ISAAC

Swagger Petstore

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

License information

License : Apache 2.0
License URL : http://www.apache.org/licenses/LICENSE-2.0.html
Terms of service : http://helloreverb.com/terms/

URI scheme

Host : petstore.swagger.io
BasePath : /v2
Schemes : HTTP

Tags

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

Paths

Add a new pet to the store

```
POST /pets
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	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/json`
- `application/xml`

Tags

- pet

Security

Type	Name	Scopes
oauth2	**petstore_auth**	write_pets,read_pets

Example HTTP request

Request path

```
/pets
```

Request body

```
{
  "id" : 0,
  "category" : {
    "id" : 0,
    "name" : "FoobarCategory"
  },
  "name" : "doggie",
  "photoUrls" : [ "string" ],
  "tags" : [ {
    "id" : 0,
    "name" : "string"
  } ],
  "status" : "string"
}
```

Update an existing pet

```
PUT /pets
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	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/json`
- `application/xml`

Tags

- pet

Security

Type	Name	Scopes
oauth2	**petstore_auth**	write_pets,read_pets

Example HTTP request

Request path

```
/pets
```

Request body

```
{
  "id" : 0,
  "category" : {
    "id" : 0,
    "name" : "FoobarCategory"
  },
  "name" : "doggie",
  "photoUrls" : [ "string" ],
  "tags" : [ {
    "id" : 0,
    "name" : "string"
  } ],
  "status" : "string"
}
```

Finds Pets by status

```
GET /pets/findByStatus
```

Description

Multiple status values can be provided with comma seperated strings

Parameters

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

Responses

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

Produces

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

Tags

- pet

Security

Type	Name	Scopes
oauth2	**petstore_auth**	write_pets,read_pets

Example HTTP request

Request path

```
/pets/findByStatus
```

Request query

```
{
  "status" : "string"
}
```

Example HTTP response

Response 200

```
"array"
```

Finds Pets by tags

```
GET /pets/findByTags
```

Description

Muliple tags can be provided with comma seperated strings. Use tag1, tag2, tag3 for testing.

Parameters

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

Responses

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

Produces

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

Tags

- pet

Security

Type	Name	Scopes
oauth2	**petstore_auth**	write_pets,read_pets

Example HTTP request

Request path

```
/pets/findByTags
```

Request query

```
{
  "tags" : "string"
}
```

Example HTTP response

Response 200

```
"array"
```

Updates a pet in the store with form data

```
POST /pets/{petId}
```

Parameters

Type	Name	Description	Schema
Path	**petId** *required*	ID of pet that needs to be updated	string
FormData	**name** *required*	Updated name of the pet	string
FormData	**status** *required*	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`

Tags

- pet

Security

Type	Name	Scopes
oauth2	**petstore_auth**	write_pets,read_pets

Example HTTP request

Request path

```
/pets/string
```

Request formData

```
"string"
```

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	**petId** *required*	ID of pet that needs to be fetched	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation **Headers** : `X-Rate-Limit-Limit` (integer) : The number of allowed requests in the current period. `X-Rate-Limit-Remaining` (integer) : The number of remaining requests in the current period. `X-Rate-Limit-Reset` (integer) : The number of seconds left in the current period.	Pet
400	Invalid ID supplied	No Content
404	Pet not found	No Content

Produces

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

Tags

- pet

Security

Type	Name	Scopes
apiKey	**api_key**	
oauth2	**petstore_auth**	write_pets,read_pets

Example HTTP request

Request path

```
/pets/0
```

Example HTTP response

Response 200

```json
{
  "id" : 0,
  "category" : {
    "id" : 0,
    "name" : "FoobarCategory"
  },
  "name" : "doggie",
  "photoUrls" : [ "string" ],
  "tags" : [ {
    "id" : 0,
    "name" : "string"
  } ],
  "status" : "string"
}
```

Deletes a pet

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

Parameters

Type	Name	Description	Schema
Header	**api_key** *required*		string
Path	**petId** *required*	Pet id to delete	integer(int64)

Responses

HTTP Code	Description	Schema
400	Invalid pet value	No Content

Produces

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

Tags

- pet

Security

Type	Name	Scopes
oauth2	**petstore_auth**	write_pets,read_pets

Example HTTP request

Request path

```
/pets/0
```

Request header

```
"string"
```

Place an order for a pet

```
POST /stores/order
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	order placed for purchasing the pet	Order

Responses

HTTP Code	Description	Schema
200	successful operation **Headers** : `X-Rate-Limit-Limit` (integer) : The number of allowed requests in the current period. `X-Rate-Limit-Remaining` (integer) : The number of remaining requests in the current period. `X-Rate-Limit-Reset` (integer) : The number of seconds left in the current period.	Order
400	Invalid Order	No Content

Produces

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

Tags

- store

Example HTTP request

Request path

```
/stores/order
```

Request body

```
{
  "id" : 0,
  "petId" : 0,
  "quantity" : 10,
  "shipDate" : "string",
  "status" : "string",
  "complete" : true
}
```

Example HTTP response

Response 200

```
{
  "id" : 0,
  "petId" : 0,
  "quantity" : 10,
  "shipDate" : "string",
  "status" : "string",
  "complete" : true
}
```

Find purchase order by ID

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

Description

For valid response try integer IDs with value <= 5 or > 10. Other values will generated exceptions

Parameters

Type	Name	Description	Schema
Path	**orderId** *required*	ID of pet that needs to be fetched	string

Responses

HTTP Code	Description	Schema
200	successful operation **Headers** : `X-Rate-Limit-Limit` (integer) : The number of allowed requests in the current period. `X-Rate-Limit-Remaining` (integer) : The number of remaining requests in the current period. `X-Rate-Limit-Reset` (integer) : The number of seconds left in the current period.	Order
400	Invalid ID supplied	No Content
404	Order not found	No Content

Produces

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

Tags

- store

Example HTTP request

Request path

```
/stores/order/string
```

Example HTTP response

Response 200

```
{
  "id" : 0,
  "petId" : 0,
  "quantity" : 10,
  "shipDate" : "string",
  "status" : "string",
  "complete" : true
}
```

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	**orderId** *required*	ID of the order that needs to be deleted	string

Responses

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

Produces

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

Tags

- store

Example HTTP request

Request path

```
/stores/order/string
```

Create user

```
POST /users
```

Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	Created user object	User

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Example HTTP request

Request path

```
/users
```

Request body

```
{
  "id" : 0,
  "username" : "string",
  "firstName" : "string",
  "lastName" : "string",
  "email" : "string",
  "password" : "string",
  "phone" : "string",
  "userStatus" : 0
}
```

Creates list of users with given input array

```
POST /users/createWithArray
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	List of user object	< User > array

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Example HTTP request

Request path

```
/users/createWithArray
```

Request body

```json
[ {
  "id" : 0,
  "username" : "string",
  "firstName" : "string",
  "lastName" : "string",
  "email" : "string",
  "password" : "string",
  "phone" : "string",
  "userStatus" : 0
} ]
```

Creates list of users with given input array

```
POST /users/createWithList
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	List of user object	< User > array

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Example HTTP request

Request path

```
/users/createWithList
```

Request body

```
[ {
  "id" : 0,
  "username" : "string",
  "firstName" : "string",
  "lastName" : "string",
  "email" : "string",
  "password" : "string",
  "phone" : "string",
  "userStatus" : 0
} ]
```

Logs user into the system

```
GET /users/login
```

Parameters

Type	Name	Description	Schema
Query	**password** *optional*	The password for login in clear text	string
Query	**username** *optional*	The user name for login	string

Responses

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

Produces

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

Tags

- user

Example HTTP request

Request path

```
/users/login
```

Request query

```
{
  "password" : "string",
  "username" : "string"
}
```

Example HTTP response

Response 200

```
"string"
```

Logs out current logged in user session

```
GET /users/logout
```

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

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

Tags

- user

Example HTTP request

Request path

```
/users/logout
```

Get user by user name

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

Parameters

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

Responses

HTTP Code	Description	Schema
200	successful operation **Headers** : `X-Rate-Limit-Limit` (integer) : The number of allowed requests in the current period. `X-Rate-Limit-Remaining` (integer) : The number of remaining requests in the current period. `X-Rate-Limit-Reset` (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`

Tags

- user

Example HTTP request

Request path

```
/users/string
```

Example HTTP response

Response 200

```
{
  "id" : 0,
  "username" : "string",
  "firstName" : "string",
  "lastName" : "string",
  "email" : "string",
  "password" : "string",
  "phone" : "string",
  "userStatus" : 0
}
```

Updated user

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

Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Schema
Path	**username** *required*	name that need to be deleted	string
Body	**body** *optional*	Updated user object	User

Responses

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

Produces

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

Tags

- user

Example HTTP request

Request path

```
/users/string
```

Request body

```
{
  "id" : 0,
  "username" : "string",
  "firstName" : "string",
  "lastName" : "string",
  "email" : "string",
  "password" : "string",
  "phone" : "string",
  "userStatus" : 0
}
```

Delete user

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

Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Schema
Path	**username** *required*	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/json`
- `application/xml`

Tags

- user

Example HTTP request

Request path

```
/users/string
```

Definitions

Category

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

Order

Name	Description	Schema
complete *optional*	**Example** : `true`	boolean
id *optional*	**Example** : `0`	integer(int64)
petId *optional*	**Example** : `0`	integer(int64)
quantity *optional*	**Default** : `0` **Minimum value** : `0` **Maximum value** : `10000` **Example** : `10`	integer(int32)
shipDate *optional*	**Example** : `"string"`	string(date-time)
status *optional*	Order Status **Example** : `"string"`	enum (Ordered, Cancelled)

Pet

Name	Description	Schema
category *optional*	**Example** : `"Category"`	Category
id *optional*	**Example** : `0`	integer(int64)
name *required*	**Example** : `"doggie"`	string
photoUrls *required*	**Example** : `[ "string" ]`	< string > array

Name	Description	Schema
status *optional*	pet status in the store, **Example** : `"string"`	enum (Dead, Alive)
tags *optional*	**Example** : `[ "Tag" ]`	< Tag > array

Tag

Name	Description	Schema
id *optional*	**Example** : `0`	integer(int64)
name *optional*	**Example** : `"string"`	string

User

Name	Description	Schema
email *optional*	**Example** : `"string"`	string
firstName *optional*	**Example** : `"string"`	string
id *optional*	**Example** : `0`	integer(int64)
lastName *optional*	**Example** : `"string"`	string
password *optional*	**Example** : `"string"`	string
phone *optional*	**Example** : `"string"`	string
userStatus *optional*	User Status **Example** : `0`	integer(int32)
username *optional*	**Example** : `"string"`	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