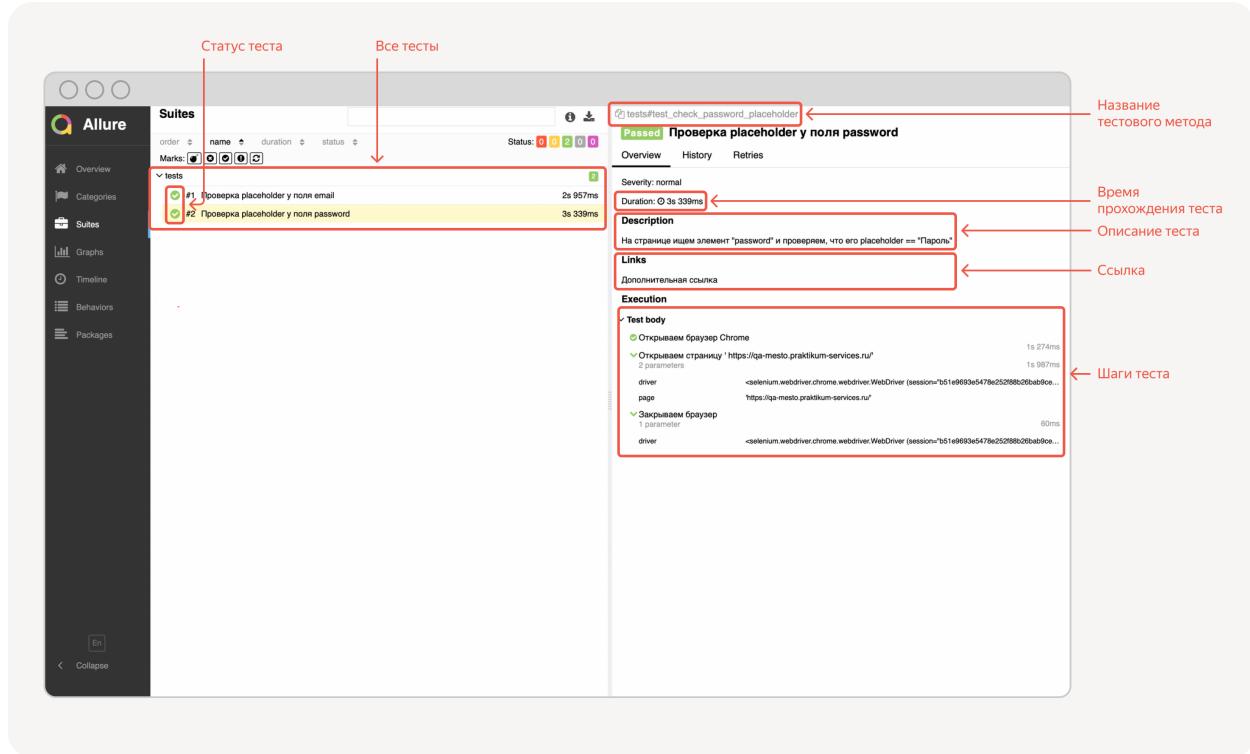


Шпаргалка: Allure

Чтобы рассказать о результатах тестов команде, можно составить отчёт о тестировании. Создавать наглядные отчёты поможет **фреймворк Allure**. В нём всё по полочкам:



Чтобы смотреть такие отчёты, нужно установить:

- фреймворк Allure;
- библиотеку `allure-pytest` — для работы в связке с Pytest;
- JDK — среду выполнения Java. Она нужна для работы с Allure.

Генерация отчёта

- Чтобы сгенерировать Allure-отчёт, введи в терминале PyCharm:

```
pytest <имя файла с тестами> --alluredir=allure_results
```

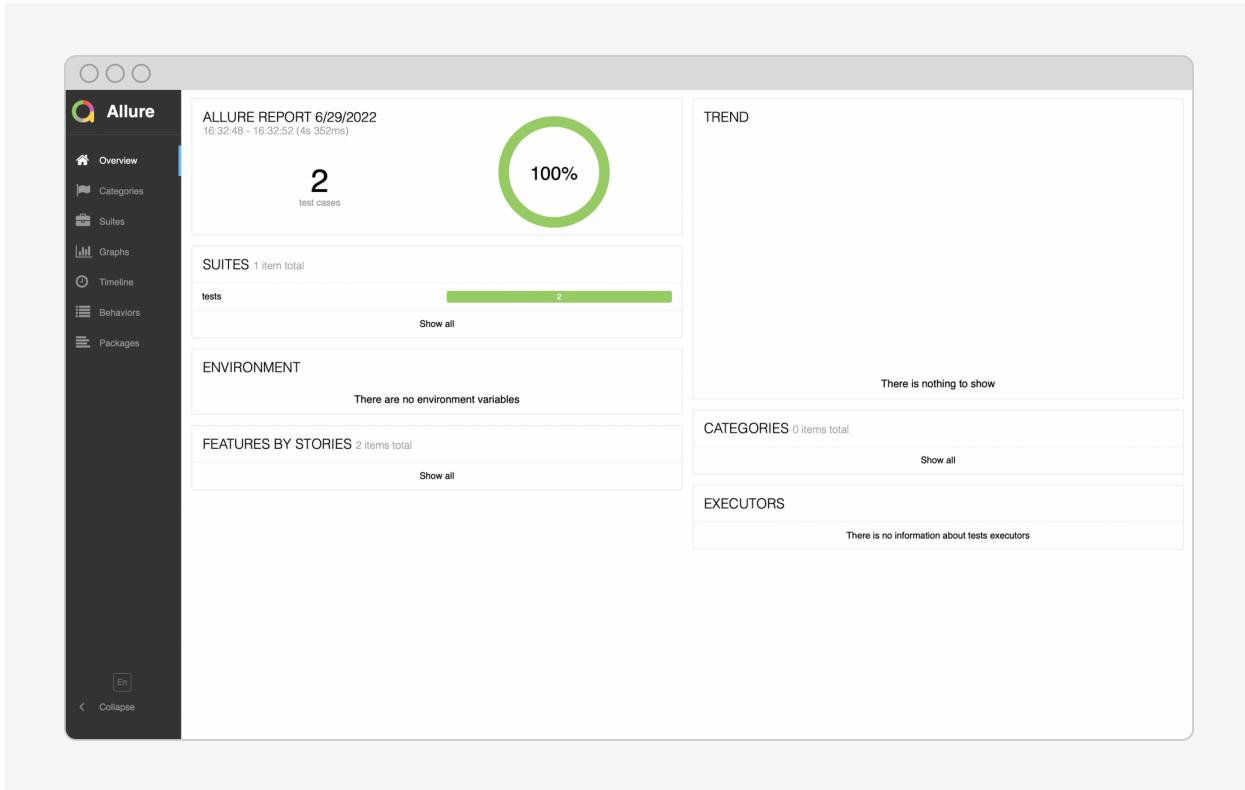
Когда тесты пройдут, в проекте автоматически появится папка `allure_results`. В неё попадут отчёты в формате JSON — по одному на каждый тест.

- Чтобы сформировать отчёт в формате веб-страницы, напиши в терминале PyCharm:

```
allure serve allure_results
```

Здесь `allure_results` указывает на папку с отчётомами, из которой нужно взять JSON-файлы.

Сформируется HTML-файл: по умолчанию он будет лежать в директории с установленным Allure. Запустится веб-сервер Allure, и в браузере откроется вкладка с результатами тестов.



Allure: декораторы

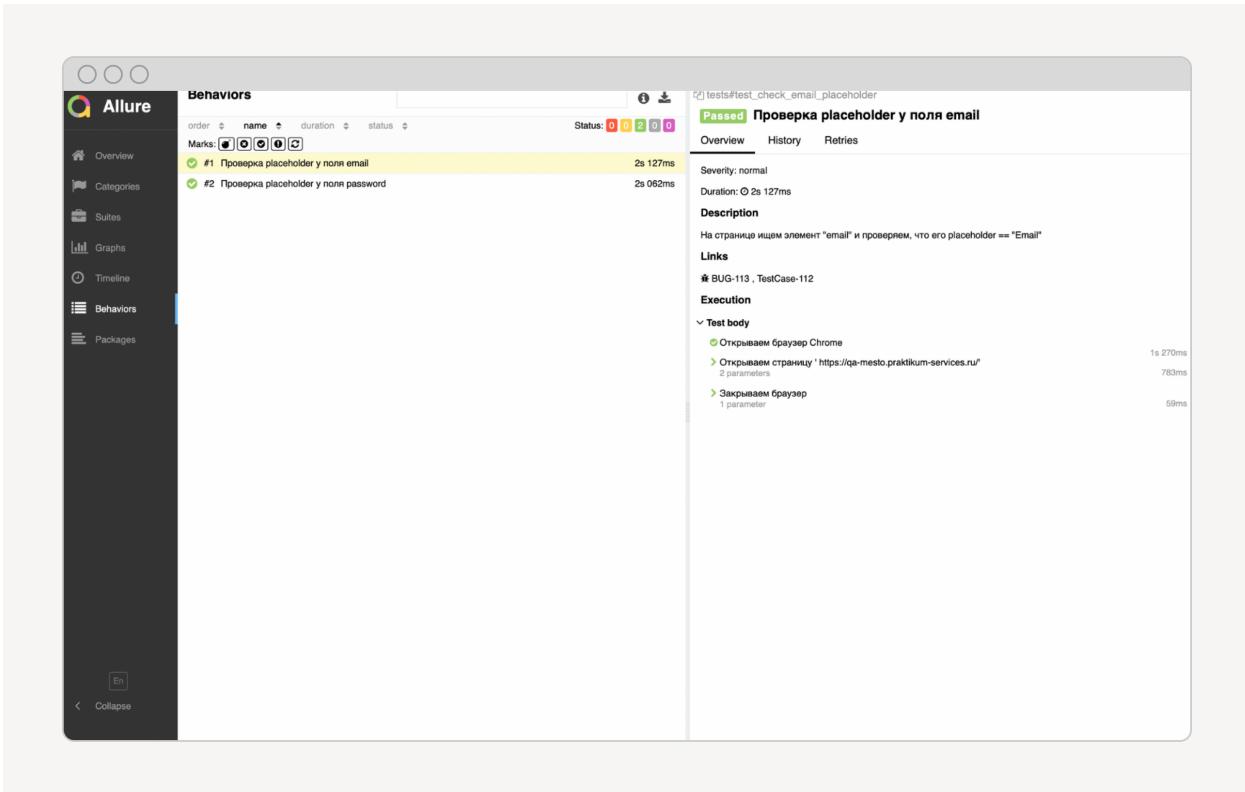
По умолчанию отчёт в Allure выглядит так:

The screenshot shows the Allure UI interface. On the left is a sidebar with navigation links: Overview, Categories, Suites (selected), Graphs, Timeline, Behaviors, and Packages. The main area is titled 'Suites' and shows a table of tests. The table has columns for 'order', 'name', 'duration', and 'status'. There are filters for 'order', 'name', 'duration', and 'status'. The status bar at the top right shows 'Status: 0 0 2 0 0' with colored icons (red, green, yellow, blue, purple). The table contains two rows:

	#1 test_check_email_placeholder	3s 679ms
	#2 test_check_password_placeholder	5s 158ms

The first test is highlighted with a yellow background. To the right of the table is a detailed view for the first test, titled 'tests#test_check_email_placeholder'. It shows the status as 'Passed' with a green icon. Below the status are tabs for 'Overview', 'History', and 'Retries', with 'Overview' selected. The overview section includes 'Severity: normal', 'Duration: 3s 679ms', and 'Execution' which states 'No information about test execution is available.' At the bottom of the sidebar is a button labeled 'Collapse'.

Его можно сделать информативнее. Например, указать понятное название теста, добавить ссылку на баг, описание и многое другое:



Чтобы делать отчёты информативнее, нужно:

- импортировать библиотеку `allure` прямо в код с тестами,
- «навесить» на функции **декораторы**.

Декоратор расширяет возможности функции. Он не изменяет в ней код, а как бы прикрепляется, как бирка. Например, если добавить декоратор `@allure.description` к функции `test_check_email_placeholder()`, в отчёте можно отобразить подробное описание, что происходит в этом кусочке теста.

Пример. Вот код автотестов, дополненный декораторами для Allure-отчёта.

```
import allure # 1. импортировали библиотеку
from selenium import webdriver
from selenium.webdriver.common.by import By

base_url = 'https://qa-mesto.praktikum-services.ru/'

@allure.step('Открываем браузер Chrome') # декоратор
def init_driver():
    return webdriver.Chrome()

@allure.step('Открываем страницу {page}') # декоратор
def open_page(driver, page):
```

```

driver.get(page)

@allure.step('Закрываем браузер') # декоратор
def quit_driver(driver):
    driver.quit()

@allure.title('Проверка placeholder у поля email') # декораторы
@allure.description('На странице ищем элемент "email" и проверяем, что его placeholder == "Email"')
@allure.testcase('ссылка на тест-кейс', 'TestCase-112')
@allure.issue('ссылка на баг', 'BUG-113')
def test_check_email_placeholder():
    driver = init_driver()
    open_page(driver, base_url)

    email = driver.find_element(By.ID, "email")
    assert email.get_attribute("placeholder") == 'Email'

    quit_driver(driver)

@allure.title('Проверка placeholder у поля password')
@allure.description('На странице ищем элемент "password" и проверяем, что его placeholder == "Пароль"')
@allure.link(base_url, name='Дополнительная ссылка')
def test_check_password_placeholder():
    driver = init_driver()
    open_page(driver, base_url)

    password = driver.find_element(By.ID, "password")
    assert password.get_attribute("placeholder") == 'Пароль'

    quit_driver(driver)

```

Описать шаг — `@allure.step`

Тест можно разбить на шаги — кусочки кода, которые обозначают отдельный этап. Шаги задают декоратором `@allure.step`. В скобках нужно написать, что именно происходит в шаге.

Пример. Вот все шаги, из которых состоит тест:

```

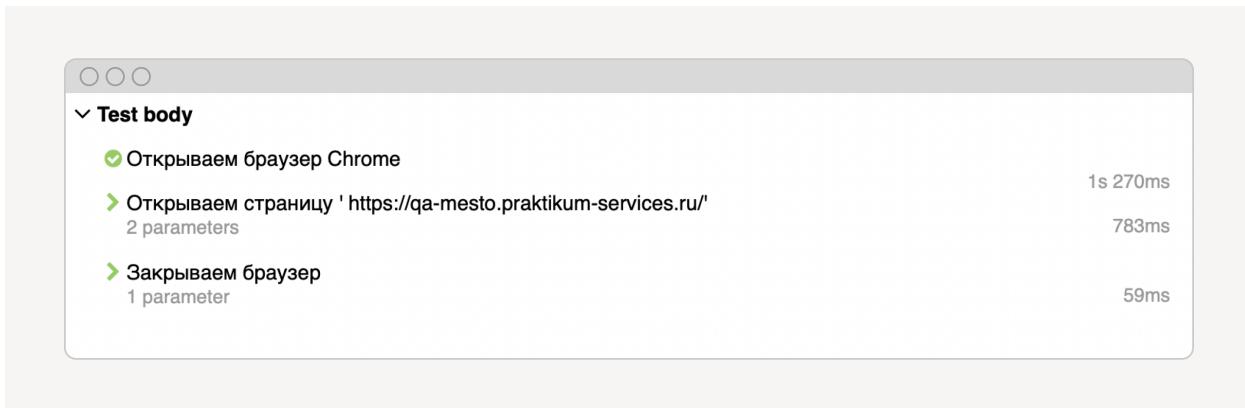
@allure.step('Открываем браузер Chrome')
def init_driver():
    return webdriver.Chrome()

@allure.step('Открываем страницу {page}')
def open_page(driver, page):
    driver.get(page)

@allure.step('Закрываем браузер')
def quit_driver(driver):
    driver.quit()

```

Именно их ты увидишь в отчёте — в Test body на панели информации о тесте:



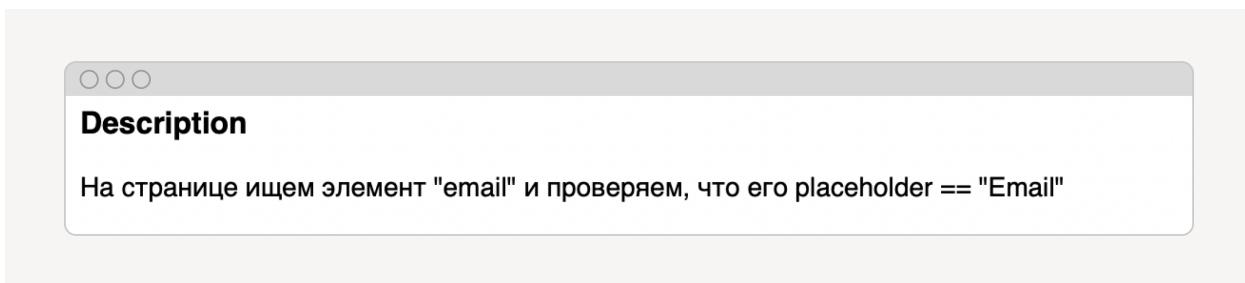
Описание теста — `@allure.description`

Добавляет подробное описание, что происходит в конкретном тесте. Например, для `test_check_email_placeholder` — «На странице ищем элемент `email` и проверяем, что его `placeholder == "Email"`».

Это удобно, когда из названия теста не очень понятно, что именно в нем происходит. Да и ориентироваться так проще.

```
@allure.description(  
    'На странице ищем элемент email и проверяем, что его placeholder == "Email"')  
def test_check_email_placeholder():  
    ...
```

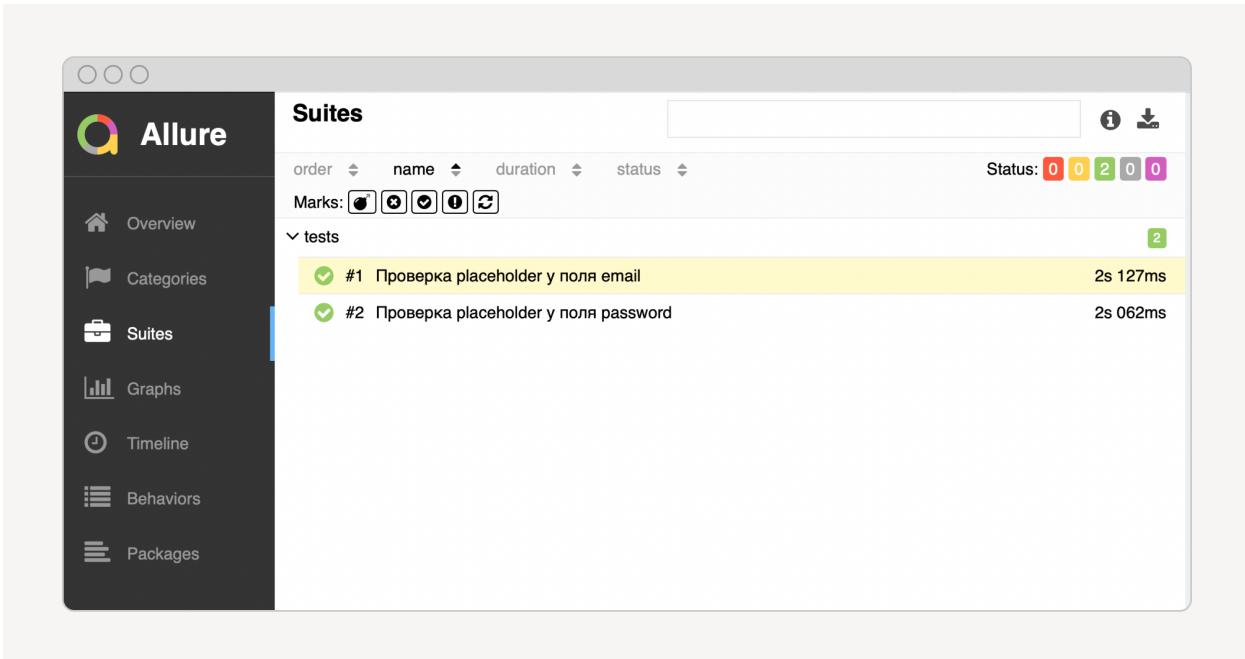
В отчёте описание отображается в секции Description:



Понятное название — `@allure.title`

Добавляет новое название тесту в отчёте. Например, «проверка `placeholder` у поля `email`». Если не писать `title`, Allure назовёт тест по имени функции — например, `test_check_email_placeholder`.

В отчёте названия отображаются на панели Suites:



Отчёт можно соединить с системами баг-трекинга и тест-менеджмента. Например, поставить ссылку на тикет или на баг. Для этого есть свои декораторы.

Ссылка на баг — `@allure.issue`

В скобках нужно написать:

- ссылку на баг — например, в Jira;
- описание бага или его id.

```
@allure.issue('ссылка на баг', 'BUG-113')
```

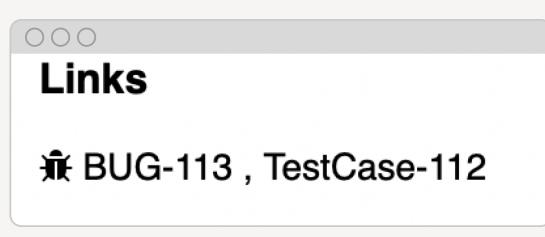
Ссылка на тест-кейс — `@allure.testcase`

В скобках:

- ссылка на тест-кейс в Test Management System,
- описание или id тест-кейса.

```
@allure.testcase('ссылка на тест-кейс', 'TestCase-112')
```

В отчёте ссылки отображаются в секции Links. Для `allure.issue` автоматически появится иконка жука:



Дополнительные ссылки — `@allure.link`

Если добавить любые другие рабочие ссылки, по ним можно будет перейти прямо из отчёта:

```
base_url = 'https://qa-mesto.praktikum-services.ru/'  
  
@allure.link(base_url, name='дополнительная ссылка')
```

В Allure есть и другие фичи. Он них можно почитать в [документации](#). На официальном сайте Allure есть [подробное описание структуры отчёта](#).