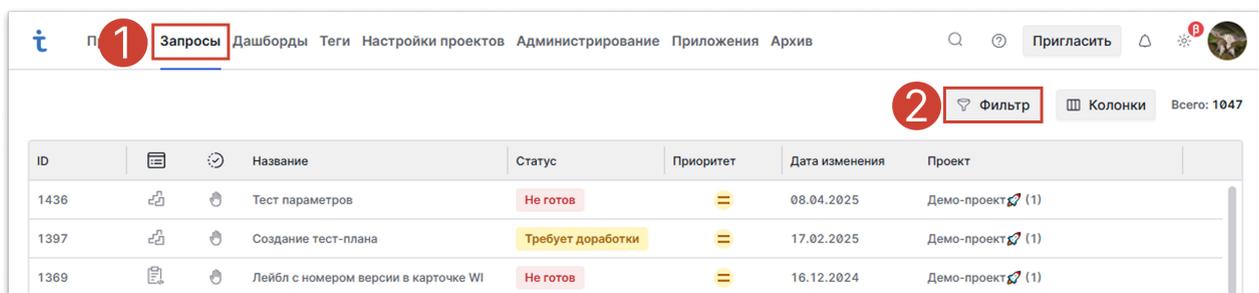
1. В верхней навигационной панели нажмите **Запросы**.
2. Нажмите **Фильтр**.



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: 26.03.2025, 20:00:23

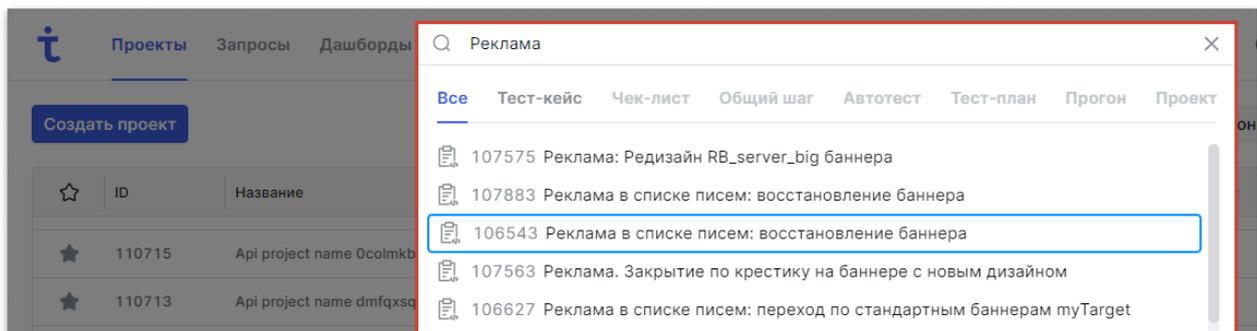
# Глобальный поиск

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**.

# Уведомления

## Просмотр уведомлений

---

### Настройте уведомления в мессенджере

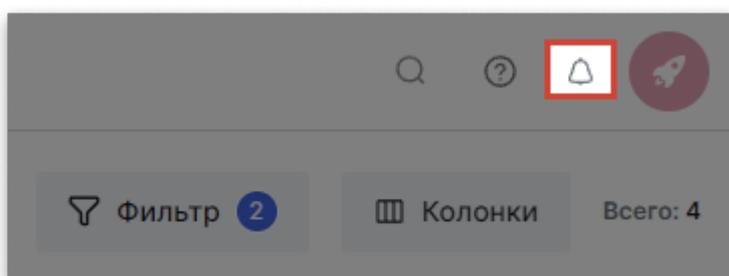
Когда пользователь получает уведомления, вы можете узнавать об этом из сообщений в мессенджере. Для этого нужно **настроить вебхук**, указав **необходимые переменные**.

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.

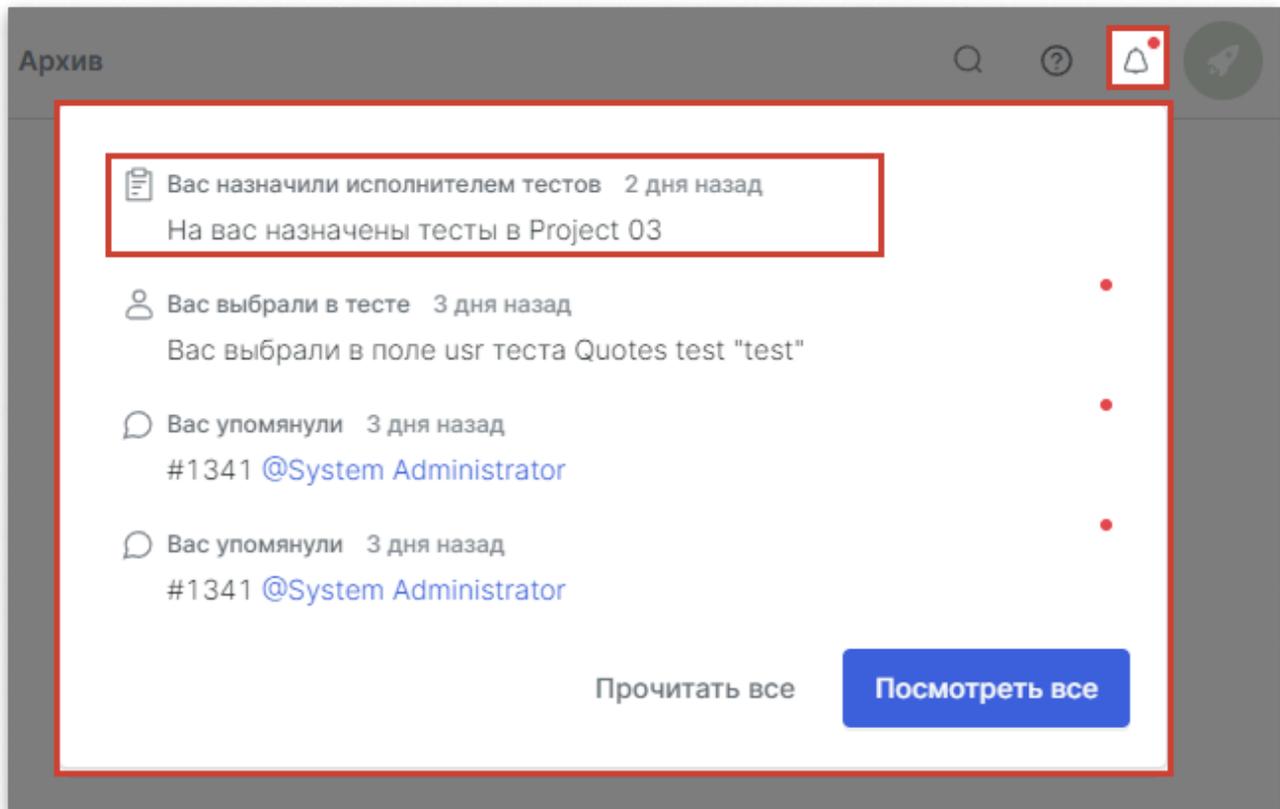
2. Нажмите **Посмотреть все**.

По умолчанию открываются все уведомления. Чтобы фильтровать их по типам, нажмите **Фильтр** и укажите необходимые вам параметры поиска.

## Вас назначили исполнителем тестов

---

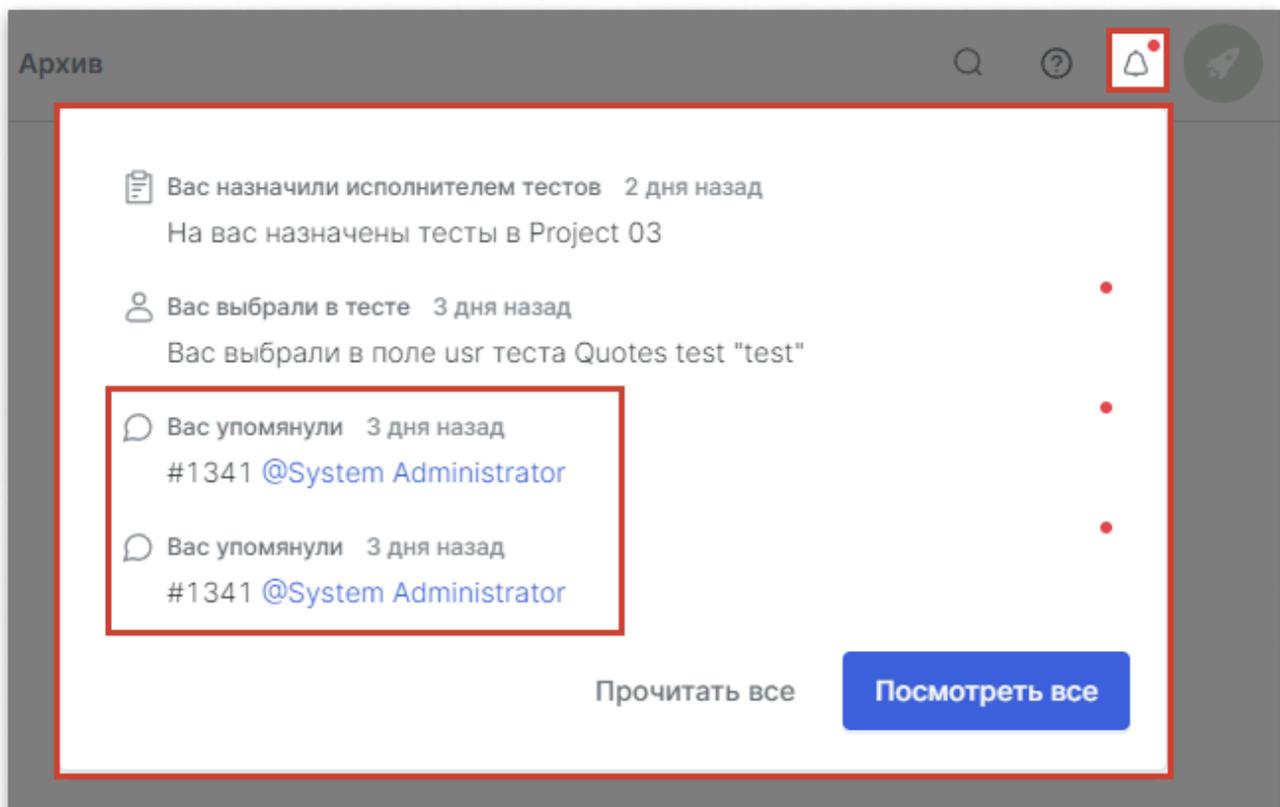
Отправляется при назначении тестов на пользователя.



При нажатии на уведомление о назначенных тестах будет открыт тест-план с назначенными на данного пользователя тестами.

## Вас упомянули

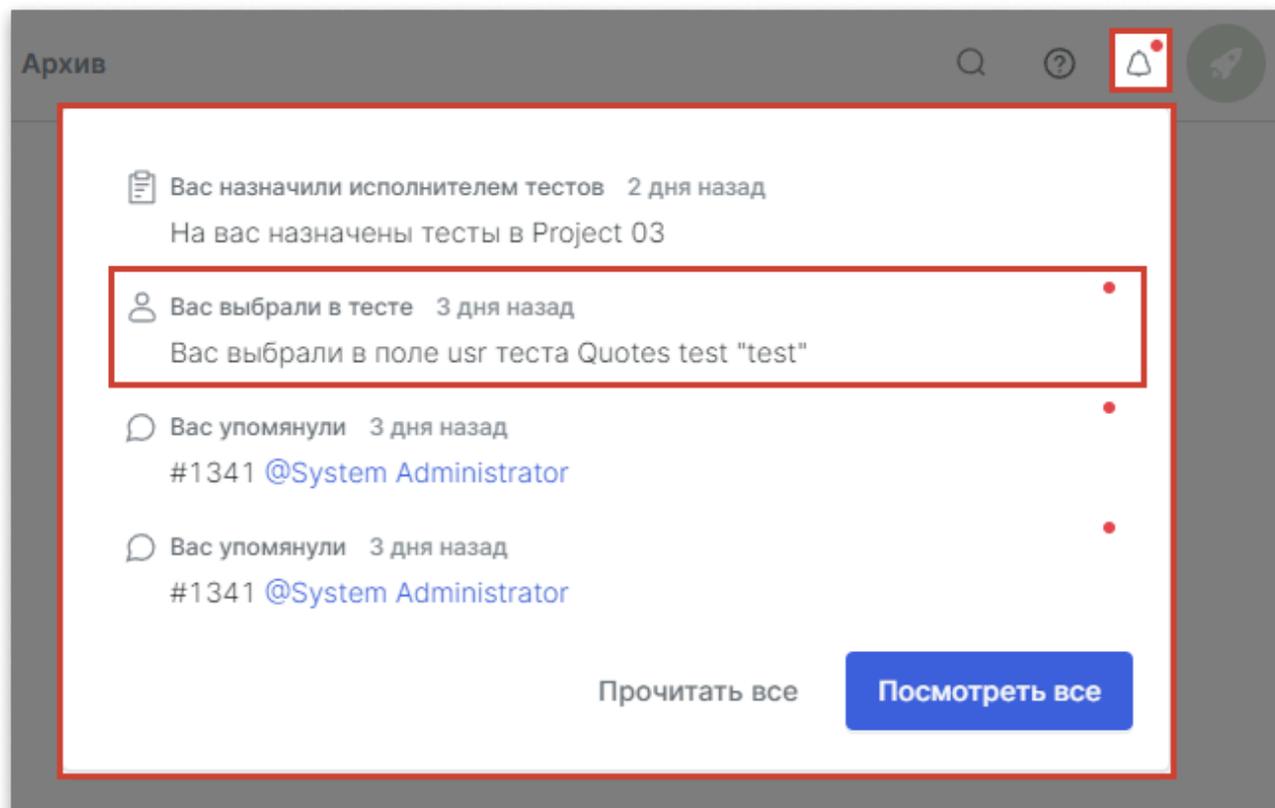
Отправляется при упоминании пользователя в комментарии к тесту.



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: 24.04.2024, 13:36:14

# 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.

## Устаревшие API-методы

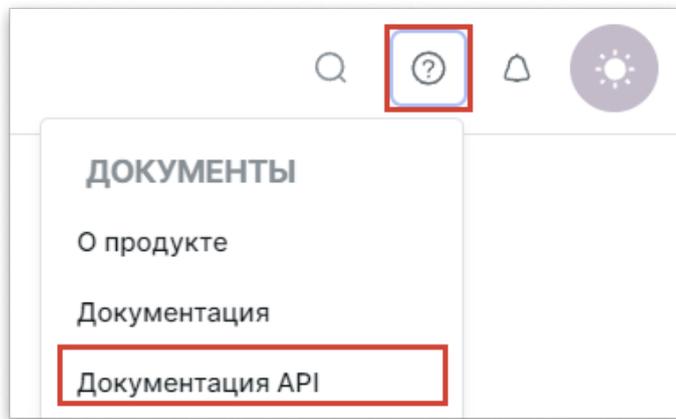
Следующие API-методы устарели и перестанут поддерживаться (дата прекращения поддержки уточняется). Рекомендуем использовать обновленные методы, добавленные в документацию **Swagger**. Список устаревших API-методов представлен в таблице ниже.

Раздел Swagger	Метод
WorkItem	get /api/v2/workItems/sharedSteps/{sharedStepId}/references
TestSuites	post /api/v2/testSuites/{id}/workItems/search
Sections	get /api/v2/sections/{id}/workItems
Projects	get /api/v2/projects post /api/v2/projects/import

## 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.

## Авторизация

---

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 NxBZ3A5d6h3bU5JLk40Q3VM` .
4. Нажмите **Authorize**.

После авторизации вы можете просматривать запросы и описания методов и делать запросы из Swagger. Для этого:

1. Откройте API-метод и нажмите **Try it out**.
2. Заполните тело запроса (если применимо).
3. Нажмите **Execute**.

После успешного выполнения запроса в системе Test IT отобразится его результат.

# Горячие клавиши

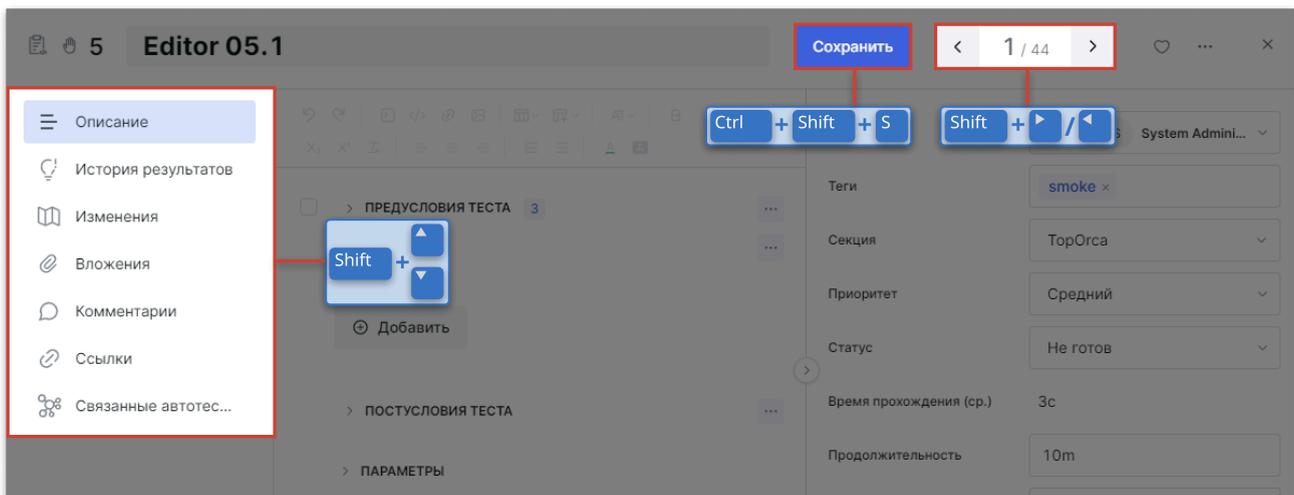
Горячие клавиши помогут вам сэкономить время и сделать работу в Test IT легкой и интуитивной. Вы можете использовать горячие клавиши для работы с модальными окнами всех разделов системы, рабочими элементами и тест-поинтами во вкладке **Выполнение тест-плана**. В горячих клавишах применяется латинский 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**.

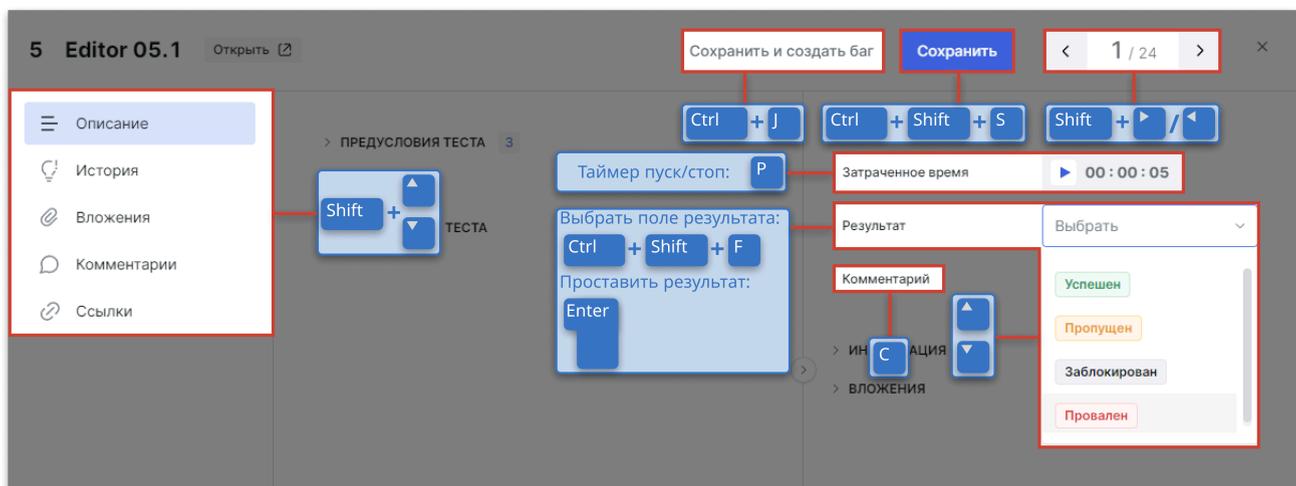
### Важно

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-пойнта

Для проставления результатов тест-пойнтов вы можете использовать следующие клавиши и их сочетания:

- **Ctrl / Cmd + Shift + F** — выбор меню результата
- **↓ / ↑** — переключение между пунктами меню выбора результата
- **Enter** — проставление тест-пойнту выбранного результата

Обновлено: 05.12.2025, 18:54:51

# 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. [Task-трекер](#).

**Test IT Enterprise** — версия системы Test IT, устанавливаемая на сервере клиента. Объем хранилища и количество пользователей не ограничены. Подробнее: [Test IT Enterprise](#).

**Test IT Lite** — облачная версия системы Test IT с ограниченным объемом хранилища, количеством пользователей и набором функций. Содержание инфраструктуры не требуется.

**Test IT Management** — плагин, для интеграции рабочих элементов Test IT в код IntelliJ IDEA. Подробнее: [Плагин Test IT Management](#).

**Test IT Standard** — облачная версия системы 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 TMS-систем. 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 Enterprise** для проверки подлинности лицензии продукта и получения доступа к его функциональности.

**Мигратор** — набор утилит (экспортер и импортер), позволяющий **автоматически перенести данные тестов** из сторонних 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). Параметры создаются в разделе **Конфигурации**.  
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 Enterprise) 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 Enterprise**, 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 или тест-плане. 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 тестирующие 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**.