



Swagger Petstore

# Table of Contents

Overview .....	1
Version information .....	1
Contact information .....	1
License information .....	1
URI scheme .....	1
Tags .....	1
Operations .....	2
Pets .....	2
Finds Pets by status .....	2
Finds Pets by tags .....	3
Updates a pet in the store with form data .....	4
Find pet by ID .....	5
Deletes a pet .....	6
Stores .....	7
Place an order for a pet .....	7
Find purchase order by ID .....	8
Delete purchase order by ID .....	9
Users .....	10
Creates list of users with given input array .....	10
Creates list of users with given input array .....	11
Logs user into the system .....	12
Logs out current logged in user session .....	13
Get user by user name .....	14
Updated user .....	15
Delete user .....	16
Definitions .....	17
Category .....	17
Order .....	17
Pet .....	17
Tag .....	18
User .....	18
Security .....	19
petstore_auth .....	19
api_key .....	19

## Overview

This is a sample server Petstore server.

Learn about [Swagger](#) or join the IRC channel [#swagger](#) on [irc.freenode.net](#).

For this sample, you can use the api key [special-key](#) to test the authorization filters

## Version information

*Version* : 1.0.0

## Contact information

*Contact* : [apiteam@swagger.io](mailto:apiteam@swagger.io)

## License information

*License* : Apache 2.0

*License URL* : <http://www.apache.org/licenses/LICENSE-2.0.html>

*Terms of service* : <http://helloverb.com/terms/>

## URI scheme

*Host* : petstore.swagger.io

*BasePath* : /v2

*Schemes* : HTTP

## Tags

- pet : Pet resource
- store : Store resource
- user : User resource

# Operations

## Pets

### Finds Pets by status

```
GET /pets/findByStatus
```

#### Description

Multiple status values can be provided with comma separated strings

#### Parameters

Type	Name	Description	Schema
Query	<b>status</b> <i>optional</i>	Status values that need to be considered for filter	< string > array(multi)

#### Responses

HTTP Code	Description	Schema
200	successful operation <b>Headers :</b> <b>X-Rate-Limit-Limit</b> (integer) : The number of allowed requests in the current period. <b>X-Rate-Limit-Remaining</b> (integer) : The number of remaining requests in the current period. <b>X-Rate-Limit-Reset</b> (integer) : The number of seconds left in the current period.	< <a href="#">Pet</a> > array
400	Invalid status value	No Content

#### Produces

- `application/json`
- `application/xml`

#### Security

Type	Name	Scopes
oauth2	<a href="#">petstore_auth</a>	write_pets,read_pets

## Finds Pets by tags

```
GET /pets/findByTags
```

### Description

Multiple tags can be provided with comma separated strings. Use tag1, tag2, tag3 for testing.

### Parameters

Type	Name	Description	Schema
Query	<b>tags</b> <i>optional</i>	Tags to filter by	< string > array(multi)

### Responses

HTTP Code	Description	Schema
200	successful operation <b>Headers :</b> <b>X-Rate-Limit-Limit</b> (integer) : The number of allowed requests in the current period. <b>X-Rate-Limit-Remaining</b> (integer) : The number of remaining requests in the current period. <b>X-Rate-Limit-Reset</b> (integer) : The number of seconds left in the current period.	< <a href="#">Pet</a> > array
400	Invalid tag value	No Content

### Produces

- `application/json`
- `application/xml`

### Security

Type	Name	Scopes
oauth2	<a href="#">petstore_auth</a>	write_pets,read_pets

## Updates a pet in the store with form data

POST /pets/{petId}

### Parameters

Type	Name	Description	Schema
Path	<b>petId</b> <i>required</i>	ID of pet that needs to be updated	string
FormData	<b>name</b> <i>required</i>	Updated name of the pet	string
FormData	<b>status</b> <i>required</i>	Updated status of the pet	string

### Responses

HTTP Code	Description	Schema
405	Invalid input	No Content

### Consumes

- `application/x-www-form-urlencoded`

### Produces

- `application/json`
- `application/xml`

### Security

Type	Name	Scopes
oauth2	<b>petstore_auth</b>	write_pets,read_pets

## Find pet by ID

```
GET /pets/{petId}
```

### Description

Returns a pet when ID < 10. ID > 10 or nonintegers will simulate API error conditions

### Parameters

Type	Name	Description	Schema
Path	<b>petId</b> <i>required</i>	ID of pet that needs to be fetched	integer(int64)

### Responses

HTTP Code	Description	Schema
200	successful operation <b>Headers :</b> <b>X-Rate-Limit-Limit</b> (integer) : The number of allowed requests in the current period. <b>X-Rate-Limit-Remaining</b> (integer) : The number of remaining requests in the current period. <b>X-Rate-Limit-Reset</b> (integer) : The number of seconds left in the current period.	<a href="#">Pet</a>
400	Invalid ID supplied	No Content
404	Pet not found	No Content

### Produces

- `application/json`
- `application/xml`

### Security

Type	Name	Scopes
apiKey	<a href="#">api_key</a>	
oauth2	<a href="#">petstore_auth</a>	write_pets,read_pets

## Deletes a pet

```
DELETE /pets/{petId}
```

### Parameters

Type	Name	Description	Schema
Header	<b>api_key</b> <i>required</i>		string
Path	<b>petId</b> <i>required</i>	Pet id to delete	integer(int64)

### Responses

HTTP Code	Description	Schema
<b>400</b>	Invalid pet value	No Content

### Produces

- `application/json`
- `application/xml`

### Security

Type	Name	Scopes
<b>oauth2</b>	<b>petstore_auth</b>	write_pets,read_pets



## Stores

### Place an order for a pet

POST /stores/order

#### Parameters

Type	Name	Description	Schema
Body	<b>body</b> <i>optional</i>	order placed for purchasing the pet	<a href="#">Order</a>

#### Responses

HTTP Code	Description	Schema
200	successful operation <b>Headers :</b> <b>X-Rate-Limit-Limit</b> (integer) : The number of allowed requests in the current period. <b>X-Rate-Limit-Remaining</b> (integer) : The number of remaining requests in the current period. <b>X-Rate-Limit-Reset</b> (integer) : The number of seconds left in the current period.	<a href="#">Order</a>
400	Invalid Order	No Content

#### Produces

- `application/json`
- `application/xml`

## Find purchase order by ID

```
GET /stores/order/{orderId}
```

### Description

For valid response try integer IDs with value  $\leq 5$  or  $> 10$ . Other values will generated exceptions

### Parameters

Type	Name	Description	Schema
Path	<b>orderId</b> <i>required</i>	ID of pet that needs to be fetched	string

### Responses

HTTP Code	Description	Schema
<b>200</b>	successful operation <b>Headers :</b> <b>X-Rate-Limit-Limit</b> (integer) : The number of allowed requests in the current period. <b>X-Rate-Limit-Remaining</b> (integer) : The number of remaining requests in the current period. <b>X-Rate-Limit-Reset</b> (integer) : The number of seconds left in the current period.	<a href="#">Order</a>
<b>400</b>	Invalid ID supplied	No Content
<b>404</b>	Order not found	No Content

### Produces

- `application/json`
- `application/xml`

## Delete purchase order by ID

```
DELETE /stores/order/{orderId}
```

### Description

For valid response try integer IDs with value < 1000. Anything above 1000 or nonintegers will generate API errors

### Parameters

Type	Name	Description	Schema
Path	<b>orderId</b> <i>required</i>	ID of the order that needs to be deleted	string

### Responses

HTTP Code	Description	Schema
<b>400</b>	Invalid ID supplied	No Content
<b>404</b>	Order not found	No Content

### Produces

- `application/json`
- `application/xml`

## Users

### Creates list of users with given input array

POST /users/createWithArray

#### Parameters

Type	Name	Description	Schema
Body	<b>body</b> <i>optional</i>	List of user object	< <a href="#">User</a> > array

#### Responses

HTTP Code	Description	Schema
<b>default</b>	successful operation	No Content

#### Produces

- `application/json`
- `application/xml`

## Creates list of users with given input array

POST /users/createWithList

### Parameters

Type	Name	Description	Schema
Body	<b>body</b> <i>optional</i>	List of user object	< <a href="#">User</a> > array

### Responses

HTTP Code	Description	Schema
default	successful operation	No Content

### Produces

- `application/json`
- `application/xml`

## Logs user into the system

GET /users/login

### Parameters

Type	Name	Description	Schema
Query	<b>password</b> <i>optional</i>	The password for login in clear text	string
Query	<b>username</b> <i>optional</i>	The user name for login	string

### Responses

HTTP Code	Description	Schema
<b>200</b>	successful operation <b>Headers :</b> <b>X-Rate-Limit-Limit</b> (integer) : The number of allowed requests in the current period. <b>X-Rate-Limit-Remaining</b> (integer) : The number of remaining requests in the current period. <b>X-Rate-Limit-Reset</b> (integer) : The number of seconds left in the current period.	string
<b>400</b>	Invalid username/password supplied	No Content

### Produces

- `application/json`
- `application/xml`

## Logs out current logged in user session

GET /users/logout

### Responses

HTTP Code	Description	Schema
<b>default</b>	successful operation	No Content

### Produces

- `application/json`
- `application/xml`

## Get user by user name

```
GET /users/{username}
```

### Parameters

Type	Name	Description	Schema
Path	<b>username</b> <i>required</i>	The name that needs to be fetched. Use user1 for testing.	string

### Responses

HTTP Code	Description	Schema
200	successful operation <b>Headers :</b> <b>X-Rate-Limit-Limit</b> (integer) : The number of allowed requests in the current period. <b>X-Rate-Limit-Remaining</b> (integer) : The number of remaining requests in the current period. <b>X-Rate-Limit-Reset</b> (integer) : The number of seconds left in the current period.	User
400	Invalid username supplied	No Content
404	User not found	No Content

### Produces

- application/json
- application/xml



## Updated user

```
PUT /users/{username}
```

### Description

This can only be done by the logged in user.

### Parameters

Type	Name	Description	Schema
Path	<b>username</b> <i>required</i>	name that need to be deleted	string
Body	<b>body</b> <i>optional</i>	Updated user object	<a href="#">User</a>

### Responses

HTTP Code	Description	Schema
400	Invalid user supplied	No Content
404	User not found	No Content

### Produces

- `application/json`
- `application/xml`

## Delete user

```
DELETE /users/{username}
```

### Description

This can only be done by the logged in user.

### Parameters

Type	Name	Description	Schema
Path	<b>username</b> <i>required</i>	The name that needs to be deleted	string

### Responses

HTTP Code	Description	Schema
<b>400</b>	Invalid username supplied	No Content
<b>404</b>	User not found	No Content

### Produces

- `application/json`
- `application/xml`

## Definitions

### Category

Name	Description	Schema
<b>id</b> <i>optional</i>		integer(int64)
<b>name</b> <i>optional</i>	The name of the category <b>Default</b> : "DefaultCategory" <b>Length</b> : 0 - 255 <b>Pattern</b> : "[A-Za-zäöüÄÖÜß]{0,255}" <b>Example</b> : "FoobarCategory"	string

### Order

Name	Description	Schema
<b>complete</b> <i>optional</i>		boolean
<b>id</b> <i>optional</i>		integer(int64)
<b>petId</b> <i>optional</i>		integer(int64)
<b>quantity</b> <i>optional</i>	<b>Default</b> : 0 <b>Minimum value</b> : 0 <b>Maximum value</b> : 10000 <b>Example</b> : 10	integer(int32)
<b>shipDate</b> <i>optional</i>		string(date-time)
<b>status</b> <i>optional</i>	Order Status	enum (Ordered, Cancelled)

### Pet

Name	Description	Schema
<b>category</b> <i>optional</i>		<a href="#">Category</a>
<b>id</b> <i>optional</i>		integer(int64)
<b>name</b> <i>required</i>	<b>Example</b> : "doggie"	string
<b>photoUrls</b> <i>required</i>		< string > array
<b>status</b> <i>optional</i>	pet status in the store,	enum (Dead, Alive)

Name	Description	Schema
<b>tags</b> <i>optional</i>		< Tag > array

## Tag

Name	Schema
<b>id</b> <i>optional</i>	integer(int64)
<b>name</b> <i>optional</i>	string

## User

Name	Description	Schema
<b>email</b> <i>optional</i>		string
<b>firstName</b> <i>optional</i>		string
<b>id</b> <i>optional</i>		integer(int64)
<b>lastName</b> <i>optional</i>		string
<b>password</b> <i>optional</i>		string
<b>phone</b> <i>optional</i>		string
<b>userStatus</b> <i>optional</i>	User Status	integer(int32)
<b>username</b> <i>optional</i>		string

# Security

## petstore\_auth

Type : oauth2

Flow : implicit

Token URL : <http://petstore.swagger.io/api/oauth/dialog>

Name	Description
write_pets	modify pets in your account
read_pets	read your pets

## api\_key

Type : apiKey

Name : api\_key

In : HEADER