

UNPUBLISHABLE SPACE

UNPUBLISHABLE SPACE

1. Bulleted & Numbered Lists
2. App Widget
3. Swagger
4. IFrame
5. Links

1. Bulleted & Numbered Lists

Simple Unordered List

- First item
- Second item
- Third item
- Fourth item

Simple Numbered List

1. First numbered item
2. Second numbered item
3. Third numbered item
4. Fourth numbered item

Numbered List with Unordered Sublists

1. First item (no sublist)
2. Second item with unordered sublist
 - Subitem 2.1
 - Subitem 2.2
 - Subitem 2.3
3. Third item with unordered sublist
 - Subitem 3.1
 - Subitem 3.2
 - Subitem 3.3
4. Fourth item (no sublist)

Complex Nested Lists

1. Top level item 1

- Nested unordered item 1.1
- Nested unordered item 1.2
 - 1. Deep nested ordered item 1.2.1
 - 2. Deep nested ordered item 1.2.2
- Nested unordered item 1.3

2. Top level item 2

- Nested unordered item 2.1
- Nested unordered item 2.2

3. Top level item 3

Mixed Content List

1. Item with **bold text**
2. Item with *italic text*
3. Item with `inline code`
4. Item with [a link](#)
 - Subitem with mixed **bold** and *italic*
 - Another subitem with `code`

List After Paragraph

This is a paragraph before the list.

- List item after paragraph
- Another list item
- Third list item

And this is a paragraph after the list.

2. App Widget

Cypress will inject the app widget script in the body of the website.

BUMBLEBEE



3. 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](https://twitter.com/swagger_api). 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

ID of pet to update

integer(\$int64)
(path)

petId

additionalMetadata

Additional data to pass to server

string
(formData)

additionalMetadata

file

file to upload

file

(formData)

Choose File No file chosen

Responses

Response content type

application/json



Code

Description

200

successful operation

Example Value | Model

```
{
  "code": 0,
  "type": "string",
  "message": "string"
}
```

POST

/pet Add a new pet to the store



Parameters

Try it out

Name

Description

body * required

Pet object that needs to be added to the store

object
(body)

Example Value | Model

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

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

Pet object that needs to be added to the store

object
(body)

Example Value | Model

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

Parameter content type

application/json



Responses

Response content type

application/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 array<string> <i>(query)</i>	Status values that need to be considered for filter <i>Available values : available, pending, sold</i>



Responses

Response content type

application/json



Code	Description
200	successful operation

Example Value | Model

```
[
  {
    "id": 0,
    "category": {
      "id": 0,
      "name": "string"
    },
  },
]
```

Code**Description**

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

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 Tags to filter by
array<string>
(query)

Responses

Response content type

application/json

**Code****Description**

200

successful operation

Example Value | Model

```
[
  {
    "id": 0,
```

Code

Description

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

400

Invalid tag value

GET

/pet/{petId} Find pet by ID



Returns a single pet

Parameters

Try it out

Name

Description

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

Responses

Response content type

application/json



Code

Description

200

successful operation

Example Value | Model

```
{
  "id": 0,
```

Code

Description

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

- 400 Invalid ID supplied
- 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 type

application/json

Code	Description
405	Invalid input

DELETE **/pet/{petId}** Deletes a pet



Parameters

Try it out

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

Responses

Response content type

application/json

Code	Description
400	Invalid ID supplied
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

order placed for purchasing the pet

object
(body)

Example Value

Model

```
{
  "id": 0,
  "petId": 0,
  "quantity": 0,
  "shipDate": "2026-06-11T20:35:19.326Z",
  "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": "2026-06-11T20:35:19.373Z", "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 integer(\$int64) (path)	ID of pet that needs to be fetched
orderId	

Responses

Response content type

application/json



Code	Description
200	successful operation
Example Value Model <pre>{ "id": 0, "petId": 0,</pre>	

Code

Description

```
"quantity": 0,  
"shipDate": "2026-06-11T20:35:19.376Z",  
"status": "placed",  
"complete": true  
}
```

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)
(path)

ID of the order that needs to be deleted

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

List of user object

array<object>
(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
------	-------------

default

successful operation

GET

/user/{username} Get user by user name



Parameters

Try it out

Name	Description
------	-------------

username * required

The name that needs to be fetched. Use user1 for testing.

string
(path)

username

Responses

Response content type

application/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 (body)	Updated user object
Example Value Model <pre>{ "id": 0, "username": "string", "firstName": "string", "lastName": "string",</pre>	

Name

Description

```
"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
(path)

username

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

The user name for login

string
(query)

username

password * required

The password for login in clear text

string
(query)

password

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

List of user object

array<object>

(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

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

Created user object

object
(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

default

successful operation

Models



ApiResponse >

Category >

Pet >

Tag >

Order >

User >

4. IFrame

WebGL is required for this map.

We are sorry, but it seems that **your browser does not support WebGL**, a technology for rendering 3D ; web.

[WebGL is required to display this map.](#)

[Read more](#)

5. Links

External Links:

External Link - Redirect to: archbee.com | OPEN IN A NEW TAB - OFF | NO FOLLOW TAG - OFF

External Link - Redirect to: archbee.com | OPEN IN A NEW TAB - ON | NO FOLLOW TAG - OFF

External Link - Redirect to: archbee.com | OPEN IN A NEW TAB - OFF | NO FOLLOW TAG - ON

External Link - Redirect to: archbee.com | OPEN IN A NEW TAB - ON | NO FOLLOW TAG - ON

Internal Links:

Bulleted & Numbered Lists - ANCHOR - OFF | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - OFF

Bulleted & Numbered Lists - ANCHOR - ON | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - OFF

Bulleted & Numbered Lists - ANCHOR - OFF | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - ON

Bulleted & Numbered Lists - ANCHOR - ON | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - ON

CUSTOM TITLE - ON - ANCHOR - OFF | CUSTOM TITLE - ON | OPEN IN A NEW TAB - OFF

CUSTOM TITLE - ANCHOR ON - ANCHOR - ON | CUSTOM TITLE - ON | OPEN IN A NEW TAB - OFF

CUSTOM TITLE - NEW TAB ON - ANCHOR - OFF | CUSTOM TITLE - ON | OPEN IN A NEW TAB - ON

CUSTOM TITLE - ALL ON - ANCHOR - ON | CUSTOM TITLE - ON | OPEN IN A NEW TAB - ON

Cross Space Links:

Crossover Links - ANCHOR - OFF | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - OFF

Crossover Links - ANCHOR - ON | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - OFF

Crossover Links - ANCHOR - OFF | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - ON

Crossover Links - ANCHOR - ON | CUSTOM TITLE - OFF | OPEN IN A NEW TAB - ON

CS CUSTOM TITLE - ON - ANCHOR - OFF | CUSTOM TITLE - ON | OPEN IN A NEW TAB - OFF

CS CUSTOM TITLE - ANCHOR ON - ANCHOR - ON | CUSTOM TITLE - ON | OPEN IN A NEW TAB - OFF

CS CUSTOM TITLE - NEW TAB ON - ANCHOR - OFF | CUSTOM TITLE - ON | OPEN IN A NEW TAB - ON

CS CUSTOM TITLE - ALL ON - ANCHOR - ON | CUSTOM TITLE - ON | OPEN IN A NEW TAB - ON