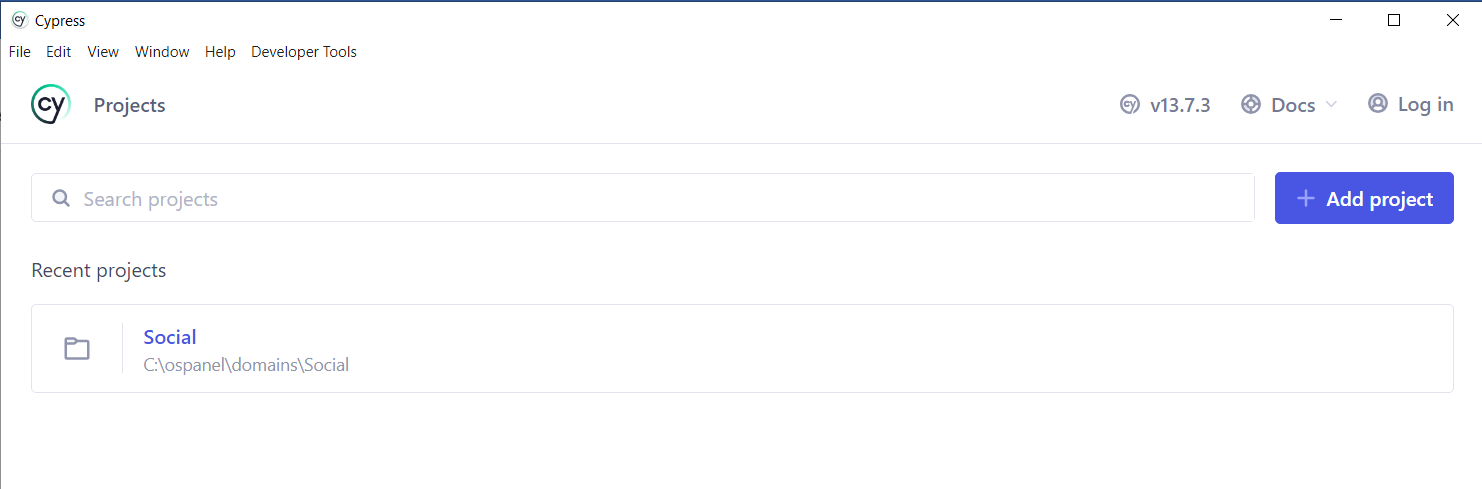
**Практическая работа Тестирование e2e с Cypress**

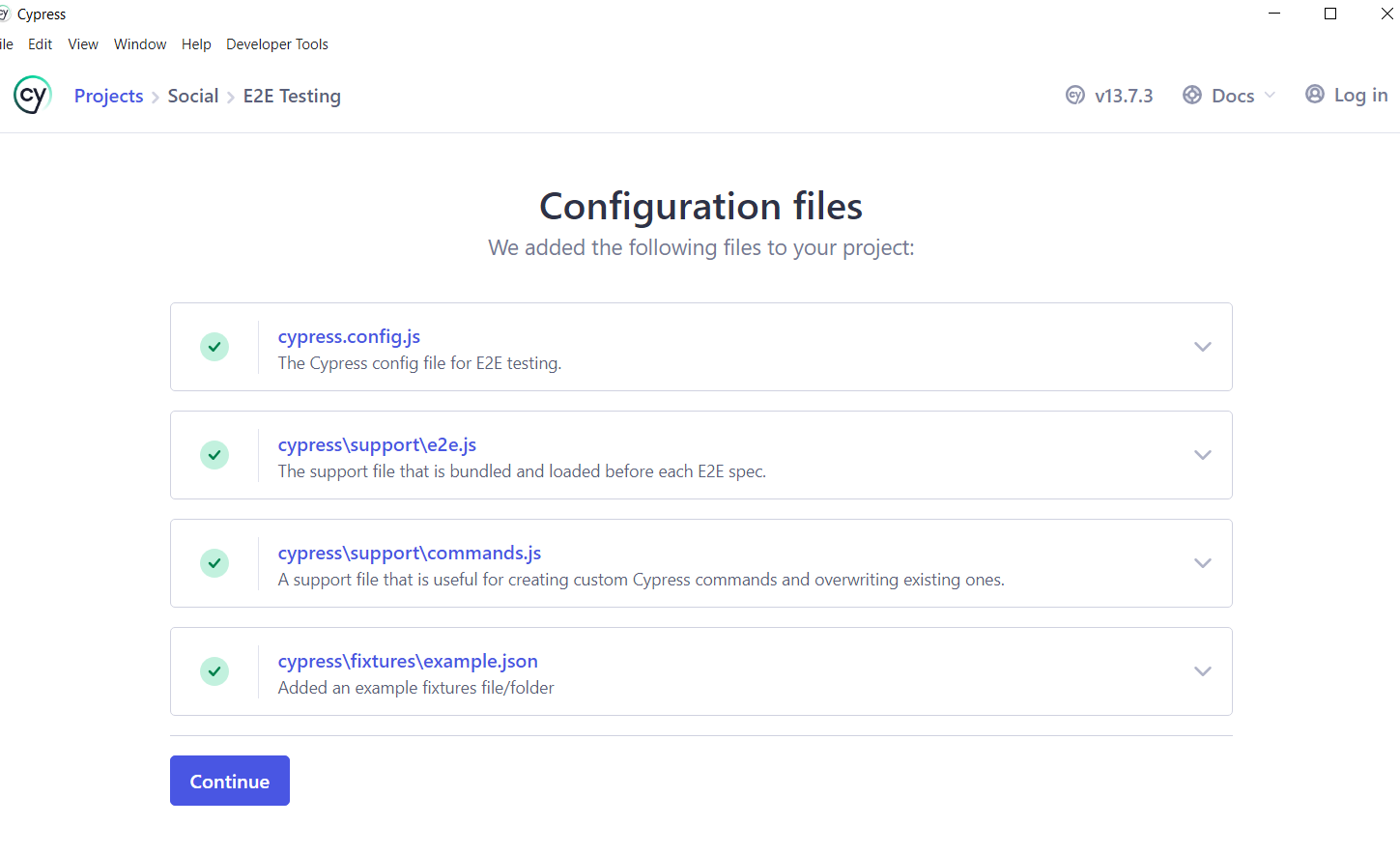
Документация <https://docs.cypress.io/guides/getting-started/installing-cypress>

**Установка и начало работы**

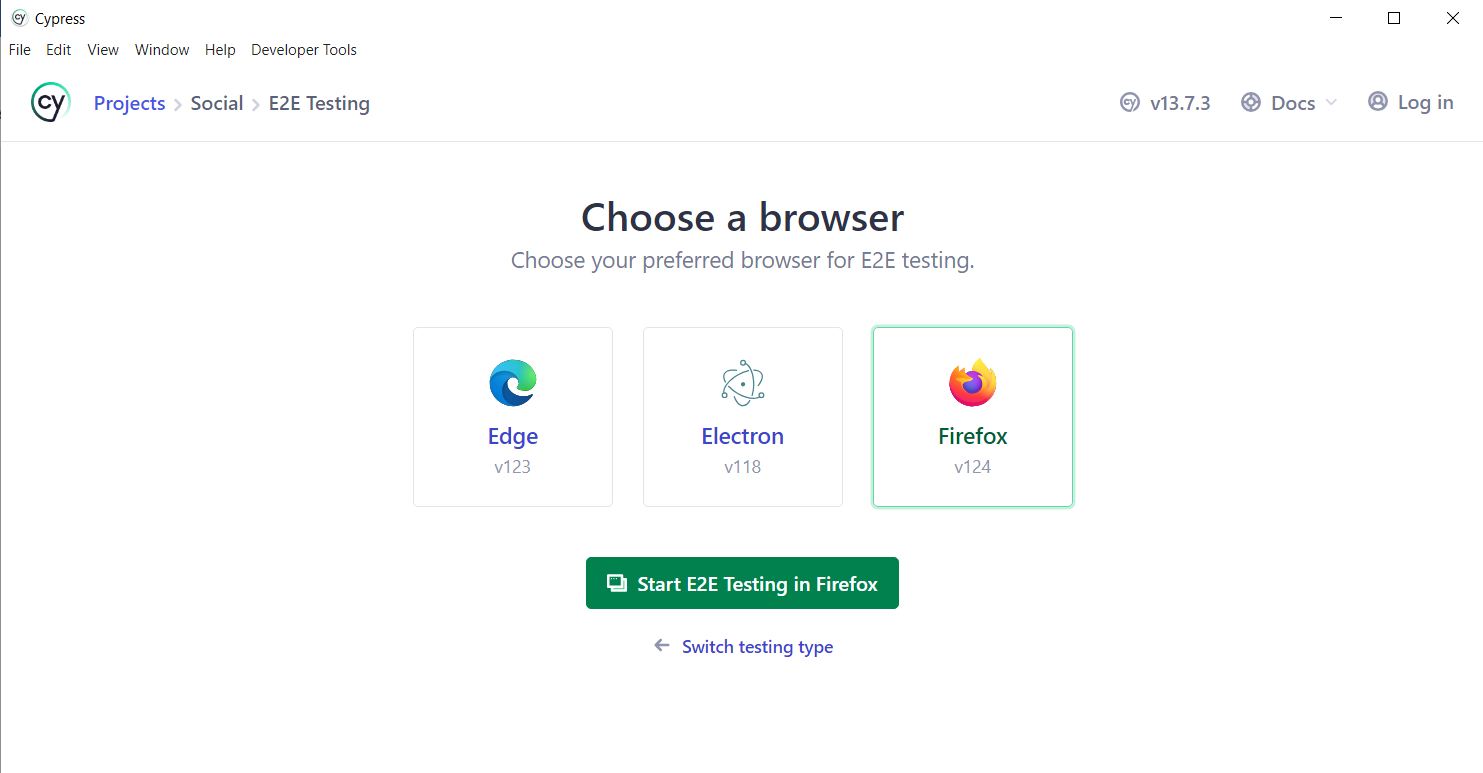
1. Скачайте zip-архив программы <https://www.cypress.io/install> и распакуйте его.
2. Запустите приложение.
3. Добавьте проект, который будете тестировать.



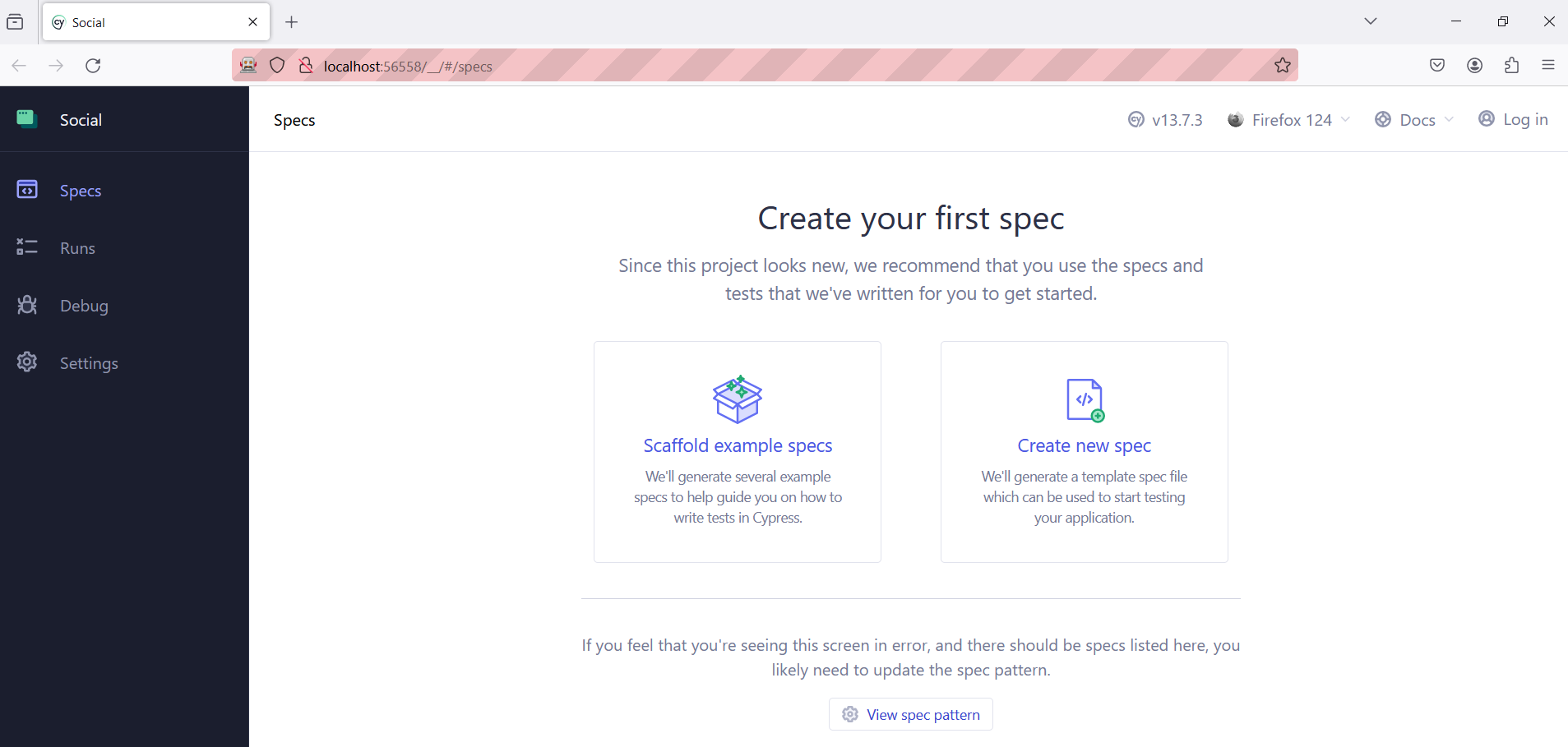
1. Щелкните по названию проекта и выберите вариант тестирования E2E Testing.



1. Файлы конфигурации оставляем без изменений и нажимаем Continue.

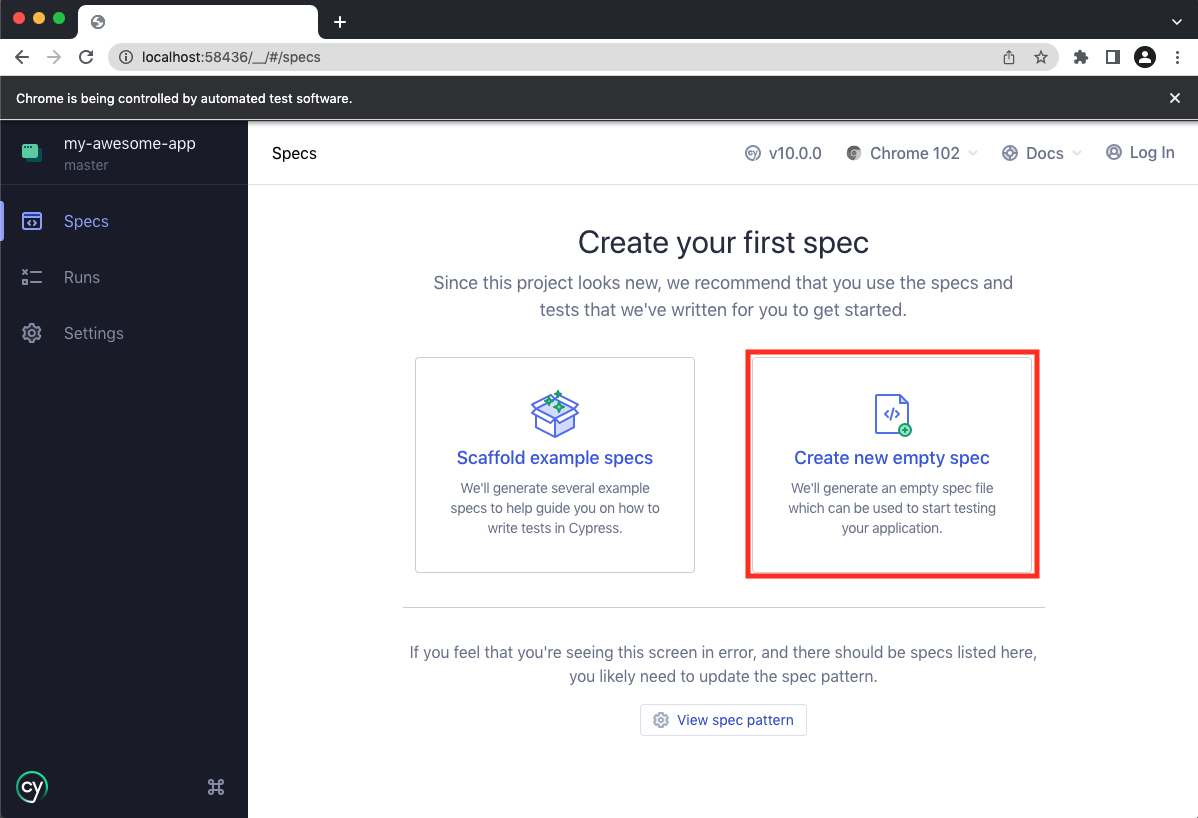


1. Выбираем браузер и нажимаем Start.

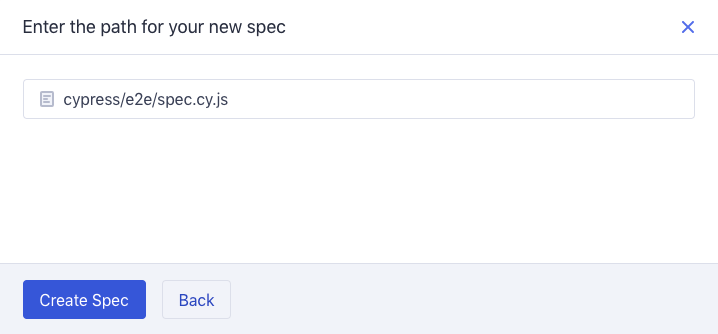


**Первый тест с Cypress**

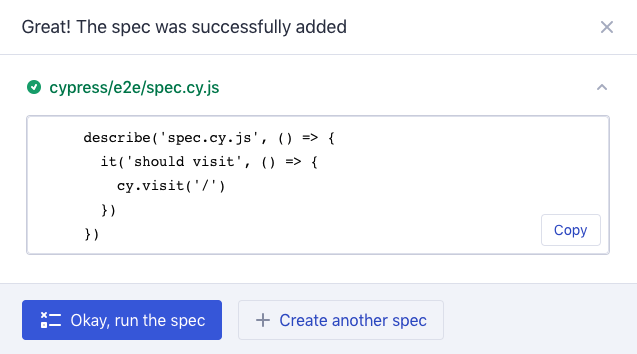
1. Создадим новую пустую спецификацию.



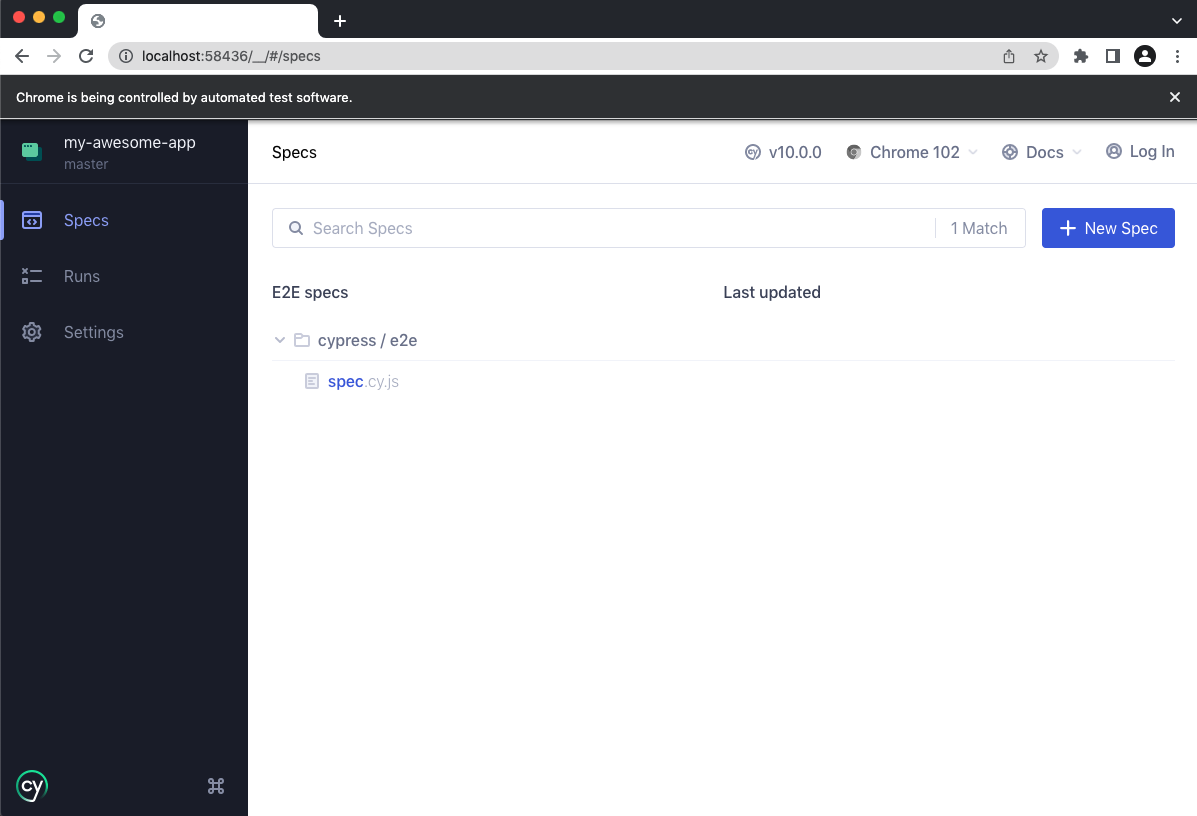
1. Щелкнув по нему, вы должны увидеть диалоговое окно, в котором вы можете ввести название вашей новой спецификации. Пока просто примите имя по умолчанию.



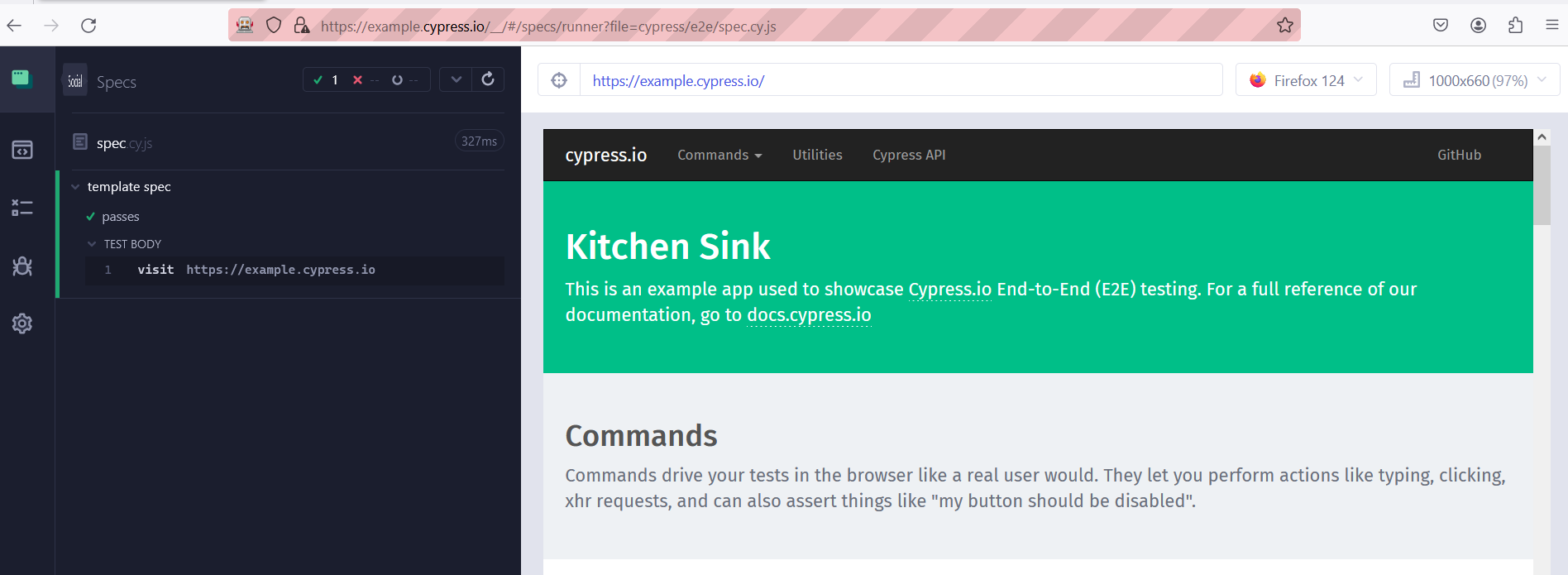
1. Недавно созданная спецификация отображается в диалоговом окне подтверждения. Просто продолжайте и закройте ее кнопкой ✕.



1. Как только мы создадим этот файл, вы должны сразу увидеть его в списке комплексных спецификаций. Cypress отслеживает ваши файлы спецификаций на предмет любых изменений и автоматически отображает их.

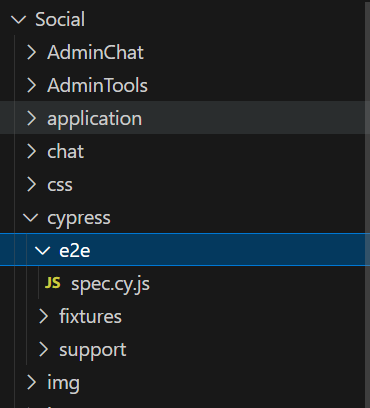


Хотя мы еще не написали никакого кода - ничего страшного - давайте кликнем по вашей новой спецификации и посмотрим, как Cypress запустит ее. Предупреждение о спойлере: вероятно, произойдет **СБОЙ**.

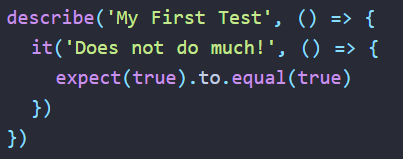


## **Напишите свой первый тест**

Все тесты Cypress сохраняет с корень вашего сайта

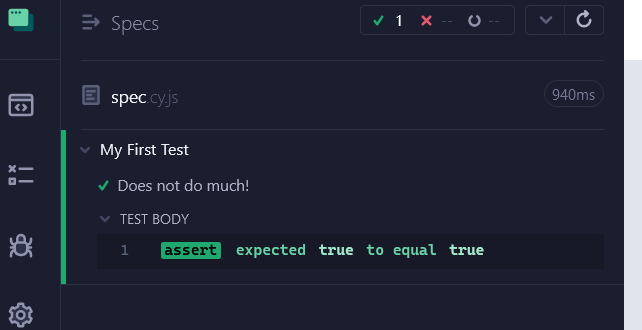


Откройте файл созданной ранее спецификации и замените содержимое вашей спецификации приведенным ниже кодом.

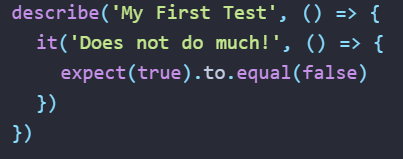


Как только вы сохраните это изменение, вы увидите перезагрузку браузера.

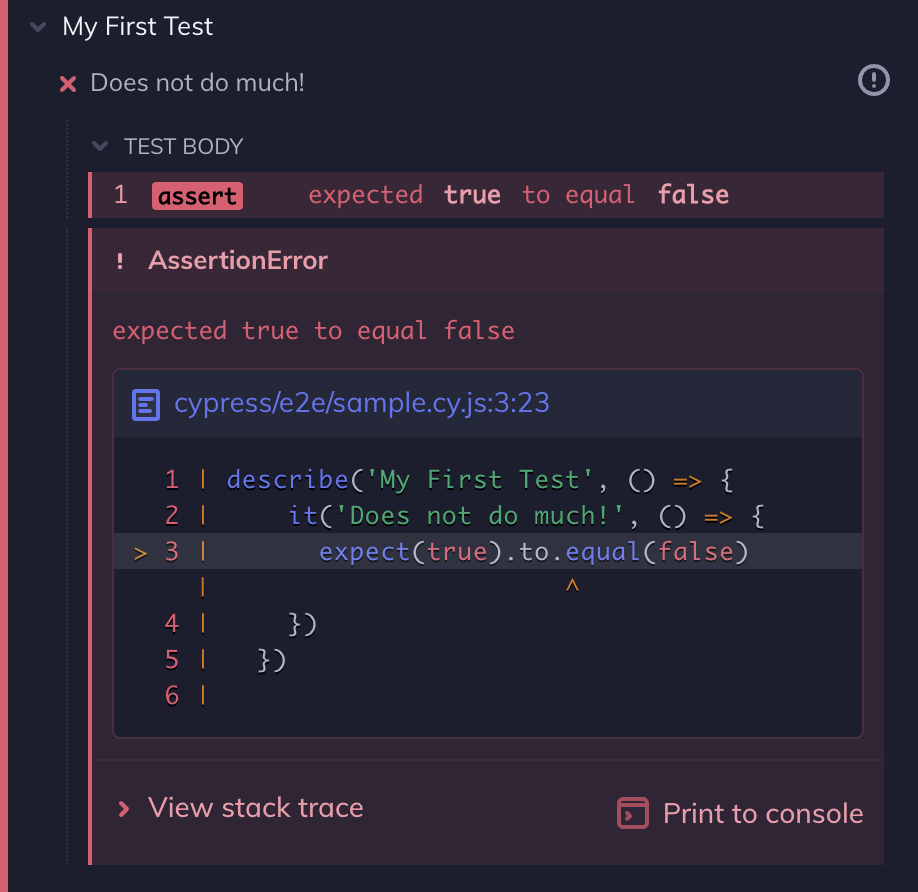
В журнале команд вы увидите, что Cypress отображает пакет, тест и ваше первое утверждение (которое должно быть выделено зеленым цветом).



Теперь давайте напишем наш первый неудачный тест.

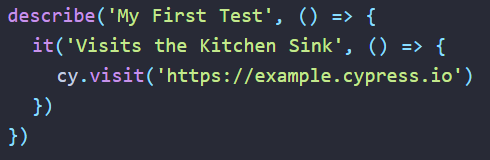


После повторного сохранения Cypress отобразит неудачный тест красным цветом, поскольку true не равно false.



**Пишем реальный тест**

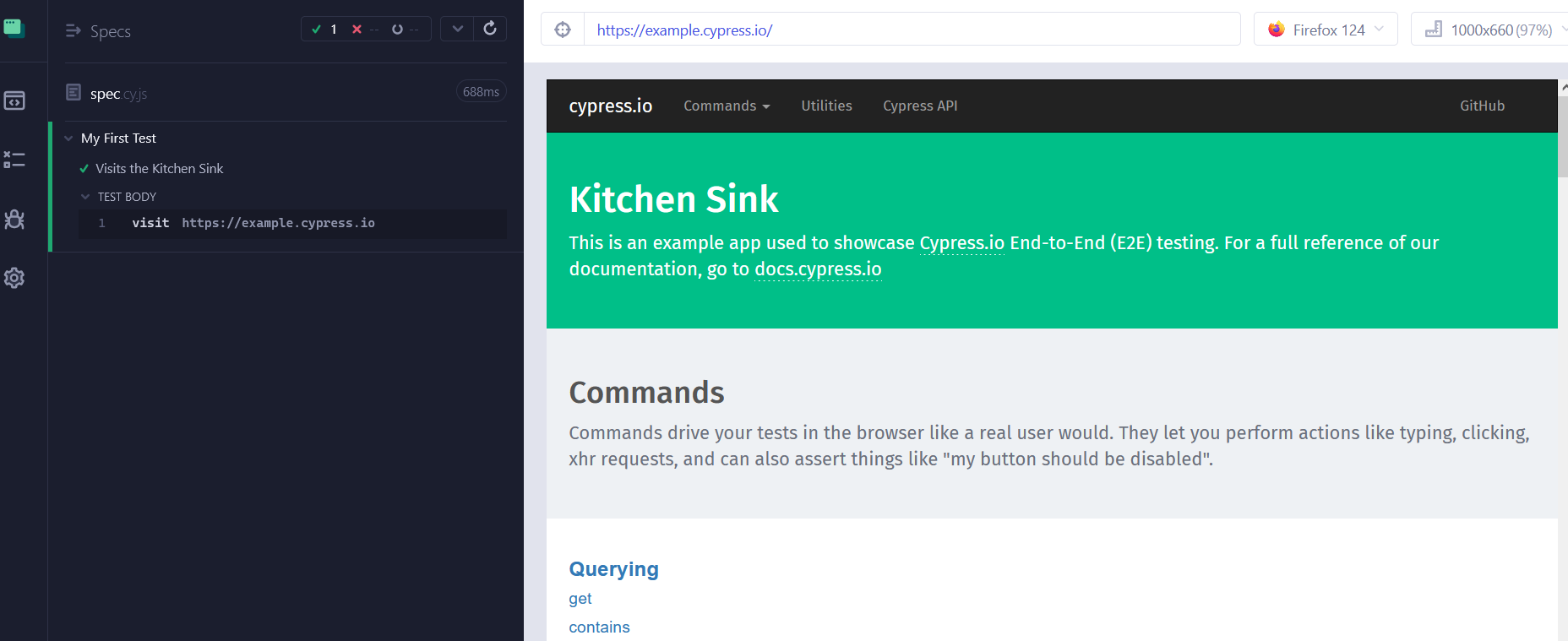
1. Перейдите на страницу. Мы можем передать URL, по которому хотим перейти [cy.visit()](https://docs.cypress.io/api/commands/visit). Давайте заменим наш предыдущий тест на приведенный ниже, который фактически позволяет перейти на страницу:



Сохраните файл и снова переключитесь на программу Cypress Test Runner. Вы можете заметить несколько моментов:

1. В журнале команд теперь отображается новое VISIT действие.
2. Приложение для кухонной мойки загружено на панель предварительного просмотра приложения.
3. Тест зеленый, хотя мы и не делали никаких утверждений.
4. VISIT отображается синее состояние ожидания, пока страница не завершит загрузку.

Если бы этот запрос вернулся с не 2xx кодом состояния, таким как 404 или 500, или если бы в коде приложения была ошибка JavaScript, тест был бы неудачным.



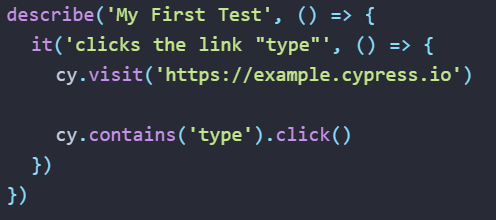
2. Запрос элемента. Теперь, когда у нас загружена страница, нам нужно предпринять некоторые действия над ней. Почему бы нам не щелкнуть ссылку на странице? Звучит достаточно просто, давайте поищем ту, которая нам нравится... как насчет type?

Чтобы найти этот элемент по его содержимому, мы будем использовать [cy.contains()](https://docs.cypress.io/api/commands/contains).

Давайте добавим это в наш тест и посмотрим, что получится:



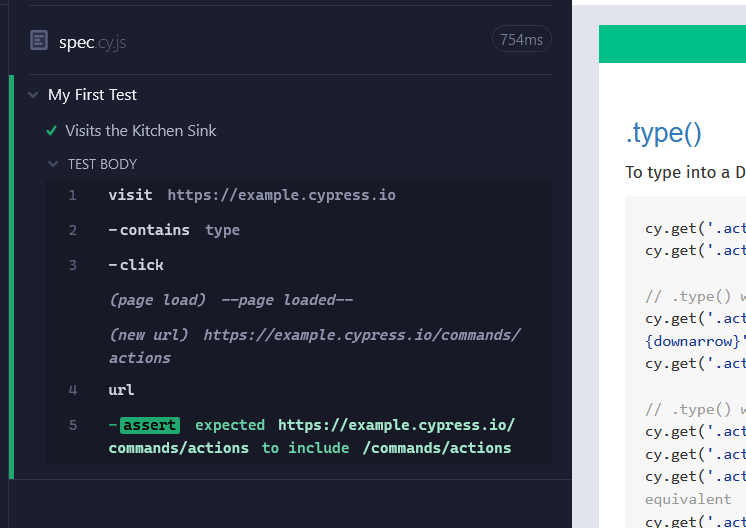
3. Щелкните элемент. Теперь мы хотим перейти по найденной ссылке. Как нам это сделать? Добавьте команду [.click()](https://docs.cypress.io/api/commands/click) в конец предыдущей команды, вот так:



4.Создайте утверждение. Давайте сделаем утверждение о чем-либо на новой странице, на которую мы перешли. Возможно, мы хотели бы убедиться, что новый URL соответствует ожидаемому URL. Мы можем сделать это, просмотрев URL-адрес и связав с ним утверждение с помощью [.should()](https://docs.cypress.io/api/commands/should).

Вот как это выглядит:



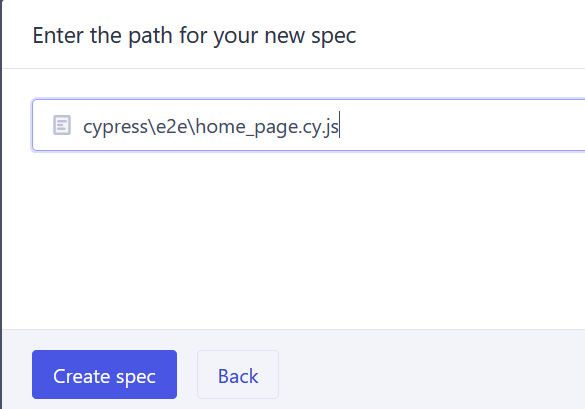


**Настраиваем Cypress под локальный сервер.**

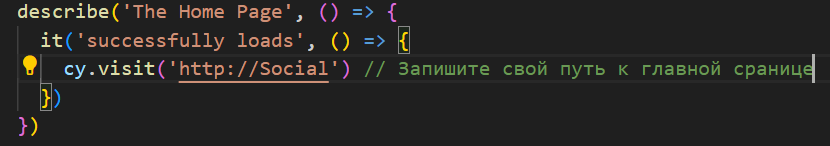
Запустите локальный сервер разработки, на котором размещено приложение.

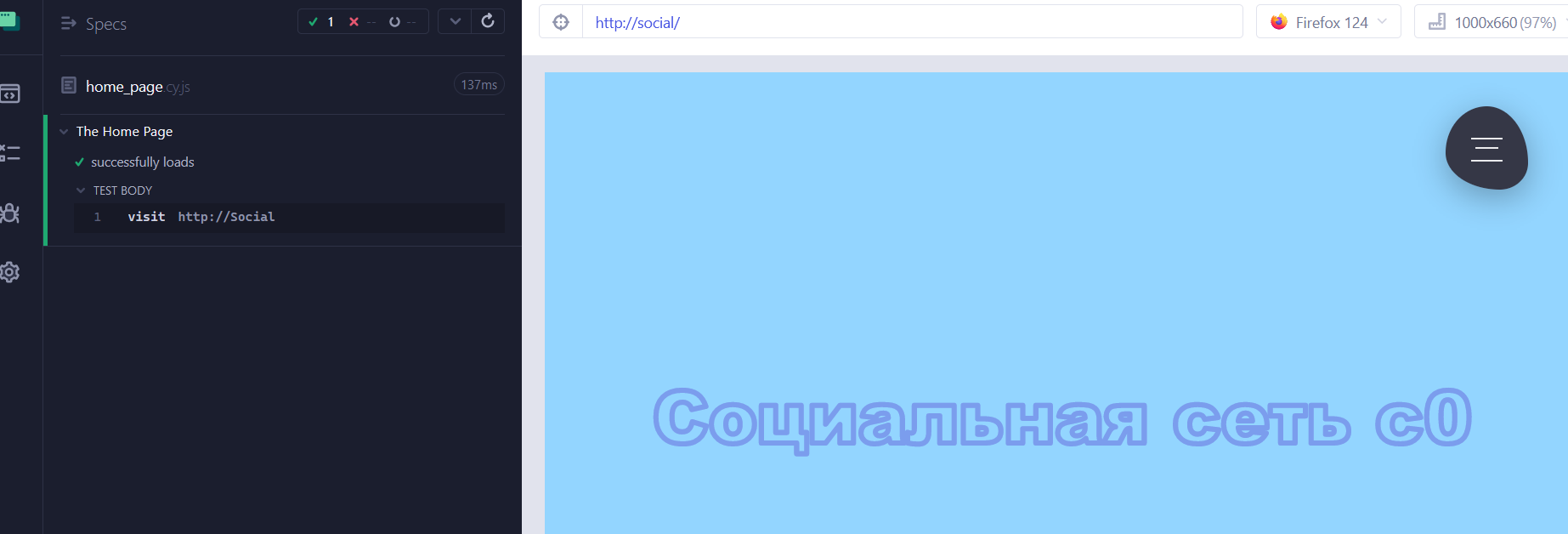
Это должно выглядеть примерно так [**http://localhost:8080**](http://localhost:8080/).

Теперь удалим ранее созданную спецификацию и создадим новую



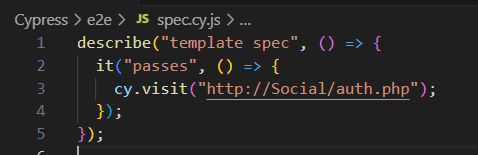
Теперь вам нужно добавить следующий код в ваш тестовый файл, чтобы перейти на ваш сервер:

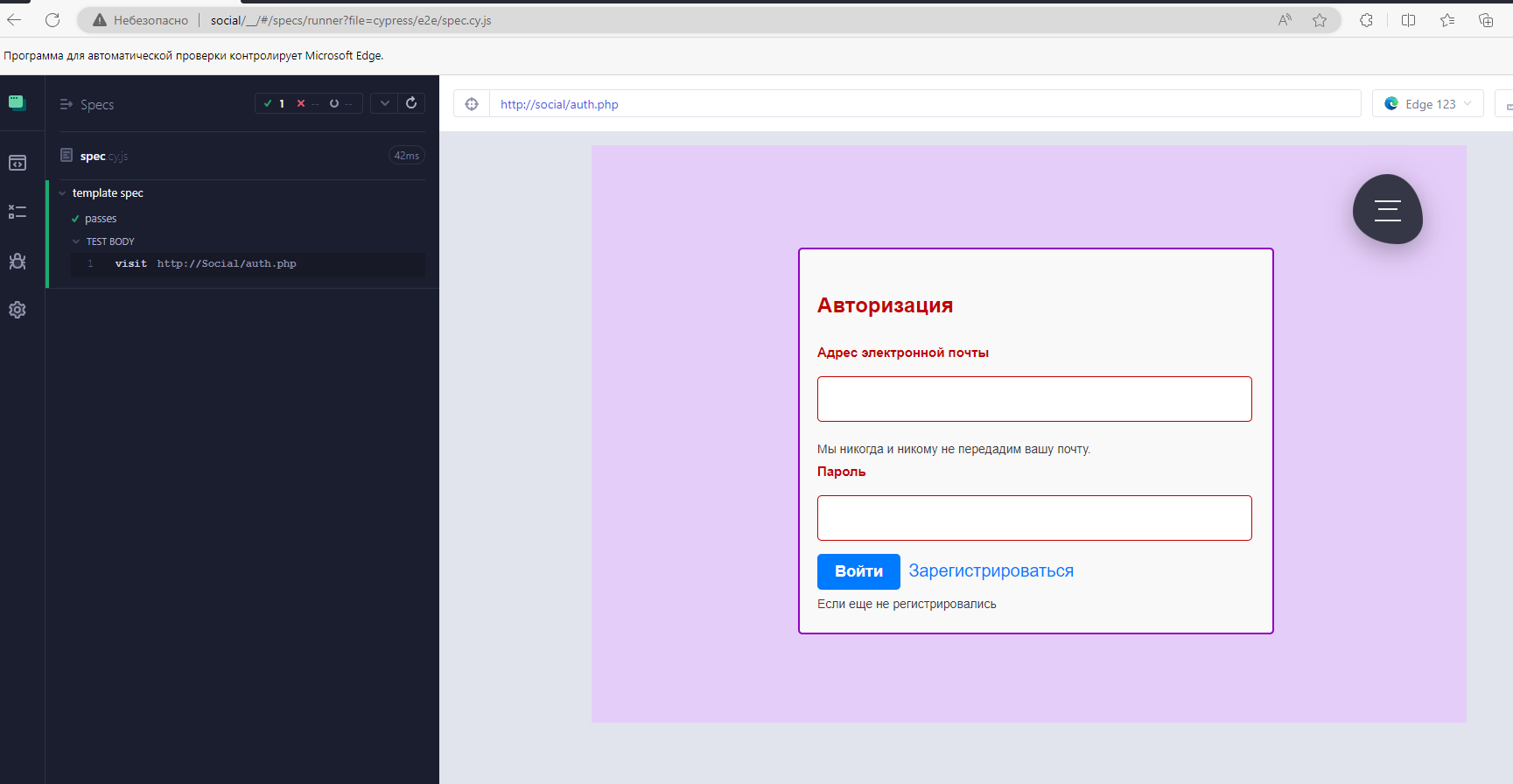




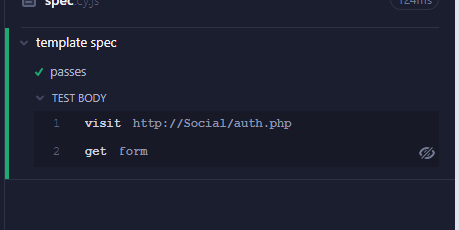
**Выполните следующие примеры:**

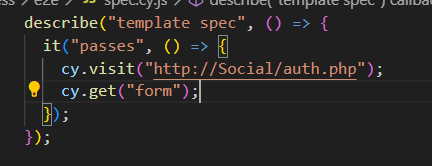
1. Посетите страницу авторизации <http://Social/auth.php>.



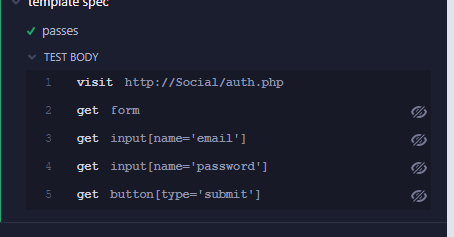


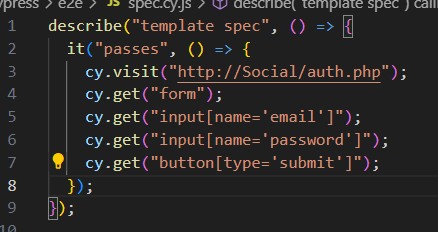
1. Проверьте наличие элемента form на этой странице. (cy.get("form");)





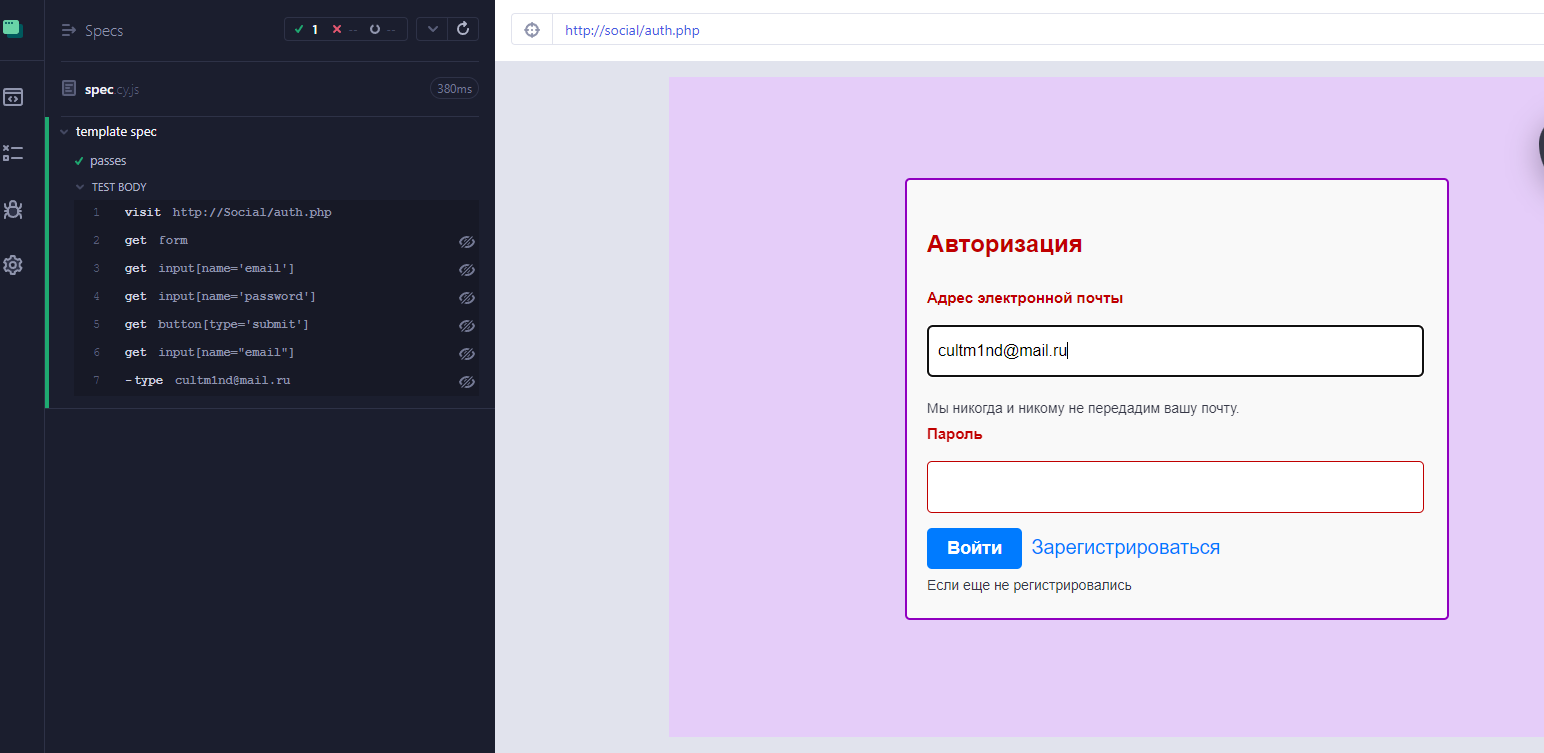
1. Проверьте наличие элементов для ввода логина и пароля, а также кнопка Войти.

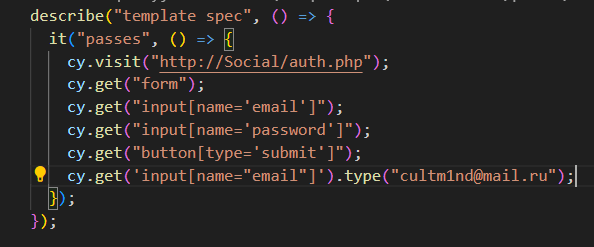




1. Проверьте, что пользователь может заполнить поле email cy.get('input[name="name"]').type("Molly");

Команда Cypress  **type**, которая вводит в input элемент указанный текст. Также обратите внимание на селектор CSS для получения элемента ввода.



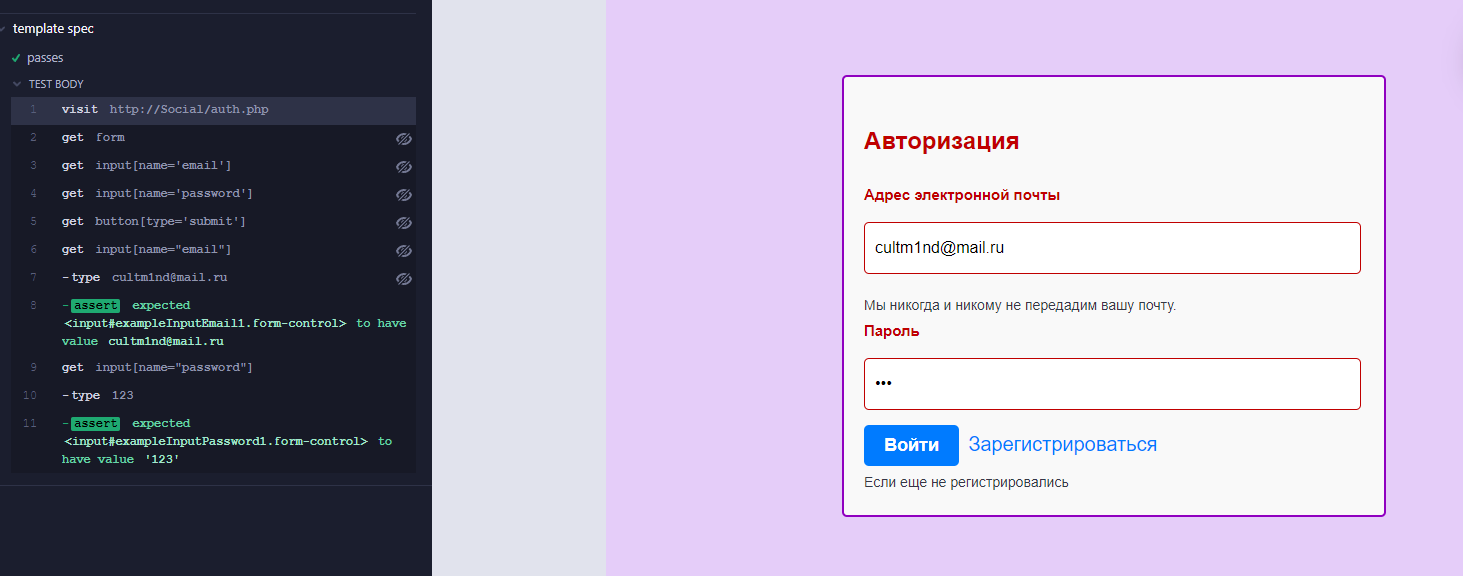


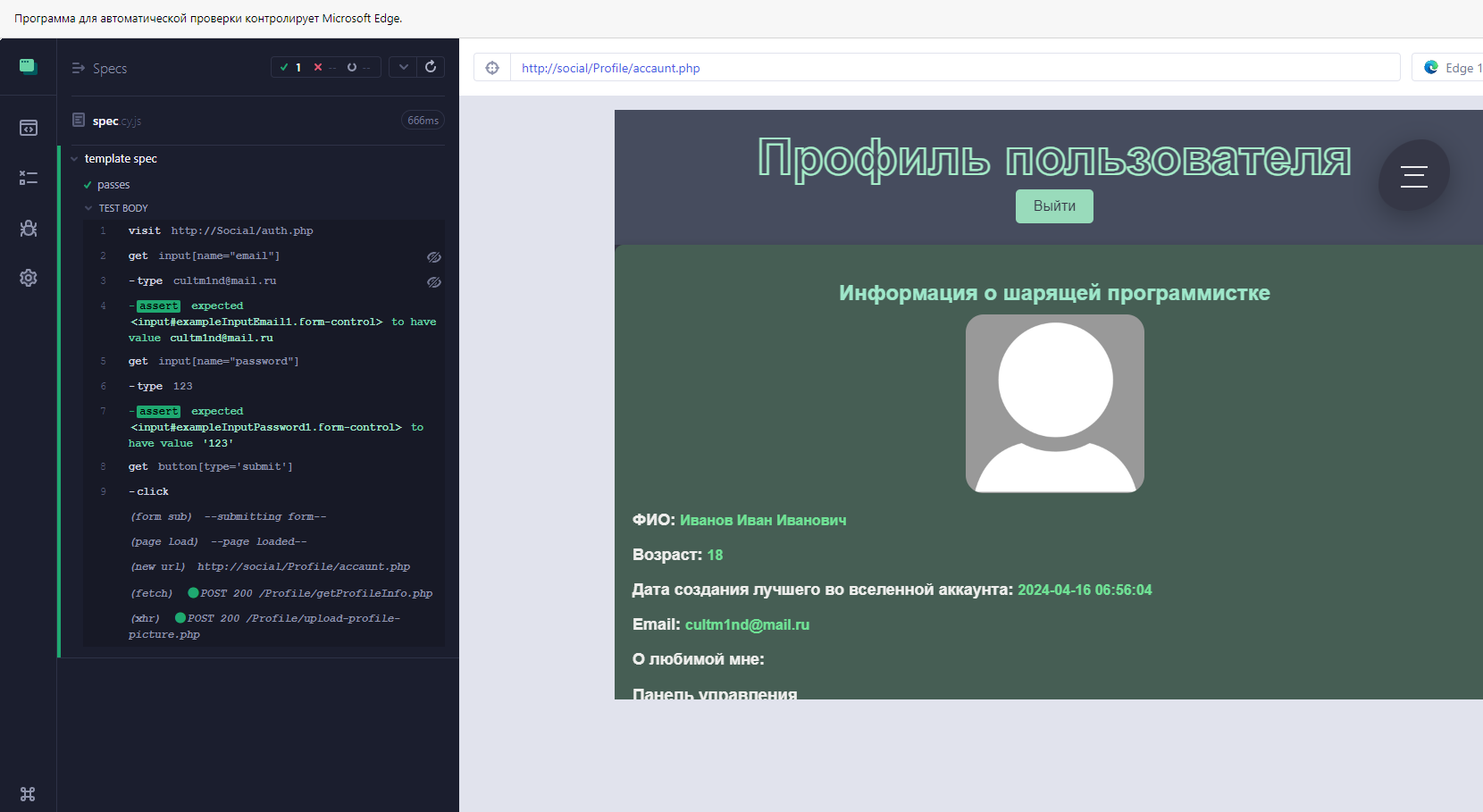
1. Добавим еще одну команду: **should**. Эта команда создает утверждение и используется, например, для проверки того, обновляет ли input элемент свое состояние должным образом:

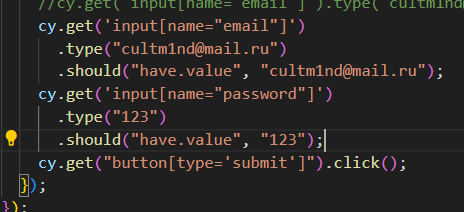
cy.get('input[name="name"]')

.type("Molly")

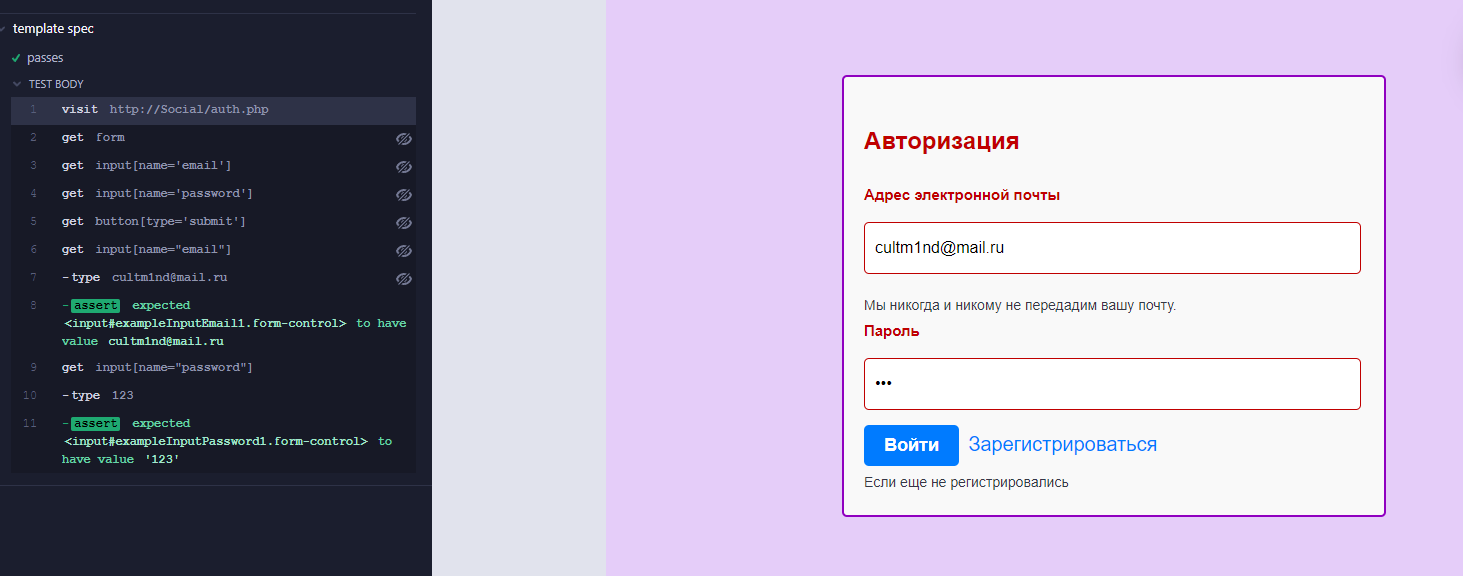
.should("have.value", "Molly");

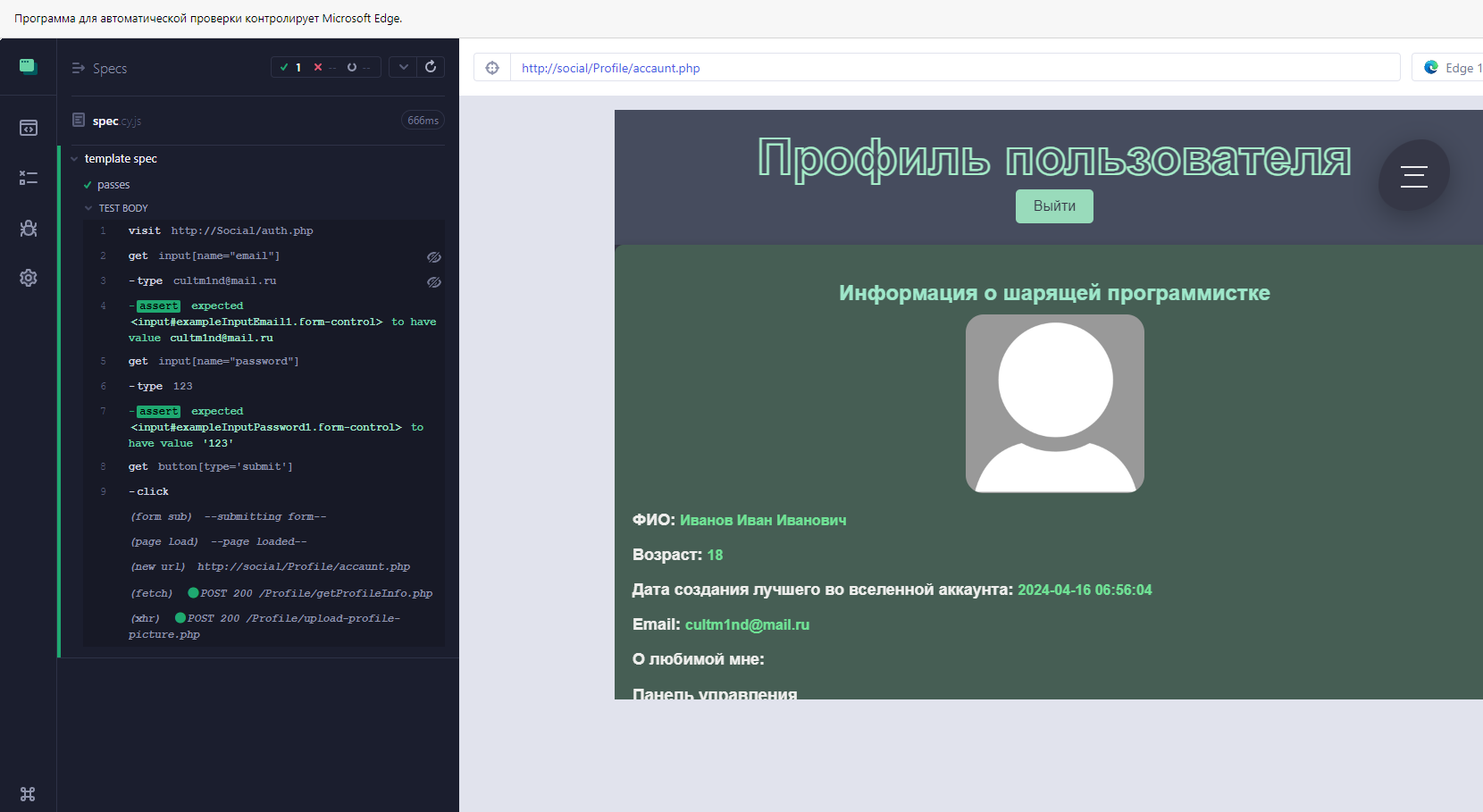


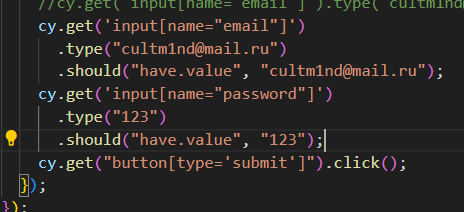




1. Аналогичным образом проверьте поле ввода пароля.

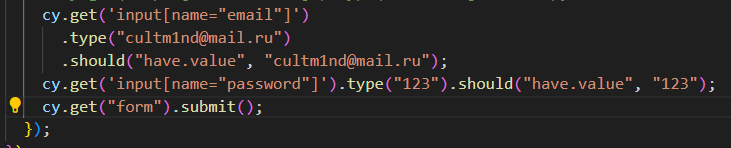


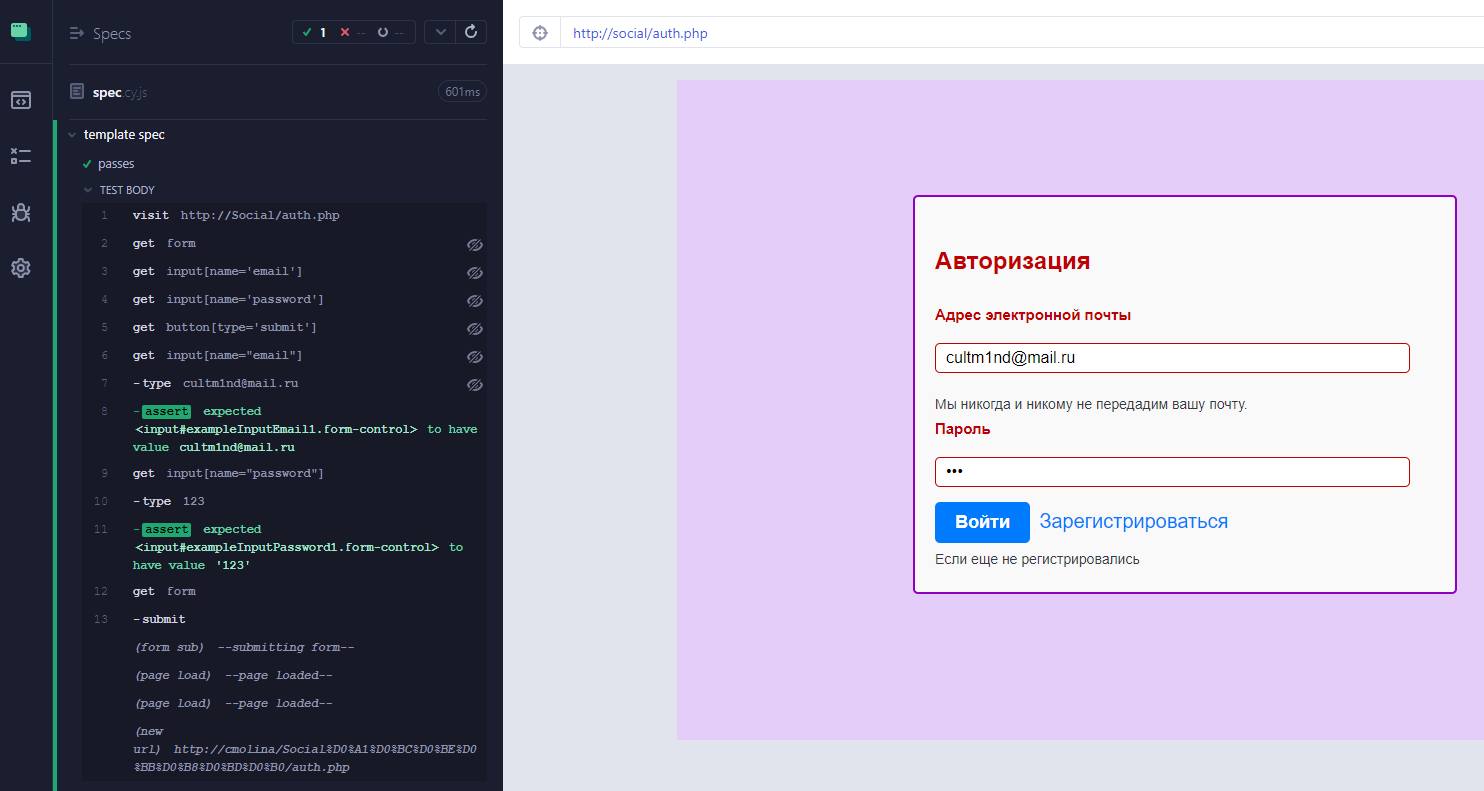


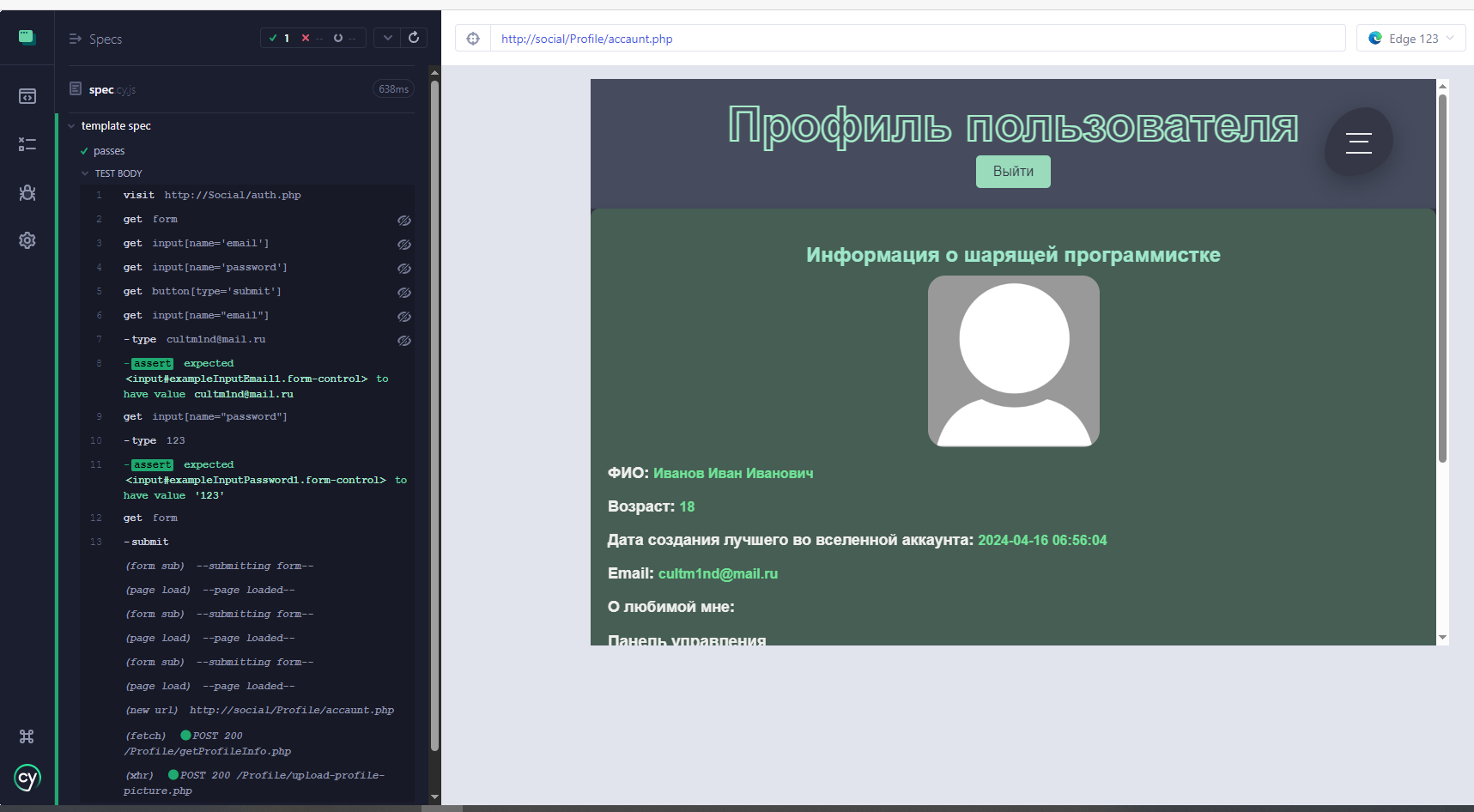
****

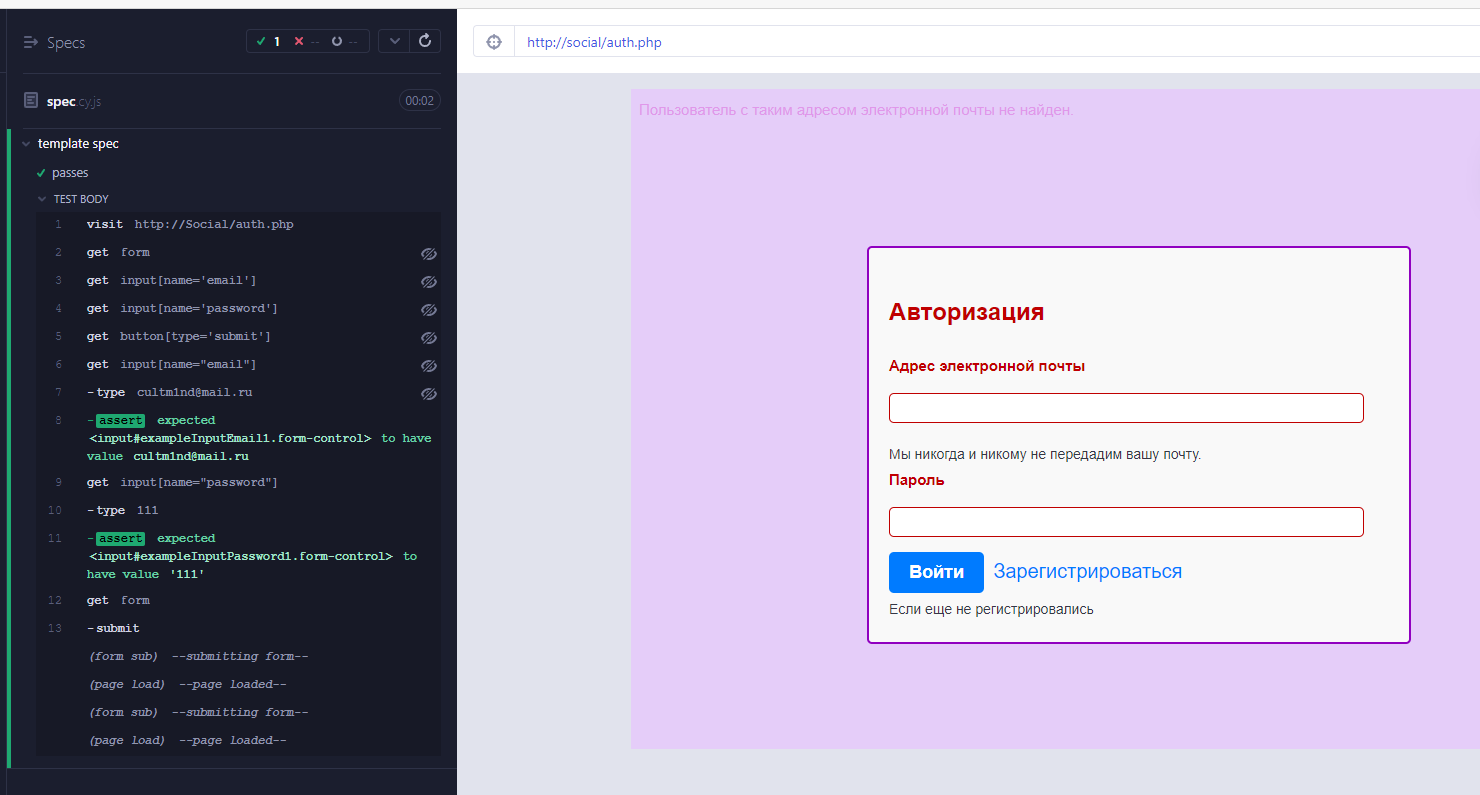
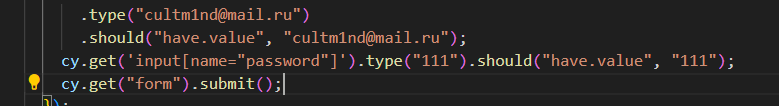
1. Протестируйте отправку формы с помощью **submit** (cy.get("form").submit();)

Проделайте тесты для удачного и неудачного входа.





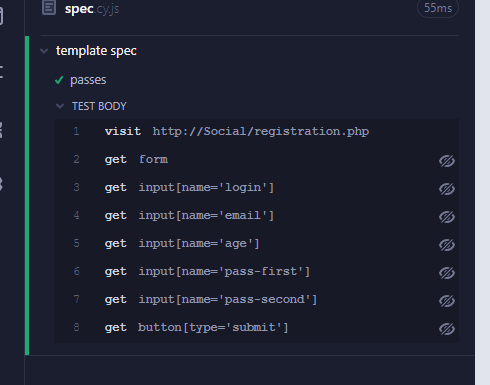




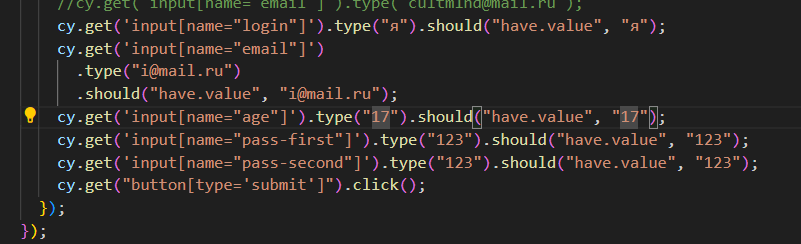
1. Разработайте тесты для проверки возможности регистрации на сайте.

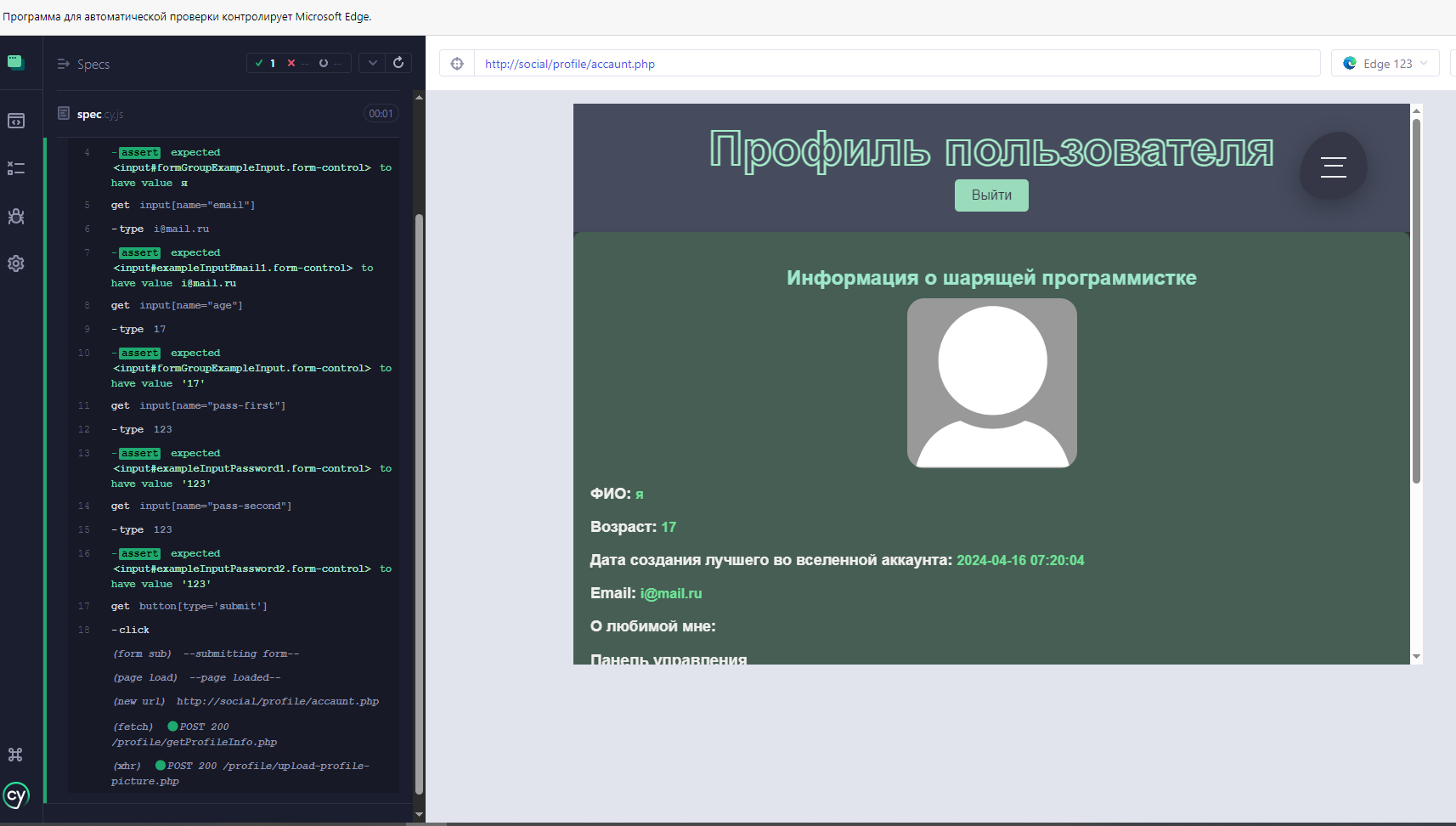
Получение форм





Возможность регистрации



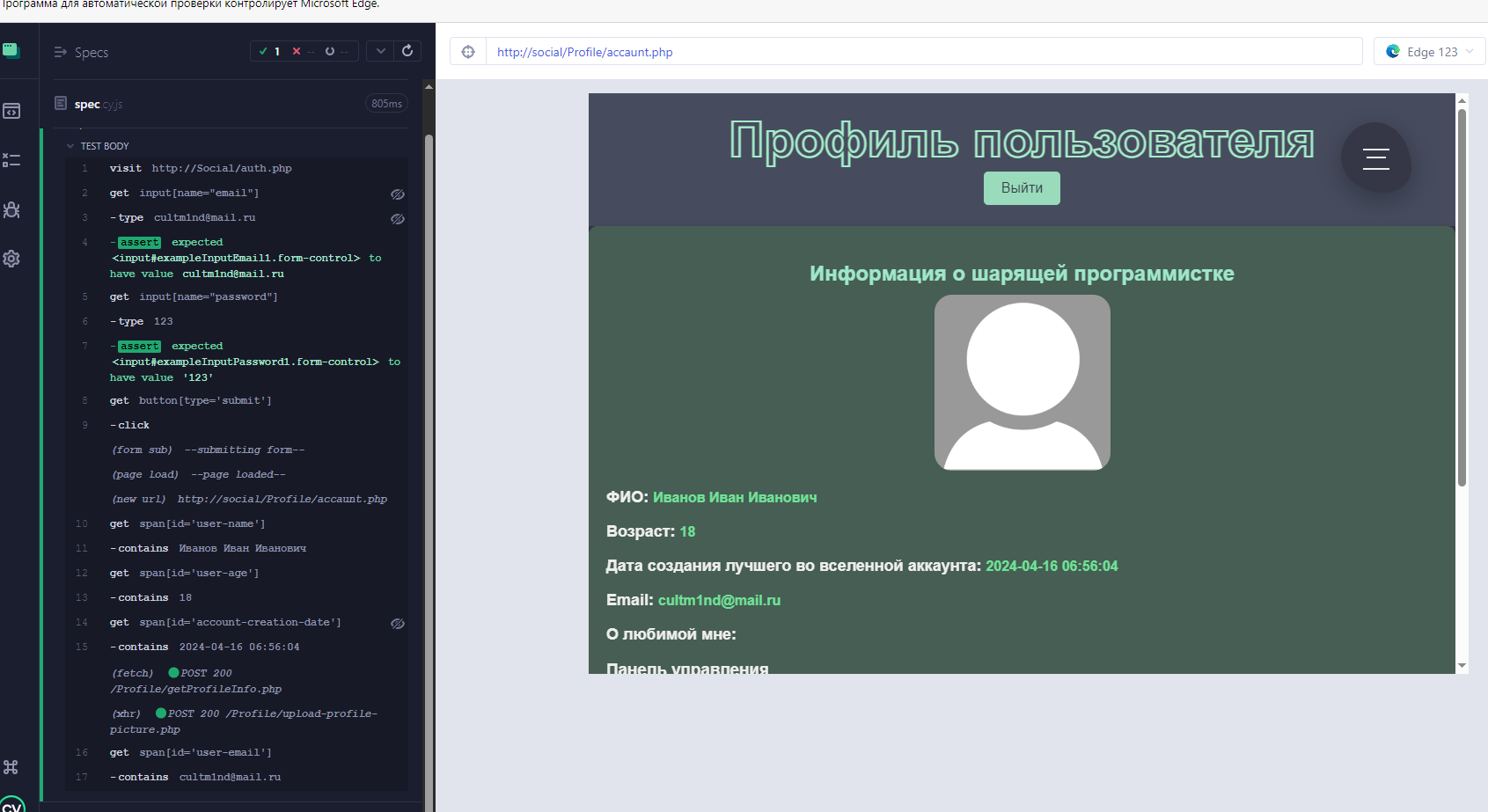


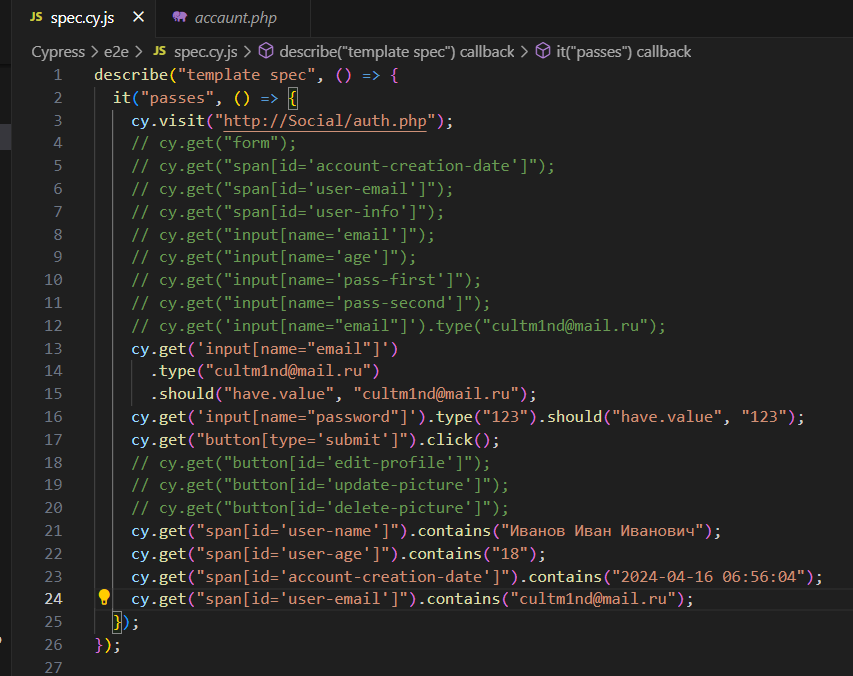
Подробнее об утверждениях Cypress написано здесь <https://docs.cypress.io/guides/core-concepts/introduction-to-cypress>

Весь набор тестовых команд здесь <https://example.cypress.io/>

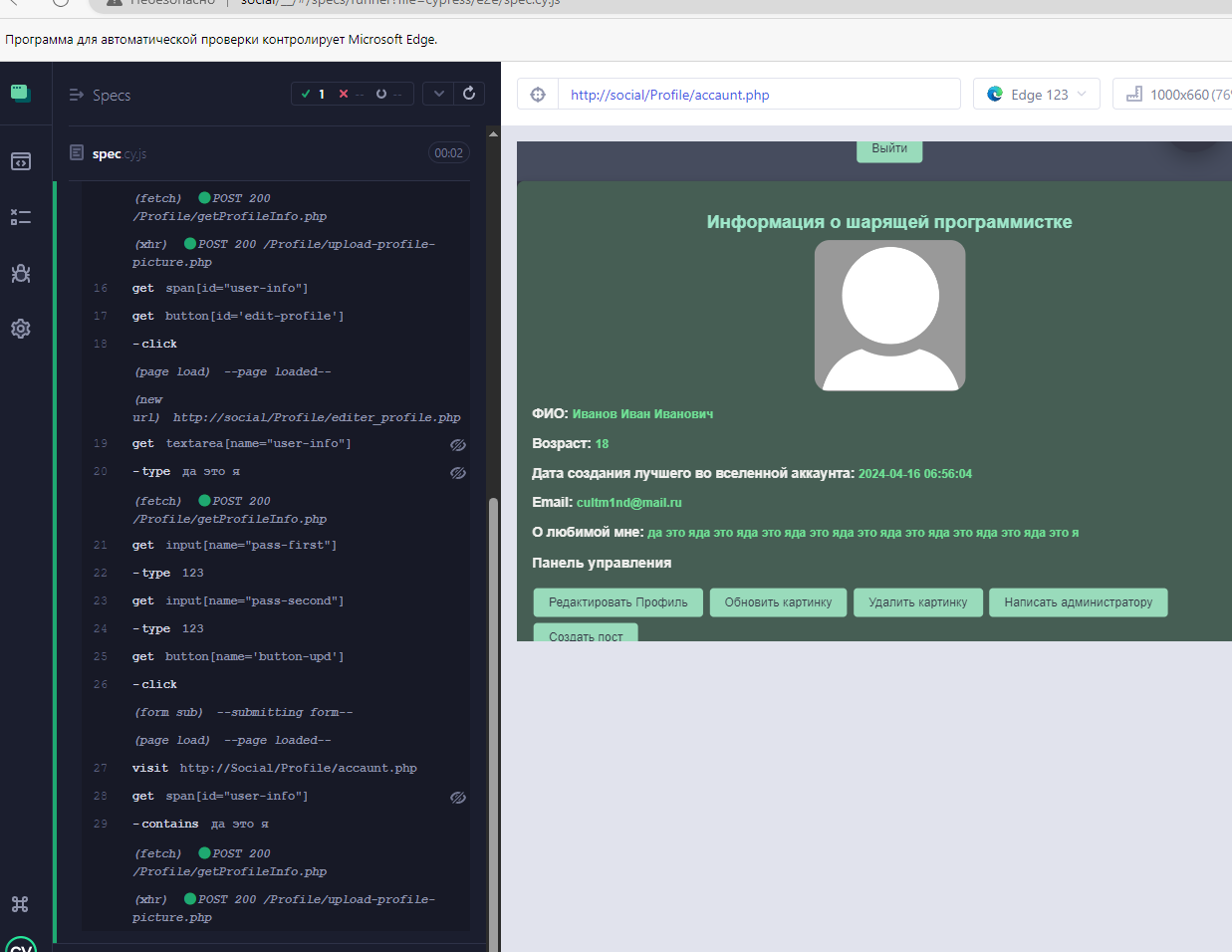
**Тестирование Личный кабинет**

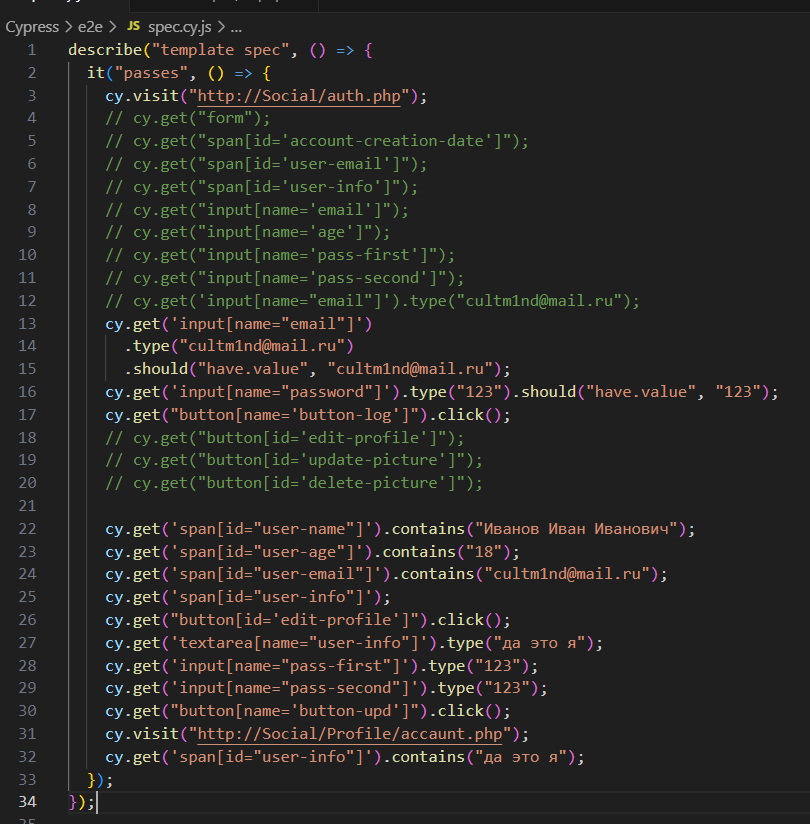
Наличие информации

****

****

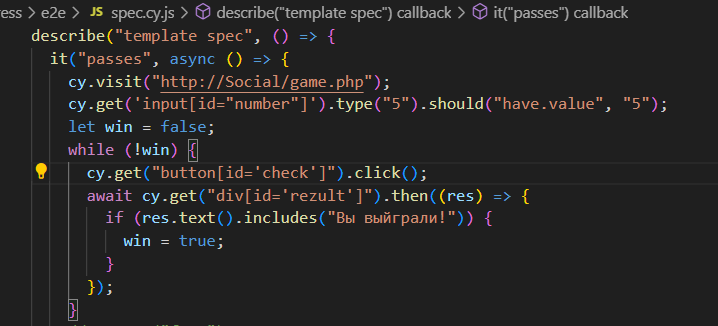
Редактирование





**Тестирование Чат**

**Тестирование Игра**

****

