Фуринов Дмитрий 4-105-1

Описание протестированных функций

В ходе тестирования были проверены следующие эндпоинты API Regres:

- 1. Получение списка пользователей (GET /api/users): возвращает список пользователей
- 2. Получение информации о конкретном пользователе (GET /api/users/{id}): возвращает данные о пользователе с указанным идентификатором.
- 3. **Создание нового пользователя (POST /api/users)**: позволяет добавить нового пользователя, передавая его имя и должность в теле запроса.
- 4. **Обновление данных пользователя (PUT /api/users/{id})**: обновляет информацию о пользователе с указанным идентификатором.
- 5. **Удаление пользователя (DELETE /api/users/{id})**: удаляет пользователя с заданным идентификатором.
- 6. **Регистрация нового пользователя (POST /api/register)**: позволяет зарегистрировать нового пользователя, передавая его email и пароль.
- 7. **Авторизация пользователя (POST /api/login)**: выполняет вход пользователя по email и паролю.

Список тест-кейсов

Α	В	С	D	Е
Nº	Объект тестирования	Название теста	Шаги	Ожидаемый результат
1	Веб-приложение	Поиск товаров	 Открыть главную страницу Wildberries Ввести "флешка" в поле поиска Нажать Enter 	Отображаются карточки товаров с запросом "флешка"
2	Веб-приложение	Добавление товара в корзину	 Найти товар Нажать "Добавить в корзину" Перейти в корзину 	Товар присутствует в корзине
3	Веб-приложение	Удаление товара из корзины	 Добавить товар Перейти в корзину Нажать "Удалить" для товара 	Корзина пуста
4	Веб-приложение	Авторизация	 Перейти на страницу логина Ввести корректные данные Нажать "Войти" 	Пользователь успешно авторизован
5	Веб-приложение	Регистрация	 Перейти на страницу регистрации Ввести данные нового пользователя Нажать "Зарегистрироваться" 	Пользователь успешно зарегистрирован
6	Веб-приложение	Оформление заказа	Добавить товар в корзину Перейти в Сheckout Заполнить форму заказа Подтвердить заказа	Заказ оформлен, появляется подтверждение
7	API (ReqRes)	Получение списка пользователей	GET /api/users?page=2	Код 200, в ответе присутствует список пользователей
8	API (ReqRes)	Получение конкретного пользователя	GET /api/users/2	Код 200, в ответе данные пользователя c id = 2
9	API (ReqRes)	Создание пользователя	POST /api/users c телом: { "name": "John", "job": "QA" }	Код 201, данные созданного пользователя возвращены
10	API (ReqRes)	Обновление данных пользователя	PUT /api/users/2 с телом: { "name": "John", "job": "Lead QA" }	Код 200, данные обновлены
11	API (ReqRes)	Удаление пользователя	DELETE /api/users/2	Код 204
12	API (ReqRes)	Авторизация (негативный сценарий)	POST /api/login c телом: { "email": "test@mail.com", "password": "1234" }	Код 400, ошибка авторизации (некорректные данные)

https://docs.google.com/spreadsheets/d/14nhW-wr92LsCgyyHwdDa YOer0I5O2Q2rCCUMoTG6Tg/edit?usp=sharing

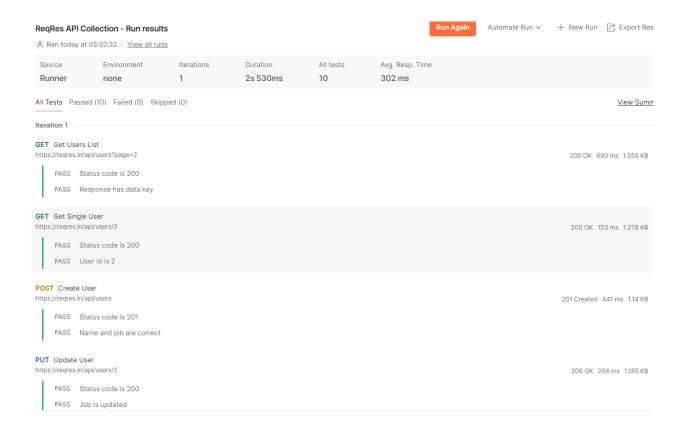
Автоматизированные тесты

Остальной код в репозитории: https://github.com/FurySun-Dev/PO 6 last

Коллекция Postman (.json) + автотесты

PS C:\Users\fdvsa\OneDrive\Pa6очий стол\66> [

В репозитории: https://github.com/FurySun-Dev/PO 6 last



```
PUT Update User
https://regres.in/api/users/2

PASS Status code is 200
PASS Job is updated

DELETE Delete User
https://regres.in/api/users/2

PASS Status code is 204

PASS Status code is 400
```

Скриншоты и логи ошибок