

[TEST DRAFT] PUBLISH

[TEST DRAFT] PUBLISH

CONTENTS

1. adsadsad	3
2. SWAGGER	4

1. adsadsad

sadasdasdsadsa

2. SWAGGER

1.0.7 OAS 2.0

Swagger Petstore

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.

- Terms of service
- Contact the developer
- Apache 2.0
- Find out more about Swagger

Schemes

HTTPS

Authorize

pet Everything about your Pets

Find out more

POST

/pet/{petId}/uploadImage uploads an image

Parameters

Try it out

Name	Description
petId required integer(\$int64) <i>(path)</i>	ID of pet to update petId
additionalMetadata string <i>(formData)</i>	Additional data to pass to server additionalMetadata
file file <i>(formData)</i>	file to upload Choose File No file chosen

Responses

Response content type application/json

Code	Description
200	successful operation Example Value Model <pre>{ "code": 0, "type": "string", "message": "string" }</pre>

POST

/pet Add a new pet to the store



Parameters

Try it out

Name	Description
body ★ required object <i>(body)</i>	Pet object that needs to be added to the store Example Value Model <pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "doggie", "photoUrls": ["string"], "tags": [{ "id": 0, "name": "string" }], "status": "available" }</pre>

Parameter content type

application/json

Responses

Response content type application/json

Code	Description
405	Invalid input

PUT

/pet Update an existing pet



Parameters

Try it out

Name	Description
body ★ required object <i>(body)</i>	Pet object that needs to be added to the store Example Value Model <pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "doggie",</pre>

Name

Description

```
"photoUrls": [
  "string"
],
"tags": [
  {
    "id": 0,
    "name": "string"
  }
],
"status": "available"
}
```

Parameter content type

application/json

Responses

Response content typeapplication/json

Code

Description

400

Invalid ID supplied

404

Pet not found

405

Validation exception

GET

/pet/findByStatus

Finds Pets by status

Multiple status values can be provided with comma separated strings

Parameters

Try it out

Name

Description

status ★ required

Status values that need to be considered for filter

array<string>

Available values : available, pending, sold

(query)

available

pending

sold

Responses

Response content typeapplication/json

Code

Description

200

successful operation

Example Value

Model

Code	Description
400	Invalid status value

GET

/pet/~~findByTags~~ Finds Pets by tags



Warning: Deprecated

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

Parameters

Try it out

Name	Description
tags ★ required array<string> <i>(query)</i>	Tags to filter by

Responses

Response content type

application/json

Code	Description
200	successful operation

Example Value

Model

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

Code	Description
400	Invalid tag value

GET

/pet/{petId} Find pet by ID

^

Returns a single pet

Parameters

Try it out

Name	Description
petId * required integer(\$int64) (path)	ID of pet to return

petId

Responses

Response content type application/json

Code	Description
200	successful operation

Example Value | Model

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

400	Invalid ID supplied
-----	---------------------

Code	Description
404	Pet not found

POST

/pet/{petId}

Updates a pet in the store with form data



Parameters

Try it out

Name	Description
petId * required integer(\$int64) (path)	ID of pet that needs to be updated
name string (formData)	Updated name of the pet
status string (formData)	Updated status of the pet

Responses

Response content typeapplication/json

Code	Description
405	Invalid input

DELETE

/pet/{petId}

Deletes a pet



Parameters

Try it out

Name	Description
api_key string (header)	
petId * required integer(\$int64) (path)	Pet id to delete

Responses

Response content typeapplication/json

Code	Description
400	Invalid ID supplied

Code	Description
404	Pet not found

store Access to Petstore orders



GET

/store/inventory Returns pet inventories by status

^

Returns a map of status codes to quantities

Parameters

Try it out

No parameters

Responses

Response content type application/json

Code	Description
200	successful operation

Example Value

Model

```
{
  "additionalProp1": 0,
  "additionalProp2": 0,
  "additionalProp3": 0
}
```

POST

/store/order Place an order for a pet

^

Parameters

Try it out

Name	Description
body ★ required object (body)	order placed for purchasing the pet

Example Value

Model

```
{
  "id": 0,
  "petId": 0,
  "quantity": 0,
  "shipDate": "2026-09-04T10:03:05.205Z",
  "status": "placed",
  "complete": true
}
```

Parameter content type
application/json

Responses		Response content type
		application/json
Code	Description	
200	successful operation	
		Example Value Model
		<pre>{ "id": 0, "petId": 0, "quantity": 0, "shipDate": "2026-09-04T10:03:14.060Z", "status": "placed", "complete": true }</pre>
400	Invalid Order	

GET	/store/order/{orderId}	Find purchase order by ID	
For valid response try integer IDs with value >= 1 and <= 10. Other values will generated exceptions			
Parameters			Try it out
Name	Description		
orderId * required	ID of pet that needs to be fetched		
integer(\$int64)			
(path)	orderId		
Responses		Response content type	
		application/json	
Code	Description		
200	successful operation		
		Example Value Model	
		<pre>{ "id": 0, "petId": 0, "quantity": 0, "shipDate": "2026-09-04T10:03:14.063Z", "status": "placed", "complete": true }</pre>	
400	Invalid ID supplied		
404	Order not found		

DELETE	/store/order/{orderId}	Delete purchase order by ID	
--------	------------------------	-----------------------------	--

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

Parameters

Try it out

Name	Description
orderId ★ required integer(\$int64) <i>(path)</i>	ID of the order that needs to be deleted

orderId

Responses

Response content type

application/json

Code	Description
400	Invalid ID supplied
404	Order not found

user

Operations about user

Find out more about our store ^

POST

/user/createWithList

Creates list of users with given input array

^

Parameters

Try it out

Name	Description
body ★ required array<object> <i>(body)</i>	List of user object

Example Value

Model

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

Parameter content type

application/json

Responses

Response content type

application/json

Code	Description
default	successful operation

GET

/user/{username}

Get user by user name

^

Parameters

Try it out

Name	Description
username * required string (path)	The name that needs to be fetched. Use user1 for testing. username

Responses

Response content typeapplication/json

Code	Description
200	successful operation Example Value Model <pre>{ "id": 0, "username": "string", "firstName": "string", "lastName": "string", "email": "string", "password": "string", "phone": "string", "userStatus": 0 }</pre>
400	Invalid username supplied
404	User not found

PUT

/user/{username}

Updated user

^

This can only be done by the logged in user.

Parameters

Try it out

Name	Description
username * required string (path)	name that need to be updated username
body * required object	Updated user object

Name

Description

(body)

Example Value

Model

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

Parameter content type

application/json

Responses

Response content type

application/json

Code

Description

400

Invalid user supplied

404

User not found

DELETE

/user/{username}

Delete user

^

This can only be done by the logged in user.

Parameters

Try it out

Name

Description

username * required

The name that needs to be deleted

string

username

(path)

Responses

Response content type

application/json

Code

Description

400

Invalid username supplied

404

User not found

GET

/user/login

Logs user into the system

^

Parameters

Try it out

Name	Description
username * required string (query)	The user name for login
password * required string (query)	The password for login in clear text

Responses

Response content type application/json

Code	Description	
200	successful operation	
Example Value Model "string"		
Headers:		
Name	Description	Type
X-Expires-After	date in UTC when token expires	string
X-Rate-Limit	calls per hour allowed by the user	integer
400	Invalid username/password supplied	

GET

/user/logout

Logs out current logged in user session

^

Parameters

Try it out

No parameters

Responses

Response content type application/json

Code	Description
default	successful operation

POST

/user/createWithArray

Creates list of users with given input array

^

Parameters

Try it out

Name	Description
body ★ required array<object> <i>(body)</i>	List of user object Example Value Model <pre>[{ "id": 0, "username": "string", "firstName": "string", "lastName": "string", "email": "string", "password": "string", "phone": "string", "userStatus": 0 }]</pre> Parameter content type application/json

Responses

Response content type application/json

Code	Description
default	successful operation

POST /user Create user

This can only be done by the logged in user.

Parameters

Try it out

Name	Description
body ★ required object <i>(body)</i>	Created user object Example Value Model <pre>{ "id": 0, "username": "string", "firstName": "string", "lastName": "string", "email": "string", "password": "string", "phone": "string", "userStatus": 0 }</pre> Parameter content type application/json

Responses

Response content type application/json

Code	Description
default	successful operation

Models

ApiResponse {
 code > [...]
 type > [...]
 message > [...]
}

Category {
 id > [...]
 name > [...]
}

```
Pet ▾ {
  id                > [...]
  category          Category > {...}
  name*             > [...]
  photoUrls*        > [...]
  tags              > [...]
  status            > [...]
}
```

```
Tag ▾ {
  id                > [...]
  name              > [...]
}
```

```
Order ▾ {
  id                > [...]
  petId             > [...]
  quantity          > [...]
  shipDate          > [...]
  status            > [...]
  complete          > [...]
}
```

```
User ▾ {
  id                > [...]
  username           > [...]
  firstName          > [...]
  lastName           > [...]
  email              > [...]
  password           > [...]
  phone              > [...]
  userStatus         > [...]
}
```