






Модуль: Pet (Питомцы)

ID	Название / Сценарий	Метод	Эндпоинт	Тестовые данные	Ожидаемый результат	Тип теста
PET-01	Создание валидного питомца	POST	/pet	{ "id": 101, "name": "Барсик", "status": "available" }	<ul style="list-style-type: none">• Status: 200• ID сохранён• Данные соответствуют отправленным	<div>✓</div> <div>Positive</div>
PET-02	Получение созданного питомца	GET	/pet/{petId}	petId = 101	<ul style="list-style-type: none">• Status: 200• Имя = "Барсик"• Статус = "available"	<div>✓</div> <div>Positive</div>
PET-03	Обновление данных питомца	PUT	/pet	{ "id": 101, "name": "Барсик Обнова" }	<ul style="list-style-type: none">• Status: 200• Имя обновилось• После GET видно изменения	<div>✓</div> <div>Positive</div>
PET-04	Удаление питомца	DELETE	/pet/{petId}	petId = 101	<ul style="list-style-type: none">• Status: 200• Подтверждение удаления	<div>✓</div> <div>Positive</div>
PET-05	Создание с невалидным ID (строка)	POST	/pet	{ "id": "mistake", "name": "Барсик" }	<ul style="list-style-type: none">• Status: 400/500• Сообщение об ошибке	<div>✗</div> <div>Negative</div>
PET-06	Получение несуществующего питомца	GET	/pet/{petId}	petId = 0	<ul style="list-style-type: none">• Status: 404• "Pet not found"	<div>✗</div> <div>Negative</div>

Модуль: Store (Магазин)

ID	Название / Сценарий	Метод	Эндпоинт	Тестовые данные	Ожидаемый результат	Тип теста
STORE-01	Размещение валидного заказа	POST	/store/order	{"id": 101, "petId": 1, "quantity": 5, "status": "placed"}	<ul style="list-style-type: none">Status: 200Статус заказа = "placed"	 Positive
STORE-02	Получение заказа по ID	GET	/store/order/{orderId}	orderId = 101	<ul style="list-style-type: none">Status: 200Данные заказа корректны	 Positive
STORE-03	Получение инвентаря	GET	/store/inventory	Без параметров	<ul style="list-style-type: none">Status: 200JSON с данными инвентаря	 Positive
STORE-04	Создание заказа с невалидным petId	POST	/store/order	{"petId": "ABC", "quantity": 5}	<ul style="list-style-type: none">Status: 400Ошибка валидации	 Negative
STORE-05	Получение несуществующего заказа	GET	/store/order/{orderId}	orderId = 10001	<ul style="list-style-type: none">Status: 404Заказ не найден	 Negative

Модуль: User (Пользователи)

ID	Название / Сценарий	Метод	Эндпоинт	Тестовые данные	Ожидаемый результат	Тип теста
USE R-01	Создание пользователя	POST	/user	{ "id": 103, "username": "andeytop", "email": "anreyrealboy123@gmail.com" }	<ul style="list-style-type: none">• Status: 200• Пользователь создан	<div>✓</div> Positive
USE R-02	Создание нескольких пользователей	POST	/user/createWithList	Массив из 2 пользователей	<ul style="list-style-type: none">• Status: 200• Оба пользователя созданы	<div>✓</div> Positive
USE R-03	Получение пользователя по имени	GET	/user/{username}	username = "matvey0205"	<ul style="list-style-type: none">• Status: 200• Данные пользователя корректны	<div>✓</div> Positive
USE R-04	Обновление данных пользователя	PUT	/user/{username}	{ "username": "matvey0205", "email": "matteus123@gmail.com" }	<ul style="list-style-type: none">• Status: 200• Данные обновлены	<div>✓</div> Positive
USE R-05	Логин пользователя	GET	/user/login	?username=matvey0205&password=admin	<ul style="list-style-type: none">• Status: 200• Сессия создана• Возвращается токен	<div>✓</div> Positive
USE R-06	Логаут пользователя	GET	/user/logout	Без параметров	<ul style="list-style-type: none">• Status: 200• Сессия завершена	<div>✓</div> Positive
USE R-07	Удаление пользователя	DELETE	/user/{username}	username = "andeytop"	<ul style="list-style-type: none">• Status: 200• Пользователь удалён	<div>✓</div> Positive

Статистика тестирования

Метрика	Значение
Всего тест-кейсов	18
Позитивных тестов	13 (72%)
Негативных тестов	5 (28%)
Покрытие эндпоинтов	10+ эндпоинтов
Проверяемые методы	GET, POST, PUT, DELETE
Модули API	Pet, Store, User

Функциональные проверки:

- CRUD операции для всех сущностей (Create, Read, Update, Delete)
- Валидация бизнес-логики (статусы заказов, инвентарь)
- Работа с аутентификацией (login/logout)

Негативные сценарии:

- Невалидные типы данных (строка вместо числа)
- Отсутствующие ресурсы (404 ошибки)
- Ошибки валидации входных данных

Структурные проверки:

- Формат JSON запросов и ответов
- Правильность HTTP методов
- Соответствие API спецификации

Структура тестов в Postman

PetStore API Testing/

