

Пермяков Вячеслав Николаевич, образование высшее, работаю программистом 1с (стаж более 15 лет).

Конфигурации:

1. УПП (доработанная) 8.2
2. Бухгалтерия 8.3
3. ЗУП 8.3

Закончил курсы:



Курсы QA-тестировщик с нуля до Junior QA за 6 месяцев

1. Тестирование web-приложений
2. Devtools, вкладки и использование
3. web-верстка, базовый уровень
4. Полезные инструменты для разработчика
5. Основные запросы по работе с данными
6. Проектирование и оптимизация реляционной базы данных
7. Тестирование API
8. Основы Python

ПРИМЕРЫ ТЕСТИРОВАНИЯ

- Приложение 1 (стр. 2 -.24)

- Приложение 2 (стр. 25-33)

- Тестирование API

Источник сервер Swagger Petstore (Базовый URL: [petstore.swagger.io /v2](https://petstore.swagger.io/v2)). Интеграционное тестирование API проводим с помощью Postman. Скачиваем необходимую версию (в зависимости от операционной системы) с официального сайта postman.com и устанавливаем приложение на компьютер. После установки создаем рабочее пространство, коллекцию и окружение. Составим план тестирования:

Вызов API	Действие
Позитивное тестирование	
POST create user	Создаем пользователя с валидными обязательными параметрами и использованием встроенной функция генерации случайных данных, что позволяет избежать парадокс «пестецида».
POST creates list with given input array	Создаем список пользователей с заданным входным массивом (два пользователя).
GET Logs user into the system	Регистрируем пользователя в системе.
GET user by user name	Находим пользователя по логину
PUT Updated user	Меняем данные существующего пользователя
Delete user	Удаляем существующего пользователя
Негативное тестирование	
GET user by user name	Попытка найти удаленного пользователя по логину
POST create user	Попытка создать уже существующего пользователя
Delete user	Попытка удалить несуществующего пользователя
POST create user	Попытка создать пользователя с неправильно заполненным адресом в заголовке.

Позитивное тестирование				
Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
TC/VPN0001	Высокий	Создание пользователя с использованием функции генерации случайных данных.	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении

				RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger добавляем переменную baseUrl и заносим значение		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос create user в коллекции SwaggerPetstore .		Открывается запрос с типом Post и заголовком {{baseUrl}}/user (см. рис. 1)	Совпадает с ожидаемым	Пройден успешно
3.Заполняем данные рандомными значениями.		Данные генерируются автоматически (см. рис. 1).	Совпадает с ожидаемым	Пройден успешно
4.Сохраняем и запускаем запрос		Должен загрузиться ответ с кодом состояния 200 и всеми обязательными параметрами, скорость ответа высокая (см. рис. 1)	Совпадает с ожидаемым	Пройден успешно
Создал: Пермяков Вячеслав 04.11.2024			Новый тест-кейс	

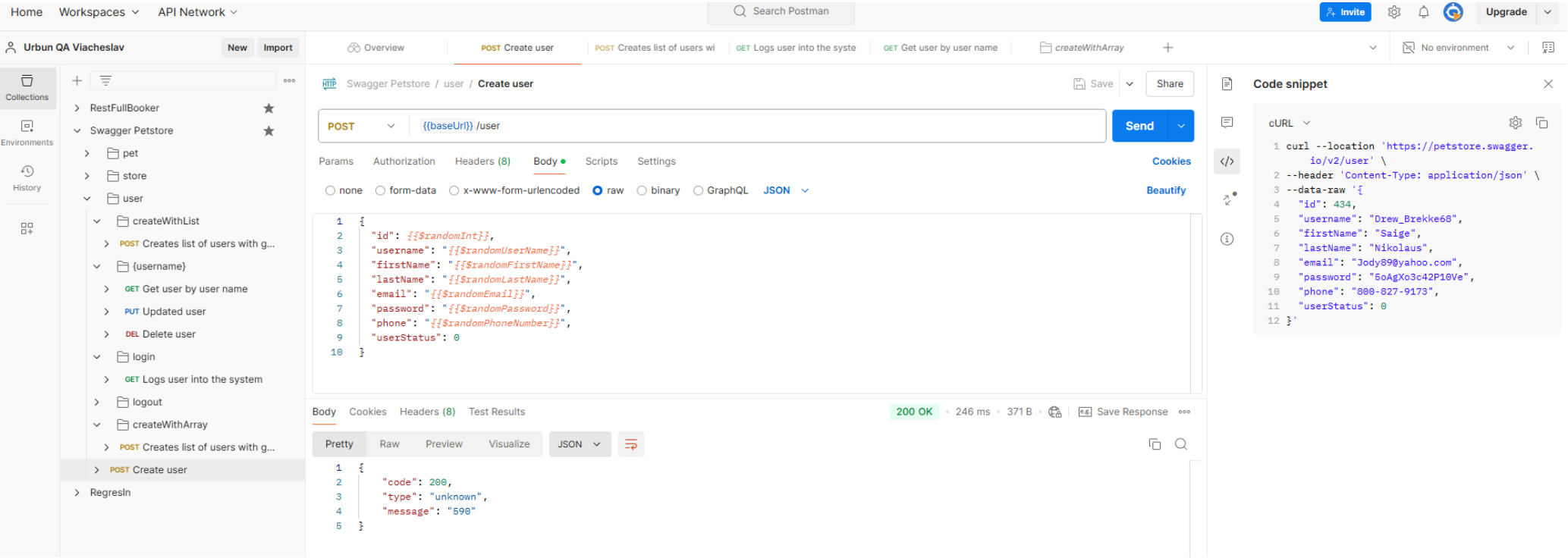


Рис. 1

Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
ТС/VPN0002	Высокий	Создание списка пользователей с использованием массива	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус

1. В окружение RestSwagger проверяем наличие переменной baseUrl	При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос “Creates list with given input array” в коллекции SwaggerPetstore	Открывается запрос с типом Post и заголовком {{baseUrl}}/user (см. рис. 2)	Совпадает с ожидаемым	Пройден успешно
3.Заполняем данные.	Данные заносятся из входного массива (см. рис. 2)	Совпадает с ожидаемым	Пройден успешно
4.Сохраняем и запускаем запрос	Должен загрузиться ответ с кодом состояния 200 и всеми обязательными параметрами, скорость ответа высокая (см. рис. 2)	Совпадает с ожидаемым	Пройден успешно
Создал: Пермяков Вячеслав 04.11.2024		Новый тест-кейс	

Home Workspaces API Network Search Postman Invite Upgrade

Urban QA Viacheslav New Import Overview POST Create user POST Creates list of users wi GET Logs user into the syste GET Get user by user name createWithArray No environment

Swagger Petstore / user / createWithArray / Creates list of users with given input array Save Share

POST {{baseUrl}}/user/createWithArray Send

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  [
2    {
3      "id": 3,
4      "username": "Tyson_Wehner89",
5      "firstName": "Gerard",
6      "lastName": "Stoltenberg",
7      "email": "Cleora.Quigley31@hotmail.com",
8      "password": "9yUc485yxwI24DK",
9      "phone": "894-888-8878",
10     "userStatus": 0
11   },
12   {
13     "id": 3,
14     "username": "Priscilla38",
15     "firstName": "Linnea",
16     "lastName": "Dickinson",
17     "email": "Green_Mosciski@hotmail.com",
18     "password": "Nd_2TJCHC2G8Jh8",
19     "phone": "381-823-0958",
20     "userStatus": 1
21   }
22 ]

```

Body Cookies Headers (8) Test Results 200 OK • 798 ms • 370 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "code": 200,
3    "type": "unknown",
4    "message": "ok"
5  }

```

Code snippet

cURL

```

1 curl --location 'https://petstore.swagger.io/v2/user/createWithArray' \
2 --header 'Content-Type: application/json' \
3 --data-raw '[
4   {
5     "id": 3,
6     "username": "Tyson_Wehner89",
7     "firstName": "Gerard",
8     "lastName": "Stoltenberg",
9     "email": "Cleora.Quigley31@hotmail.com",
10    "password": "9yUc485yxwI24DK",
11    "phone": "894-888-8878",
12    "userStatus": 0
13  },
14  {
15    "id": 3,
16    "username": "Priscilla38",
17    "firstName": "Linnea",
18    "lastName": "Dickinson",
19    "email": "Green_Mosciski@hotmail.com",
20    "password": "Nd_2TJCHC2G8Jh8",
21    "phone": "381-823-0958",
22    "userStatus": 1
23  }
24 ]'

```

Рис. 2

Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
ТС/VPN0003	Высокий	Регистрация пользователя	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружение RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger проверяем наличие переменной baseUrl		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос “Logs user into the system” в коллекции SwaggerPetstore		Открывается запрос с типом GET и заголовком {{baseUrl}}/user/Login (см. рис. 3)	Совпадает с ожидаемым	Пройден успешно
3.Заполняем логин и пароль через параметры.		Данные заносятся при соблюдении оговоренных ограничений (см. рис. 3)	Совпадает с ожидаемым	Пройден успешно
4.Сохраняем и запускаем запрос		Должен загрузиться ответ с кодом состояния 200 и всеми обязательными параметрами, скорость ответа высокая (см.	Совпадает с ожидаемым	Пройден успешно

	рис. 3)		
Создал: Пермяков Вячеслав 04.11.2024		Новый тест-кейс	

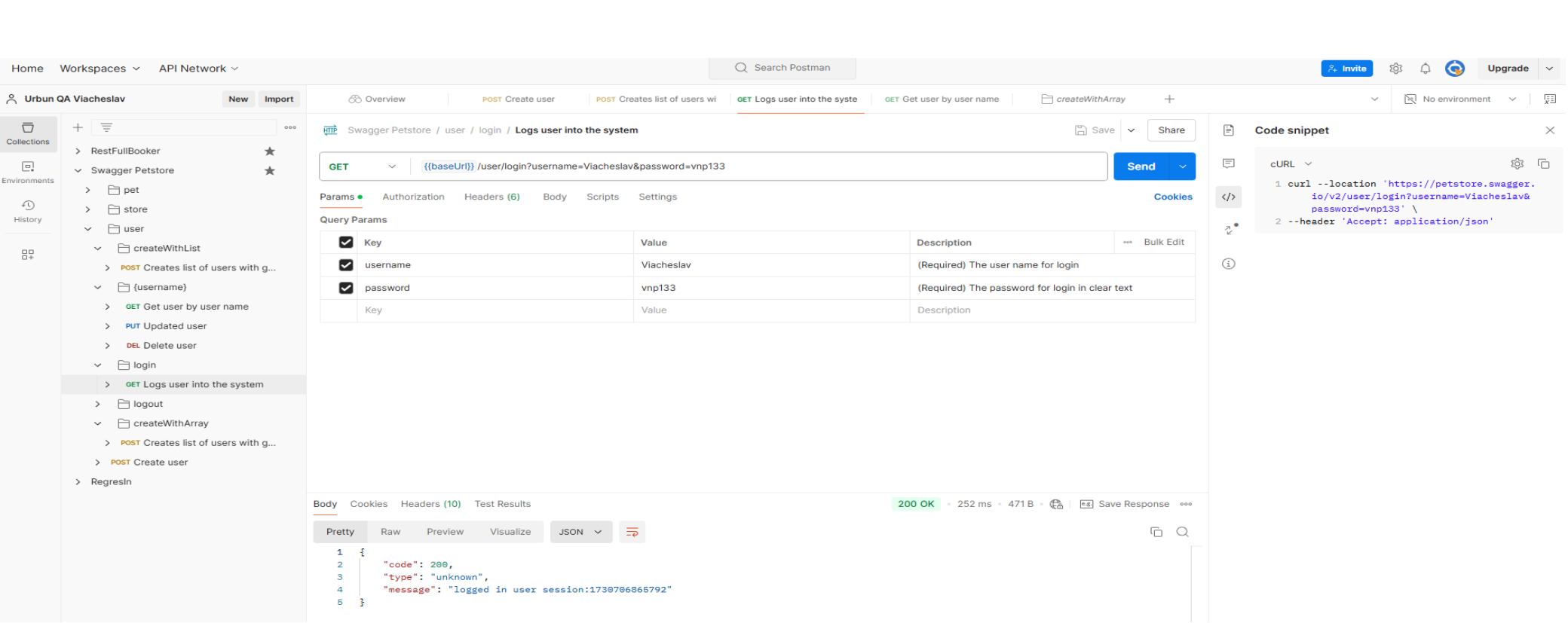


Рис. 3

Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
ТС/VPN0004	Высокий	Получить пользователя по логину	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger проверяем наличие переменной baseUrl		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос “Get user by user name” в коллекции SwaggerPetstore		Открывается запрос с типом Get и заголовком {{baseUrl}}/user/:username (см. рис. 4)	Совпадает с ожидаемым	Пройден успешно
3.В параметры заносим существующий логин.		Значение существующего логина заносится. (см. рис. 4)	Совпадает с ожидаемым	Пройден успешно

4.Сохраняем и запускаем запрос	Должен загрузиться пользователь с введенным логином, код ответа 200 и всеми обязательными параметрами, скорость ответа высокая (см. рис. 4)	Совпадает с ожидаемым	Пройден успешно
Создал: Пермяков Вячеслав 04.11.2024		Новый тест-кейс	

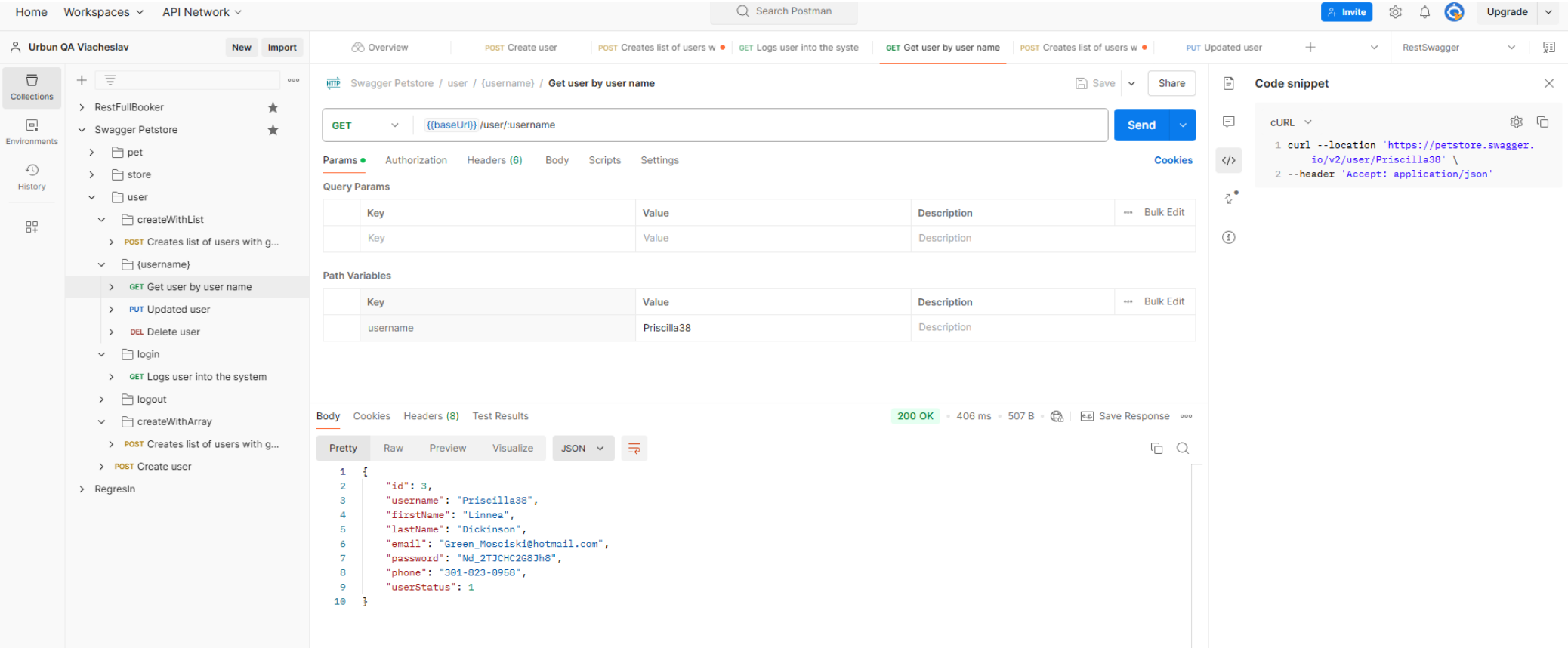


Рис. 4

Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
ТС/VPN0005	Высокий	Меняем данные существующего пользователя	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger проверяем наличие переменной baseUrl		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос "Updated user" в коллекции SwaggerPetstore		Открывается запрос с типом Put и заголовком {{baseUrl}}/user/:username (см. рис. 5)	Совпадает с ожидаемым	Пройден успешно
3.В параметры заносим существующий логин.		Значение существующего логина заносится. (см. рис. 5)	Совпадает с ожидаемым	Пройден успешно
4.Меняем значение password		Значение меняется, было «пустое место», стало «жаль» (см. рис. 6)	Совпадает с ожидаемым	Пройден успешно
5.Сохраняем и запускаем		Должен загрузиться пользователь с введенным	Совпадает с ожидаемым	Пройден успешно

запрос	логином, и измененным значением passwd код ответа 200 и всеми обязательными параметрами, скорость ответа высокая (см. рис. 6)		
Создал: Пермяков Вячеслав 04.11.2024		Новый тест-кейс	

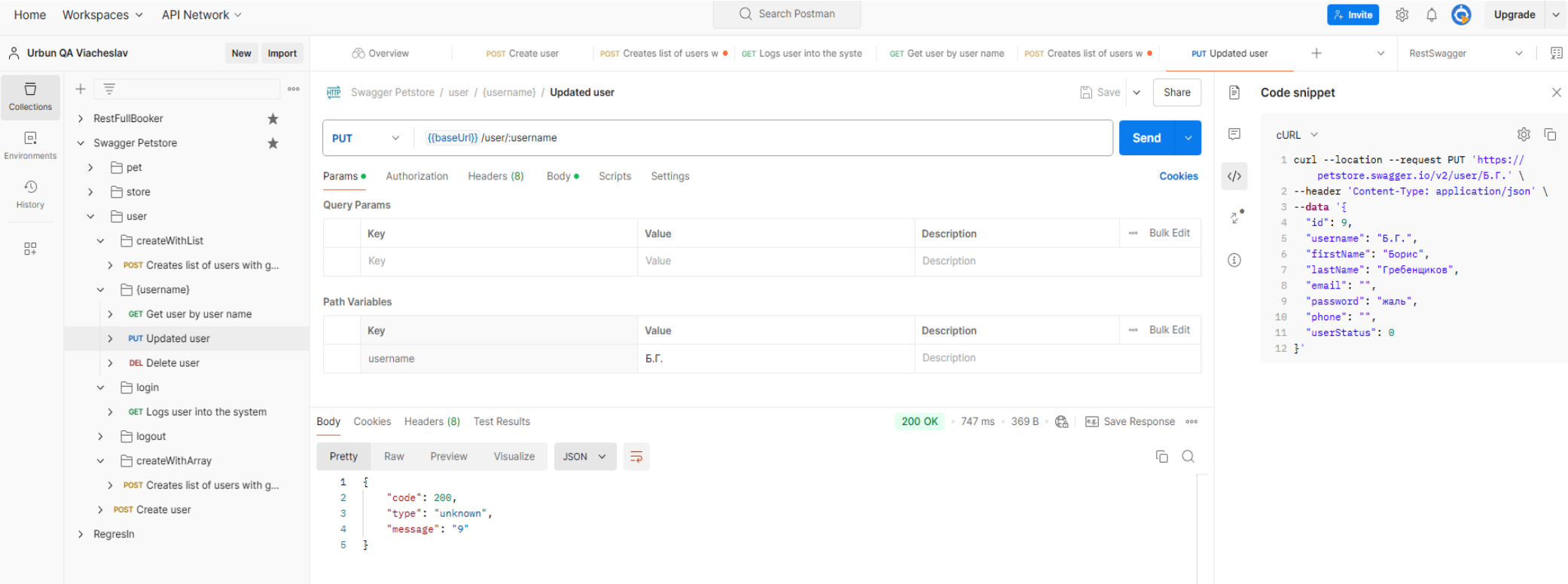


Рис. 5

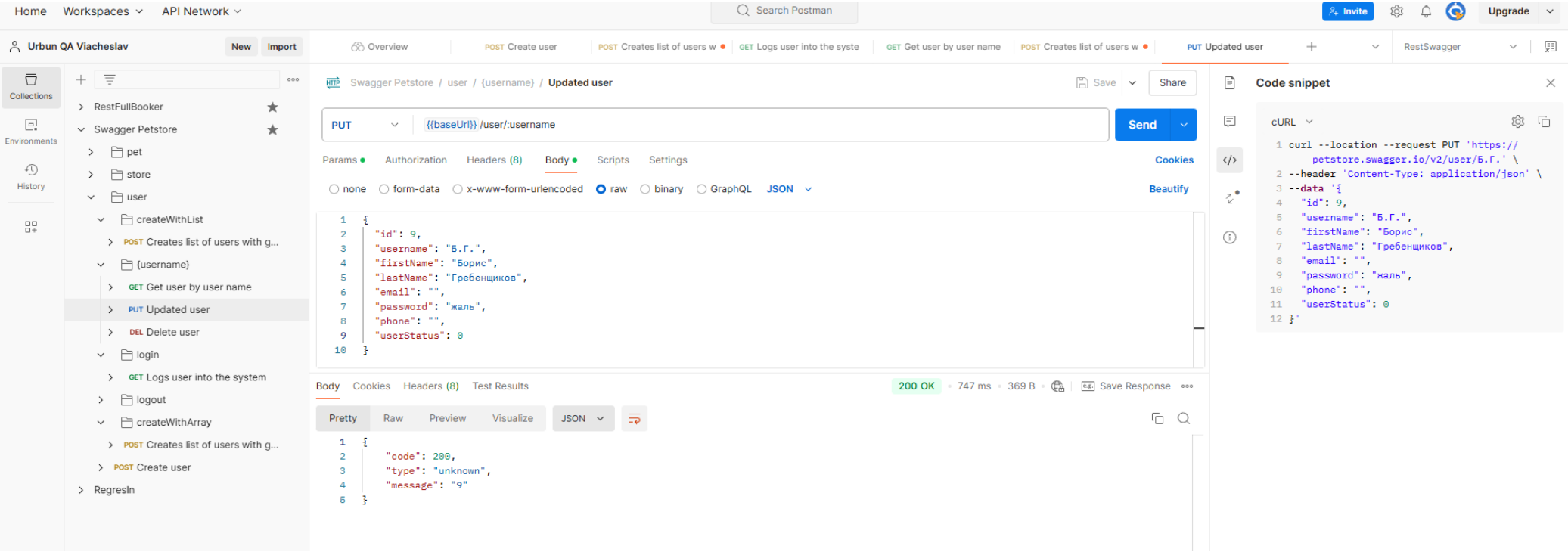


Рис. 6

Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
----------------------------------	-----------	---------------------	----------	-------------

TC/VPN0006	Высокий	Удаляем существующего пользователя	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger проверяем наличие переменной baseUrl		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос “Delete user” в коллекции SwaggerPetstore		Открывается запрос с типом Delete и заголовком {{baseUrl}}/user/:username (см. рис. 7)	Совпадает с ожидаемым	Пройден успешно
3.В параметры заносим существующий логин.		Значение существующего логина заносится. (см. рис. 7)	Совпадает с ожидаемым	Пройден успешно
4.Сохраняем и запускаем запрос		Пользователь с введенным логином удаляется, код ответа 200 и всеми обязательными параметрами, скорость ответа высокая (см. рис. 7)	Совпадает с ожидаемым	Пройден успешно
Создал: Пермяков Вячеслав 04.11.2024			Новый тест-кейс	

HomeWorkspaces ▾API Network ▾

Search Postman

Invite⚙️🔔🔄Upgrade ▾

Urbun QA ViacheslavNewImport

OverviewPOST Create userPOST Creates list of user. •GET Logs user into the syGET Get user by user namPOST Creates list of user. •PUT Updated userDEL Delete user+ ▾RestSwagger ▾📄

🗑️Collections

+

☰

...

RestFullBooker★

Swagger Petstore★

pet

store

user

createWithList

POST Creates list of users with g...

(username)

GET Get user by user name

PUT Updated user

DEL Delete user

login

GET Logs user into the system

logout

createWithArray

POST Creates list of users with g...

POST Create user

Regreslin

Swagger Petstore / user / {username} / Delete user

Save ▾Share

DELETE ▾{{baseUrl}}/user/username

Send ▾

Params •AuthorizationHeaders (5)BodyScriptsSettings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Path Variables

	Key	Value	Description	...	Bulk Edit
	username	4X	(Required) The name that needs to be deleted		

BodyCookiesHeaders (8)Test Results

200 OK • 1313 ms • 372 B • 🌐📄 Save Response ...

PrettyRawPreviewVisualizeJSON ▾🔧

```
1 {
2   "code": 200,
3   "type": "unknown",
4   "message": "4X"
5 }
```

Code snippet

✕

cURL ▾⚙️📄

```
1 curl --location --request DELETE 'https://petstore.swagger.io/v2/user/4X'
```

</>

🔗

📘

Негативное тестирование

Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
ТС/VPN0007	Высокий	Получить удаленного пользователя по логину	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger проверяем наличие переменной baseUrl		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос “Get user by user name” в коллекции SwaggerPetstore		Открывается запрос с типом Get и заголовком {{baseUrl}}/user/:username (см. рис. 8)	Совпадает с ожидаемым	Пройден успешно
3.В параметры заносим существующий логин.		Значение существующего логина заносится. (см. рис. 8)	Совпадает с ожидаемым	Пройден успешно
4.Сохраняем и запускаем запрос		Пользователь с введенным логином не находится, код	Совпадает с ожидаемым	Пройден успешно

	ответа 404, скорость ответа высокая (см. рис. 8)		
Создал: Пермяков Вячеслав 04.11.2024		Новый тест-кейс	

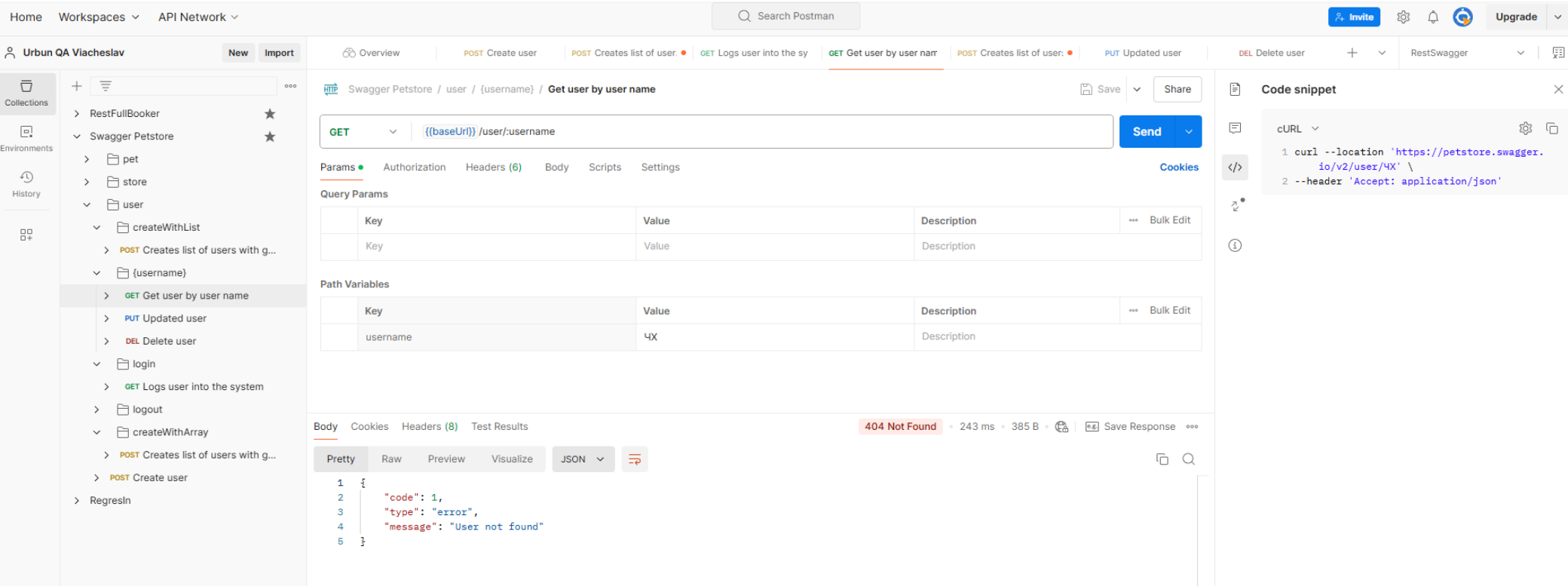


Рис. 8

Идентифика тор тест-	Приоритет	Название тест-кейса	Описание	Предусловия
-------------------------	-----------	---------------------	----------	-------------

кейса				
TC/VPN0008	Высокий	Повторное создание существующего пользователя.	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger добавляем переменную baseURL и заносим значение		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос create user в коллекции SwaggerPetstore .		Открывается запрос с типом Post и заголовком {{baseUrl}}/user (см. рис. 9)	Совпадает с ожидаемым	Пройден успешно
3.Заполняем данные значениями существующего пользователя.		Данные заполняются (см. рис. 9).	Совпадает с ожидаемым	Пройден успешно
4.Сохраняем и запускаем запрос		Должен загрузиться ответ с кодом состояния 403, скорость ответа высокая (см. рис. 9)	НЕ совпадает с ожидаемым, код ответа 200	Ошибка!
Создал: Пермяков Вячеслав 04.11.2024			Новый тест-кейс	

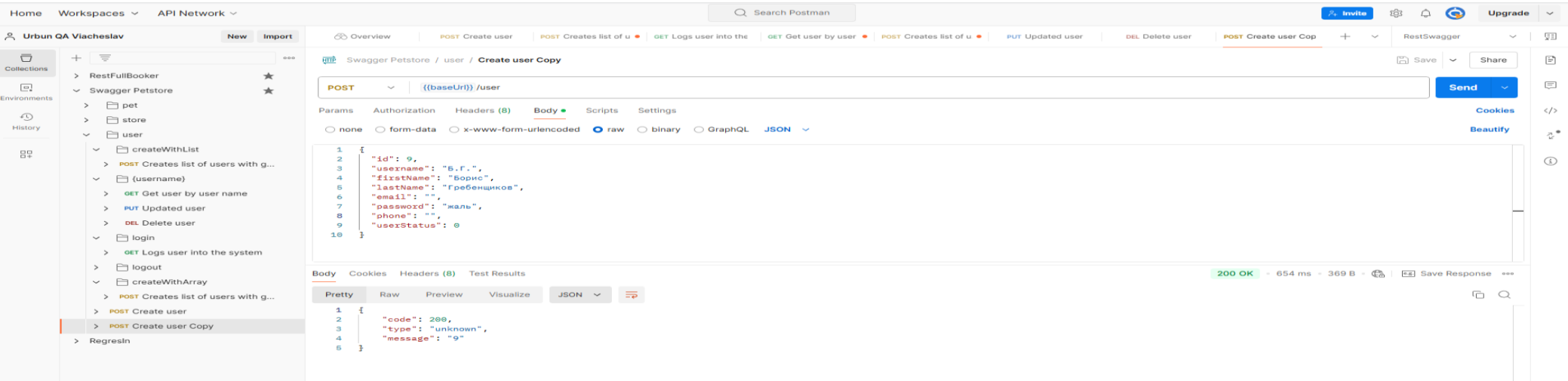


Рис. 9

Баг-репорт

Заголовок	Разрешается повторно вводить уже пользователя
Описание	При вводе уже существующего пользователя нет предупреждающего сообщения, в ответе код сообщения 200, должен быть 403
Шаги воспроизведения	1.Открывается запрос с типом Post и заголовком <code>{{baseUrl}}/user</code> 2. Заполняем данные значениями существующего пользователя.
Фактический результат	Данные заполняются, запрос выполняется, код ответа 200 (см. рис 9)
Ожидаемый результат	Сообщение «Такой пользователь уже введен!», запрос не выполняется, код ответа 403
Окружение	Проблема наблюдается как на мобильных, так и на десктоп устройствах и в разных браузерах (Google Chrome, Yandex)
Серьезность	Средняя

Приоритет		Средний		
Дата		06.11.2024		
Автор		Пермяков Вячеслав		
Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
TC/VPN0009	Высокий	Удаляем не существующего пользователя	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger проверяем наличие переменной baseURL		При открытии окружение видим переменную и ее значение: <code>https://petstore.swagger.io/v2</code>	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос "Delete user" в коллекции SwaggerPetstore		Открывается запрос с типом Delete и заголовком <code>{{baseUrl}}/user/:username</code> (см. рис. 10)	Совпадает с ожидаемым	Пройден успешно
3.В параметры заносим несуществующий логин.		Значение несуществующего логина заносится. (см. рис. 10)	Совпадает с ожидаемым	Пройден успешно
4.Сохраняем и запускаем запрос		Пользователь не найден, код ответа 404, скорость ответа	Совпадает с ожидаемым	Пройден успешно

	высокая (см. рис. 10)		
Создал: Пермяков Вячеслав 04.11.2024		Новый тест-кейс	

HomeWorkspacesAPI Network

Search Postman

Invite

Upgrade

Urbun QA Viacheslav

NewImport

OverviewPOST Create userPOST Creates list oGET Logs user intoGET Get user by usPOST Creates list oPUT Updated userDEL Delete userPOST Create user CDEL Delete user Coj+RestSwagger

Collections

RestFullBooker

Swagger Petstore

pet

store

user

createWithList

POST Creates list of users with g...

{username}

GET Get user by user name

PUT Updated user

DEL Delete user

DEL Delete user Copy

login

GET Logs user into the system

logout

createWithArray

POST Creates list of users with g...

POST Create user

POST Create user Copy

Regresin

Swagger Petstore / user / {username} / Delete user Copy

DELETE{{baseUrl}}/user/:username

Send

ParamsAuthorizationHeaders (5)BodyScriptsSettingsCookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Path Variables

Key	Value	Description	Bulk Edit
username	Винни-Пух	(Required) The name that needs to be deleted	

BodyCookiesHeaders (7)Test Results

404 Not Found1284 ms292 BSave Response

Key	Value
Date	Mon, 04 Nov 2024 09:41:44 GMT
Content-Length	0
Connection	keep-alive
Access-Control-Allow-Origin	*
Access-Control-Allow-Methods	GET, POST, DELETE, PUT
Access-Control-Allow-Headers	Content-Type, api_key, Authorization
Server	Jetty(9.2.9.v20150224)

Code snippet

cURL

curl --location --request DELETE 'https://petstore.swagger.io/v2/user/Винни-Пух'

Рис.10

Идентифика тор тест- кейса	Приоритет	Название тест-кейса	Описание	Предусловия
TC/VPN0010	Высокий	Создание пользователя с неправильным заголовком.	Тест проверяет код состояния, загрузку ответа, идемпотентность, заголовки и производительность (разумно ожидаемые сроки)	Пользователь находится в рабочем пространстве Postman Urban QA Viacheslav, в коллекции SwaggerPetstore окружении RestSwagger
Шаги		Ожидаемый результат	Фактический результат	Статус
1. В окружение RestSwagger добавляем переменную baseURL и заносим значение		При открытии окружение видим переменную и ее значение: https://petstore.swagger.io/v2	Совпадает с ожидаемым	Пройден успешно
2.Находим запрос create user в коллекции SwaggerPetstore .		Открывается запрос с типом Post и заголовком {{baseUrl}}/user (см. рис. 9)	Совпадает с ожидаемым	Пройден успешно
3. Меняем в заголовке адрес на несуществующий		Адрес меняется (см. рис.11)	Совпадает с ожидаемым	Пройден успешно
4.Заполняем данные		Данные заполняются (см. рис. 11).	Совпадает с ожидаемым	Пройден успешно
5.Сохраняем и запускаем запрос		Должен загрузиться ответ с кодом состояния 404, скорость ответа высокая (см. рис. 11)	Совпадает с ожидаемым	Пройден успешно
Создал: Пермяков Вячеслав 04.11.2024			Новый тест-кейс	

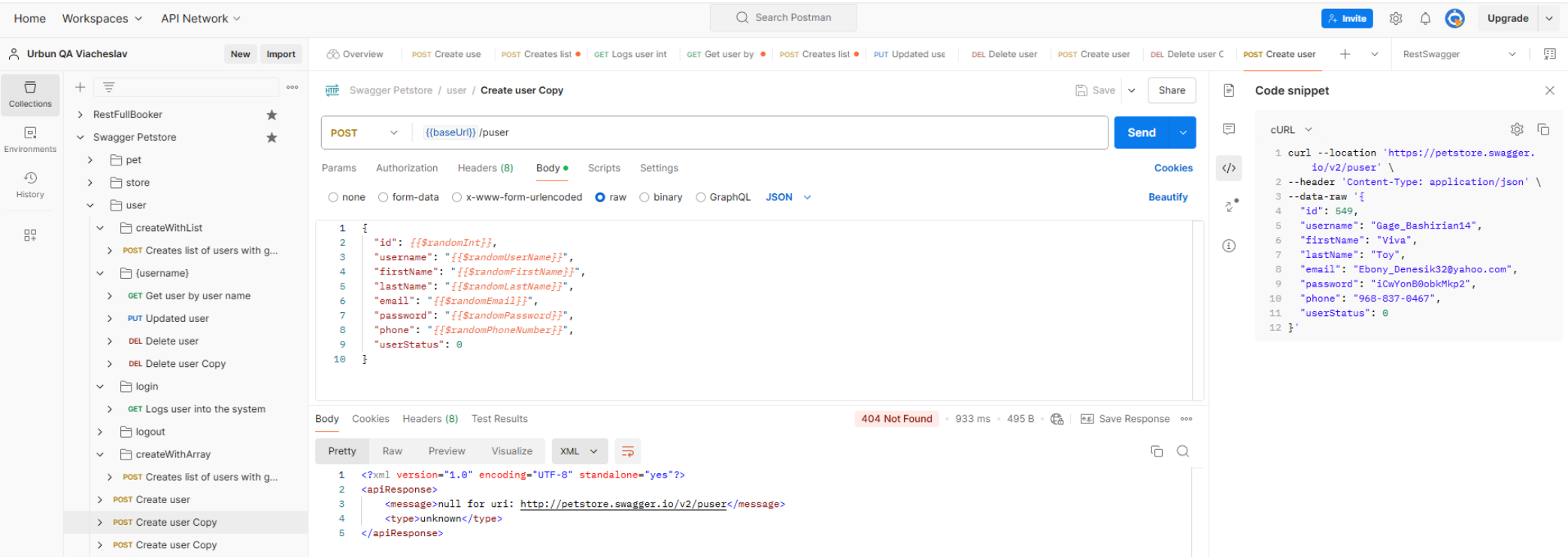


Рис. 11

Тестирование посадочной страницы:

<https://urban-university.ru/kyp-fouth-part>

Это форма опроса.

Функциональные возможности

1. Выбор вариантов ответа (radiobutton)
2. Ввод своих данных (Имя, телефон, Email).
3. Получение рекомендаций.

План работ:

Открываем форму: https://urban-university.ru/kyp-fouth-part	Действие
Позитивное тестирование	
Проверяем возможность перехода на следующую страницу	Открываем любую страницу формы (кроме последней) и пытаемся перейти на следующую
Проверяем возможность выбора ответа	Открываем любую страницу формы (кроме последней) и пытаемся выбрать вариант ответа
Проверяем возможность ввода данных пользователем	Открываем последнюю страницу формы и вводим «Имя», «№ телефона», «Email» (вводим существующие значения) и соглашаемся на обработку данных.
Проверяем возможность получить рекомендацию	После ввода данных пользователя и согласия на обработку данных по кнопке «Завершить опрос» пытаемся получить рекомендации.
Меняя варианты ответов, пытаемся получить разные рекомендации	Открывая последовательно страницы формы (кроме последней) выбираем другие варианты ответов, заполняем последнюю страницу (можно данными того же пользователя) и по кнопке «Завершить опрос» пытаемся получить другие рекомендации.
Негативное тестирование	
Проверяем возможность не выбирать ни один из вариантов ответа	Открываем последовательно все страницы формы (кроме последней) и, не выбрав ни один из вариантов ответа пытаемся перейти на следующую страницу.
Проверяем возможность ввода несуществующих данных	Открываем последнюю страницу формы и вводим в поле «Имя» несуществующие данные (цифры), в поле «№ телефона» вводим символы не соответствующие российскому номеру телефона и в поле «Email» символы не соответствующие типу.

Тест-кейсы**1. Проверяем возможность перехода на следующую страницу**

Идентификатор тест-кейса	Приоритет	Название тест-кейса	Описание	Предусловия	
SWTK/VNP2	Высокий	Возможность перехода на следующую страницу.	Тест проверяет возможность перехода с одной страницы формы на следующую	Пользователь находится на странице https://urban-university.ru/kyp-fourth-part .	
Шаги	Ожидаемый результат		Фактический результат		Статус
1.Открываем первую страницу формы	Открывается первая страница с вариантами ответов		Совпадает с ожидаемым		Пройден успешно
2.Щелкаем по кнопке «Переход к следующему вопросу»,	Открывается следующая страница с вариантами ответов		Совпадает с ожидаемым		Пройден успешно
3.Повторяем пункт 2 и так до последней страницы	Открывается следующая страница с вариантами ответов.		Совпадает с ожидаемым		Пройден успешно

2. Проверяем возможность выбора ответа

Идентификатор тест-кейса	Приоритет	Название тест-кейса	Описание	Предусловия	
SWTK/VNP1	Высокий	Возможность выбора варианта ответа.	Тест проверяет возможность выбора варианта ответа на любой странице формы	Пользователь находится на странице https://urban-university.ru/kyp-fourth-part .	
Шаги	Ожидаемый результат		Фактический результат		Статус

1.Открываем первую страницу формы	Открывается первая страница с вариантами ответов	Совпадает с ожидаемым	Пройден успешно
2.Выбираем любой вариант ответа.	У выбранного варианта появляется точка в radiobutton	Совпадает с ожидаемым	Пройден успешно
3.Открываем следующую страницу,	Открывается следующая страница с вариантами ответов	Совпадает с ожидаемым	Пройден успешно
4.Выбираем любой вариант ответа.	У выбранного варианта появляется точка в radiobutton	Совпадает с ожидаемым	Пройден успешно
5.Повторяем пункт 3 и 4 и так до последней страницы	Открывается следующая страница с вариантами ответов, У выбранного варианта появляется точка в radiobutton	Совпадает с ожидаемым	Пройден успешно

3. Проверяем ввод личных данных на последней странице форма

Идентификатор тест-кейса	Приоритет	Название тест-кейса	Описание	Предусловия
SWTK/VNP3	Высокий	Возможность заполнения личных данных.	Тест проверяет возможность правильного заполнения личных данных на последней странице формы.	Пользователь находится на последней странице https://urban-university.ru/kyp-fourth-part .
Шаги	Ожидаемый результат		Фактический результат	Статус
1.Открываем последнюю страницу формы	Открывается последняя страница формы с полями для ввода Имени, телефона, Email и чек-боксом для согласия на обработку данных.		Совпадает с ожидаемым	Пройден успешно
2.Заполняем поле «Имя» пробуя разные варианты	Позволяет ввести только буквы русского или латинского алфавита		Совпадает с ожидаемым	Пройден успешно

3.Вводим номер телефона (для России)	Есть проверка на несуществующие в России номера	Совпадает с ожидаемым	Пройден успешно
4.Вводим Email	Есть необходимые проверки: <ul style="list-style-type: none"> • Название (test) — один или много символов; • Знак собаки (@); • Доменное имя почтового сервера (mail) — один или много символов; • Точка (.); • Доменное имя первого уровня (ru) от двух до пяти букв. 	Совпадает с ожидаемым	Пройден успешно
5.Проверяем отсутствие галочки «я соглашаюсь на обработку персональных данных»	Вызывает ошибку, кнопка «Завершить опрос» не срабатывает, поле в красной рамке и надпись «обязательное поле».	Совпадает с ожидаемым	Пройден успешно
6.Ставим галочку «я соглашаюсь на обработку персональных данных»	Ошибки не возникает, кнопка «Завершить опрос» срабатывает	Совпадает с ожидаемым	Пройден успешно

Баг-репорт № 1

id	Название	Описание	Окружение	Шаги	Ожидаемый результат	Фактический результат
SW/VNP1	Разрешает переход на следующую	На любой странице формы срабатывает кнопка "Переход к	Windows 10, десктоп			

	страницу, если не выбран ответ.	следующему вопросу", даже если не выбран ни один вариант ответа.				
				1 Открываем форму по ссылке https://urban-university.ru/kyp-fouth-part	Форма открывается	Совпадает с ожидаемым
				2 Не выбираем ни один из вариантов ответа	Переход на следующую страницу не происходит. Сообщение "Не выбран вариант ответа!"	Переходим на следующую страницу
				3 Щелкаем по кнопке "К следующему вопросу"		

Приоритет: высокий (если мы проходим опрос – ответы надо выбирать, иначе теряется смысл).

Баг-репорт № 2

id	Название	Описание	Окружение	Шаги	Ожидаемый результат	Фактический результат
SW/VNP2	Независимо от вариантов ответа результат одинаковый	На любые варианты ответов на любой странице формы результат одинаковый «вам идеально подходит профессия: Python - разработчик»	Windows 10, десктоп			
				1 Открываем форму по ссылке https://urban-university.ru/kyp-fouth-part	Форма открывается	Совпадает с ожидаемым
				2 Выбираем любой из вариантов ответа	Переход на следующую страницу	Совпадает с ожидаемым
				3 Щелкаем по кнопке "К следующему вопросу"		
				4 На следующей странице снова		

				выбираем любой вариант ответа и т.д.		
				5 Доходим до последней страницы заполняем имя, Email, ставим галочку, что согласны и щелкаем по кнопке «Завершить опрос»	Получить вариант рекомендации, какая профессия нам подходит	Совпадает с ожидаемым
				6 По новой открываем форму по ссылке https://urban-university.ru/kyp-fourth-part	Форма открывается	Совпадает с ожидаемым
				7 На каждой странице выбираем ДРУГИЕ варианты ответов		
				8 Доходим до последней страницы заполняем имя, телефон, Email, ставим галочку, что согласны и щелкаем по кнопке «Завершить опрос»	Получить ДРУГОЙ вариант рекомендации	Получаем все ту же рекомендацию,

Приоритет: высокий (подразумевается, что в зависимости от ответов, рекомендации будут РАЗНЫЕ, иначе можно сразу выводить «вам идеально подходит профессия: Python - разработчик»).

Баг-репорт № 3

id	Название	Описание	Окружение	Шаги	Ожидаемый результат	Фактический результат
SW/VNP3	Отсутствует кнопка «Возврат».	Находясь на любой странице формы мы не можем вернуться на предыдущую страницу, нет кнопки «Возврат».	Windows 10, десктоп			
				1 Открываем форму по	Форма	Совпадает с

				ссылке https://urban-university.ru/kyp-fourth-part	открывается	ожидаемым
				2 Выбираем вариант ответа и по кнопке "К следующему вопросу" переходим на следующую страницу и так на всех страницах с вариантами ответов	Переходим на следующую страницу	Совпадает с ожидаемым
				3 На последней странице заполняем личные данные: Имя, телефон, Email и чек-бокс - согласия на обработку данных.	Данные заполняются правильно	Совпадает с ожидаемым
				4 щелкаем по кнопке «Завершить опрос» срабатывает	Открывается результат опроса	Совпадает с ожидаемым
				5 Снова повторяем пункты 1, 2, 3 личные данные вводим те же самые, как в пункте 4		
				6 щелкаем по кнопке «Завершить опрос» срабатывает	Должно появиться сообщение «Пользователь с такими данными уже проходил этот опрос!» Остаемся на странице ввода данных	Открывается результат опроса

Приоритет: низкий, ну прошел опрос еще раз, это не критично.

Баг-репорт № 4

id	Название	Описание	Окружение	Шаги	Ожидаемый результат	Фактический результат
SW/VNP4	Нет проверки, что пользователь уже проходил этот опрос.	Можно бесконечно число раз проходит опрос под одним и тем же пользователем.	Windows 10, десктоп			
				1 Открываем форму по ссылке https://urban-university.ru/kyp-fourth-part	Форма открывается	Совпадает с ожидаемым
				2 Щелкаем по кнопке "К следующему вопросу"	Переходим на следующую страницу	Совпадает с ожидаемым
				3 Пытаемся вернуться на предыдущую страницу	Открывается предыдущая страница формы	Нет возможности вернуться, отсутствует кнопка «Возврат»

Приоритет: высокий, у пользователя должна быть возможность вернуться на предыдущую страницу.

Баг-репорт № 5

id	Название	Описание	Окружение	Шаги	Ожидаемый результат	Фактический результат
SW/VNP5	Полоска прогресс-бара на первой странице показывает 50.	При открытии первой страницы форма полоска прогресс-бара сразу показывает 50%, хотя у нас 5 страниц с вопросами и логичнее было показать 20%	Windows 10, десктоп			
				1 Открываем форму по ссылке https://urban-university.ru/kyp-fourth-part	Форма открывается	Совпадает с ожидаемым

				2 На первой странице	Полоска прогресс-бара показывает 20%	Показывает 50%
				3 По кнопке «К следующему вопросу» переходим на вторую страницу	Полоска прогресс-бара показывает 40%	Показывает 62,5%
				4 По кнопке «К следующему вопросу» переходим на третью страницу	Полоска прогресс-бара показывает 60%	Показывает 75%
				5 По кнопке «К следующему вопросу» переходим на четвертую страницу	Полоска прогресс-бара показывает 80%	Показывает 87.5%
				По кнопке «К следующему вопросу» переходим на пятую страницу	Полоска прогресс-бара показывает 100%	Показывает 100%

Приоритет: средний, основному функционалу не мешает.

Теперь замечания и пожелания к оформлению:

1. Я бы на первой странице на рисунке добавил бабочку (красиво!)
2. На второй странице нарисовал гвоздь (вдруг кто-то забыл, что это и как выглядит!)