

Swagger Petstore

Table of Contents

1. Overview	1
1.1. Version information	1
1.2. Contact information	1
1.3. License information	1
1.4. URI scheme	1
1.5. Tags	1
2. Definitions	2
2.1. Category	2
2.2. Order	2
2.3. Pet	2
2.4. Tag	2
2.5. User	2
3. Resources	4
3.1. Pets	4
3.1.1. Update an existing pet	4
3.1.2. Add a new pet to the store	4
3.1.3. Finds Pets by tags	5
3.1.4. Finds Pets by status	6
3.1.5. Find pet by ID	7
3.2. Stores	7
3.2.1. Place an order for a pet	7
3.2.2. Find purchase order by ID	8
3.2.3. Delete purchase order by ID	9
3.3. Users	10
3.3.1. Creates list of users with given input array	10
3.3.2. Creates list of users with given input array	10
3.3.3. Logs out current logged in user session	11
3.3.4. Logs user into the system	11
3.3.5. Updated user	12
3.3.6. Get user by user name	13
3.3.7. Delete user	14
3.3.8. Create user	14

Chapter 1. Overview

Petstore API Description

1.1. Version information

Version: 1.0.0

1.2. Contact information

Contact: TestName

Contact Email: test@test.de

1.3. License information

License: Apache 2.0

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

1.4. URI scheme

Host: localhost:8080

BasePath: /

1.5. Tags

- Pets: Operations about pets
- Stores: Operations about store
- Users: Operations about user

Chapter 2. Definitions

2.1. Category

Name	Description	Required	Schema	Default
id		false	integer (int64)	
name		false	string	

2.2. Order

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

2.3. Pet

Name	Description	Required	Schema	Default
category		false	Category	
id		false	integer (int64)	
name		false	string	
photoUrls		false	string array	
status	pet status in the store	false	enum (available, pending, sold)	
tags		false	Tag array	

2.4. Tag

Name	Description	Required	Schema	Default
id		false	integer (int64)	
name		false	string	

2.5. User

Name	Description	Required	Schema	Default
email		false	string	
firstName		false	string	
id		false	integer (int64)	
lastName		false	string	
password		false	string	
phone		false	string	
userStatus	User Status	false	integer (int32)	
username		false	string	

Chapter 3. Resources

3.1. Pets

Operations about pets

3.1.1. Update an existing pet

PUT /pets

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	pet	Pet object that needs to be added to the store	true	Pet	

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
400	Invalid ID supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Pet not found	No Content
405	Validation exception	No Content

Consumes

- application/json

Produces

- application/xml
- application/x-smile
- application/json

3.1.2. Add a new pet to the store

POST /pets

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	pet	Pet object that needs to be added to the store	true	Pet	

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content
405	Invalid input	No Content

Consumes

- application/json

Produces

- application/xml
- application/x-smile
- application/json

3.1.3. 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	Required	Schema	Default
QueryParamter	tags	Tags to filter by	true	string	

Responses

HTTP Code	Description	Schema
200	OK	Pet array
400	Invalid tag value	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- application/xml
- application/x-smile
- application/json

3.1.4. Finds Pets by status

```
GET /pets/findByStatus
```

Description

Multiple status values can be provided with comma seperated strings

Parameters

Type	Name	Description	Required	Schema	Default
QueryParamet er	status	Status values that need to be considered for filter	true	enum (available, pending, sold)	available

Responses

HTTP Code	Description	Schema
200	OK	Pet array
400	Invalid status value	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- application/xml
- application/x-smile
- application/json

3.1.5. 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	Required	Schema	Default
PathParameter	petId	ID of pet that needs to be fetched	true	string	

Responses

HTTP Code	Description	Schema
200	OK	Pet
400	Invalid ID supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Pet not found	No Content

Produces

- application/xml
- application/x-smile
- application/json

3.2. Stores

Operations about store

3.2.1. Place an order for a pet

```
POST /stores/order
```

Parameters

Type	Name	Description	Required	Schema	Default
QueryParam	complete		false	boolean	

Type	Name	Description	Required	Schema	Default
QueryParamter	id		false	integer (int64)	
QueryParamter	identifier		false	integer (int64)	
QueryParamter	petId		false	integer (int64)	
QueryParamter	quantity		false	integer (int32)	
QueryParamter	shipDate		false	string (date-time)	
QueryParamter	status	Order Status	false	enum (placed, approved, delivered)	

Responses

HTTP Code	Description	Schema
200	OK	Order
201	Created	No Content
400	Invalid Order	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

3.2.2. 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	Required	Schema	Default
PathParameter	orderId	ID of pet that needs to be fetched	true	string	

Responses

HTTP Code	Description	Schema
200	OK	Order
400	Invalid ID supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Order not found	No Content

Produces

- application/json

3.2.3. 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	Required	Schema	Default
PathParameter	orderId	ID of the order that needs to be deleted	true	string	

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
400	Invalid ID supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Order not found	No Content

Produces

- application/json

3.3. Users

Operations about user

3.3.1. Creates list of users with given input array

POST /users/createWithArray

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	users	List of user object	true	User array	

Responses

HTTP Code	Description	Schema
200	OK	User
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

3.3.2. Creates list of users with given input array

POST /users/createWithList

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	users	List of user object	true	User array	

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

3.3.3. Logs out current logged in user session

GET /users/logout

Responses

HTTP Code	Description	Schema
200	OK	string
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- application/json

3.3.4. Logs user into the system

GET /users/login

Parameters

Type	Name	Description	Required	Schema	Default
QueryParamet er	password	The password for login in clear text	true	string	

Type	Name	Description	Required	Schema	Default
QueryParamter	username	The user name for login	true	string	

Responses

HTTP Code	Description	Schema
200	OK	string
400	Invalid username/password supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- application/json

3.3.5. Updated user

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

Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Required	Schema	Default
QueryParamter	email		false	string	
QueryParamter	firstName		false	string	
QueryParamter	id		false	integer (int64)	
QueryParamter	identifier		false	string	
QueryParamter	lastName		false	string	
QueryParamter	password		false	string	
QueryParamter	phone		false	string	

Type	Name	Description	Required	Schema	Default
QueryParamter	userStatus	User Status	false	integer (int32)	
PathParameter	username	name that need to be deleted	true	string	
QueryParamter	username		false	string	

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
400	Invalid user supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	User not found	No Content

Consumes

- application/json

Produces

- application/json

3.3.6. Get user by user name

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

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	username	The name that needs to be fetched. Use user1 for testing.	true	string	

Responses

HTTP Code	Description	Schema
200	OK	User

HTTP Code	Description	Schema
400	Invalid username supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	User not found	No Content

Produces

- application/json

3.3.7. Delete user

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

Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	username	The name that needs to be deleted	true	string	

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
400	Invalid username supplied	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	User not found	No Content

Produces

- application/json

3.3.8. Create user

```
POST /users
```


Description

This can only be done by the logged in user.

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	user	Created user object	true	User	

Responses

HTTP Code	Description	Schema
200	OK	User
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json