**Документация API Pet Store**

Этот документ описывает API для Pet Store, пример сервера, основанного на спецификации OpenAPI 3.0.

**1. Введение**

API Pet Store позволяет вам управлять домашними животными, их заказами и учетными записями пользователей. Вы можете добавлять новых питомцев, обновлять существующих, находить питомцев по статусу или тегам, размещать заказы и управлять информацией о пользователях.

**2. Базовый URL**

https://petstore3.swagger.io/api/v3

**3. Общая информация**

* **Условия использования:** <http://swagger.io/terms/>
* **Контактная информация:** [apiteam@swagger.io](mailto:apiteam@swagger.io)
* **Лицензия:** [Apache 2.0](http://www.apache.org/licenses/LICENSE-2.0.html)
* **Версия:** 1.0.11
* **Внешняя документация:** [http://swagger.io](http://swagger.io/)

**4. Аутентификация**

API использует следующие схемы аутентификации:

* **petstore\_auth (OAuth 2.0):** Используется для операций, связанных с домашними животными. Требует области видимости write:pets и read:pets.
  + **URL авторизации:** https://petstore3.swagger.io/oauth/authorize
* **api\_key (Ключ API):** Используется для получения информации об инвентаре. Ключ API должен быть передан в заголовке api\_key.

**5. API Endpoints**

В следующих разделах подробно описаны доступные конечные точки API, их параметры, тела запросов и ответы.

**5.1. Pet (Домашние животные)**

**Тег:** pet

Операции, связанные с управлением домашними животными.

**5.1.1. PUT /pet**

* **Краткое описание:** Обновить существующего питомца
* **Описание:** Обновить существующего питомца по ID.
* **ID операции:** updatePet
* **Тело запроса:**
  + **Описание:** Обновить существующего питомца в магазине.
  + **Типы контента:** application/json, application/xml, application/x-www-form-urlencoded
  + **Схема:** [Pet](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Pet) (Обязательно)
* **Ответы:**
  + 200 OK: Успешная операция.
    - **Типы контента:** application/json, application/xml
    - **Схема:** [Pet](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Pet)
  + 400 Bad Request: Предоставлен неверный ID
  + 404 Not Found: Питомец не найден
  + 422 Unprocessable Entity: Ошибка валидации
* **Безопасность:**
  + petstore\_auth: write:pets, read:pets

**5.1.2. POST /pet**

* **Краткое описание:** Добавить нового питомца в магазин
* **Описание:** Добавить нового питомца в магазин.
* **ID операции:** addPet
* **Тело запроса:**
  + **Описание:** Создать нового питомца в магазине.
  + **Типы контента:** application/json, application/xml, application/x-www-form-urlencoded
  + **Схема:** [Pet](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Pet) (Обязательно)
* **Ответы:**
  + 200 OK: Успешная операция.
    - **Типы контента:** application/json, application/xml
    - **Схема:** [Pet](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Pet)
  + 400 Bad Request: Неверный ввод
  + 422 Unprocessable Entity: Ошибка валидации
* **Безопасность:**
  + petstore\_auth: write:pets, read:pets

**5.1.3. GET /pet/findByStatus**

* **Краткое описание:** Найти питомцев по статусу
* **Описание:** Можно предоставить несколько значений статуса, разделенных запятыми.
* **ID операции:** findPetsByStatus
* **Параметры:**
  + status (query, string, optional): Значения статуса, которые необходимо учитывать для фильтрации. Допустимые значения: available, pending, sold. По умолчанию: available.
* **Ответы:**
  + 200 OK: Успешная операция
    - **Типы контента:** application/json, application/xml
    - **Схема:** Массив [Pet](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Pet)
  + 400 Bad Request: Неверное значение статуса
* **Безопасность:**
  + petstore\_auth: write:pets, read:pets

**5.1.4. GET /pet/findByTags**

* **Краткое описание:** Найти питомцев по тегам
* **Описание:** Можно предоставить несколько тегов, разделенных запятыми. Используйте tag1, tag2, tag3 для тестирования.
* **ID операции:** findPetsByTags
* **Параметры:**
  + tags (query, массив строк, optional): Теги для фильтрации.
* **Ответы:**
  + 200 OK: Успешная операция
    - **Типы контента:** application/json, application/xml
    - **Схема:** Массив [Pet](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Pet)
  + 400 Bad Request: Неверное значение тега
* **Безопасность:**
  + petstore\_auth: write:pets, read:pets

**5.1.5. GET /pet/{petId}**

* **Краткое описание:** Найти питомца по ID
* **Описание:** Возвращает одного питомца.
* **ID операции:** getPetById
* **Параметры:**
  + petId (path, integer, required): ID питомца для возврата.
* **Ответы:**
  + 200 OK: Успешная операция
    - **Типы контента:** application/json, application/xml
    - **Схема:** [Pet](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Pet)
  + 400 Bad Request: Предоставлен неверный ID
  + 404 Not Found: Питомец не найден
* **Безопасность:**
  + api\_key
  + petstore\_auth: write:pets, read:pets

**5.1.6. POST /pet/{petId}**

* **Краткое описание:** Обновить питомца в магазине с помощью данных формы
* **Описание:**
* **ID операции:** updatePetWithForm
* **Параметры:**
  + petId (path, integer, required): ID питомца, который необходимо обновить.
  + name (query, string, optional): Имя питомца, которое необходимо обновить.
  + status (query, string, optional): Статус питомца, который необходимо обновить.
* **Ответы:**
  + 400 Bad Request: Неверный ввод
* **Безопасность:**
  + petstore\_auth: write:pets, read:pets

**5.1.7. DELETE /pet/{petId}**

* **Краткое описание:** Удалить питомца
* **Описание:** Удалить питомца.
* **ID операции:** deletePet
* **Параметры:**
  + api\_key (header, string, optional):
  + petId (path, integer, required): ID питомца для удаления.
* **Ответы:**
  + 400 Bad Request: Неверное значение питомца
* **Безопасность:**
  + petstore\_auth: write:pets, read:pets

**5.1.8. POST /pet/{petId}/uploadImage**

* **Краткое описание:** Загрузить изображение
* **Описание:**
* **ID операции:** uploadFile
* **Параметры:**
  + petId (path, integer, required): ID питомца для обновления.
  + additionalMetadata (query, string, optional): Дополнительные метаданные.
* **Тело запроса:**
  + **Тип контента:** application/octet-stream
  + **Схема:** string (binary)
* **Ответы:**
  + 200 OK: Успешная операция
    - **Тип контента:** application/json
    - **Схема:** [ApiResponse](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-ApiResponse)
* **Безопасность:**
  + petstore\_auth: write:pets, read:pets

**5.2. Store (Магазин)**

**Тег:** store

Доступ к заказам Petstore.

**5.2.1. GET /store/inventory**

* **Краткое описание:** Возвращает инвентарь питомцев по статусу
* **Описание:** Возвращает карту кодов статусов и их количества.
* **ID операции:** getInventory
* **Ответы:**
  + 200 OK: Успешная операция
    - **Тип контента:** application/json
    - **Схема:** object (карта строк в целые числа)
* **Безопасность:**
  + api\_key

**5.2.2. POST /store/order**

* **Краткое описание:** Разместить заказ на питомца
* **Описание:** Разместить новый заказ в магазине.
* **ID операции:** placeOrder
* **Тело запроса:**
  + **Типы контента:** application/json, application/xml, application/x-www-form-urlencoded
  + **Схема:** [Order](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Order)
* **Ответы:**
  + 200 OK: Успешная операция
    - **Тип контента:** application/json
    - **Схема:** [Order](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Order)
  + 400 Bad Request: Неверный ввод
  + 422 Unprocessable Entity: Ошибка валидации

**5.2.3. GET /store/order/{orderId}**

* **Краткое описание:** Найти заказ на покупку по ID
* **Описание:** Для действительного ответа попробуйте целочисленные ID со значением <= 5 или > 10. Другие значения вызовут исключения.
* **ID операции:** getOrderById
* **Параметры:**
  + orderId (path, integer, required): ID заказа, который необходимо получить.
* **Ответы:**
  + 200 OK: Успешная операция
    - **Тип контента:** application/json, application/xml
    - **Схема:** [Order](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f" \l "schema-Order)
  + 400 Bad Request: Предоставлен неверный ID
  + 404 Not Found: Заказ не найден

**5.2.4. DELETE /store/order/{orderId}**

* **Краткое описание:** Удалить заказ на покупку по ID
* **Описание:** Для действительного ответа попробуйте целочисленные ID со значением < 1000. Все, что выше 1000 или не является целым числом, вызовет ошибки API.
* **ID операции:** deleteOrder
* **Параметры:**
  + orderId (path, integer, required): ID заказа, который необходимо удалить.
* **Ответы:**
  + 400 Bad Request: Предоставлен неверный ID
  + 404 Not Found: Заказ не найден

**5.3. User (Пользователи)**

**Тег:** user

Операции, связанные с пользователями.

**5.3.1. POST /user**

* **Краткое описание:** Создать пользователя
* **Описание:** Это может быть сделано только вошедшим в систему пользователем.
* **ID операции:** createUser
* **Тело запроса:**
  + **Описание:** Созданный объект пользователя
  + **Типы контента:** application/json, application/xml, application/x-www-form-urlencoded
  + **Схема:** [User](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f#schema-User)
* **Ответы:**
  + Default: Успешная операция
    - **Типы контента:** application/json, application/xml
    - **Схема:** [User](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f#schema-User)

**5.3.2. POST /user/createWithList**

* **Краткое описание:** Создает список пользователей с заданным входным массивом
* **Описание:** Создает список пользователей с заданным входным массивом.
* **ID операции:** createUsersWithListInput
* **Тело запроса:**
  + **Типы контента:** application/json
  + **Схема:** Массив [User](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f#schema-User)
* **Ответы:**
  + 200 OK: Успешная операция
    - **Типы контента:** application/json, application/xml
    - **Схема:** [User](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f#schema-User)
  + Default: Успешная операция

**5.3.3. GET /user/login**

* **Краткое описание:** Вход пользователя в систему
* **Описание:**
* **ID операции:** loginUser
* **Параметры:**
  + username (query, string, optional): Имя пользователя для входа.
  + password (query, string, optional): Пароль для входа в виде обычного текста.
* **Ответы:**
  + 200 OK: Успешная операция
    - **Заголовки:**
      * X-Rate-Limit (integer): Количество вызовов в час, разрешенное пользователю.
      * X-Expires-After (string, date-time): Дата в формате UTC, когда истекает срок действия токена.
    - **Типы контента:** application/xml, application/json
    - **Схема:** string
  + 400 Bad Request: Предоставлено неверное имя пользователя/пароль

**5.3.4. GET /user/logout**

* **Краткое описание:** Выход текущего вошедшего в систему пользователя
* **Описание:**
* **ID операции:** logoutUser
* **Ответы:**
  + Default: Успешная операция

**5.3.5. GET /user/{username}**

* **Краткое описание:** Получить пользователя по имени пользователя
* **Описание:**
* **ID операции:** getUserByName
* **Параметры:**
  + username (path, string, required): Имя, которое необходимо получить. Используйте user1 для тестирования.
* **Ответы:**
  + 200 OK: Успешная операция
    - **Типы контента:** application/json, application/xml
    - **Схема:** [User](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f#schema-User)
  + 400 Bad Request: Предоставлено неверное имя пользователя
  + 404 Not Found: Пользователь не найден

**5.3.6. PUT /user/{username}**

* **Краткое описание:** Обновить пользователя
* **Описание:** Это может быть сделано только вошедшим в систему пользователем.
* **ID операции:** updateUser
* **Параметры:**
  + username (path, string, required): Имя, которое необходимо удалить.
* **Тело запроса:**
  + **Описание:** Обновить существующего пользователя в магазине
  + **Типы контента:** application/json, application/xml, application/x-www-form-urlencoded
  + **Схема:** [User](https://chat.screwltd.com/chat?id=ca924880-b1e0-4bf2-8592-06aec0fd050f#schema-User)
* **Ответы:**
  + Default: Успешная операция

**5.3.7. DELETE /user/{username}**

* **Краткое описание:** Удалить пользователя
* **Описание:** Это может быть сделано только вошедшим в систему пользователем.
* **ID операции:** deleteUser
* **Параметры:**
  + username (path, string, required): Имя, которое необходимо удалить.
* **Ответы:**
  + 400 Bad Request: Предоставлено неверное имя пользователя
  + 404 Not Found: Пользователь не найден

6. Схемы  
В этом разделе описаны структуры данных, используемые в API.

6.1. Order

* **JSON**
* {
* "id": {
* "type": "integer",
* "format": "int64",
* "example": 10
* },
* "petId": {
* "type": "integer",
* "format": "int64",
* "example": 198772
* },
* "quantity": {
* "type": "integer",
* "format": "int32",
* "example": 7
* },
* "shipDate": {
* "type": "string",
* "format": "date-time"
* },
* "status": {
* "type": "string",
* "description": "Order Status",
* "example": "approved",
* "enum": [
* "placed",
* "approved",
* "delivered"
* ]
* },
* "complete": {
* "type": "boolean"
* }
* }

6.2. Customer

* **JSON**
* {
* "id": {
* "type": "integer",
* "format": "int64",
* "example": 100000
* },
* "username": {
* "type": "string",
* "example": "fehguy"
* },
* "address": {
* "type": "array",
* "items": {
* "$ref": "#/components/schemas/Address"
* }
* }
* }

6.3. Address

* **JSON**
* {
* "street": {
* "type": "string",
* "example": "437 Lytton"
* },
* "city": {
* "type": "string",
* "example": "Palo Alto"
* },
* "state": {
* "type": "string",
* "example": "CA"
* },
* "zip": {
* "type": "string",
* "example": "94301"
* }
* }

6.4. Category

* **JSON**
* {
* "id": {
* "type": "integer",
* "format": "int64",
* "example": 1
* },
* "name": {
* "type": "string",
* "example": "Dogs"
* }
* }

6.5. User

* **JSON**
* {
* "id": {
* "type": "integer",
* "format": "int64",
* "example": 10
* },
* "username": {
* "type": "string",
* "example": "theUser"
* },
* "firstName": {
* "type": "string",
* "example": "John"
* },
* "lastName": {
* "type": "string",
* "example": "James"
* },
* "email": {
* "type": "string",
* "example": "john@email.com"
* },
* "password": {
* "type": "string",
* "example": "12345"
* },
* "phone": {
* "type": "string",
* "example": "12345"
* },
* "userStatus": {
* "type": "integer",
* "description": "User Status",
* "format": "int32",
* "example": 1
* }
* }

6.6 Tag

* **JSON**
* {
* "id": {
* "type": "integer",
* "format": "int64"
* },
* "name": {
* "type": "string"
* }
* }

6.7. Pet

* **JSON**
* {
* "id": {
* "type": "integer",
* "format": "int64",
* "example": 10
* },
* "name": {
* "type": "string",
* "example": "doggie"
* },
* "category": {
* "$ref": "#/components/schemas/Category"
* },
* "photoUrls": {
* "type": "array",
* "items": {
* "type": "string"
* }
* },
* "tags": {
* "type": "array",
* "items": {
* "$ref": "#/components/schemas/Tag"
* }
* },
* "status": {
* "type": "string",
* "description": "pet status in the store",
* "enum": [
* "available",
* "pending",
* "sold"
* ]
* }
* }

6.8. ApiResponse

* **JSON**
* {
* "code": {
* "type": "integer",
* "format": "int32"
* },
* "type": {
* "type": "string"
* },
* "message": {
* "type": "string"
* }
* }

**7. Заключение**

Этот документ предоставляет исчерпывающий обзор API Pet Store. Следуя этим рекомендациям, разработчики могут легко интегрировать API в свои приложения и воспользоваться его функциями.

**Примечание:** Эта документация основана на предоставленной спецификации OpenAPI и может быть изменена. Всегда обращайтесь к последней версии документации для получения самой актуальной информации.