

Swagger Petstore

Overview

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on "irc.freenode.net, [#swagger](#)". 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 Email : apiteam@swagger.io

License information

License : Apache 2.0

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

Terms of service : <http://swagger.io/terms/>

URI scheme

Host : petstore.swagger.io

BasePath : /v2

Schemes : HTTPS, HTTP

Tags

- pet : Everything about your Pets
- store : Access to Petstore orders
- user : Operations about user

External Docs

Description : Find out more about Swagger

URL : <http://swagger.io>

Paths

Add a new pet to the store

POST /pet

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Pet object that needs to be added to the store	Pet

Responses

HTTP Code	Description	Schema
405	Invalid input	No Content

Consumes

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

Produces

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

Tags

- `pet`

Security

Type	Name	Scopes
oauth2	petstore_auth	write:pets,read:pets

Update an existing pet

PUT /pet

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Pet object that needs to be added to the store	Pet

Responses

HTTP Code	Description	Schema
400	Invalid ID supplied	No Content
404	Pet not found	No Content
405	Validation exception	No Content

Consumes

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

Produces

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

Tags

- pet

Security

Type	Name	Scopes
oauth2	petstore_auth	write:pets,read:pets

Finds Pets by status

```
GET /pet/findByStatus
```

Description

Multiple status values can be provided with comma separated strings

Parameters

Type	Name	Description	Schema
Query	status <i>required</i>	Status values that need to be considered for filter	< enum (available, pending, sold) > array(multi)

Responses

HTTP Code	Description	Schema
200	successful operation	< Pet > array
400	Invalid status value	No Content

Produces

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

Tags

- `pet`

Security

Type	Name	Scopes
oauth2	petstore_auth	write:pets,read:pets

Finds Pets by tags

GET /pet/findByTags

CAUTION | operation.deprecated

Description

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

Parameters

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

Responses

HTTP Code	Description	Schema
200	successful operation	< Pet > array
400	Invalid tag value	No Content

Produces

- [application/xml](#)
- [application/json](#)

Tags

- pet

Security

Type	Name	Scopes
oauth2	petstore_auth	write:pets,read:pets

Updates a pet in the store with form data

POST /pet/{petId}

Parameters

Type	Name	Description	Schema
Path	petId <i>required</i>	ID of pet that needs to be updated	integer (int64)
FormData	name <i>optional</i>	Updated name of the pet	string
FormData	status <i>optional</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/xml`
- `application/json`

Tags

- pet

Security

Type	Name	Scopes
oauth2	petstore_auth	write:pets,read:pets

Find pet by ID

```
GET /pet/{petId}
```

Description

Returns a single pet

Parameters

Type	Name	Description	Schema
Path	petId <i>required</i>	ID of pet to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	successful operation	Pet
400	Invalid ID supplied	No Content
404	Pet not found	No Content

Produces

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

Tags

- pet

Security

Type	Name
apiKey	<code>api_key</code>

Deletes a pet

```
DELETE /pet/{petId}
```

Parameters

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

Responses

HTTP Code	Description	Schema
400	Invalid ID supplied	No Content
404	Pet not found	No Content

Produces

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

Tags

- pet

Security

Type	Name	Scopes
oauth2	petstore_auth	write:pets,read:pets

uploads an image

POST /pet/{petId}/uploadImage

Parameters

Type	Name	Description	Schema
Path	petId <i>required</i>	ID of pet to update	integer (int64)
FormData	additionalMetadata <i>optional</i>	Additional data to pass to server	string
FormData	file <i>optional</i>	file to upload	file

Responses

HTTP Code	Description	Schema
200	successful operation	ApiResponse

Consumes

- `multipart/form-data`

Produces

- `application/json`

Tags

- pet

Security

Type	Name	Scopes
oauth2	petstore_auth	write:pets,read:pets

Returns pet inventories by status

GET /store/inventory

Description

Returns a map of status codes to quantities

Responses

HTTP Code	Description	Schema
200	successful operation	< string, integer (int32) > map

Produces

- `application/json`

Tags

- store

Security

Type	Name
apiKey	api_key

Place an order for a pet

POST /store/order

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	order placed for purchasing the pet	Order

Responses

HTTP Code	Description	Schema
200	successful operation	Order
400	Invalid Order	No Content

Produces

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

Tags

- store

Find purchase order by ID

```
GET /store/order/{orderId}
```

Description

For valid response try integer IDs with value ≥ 1 and ≤ 10 . Other values will generated exceptions

Parameters

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

Responses

HTTP Code	Description	Schema
200	successful operation	Order

HTTP Code	Description	Schema
400	Invalid ID supplied	No Content
404	Order not found	No Content

Produces

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

Tags

- store

Delete purchase order by ID

```
DELETE /store/order/{orderId}
```

Description

For valid response try integer IDs with positive integer value. Negative or non-integer values will generate API errors

Parameters

Type	Name	Description	Schema
Path	orderId <i>required</i>	ID of the order that needs to be deleted	integer (int64)

Responses

HTTP Code	Description	Schema
400	Invalid ID supplied	No Content
404	Order not found	No Content

Produces

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

Tags

- store

Create user

POST /user

Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Created user object	User

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Creates list of users with given input array

POST /user/createWithArray

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	List of user object	< User > array

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Creates list of users with given input array

POST /user/createWithList

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	List of user object	< User > array

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Logs user into the system

GET /user/login

Parameters

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

Responses

HTTP Code	Description	Schema
200	successful operation Headers : X-Rate-Limit (integer (int32)) : calls per hour allowed by the user. X-Expires-After (string (date-time)) : date in UTC when token expires.	string
400	Invalid username/password supplied	No Content

Produces

- application/xml
- application/json

Tags

- user

Logs out current logged in user session

GET /user/logout

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Get user by user name

```
GET /user/{username}
```

Parameters

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

Responses

HTTP Code	Description	Schema
200	successful operation	User
400	Invalid username supplied	No Content
404	User not found	No Content

Produces

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

Tags

- user

Updated user

```
PUT /user/{username}
```

Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Schema
Path	username <i>required</i>	name that need to be updated	string
Body	body <i>required</i>	Updated user object	User

Responses

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

Produces

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

Tags

- user

Delete user

```
DELETE /user/{username}
```

Description

This can only be done by the logged in user.

Parameters

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

Responses

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

Produces

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

Tags

- user

Definitions

ApiResponse

Name	Schema
code <i>optional</i>	integer (int32)
message <i>optional</i>	string
type <i>optional</i>	string

Category

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

Order

Name	Description	Schema
complete <i>optional</i>	Default : false	boolean
id <i>optional</i>		integer (int64)
petId <i>optional</i>		integer (int64)
quantity <i>optional</i>		integer (int32)
shipDate <i>optional</i>		string (date-time)
status <i>optional</i>	Order Status	enum (placed, approved, delivered)

Pet

Name	Description	Schema
category <i>optional</i>		Category
id <i>optional</i>		integer (int64)
name <i>required</i>	Example : "doggie"	string

Name	Description	Schema
photoUrls <i>required</i>		< string > array
status <i>optional</i>	pet status in the store	enum (available, pending, sold)
tags <i>optional</i>		< Tag > array

Tag

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

User

Name	Description	Schema
email <i>optional</i>		string
firstName <i>optional</i>		string
id <i>optional</i>		integer (int64)
lastName <i>optional</i>		string
password <i>optional</i>		string
phone <i>optional</i>		string

Name	Description	Schema
userStatus <i>optional</i>	User Status	integer (int32)
username <i>optional</i>		string

Security

petstore_auth

Type : oauth2

Flow : implicit

Token URL : <http://petstore.swagger.io/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