

PUBLISHABLE SPACE

PUBLISHABLE SPACE

CONTENTS

1. Bulleted & Numbered Lists	3
2. App Widget	5
3. Swagger	8
4. IFrame	24
5. Links	25

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.





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. 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) (path)	ID of pet to update
additionalMetadata string (formData)	Additional data to pass to server
file file (formData)	file to upload

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 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

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) <i>(path)</i>	ID of pet that needs to be updated
name string <i>(formData)</i>	Updated name of the pet
status string <i>(formData)</i>	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 <i>(header)</i>	
petId ★ required integer(\$int64) <i>(path)</i>	Pet id to delete

Responses

Response content type

application/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": "2026-09-28T17:45:29.059Z",
  "status": "placed",
  "complete": true
}
```

Parameter content type
application/json

Responses

Response content typeapplication/json

Code	Description
200	successful operation
Example Value Model <pre>{ "id": 0, "petId": 0, "quantity": 0, "shipDate": "2026-09-28T17:45:37.958Z", "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 typeapplication/json

Code	Description
200	successful operation
Example Value Model <pre>{ "id": 0, "petId": 0, "quantity": 0, "shipDate": "2026-09-28T17:45:37.960Z", "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) (path)	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> (body)	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 type

application/json

Code	Description
200	successful operation
400	Invalid username supplied
404	User not found

Example Value

Model

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

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
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

(body)

Created 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

Models

^

ApiResponse

▼ {

code

▼ integer(\$int32)

type

▼ string

message

▼ string

}

Category

▼ {

id

▼ integer(\$int64)

name

▼ string

}

```
Pet ∨ {
  id
    ∨ integer($int64)
  category
    Category ∨ {
      id
        ∨ integer($int64)
      name
        ∨ string
    }
  name*
    ∨ string
    example: doggie
  photoUrls*
    ∨ [
      xml: { "wrapped": true }
      ∨ string
      xml: { "name": "photoUrl" }
      xml:
        name: photoUrl
    ]
  tags
    ∨ [
      xml: { "wrapped": true }
      Tag ∨ {
        id
          ∨ integer($int64)
        name
          ∨ string
      }]
  status
    ∨ string
    pet status in the store
    Enum:
      ∨ [ available, pending, sold ]
}
```

```
Tag ∨ {
  id
    ∨ integer($int64)
  name
    ∨ string
}
```

```
Order ∨ {
  id
    ∨ integer($int64)
  petId
    ∨ integer($int64)
  quantity
    ∨ integer($int32)
  shipDate
    ∨ string($date-time)
  status
    ∨ string
    Order Status
    Enum:
      ∨ [ placed, approved, delivered ]
  complete
    ∨ boolean
}
```

```
User {
  id              integer($int64)
  username        string
  firstName       string
  lastName        string
  email           string
  password        string
  phone           string
  userStatus      integer($int32)
  User Status
}
```

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 graphics on the 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