Отчет о результатах тестирования API для сущности Pet

Объект тестирования:
API сущности **Pet** из OpenAPI Swagger Petstore. Тестирование проводится на основные операции CRUD (Create, Read, Update, Delete);

Цель тестирования:
Проверить корректную работу ручек API для сущности Pet, а также проверить обработку позитивных и негативных сценариев; Убедиться, что API правильно обрабатывает запросы и возвращает корректные коды ошибок и ответы;

Окружение:

- **URL**: https://petstore.swagger.io/v2;
- **Методы**: GET, POST, PUT, DELETE;
- **Инструменты тестирования**:
  - Библиотека Playwright JS;
  - Allure;
- **Операционная система**: MacOS Sequoia 15.1;
- **NodeJS**: 20.11.1;
- **Дата проведения**: 13-14.12.2024;

Описание тестов:

1. Тест на создание питомца (POST /pet)
   a. Описание: Тест проверяет успешное создание нового питомца с уникальными параметрами.
   b. Ожидаемый результат: Код ответа 200 OK, данные о питомце совпадают с переданными в запросе.
   c. Фактический результат: Код ответа 200 OK, питомец успешно создан, данные совпадают с ожидаемыми.

2. Тест на получение питомца по ID (GET /pet/{petId})
   a. Описание: Тест проверяет успешное получение данных о питомце по его уникальному ID.
   b. Ожидаемый результат: Код ответа 200 OK, возвращены корректные данные о питомце.
   c. Фактический результат: Код ответа 200 OK, данные питомца соответствуют ID и содержимому.

3. Тест на обновление статуса питомца (PUT /pet)
   a. Описание: Тест проверяет возможность изменения статуса питомца (с "available" на "sold").
   b. Ожидаемый результат: Код ответа 200 OK, статус питомца обновлен на "sold".
   c. Фактический результат: Код ответа 200 OK, статус питомца успешно обновлен на "sold".

4. Тест на удаление питомца (DELETE /pet/{petId})
   a. Описание: Тест проверяет удаление питомца по его уникальному ID.
   b. Ожидаемый результат: Код ответа 200 OK, питомец удален, повторный запрос на получение возвращает код 404 Not Found.
   c. Фактический результат: Код ответа 200 OK, питомец удален, повторный запрос возвращает код 404 Not Found.
5. Негативный тест на получение несуществующего питомца (GET /pet/{nonexistent-id})
   a. Описание: Тест проверяет, что запрос на получение данных о несуществующем питомце возвращает корректную ошибку.

b. Ожидаемый результат: Код ответа 404 Not Found.
c. Фактический результат: Код ответа 404 Not Found.

Тестовые данные:

- Pet ID: 12345
- Имя питомца: Rex
- Категория: Dogs
- Фото URL: ["https://example.com/photo"]
- Теги: [{"id": 1, "name": "tag1"}]
- Статус: «available» на «sold» (при обновлении)

Результаты тестирования:

№	Тест	Ожидаемый результат	Фактический результат	Статус
1	Создание (POST /pet)	200, ОК	200, ОК	Успешно
2	Получение (GET /pet/{id})	200, ОК	200, ОК	Успешно
3	Обновление (PUT /pet)	200, ОК	200, ОК	Успешно
4	Удаление (DELETE /pet/{id})	200, ОК	200, ОК	Успешно
5	Негативный сценарий	404, Not Found	404, Not Found	Успешно

Выводы:

1. Все тесты прошли успешно. Система корректно обрабатывает операции создания, получения, обновления и удаления сущностей.
2. Негативные тесты также работают корректно: попытка получения несуществующего питомца возвращает ожидаемый код ошибки 404.
3. API стабильно: ни один запрос не вызвал ошибки сервера, время ответа соответствует ожиданиям (в пределах 1 секунды).

Рекомендации:

1. Провести дополнительные тесты для работы с большим количеством питомцев, чтобы проверить производительность системы.
2. Проверить работу API с различными типами данных (например, нагрузочное тестирование с большими объемами информации или с некорректными форматами данных).

Заключение:

Функциональность API для сущности **Pet** работает в полном соответствии с ожиданиями. Все тестовые сценарии выполнены успешно, результаты соответствуют требованиям.

Отчет о результатах тестирования API для сущности User

Объект тестирования:
API сущности **User** на сайте Swagger Petstore. Тестируются основные операции CRUD.

Цель тестирования:
Проверить корректную работу эндпоинтов API для сущности **User**, а также проверить обработку как позитивных, так и негативных сценариев. Убедиться, что API правильно обрабатывает запросы, возвращает корректные ответы и коды ошибок.

Окружение:

- **URL**: https://petstore.swagger.io/v2;
- **Методы**: GET, POST, PUT, DELETE;
- **Инструменты тестирования**:
  - Библиотека Playwright JS;
  - Allure;
- **Операционная система**: MacOS Sequoia 15.1;
- **NodeJS**: 20.11.1;
- **Дата проведения**: 13-14.12.2024;

Описание тестов:

1. Тест на создание пользователя (POST /user)
   o Описание: Тест проверяет успешное создание нового пользователя с уникальными параметрами.
   o Ожидаемый результат: Код ответа 200 OK, данные о пользователе совпадают с переданными в запросе.
   o Фактический результат: Код ответа 200 OK, пользователь успешно создан, данные совпадают с ожидаемыми.
2. Тест на получение пользователя по имени (GET /user/{username})
   o Описание: Тест проверяет успешное получение данных о пользователе по его уникальному имени.
   o Ожидаемый результат: Код ответа 200 OK, возвращены корректные данные о пользователе.
   o Фактический результат: Код ответа 200 OK, данные пользователя соответствуют переданному имени.
3. Тест на обновление пользователя (PUT /user/{username})
   o Описание: Тест проверяет возможность обновления данных пользователя.
   o Ожидаемый результат: Код ответа 200 OK, данные пользователя обновлены.
   o Фактический результат: Код ответа 200 OK, данные пользователя успешно обновлены.
4. Тест на удаление пользователя (DELETE /user/{username})
   o Описание: Тест проверяет удаление пользователя по его уникальному имени.
   o Ожидаемый результат: Код ответа 200 OK, пользователь удален, повторный запрос на получение возвращает код 404 Not Found.
   o Фактический результат: Код ответа 200 OK, пользователь удален, повторный запрос возвращает код 404 Not Found.
5. Негативный тест на получение несуществующего пользователя (GET /user/{non-existent-username})
   o Описание: Тест проверяет, что запрос на получение данных о несуществующем пользователе возвращает корректную ошибку.
   o Ожидаемый результат: Код ответа 404 Not Found.
   o Фактический результат: Код ответа 404 Not Found.

Тестовые данные:

- Username: john_doe _junior
- Первое имя: John
- Фамилия: Doe
- Email: johndoe@example.com
- Новый Email: newemail@example.com
- Пароль: password123
- Телефон: +1234567890
- Статус пользователя: active

Результаты тестирования

№	Тест	Ожидаемый результат	Фактический результат	Статус
1	Создание (POST /user)	200, ОК	200, ОК	Успешно
2	Получение (GET /user/{username})	200, ОК	200, ОК	Успешно
3	Обновление (PUT /user/{username})	200, ОК	200, ОК	Успешно
4	Удаление (DELETE /user/{username})	200, ОК	200, ОК	Успешно
5	Негативный сценарий	404, Not Found	404, Not Found	Успешно

Выводы:

1. **Все тесты прошли успешно**. API для управления пользователями корректно обрабатывает операции создания, получения, обновления и удаления.
2. **Негативные тесты работают корректно**. Попытка получения несуществующего пользователя возвращает код ошибки 404.
3. **API стабильно**: ни один запрос не вызвал ошибки сервера, и время отклика соответствует ожиданиям.

Рекомендации:

1. Провести тесты на производительность, включая массовое создание пользователей.
2. Проверить работу API с различными типами данных, включая неверные или частично корректные запросы.

Отчет о результатах тестирования API для сущности Store

Объект тестирования:
API сущности **Store** на сайте Swagger Petstore. Тестируются основные операции CRUD.

Цель тестирования:
Проверить корректную работу эндпоинтов API для сущности **Store**, а также проверить обработку как позитивных, так и негативных сценариев. Убедиться, что API правильно обрабатывает запросы, возвращает корректные ответы и коды ошибок.

Окружение:

- **URL**: https://petstore.swagger.io/v2;
- **Методы**: GET, POST, PUT, DELETE;
- **Инструменты тестирования**:
  - Библиотека Playwright JS;
  - Allure;
- **Операционная система**: MacOS Sequoia 15.1;
- **NodeJS**: 20.11.1;
- **Дата проведения**: 13-14.12.2024;

Описание тестов:

1. Тест на создание заказа (POST /store/order)
   o Описание: Тест проверяет успешное создание нового заказа с уникальными параметрами.
   o Ожидаемый результат: Код ответа 200 OK, данные о заказе совпадают с переданными в запросе.
   o Фактический результат: Код ответа 200 OK, заказ успешно создан, данные совпадают с ожидаемыми.
2. Тест на получение пользователя по имени (GET /store/order/{orderId})
   o Описание: Тест проверяет успешное получение данных о заказе по его уникальному ID.
   o Ожидаемый результат: Код ответа 200 OK, возвращены корректные данные о заказе.
   o Фактический результат: Код ответа 200 OK, данные зказа соответствуют переданному ID.
3. Тест на удаление заказа (DELETE /store/order/{orderId})
   o Описание: Тест проверяет удаление заказа по его уникальному ID.
   o Ожидаемый результат: Код ответа 200 OK, заказ удален, повторный запрос на получение возвращает код 404 Not Found.
   o Фактический результат: Код ответа 200 OK, заказ удален, повторный запрос возвращает код 404 Not Found.
4. Тест на неверный заказ (POST /store/order)
   o Описание: Тест проверяет соответствие контракту;
   o Ожидаемый результат: Код ответа 500, заказ не создан;
   o Фактический результат: Код ответа 500, заказ не создан;
5. Негативный тест на получение несуществующего заказа (GET /store/order/{non-existent-orderId})
   o Описание: Тест проверяет, что запрос на получение данных о несуществующем заказе возвращает корректную ошибку.
   o Ожидаемый результат: Код ответа 400 Bad Request.
   o Фактический результат: Код ответа 400 Bad Request.

Тестовые данные: 500

- OrderID: 12345
- Pet Id: 121416
- Status: placed
- Complete: false

Результаты тестирования

№	Тест	Ожидаемый результат	Фактический результат	Статус
1	Создание (POST /store/order)	200, OK	200, OK	Успешно
2	Получение (GET /store/order/{id})	200, OK	200, OK	Успешно
3	Неверный заказ (/store/order)	500, ISE	500, ISE	Успешно
4	Удаление (DELETE /store/order/{id})	200, OK	200, OK	Успешно
5	Негативный сценарий	400, Bad Request	400, Bad Request	Успешно

Выводы:

1. **Все тесты прошли успешно**. API для управления пользователями корректно обрабатывает операции создания, получения, обновления и удаления.
2. **Негативные тесты работают корректно**. Попытка получения несуществующего пользователя возвращает код ошибки 404.
3. **API стабильно**: ни один запрос не вызвал ошибки сервера, и время отклика соответствует ожиданиям.

Рекомендации:

1. Провести тесты на производительность, включая массовое создание заказов.
2. Проверить работу API с различными типами данных, включая неверные или частично корректные запросы.