Отчет по ручному тестированию API сервиса **https://dog.ceo/api**

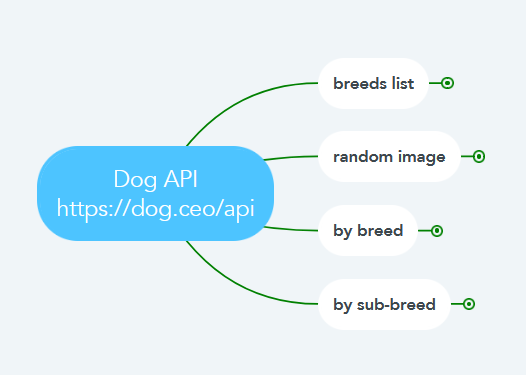
**Введение**

Целью данного тестирования является проверка работоспособности сервиса API.

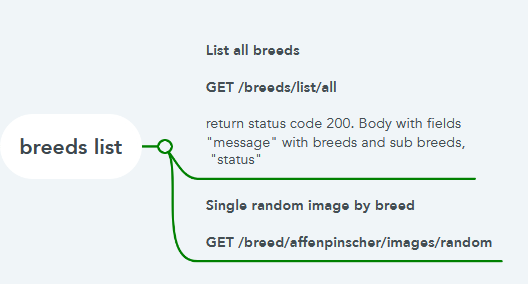
Тестирование проводилось на основе mind map:

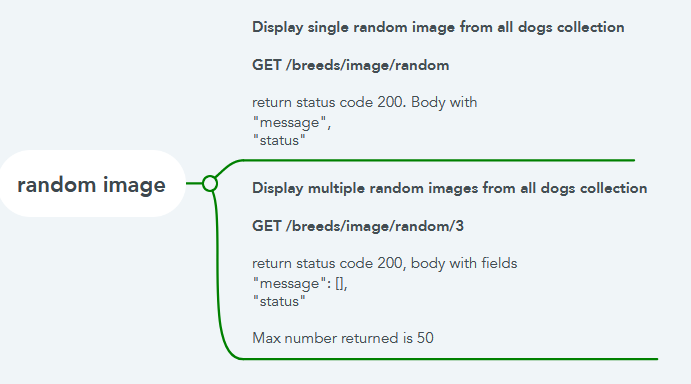
**Mind map**

Base

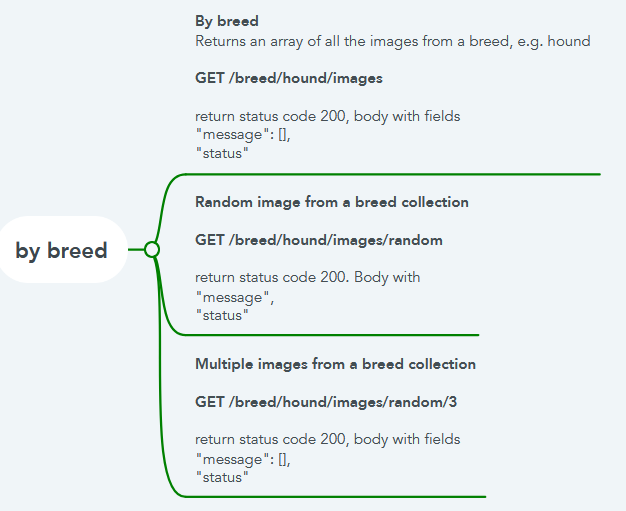
****

Breeds list

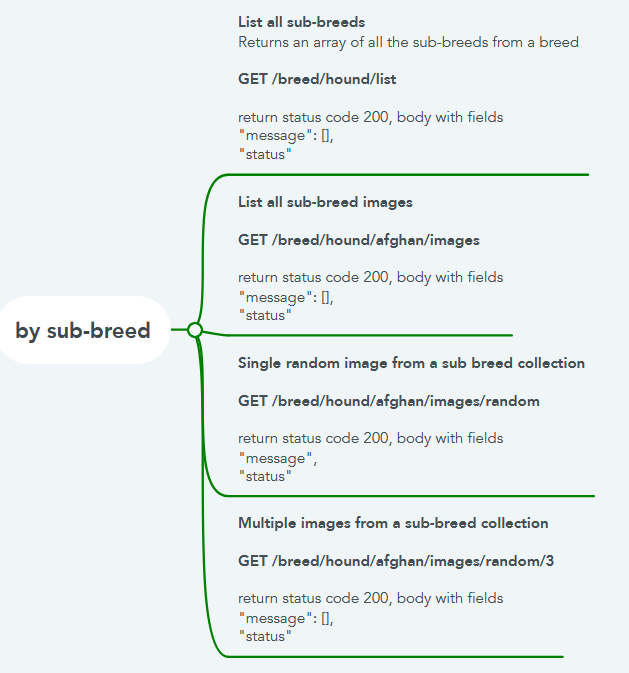
Random image



By breed



By sub-breed



Тестирование проводилось с использованием следующих инструментов и ресурсов:

Документация с описанием эндпоинтов: <https://dog.ceo/dog-api/documentation>

Ручное тестирование API:

* Postman
* Количестов ручных API тест-кейсов: 25

Автоматизированное тестирование API:

* Pytest
* Allure
* Количество автоматизированных API тест-кейсов: 25

**Сводка**

По результатам тестирования были получены следующие данные:

Критерии

Общее количество тест-кейсов 25

Пройденные тест-кейсы 19

Не пройденные тест-кейсы 6

Пропущенные тест-кейсы 0

Заблокированные тест-кейсы 0

Общее количество дефектов 6

Критические дефекты 0

Высокоприоритетные дефекты 0

Среднеприоритетные дефекты 6

Низкоприоритетные дефекты 0

**Детальное описание проверок**

В этом разделе приведены подробные данные по каждому тест-кейсу.

**Тест-кейсы**

Ниже представлена таблица с детальным описанием проверок.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Название | Шаги | Ожидаемый результат | Статус | Приоритет |
| List all breeds | | | | | |
| TC-1 | Проверка получения списка собак | 1. Отправить запрос **GET /breeds/list/all** | 1. В результате получен ответ со списком пород в формате JSON. | **Пройден** | Средний |
| 1. Проверить статус код и тело ответа. | 1. Получен статус код 200 и тело ответа содержит список пород. |
| TC-2 | Проверка получения ошибки при не поддерживаемом методе | 1. Отправить запрос **POST /breeds/list/all** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 1. Проверить статус код и тело ответа | 2. Статус код 405. Формат ответа с ошибкой соответствует ожидаемому, содержит поля: error, meassage, code. |
| TC-3 | Проверка получения рандомного изображения по названию собаки | 1. Отправить запрос **GET /breed/affenpinscher/images/random** | 1. В результате получен ответ со списком пород в формате JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message, status. |
| TC-4 | Проверка получения ошибки, когда название собаки не найдено | 1. Отправить запрос **GET /breed/dog/images/random** | 1. В результате получен ответ со списком пород в формате JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 404. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| TC-5 | Проверка получения ошибки, когда название собаки = 1 | 1. Отправить запрос **GET /breed/1/images/random** | 1. В результате получен ответ со списком пород в формате JSON. | **Не пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 400. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. Должна отрабатывать валидация. |
| Random image | | | | | |
| TC-6 | Проверка получения рандомного изображения из всего списка собак | 1. Отправить запрос **GET /breeds/image/random** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message, status |
| TC-7 | Проверка получения рандомного списка изображений из всего списка собак по количеству=1 | 1. Отправить запрос **GET /breeds/image/random/1** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status. В массиве message список из 1 элементов. |
| TC-8 | Проверка получения ошибки рандомного списка изображений из всего списка собак по количеству = 0 | 1. Отправить запрос **GET /breeds/image/random/0** | 1. В результате получен ответ с телом в формте JSON. | **Не пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 400. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| TC-9 | Проверка получения ошибки рандомного списка изображений из всего списка собак по количеству = -1 | 1. Отправить запрос **GET /breeds/image/random/-1** | 1. В результате получен ответ с телом в формте JSON. | **Не пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 400. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| TC-10 | Проверка получения рандомного списка изображений из всего списка собак по количеству=2 | 1. Отправить запрос **GET /breeds/image/random/2** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status.  В массиве message список из 2 элементов. |
| TC-11 | Проверка получения рандомного списка изображений из всего списка собак по количеству=49 | 1. Отправить запрос **GET /breeds/image/random/49** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status. В массиве message список из 49 элементов. |
| TC-12 | Проверка получения рандомного списка изображений из всего списка собак по количеству=50 | 1.Отправить запрос **GET /breeds/image/random/49** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status. В массиве message список из 50 элементов. |
| TC-13 | Проверка получения ошибки валидации по количеству=51 | 1.Отправить запрос **GET /breeds/image/random/51** | 1. В результате получен ответ с телом в формте JSON. | **Не пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 400. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| By breed | | | | | |
| TC-14 | Проверка получения всего списка изображений по названию породы | 1. Отправить запрос **GET /breed/hound/images** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status. |
| TC-15 | Проверка получения рандомного изображения по названию породы | 1. Отправить запрос **GET /breed/hound/images/random** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message, status. |
| TC-16 | Проверка получения списка изображений по названию породы по количеству | 1. Отправить запрос **GET /breed/hound/images/random/50** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status. В массиве message список из 50 элементов. |
| TC-17 | Проверка получения ошибки, когда название породы не найдено при получении всего списка | 1. Отправить запрос **GET /breed/ddd/images** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 404. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| TC-18 | Проверка получения ошибки, когда название породы не найдено при получении рандомного изображения | 1. Отправить запрос **GET /breed/sss/images/random** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 404. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| TC-19 | Проверка валидации при получении списка изображений по количеству = -1 | 1. Отправить запрос **GET /breed/hound/images/random/-1** | 1. В результате получен ответ с телом в формте JSON. | **Не пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 400. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| TC-20 | Проверка валидации при получении списка изображений по количеству = k | 1. Отправить запрос **GET /breed/hound/images/random/k** | 1. В результате получен ответ с телом в формте JSON. | **Не пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 400. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| By sub-breed | | | | | |
| TC-21 | Проверка получения списка подпород | 1. Отправить запрос **GET /breed/hound/list** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status. В массиве message содержится список под пород. |
| TC-22 | Проверка получения всего списка изображение по подпороде | 1. Отправить запрос **GET /breed/hound/afghan/images** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message [], status. В массиве message содержится список изображений под пород. |
| TC-23 | Проверка получения рандомного изображения по подпороде | 1. Отправить запрос **GET /breed/hound/afghan/images/random** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 200. Тело ответа соответствует ожидаемому, содержит поля: message, status. |
| TC-24 | Проверка получения ошибки при получении списка подпород, когда подпорода не найдена | 1. Отправить запрос **GET /breed/ddd/list** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 404. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |
| TC-25 | Проверка получения ошибки при получении всего списка изображений, когда под порода не найдена | 1. Отправить запрос **GET /breed/hound/afghan/images** | 1. В результате получен ответ с телом в формте JSON. | **Пройден** | Средний |
| 2. Проверить статус код и тело ответа. | 2. Статус код = 404. Тело ответа соответствует ожидаемому, содержит поля: status, message, code. |

**Дефекты**

Ниже перечислены дефекты , обнаруженные в ходе тестирования, с их описанием , приоритетом, критичностью, шагами воспроизведения , ожидаемым и фактическим результатами.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Название | Шаги | Фактический результат | Ожидаемый результат | Статус |
| BR-1 | Отсутствует валидация, когда название собаки = 1 | 1. Отправить запрос **GET /breed/1/images/random** | В результате получен ответ в формате JSON, с кодом 404 | Получен ответ в формате JSON с кодом 404 и описанием | Средний |
| BR-2 | Приходит рандомный списка изображений из всего списка собак по количеству = 0 | 1. Отправить запрос **GET /breeds/image/random/0** | Статус код = 200. Тело ответа содержит поля: message [], status. | Получен ответ в формате JSON с кодом 404 и описанием | Средний |

**Заключение**

По результатам тестирования сервиса Dog API можно сделать следующие выводы:

* сервис обладает продуманной бизнес логикой, соответствующим стандартам и ожиданиям пользователей.
* сервси имеет низкий уровень дефектов, большинство из которых не критичны и не влияют на основные сценарии использования.
* рекомендуется провести нагрузочное тестирование и тестирование безопасности.

**Рекомендации по улучшению:**

* Добавить кнопку Fetch у всех запросов в документации <https://dog.ceo/dog-api/documentation>.
* Во всех ошибках связанных с не найденными породами или подпородами необходимо хранить корректное имя поля, по которому не найдены изображения, в поле message. Если не найдено по породе, то поле текст в message содержит breed, если не найдено по подпороде – sub-breed.

**Риски :**

Технические проблемы: Риски связанные с возможными техническими сбоями, серверными проблемами, аппаратными сбоями, неправильной работой баз данных и т.д.