Main Custom Domain Space

1. New doc
2. First docc
3. Prima categ1
    3.1. Primul docc
4. Some page
5. Category page
    5.1. sub categ
        5.1.1. sub doc
    5.2. linke
    5.3. Un document
6. am un doccc
7. Sample API
    7.1. Add an item.
8. Swagger Petstore REDOCLY 3.1
    8.1. Introduction
    8.2. pet
        8.2.1. Add a new pet to the store JUST FOR TEST BODY ARRay
        8.2.2. Add a new pet to the store
        8.2.3. Update an existing pet
        8.2.4. OperationId with quotes
        8.2.5. OperationId with backslash
        8.2.6. Find pet by ID
        8.2.7. Updates a pet in the store with form data
        8.2.8. Deletes a pet
        8.2.9. uploads an image
        8.2.10. Finds Pets by status
        8.2.11. Finds Pets by tags
    8.3. store

8.3.1. Returns pet inventories by status
8.3.2. Place an order for a pet
8.3.3. Find purchase order by ID
8.3.4. Delete purchase order by ID
8.3.5. Subscribe to the Store events
8.4. user
8.4.1. Create user XEvi6LSOEr8dueV1Ecvlf
8.4.2. Get user by user name
8.4.3. Updated user
8.4.4. Delete user
8.4.5. Creates list of users with given input array
8.4.6. Creates list of users with given input array
8.4.7. Logs user into the system
8.4.8. Logs out current logged in user session
8.5. Download OpenAPI file
8.6. Models
8.6.1. Model: ApiResponse
8.6.2. Model: Cat
8.6.3. Model: Category
8.6.4. Model: Dog
8.6.5. Model: HoneyBee
8.6.6. Model: Id
8.6.7. Model: Order
8.6.8. Model: Pet

1. New doc

First docc

BrightSign players in large, distributed networks should meet certain provisioning and recovery requirements. Partners who want to set up their own provisioning and recovery systems, rather than using BSN.cloud, should make sure that those systems meet the following requirements.

Each player must be able to provision itself, and once provisioned, periodically check in with the server for updates if configured to do so. The player and server must also be able to initiate a recovery process should the player enter an error state that cannot be fixed by rebooting (for example, filesystem corruption caused by power loss or failure of the storage device).

To meet these requirements, two separate but complementary systems must be developed:

- A set of *autorun.brs* files that can play content, facilitate fault recovery, or provision the player for an HTML/JavaScript playback environment
- A *request handler* server to process recovery/provisioning HTTP requests sent by the player. The request handler delivers an autorun file or other HTTP response depending on player information contained within the header of the request. The request handler database can also contain conditional flags for each player, that enable new autorun scripts to be pushed to specific players in the field. If a partner-specific request handler is not specified in the registry, BSN.cloud provides a default request handler.

Player Provisioning

BrightSign player provisioning integrates the player into your environment, primarily by changing registry settings to reflect that environment and usually by downloading an application or Setup package to the player. The following settings are configured within the registry during BrightAuthor:connected or BSN.cloud provisioning:

- BSN.cloud account, username, password, and group
- Recovery/provisioning URL, error notification URL, crash report URL

- Timezone

Player Registry Entries

The following player registry entries are associated with the provisioning and recovery process. These keys can be modified by accessing the registry's "networking" section using the **roRegistry** BrightScript object.

Registry Key	Description
`ub`	The URL prefix that is applied to the URLs of the ru, eu, and cu key values. This prefix is not applied to a key value containing a colon.
`ru`	The provisioning/recovery URL used by the player to download an autorun script
`eu`	The URL to which error notifications are posted
`cu`	The URL to which crash reports are posted
`p`	The *password* to the recovery handler (`ru`) that is specified above

The following registry keys are specific to BSN.cloud but can be modified by partners to affect the BSN.cloud registration. You should not store your own information in these sections because it can affect BSN.cloud operations:

Registry Key	Description
`a`	The name of the BSN.cloud *account* to which the player belongs.
`u`	The name of the BSN.cloud *user* in the account to which the player belongs
`g`	The BSN.cloud *group* within the account to which the player belongs

> The account, username, and group registry keys are used for provisioning/recovery with BSN.cloud. A request handler may use these keys, but they are not required.

Player Startup Processes

These processes occur when a BrightSign player boots:

1. The player connects to a network and obtains an IP address.
2. The player checks the storage device for an autorun. If an autorun is found, the player will execute the autorun and if none is found, it will proceed with the steps below. See **Player Storage** for more information about storage devices and device priority (if you have multiple storage devices).
3. If an IPv4 address exists, the player will attempt to provision over the network.
4. The player will check for a recovery URL (ru):
    - If a DHCP Option 43 recovery URL (ru) is found, it will override any registry value for `ru` until the end of the DHCP lease. This URL can be set through DHCP Option 43. See **Option 43** to configure a DHCP server to communicate this URL to a BrightSign player.
    - If DHCP Option 43 ru is not found, the ru that is set in the registry will be used.
    - If there is no registry value for the ru, this step will be skipped.
5. If the recovery URL is blank, the player will attempt:
    - **Well-known host provisioning**: Also known as custom name mapping in your network's DNS server. If an IP address is mapped to *brightsign-b-deploy*, then the player will attempt to use the associated IP to provision itself automatically.
    - **B-Deploy cloud provisioning**: See **B-Deploy/Provisioning APIs**.
    - **multicast DNS (mDNS)**: The player will automatically check the standard mDNS location (*http://brightsign-b-deploy.local//)* and if there is a server on your network with that address, the player will automatically provision itself. However, no other provisioning methods can be active if you want to provision with mDNS. See the Appendix at the end of this page for more information about this option.
6. The player will retry the provisioning process every 30 seconds until there is a successful response or interruption.

URL Request Parameters

These parameters are used for communicating between a player and the request handler.

Player Request Parameters

2. First docc

nu ar trebui sa vadddd

Tab Name | Second Tab Name

goodbie

Tab Name | Second Tab Name | New Tab

we got gainz

3. Prima categ1

3.1. Primul docc

4. Some page

a page here ooo

hello

5. Category page

5.1. sub categ

5.1.1. sub doc

fake fake fakeeee

5.2. linke

5.3. Un document

Un mic document

6. am un doccc

salut saluti

e un docki

Head 1	Head2	Head3	Th4
Td 1	Td 2	Td 3	Td 4
Td 5	Td 6	Td 7	Td 8
Td 9	Td 10	Td 11	Td 12
Td 13	Td 14	Td 15	Td 16
Td 17	Td 18	Td 19	Td 20

7. Sample API

7.1. Add an item.

POST https://api.example.com/v1/item Try it

Optional server description, e.g. Main (production) server

Body Parameters

body `item[]`

title `string`

Responses

200 OK

Curl JS Ruby Python

```
curl --location --request POST 'https://api.example.com/v1/item' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200

```
// OK
```

8. Swagger Petstore REDOCLY 3.1

8.1. Introduction

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.

Introduction

This API is documented in **OpenAPI format** and is based on
Petstore sample provided by **swagger.io** team.
It was **extended** to illustrate features of **generator-openapi-repo**
tool and **ReDoc** documentation. In addition to standard
OpenAPI syntax we use a few **vendor extensions**.

OpenAPI Specification

This API is documented in **OpenAPI format** and is based on
Petstore sample provided by **swagger.io** team.
It was **extended** to illustrate features of **generator-openapi-repo**
tool and **ReDoc** documentation. In addition to standard
OpenAPI syntax we use a few **vendor extensions**.

Cross-Origin Resource Sharing

This API features Cross-Origin Resource Sharing (CORS) implemented in compliance with **W3C spec**.
And that allows cross-domain communication from the browser.
All responses have a wildcard same-origin which makes them completely public and accessible to everyone, including any code on any site.

Authentication

Petstore offers two forms of authentication:

- API Key
- OAuth2
  OAuth2 - an open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.

8.2. pet

8.2.1. Add a new pet to the store JUST FOR TEST BODY ARRay

PUT //petstore.swagger.io/v2/example Try it

Default server

Add new pet to the store inventory.

Body Parameters

body object — One Of: Pet

- id Id
- category Category
- name string * — FluffyIsDefault
- photoUrls string,integer,null *
- friend Pet
- tags Tag[]
- status string<available | pending | sold> — Select...
- petType string
- huntingSkill integer

Responses

200 successful operation

ApiResponse

- code integer<int32>
- type string
- message string

Curl | JS | Ruby | Python

```
curl --location --request PUT '//petstore.swagger.io/v2/example' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200

```
// successful operation
{
  "code": "",
  "type": "",
  "message": ""
}
```

8.2.2. Add a new pet to the store

POST //petstore.swagger.io/v2/pet Try it

Default server

Add new pet to the store inventory.

Header Parameters

Accept-Language `string` en-AU

Body Parameters

body `Pet` *

Description

Pet object that needs to be added to the store

category `Category`

name `string` * FluffyIsDefault

photoUrls `string,integer,null` *

friend `Pet`

tags `Tag[]`

status `string<available | pending | sold>` Select...

petType `string`

huntingSkill `integer`

Responses

405 Invalid input

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/pet' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'Accept-Language: en-AU' \
--data '{"body":"Pettie"}'
```

Responses

405

{"user": "SuperAwesomeSprenUser", "readingData": "spren_finger_camera_sdk_reading_data"}

8.2.3. Update an existing pet

PUT //petstore.swagger.io/v2/pet Try it

Default server

Header Parameters

> Accept-Language `string` en-AU

Body Parameters

body `Pet` *

Description

Pet object that needs to be added to the store

> category `Category`

> name `string` * FluffyIsDefault

> photoUrls `string,integer,null` *

> friend `Pet`

> tags `Tag[]`

> status `string<available | pending | sold>` Select...

> petType `string`

huntingSkill `integer`

Responses

- 400 Invalid ID supplied
- 404 Pet not found
- 405 Validation exception

Curl JS Ruby Python

```
curl --location --request PUT '//petstore.swagger.io/v2/pet' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'Accept-Language: en-AU' \
--data '{"body":"Pettie"}'
```

Responses

400 404 405

```
// Invalid ID supplied
```

8.2.4. OperationId with quotes

DELETE //petstore.swagger.io/v2/pet Try it

Default server

Header Parameters

Accept-Language `string` en-AU

Responses

200

Curl JS Ruby Python

```
curl --location --request DELETE '//petstore.swagger.io/v2/pet' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'Accept-Language: en-AU'
```

Responses

200

8.2.5. OperationId with backslash

GET //petstore.swagger.io/v2/pet Try it

Default server

Header Parameters

Accept-Language `string` en-AU

Responses

200

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/pet' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'Accept-Language: en-AU'
```

Responses

200

8.2.6. Find pet by ID

GET //petstore.swagger.io/v2/pet/{petId} Try it

Default server

Returns a single pet

Authentication

> api_key apiKey

Path Parameters

> petId integer<int64> *

Responses

- 200 successful operation
  - Pet
    - id Id
    - category Category
    - name string *
    - photoUrls string,integer,null *
    - friend Pet
    - tags Tag[]
    - status string<available | pending | sol…
    - petType string
    - huntingSkill integer
- 400 Invalid ID supplied
- 404 Pet not found
- 405 Invalid input

Tag

id Id

name string

Curl JS Ruby Python

```
curl --location --globoff '//petstore.swagger.io/v2/pet/{petId}' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200 400 404 405

```
// successful operation
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "",
    "sub": {
      "prop1": ""
    }
  },
  "name": "GuruIsExample",
  "photoUrls": "",
  "friend": {
    "": ""
  },
  "tags": [
    {
      "id": 0,
      "name": ""
    }
  ],
  "status": "",
  "petType": "k9",
  "huntingSkill": 0
}
```

8.2.7. Updates a pet in the store with form data

POST //petstore.swagger.io/v2/pet/{petId} Try it

Default server

Path Parameters

petId `integer<int64>` *

Form Parameters

name `string`

status `string`

Responses

405 Invalid input

Curl JS Ruby Python

```
curl --location --globoff --request POST '//petstore.swagger.io/v2/pet/{petId}' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

405

```
// Invalid input
```

8.2.8. Deletes a pet

DELETE //petstore.swagger.io/v2/pet/{petId} Try it

Default server

Path Parameters

petId integer<int64> *

Header Parameters

api_key string

Responses

400 Invalid pet value

Curl JS Ruby Python

```
curl --location --globoff --request DELETE '//petstore.swagger.io/v2/pet/{petId}' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

400

```
// Invalid pet value
```

8.2.9. uploads an image

POST //petstore.swagger.io/v2/pet/{petId}/uploadImage Try it

Default server

Path Parameters

petId integer<int64> *

Responses

200 successful operation

ApiResponse

code integer<int32>

type string

message string

Curl JS Ruby Python

```
curl --location --globoff --request POST '//petstore.swagger.io/v2/pet/{petId}/uploadIma
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200

```
// successful operation
{
  "code": "",
  "type": "",
  "message": ""
}
```

8.2.10. Finds Pets by status

GET //petstore.swagger.io/v2/pet/findByStatus Try it

Default server

Multiple status values can be provided with comma separated strings

Query Parameters

status `string[]` *

Responses

200 successful operation

`Pet[]`

- id `Id`
- category `Category`
- name `string` *
- photoUrls `string,integer,null` *
- friend `Pet`
- tags `Tag[]`
- status `string<available | pending | sol…`
- petType `string`
- huntingSkill `integer`

400 Invalid status value

Curl | JS | Ruby | Python

```
curl --location '//petstore.swagger.io/v2/pet/findByStatus?status=array' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200 | 400

```
// successful operation
[
  {
    "id": 0,
    "category": {
      "id": 0,
      "name": "",
      "sub": {
        "prop1": ""
      }
    },
    "name": "GuruIsExample",
    "photoUrls": "",
    "friend": {
      "": ""
    },
    "tags": [
      {
        "id": 0,
        "name": ""
      }
    ],
    "status": "",
    "petType": "k9",
    "huntingSkill": 0
  }
]
```

8.2.11. Finds Pets by tags

GET //petstore.swagger.io/v2/pet/findByTags Try it

Default server

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

Query Parameters

tags `string[]` *

+

Responses

200 successful operation

`Pet[]`

- id `Id`
- category `Category`
- name `string` *
- photoUrls `string,integer,null` *
- friend `Pet`
- tags `Tag[]`
- status `string<available | pending | sol…`
- petType `string`
- huntingSkill `integer`

400 Invalid tag value

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/pet/findByTags?tags=array' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200 400

```
// successful operation
[
  {
    "id": 0,
    "category": {
      "id": 0,
      "name": "",
      "sub": {
        "prop1": ""
      }
    },
    "name": "GuruIsExample",
    "photoUrls": "",
    "friend": {
      "": ""
    },
    "tags": [
      {
        "id": 0,
        "name": ""
      }
    ],
    "status": "",
    "petType": "k9",
    "huntingSkill": 0
  }
]
```

8.3. store

8.3.1. Returns pet inventories by status

GET //petstore.swagger.io/v2/store/inventory Try it

Default server

Returns a map of status codes to quantities

Authentication

api_key apiKey

Responses

200 successful operation

```
map<string, integer>
```

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/store/inventory' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200

```
// successful operation
{}
```

8.3.2. Place an order for a pet

POST //petstore.swagger.io/v2/store/order Try it

Default server

Body Parameters

body Order *

Description

order placed for purchasing the pet

quantity integer<int32> 1

shipDate string<date-time> mm/dd/yyyy, --:-- -

status string<placed | approved | delivere… Select...

requestId string

Responses

200 successful operation

Order

id Id

petId Id

quantity integer<int32>

shipDate string<date-time>

status string<placed | approved | deliv…

complete boolean

400 Invalid Order

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/store/order' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{"body":{"quantity":1}}'
```

Responses

200 400

```
// successful operation
{
  "id": 0,
  "petId": 0,
  "quantity": "",
  "shipDate": "",
  "status": "",
  "complete": false
}
```

8.3.3. Find purchase order by ID

GET //petstore.swagger.io/v2/store/order/{orderId} Try it

Default server

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

Path Parameters

orderId integer<int64> *

Responses

200 successful operation

Order

id Id

petId Id

quantity integer<int32>

shipDate string<date-time>

status string<placed | approved | deliv…

complete boolean

400 Invalid ID supplied

404 Order not found

Curl JS Ruby Python

```
curl --location --globoff '//petstore.swagger.io/v2/store/order/{orderId}' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200 400 404

```
// successful operation
{
  "id": 0,
  "petId": 0,
  "quantity": "",
  "shipDate": "",
  "status": "",
  "complete": false
}
```

8.3.4. Delete purchase order by ID

DELETE //petstore.swagger.io/v2/store/order/{orderId} Try it

Default server

For valid response try integer IDs with value < 1000. Anything above 1000 or nonintegers will generate API errors

Path Parameters

> orderId string *

Responses

- 400 Invalid ID supplied
- 404 Order not found

Curl JS Ruby Python

```
curl --location --globoff --request DELETE '//petstore.swagger.io/v2/store/order/{orderI
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

400 404

```
// Invalid ID supplied
```

8.3.5. Subscribe to the Store events

POST //petstore.swagger.io/v2/store/subscribe Try it

Default server

Add subscription for a store events

Body Parameters

body object

callbackUrl string<uri> * Select...

eventName string<orderInProgress | order… * Select...

Responses

200 Successful operation

array[]

201 Subscription added

object

subscriptionId string

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/store/subscribe' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{"body":{"callbackUrl":"string<uri>","eventName":"string<orderInProgress | orders
```

Responses

200 201

```
// Successful operation
[
  [
    ""
  ]
]
```

8.4. user

8.4.1. Create user XEvi6LSOEr8dueV1Ecvlf

POST //petstore.swagger.io/v2/user Try it

Default server

This can only be done by the logged in user.

Body Parameters

body `User` *

Description

Created user object

- **pet** `object` — One Of **Pet**
- **username** `string`
- **firstName** `string`
- **lastName** `string`
- **email** `string<email>` — Select...
- **password** `string<password>` — Select...
- **phone** `string`
- **userStatus** `integer<int32>`
- **image** `string`
- **addresses** `string[]`

+

Responses

default successful operation

Curl | JS | Ruby | Python

```
curl --location '//petstore.swagger.io/v2/user' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{"body":{"pet":{}}}'
```

Responses

default

```
// successful operation
```

8.4.2. Get user by user name

GET //petstore.swagger.io/v2/user/{username} Try it

Default server

Path Parameters

username string *

Responses

200 successful operation

User

id Id

pet object One Of Pet

username string

firstName string

lastName string

email string<email>

password string<password>

phone string

userStatus integer<int32>

image string

addresses string[]

400 Invalid username supplied

404 User not found

Curl JS Ruby Python

```
curl --location --globoff '//petstore.swagger.io/v2/user/{username}' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200 400 404

```
// successful operation
{
  "id": 0,
  "pet": {},
  "username": "John78",
  "firstName": "John",
  "lastName": "Smith",
  "email": "john.smith@example.com",
  "password": "drowssaP123",
  "phone": "+1-202-555-0192",
  "userStatus": "",
  "image": "",
  "addresses": [
    ""
  ]
}
```

8.4.3. Updated user

PUT //petstore.swagger.io/v2/user/{username} Try it

Default server

This can only be done by the logged in user.

Path Parameters

- username string *

Body Parameters

- body User *

Description

Updated user object

- pet object — One Of: Pet
- username string
- firstName string
- lastName string
- email string<email> — Select...
- password string<password> — Select...
- phone string
- userStatus integer<int32>
- image string
- addresses string[]

+

Responses

- 400 Invalid user supplied

● **404** User not found

Curl JS Ruby Python

```
curl --location --globoff --request PUT '//petstore.swagger.io/v2/user/{username}' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{"body":{"pet":{}}}'
```

Responses

● 400 ● 404

```
// Invalid user supplied
```

8.4.4. Delete user

DELETE //petstore.swagger.io/v2/user/{username} Try it

Default server

This can only be done by the logged in user.

Path Parameters

> username string *

Responses

- 400 Invalid username supplied
- 404 User not found

Curl JS Ruby Python

```
curl --location --globoff --request DELETE '//petstore.swagger.io/v2/user/{username}' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

400 404

```
// Invalid username supplied
```

8.4.5. Creates list of users with given input array

POST //petstore.swagger.io/v2/user/createWithArray Try it

Default server

Body Parameters

body `User[]` *

Description

List of user object

- id `Id`
- pet `object` — One Of: Pet
- username `string`
- firstName `string`
- lastName `string`
- email `string<email>` — Select...
- password `string<password>` — Select...
- phone `string`
- userStatus `integer<int32>`
- image `string`
- addresses `string[]`

Responses

default successful operation

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/user/createWithArray' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{"body":[{}]}'
```

Responses

default

```
// successful operation
```

8.4.6. Creates list of users with given input array

POST //petstore.swagger.io/v2/user/createWithList Try it

Default server

Body Parameters

body `User[]` *

Description

List of user object

- id `Id`
- pet `object` — One Of: Pet
- username `string`
- firstName `string`
- lastName `string`
- email `string<email>` — Select...
- password `string<password>` — Select...
- phone `string`
- userStatus `integer<int32>`
- image `string`
- addresses `string[]`

Responses

default successful operation

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/user/createWithList' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data '{"body":[{}]}'
```

Responses

default

```
// successful operation
```

8.4.7. Logs user into the system

GET //petstore.swagger.io/v2/user/login Try it

Default server

Query Parameters

username `string` *

password `string` *

Responses

200 successful operation

`string`

400 Invalid username/password supplied

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/user/login?username=string&password=string' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

200 400

```
// successful operation
""
```

8.4.8. Logs out current logged in user session

GET //petstore.swagger.io/v2/user/logout Try it

Default server

Responses

default successful operation

Curl JS Ruby Python

```
curl --location '//petstore.swagger.io/v2/user/logout' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json'
```

Responses

default

```
// successful operation
```

8.5. Download OpenAPI file

8.6. Models

8.6.1. Model: ApiResponse

Model Parameters

ApiResponse

- code integer<int32>
- type string
- message string

JSON


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

8.6.2. Model: Cat

Model Parameters

Cat

All Of
Pet +

- id Id
- category Category
- name string *
- photoUrls string,integer,null *
- friend Pet
- tags Tag[]
- status string<available | pending | sold>
- petType string
- huntingSkill integer
- huntingSkill string,boolean *

JSON


```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "",
    "sub": {
      "prop1": ""
    }
  },
  "name": "FluffyIsDefault",
  "photoUrls": "",
  "friend": {
    "": ""
  },
  "tags": [
    {
      "id": 0,
      "name": ""
    }
  ],
  "status": "pending",
  "petType": "",
  "huntingSkill": "lazy"
}
```


A representation of a cat

8.6.3. Model: Category

Model Parameters

- Category
  - id Id
  - name string
  - sub object

JSON


```
{
  "id": 0,
  "name": "",
  "sub": {
    "prop1": ""
  }
}
```

8.6.4. Model: Dog

Model Parameters

Dog

All Of
Pet +

- id `Id`
- category `Category`
- name `string` *
- photoUrls `string,integer,null` *
- friend `Pet`
- tags `Tag[]`
- status `string<available | pending | sold>`
- petType `string`
- huntingSkill `integer`
- packSize `integer<int32>` *

JSON


```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "",
    "sub": {
      "prop1": ""
    }
  },
  "name": "FluffyIsDefault",
  "photoUrls": "",
  "friend": {
    "": ""
  },
  "tags": [
    {
      "id": 0,
      "name": ""
    }
  ],
  "status": "pending",
  "petType": "",
  "huntingSkill": 0,
  "packSize": 1
}
```


A representation of a dog

8.6.5. Model: HoneyBee

Model Parameters

HoneyBee

All Of
Pet +

- id `Id`
- category `Category`
- name `string` *
- photoUrls `string,integer,null` *
- friend `Pet`
- tags `Tag[]`
- status `string<available | pending | sold>`
- petType `string`
- huntingSkill `integer`
- honeyPerDay `number` *

JSON


```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "",
    "sub": {
      "prop1": ""
    }
  },
  "name": "FluffyIsDefault",
  "photoUrls": "",
  "friend": {
    "": ""
  },
  "tags": [
    {
      "id": 0,
      "name": ""
    }
  ],
  "status": "pending",
  "petType": "",
  "huntingSkill": 0,
  "honeyPerDay": 0
}
```


A representation of a honey bee

8.6.6. Model: Id

Model Parameters

Id

JSON

{}

8.6.7. Model: Order

Model Parameters

- Order
  - id Id
  - petId Id
  - quantity integer<int32>
  - shipDate string<date-time>
  - status string<placed | approved | deliver…
  - complete boolean
  - requestId string

JSON


```
{
  "id": 0,
  "petId": 0,
  "quantity": 1,
  "shipDate": "",
  "status": "",
  "complete": false,
  "requestId": ""
}
```

8.6.8. Model: Pet

Model Parameters

- Pet
  - id `Id`
  - category `Category`
  - name `string` *
  - photoUrls `string,integer,null` *
  - friend `Pet`
  - tags `Tag[]`
  - status `string<available | pending | sold>`
  - petType `string`
  - huntingSkill `integer`

JSON


```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "",
    "sub": {
      "prop1": ""
    }
  },
  "name": "FluffyIsDefault",
  "photoUrls": "",
  "friend": {
    "": ""
  },
  "tags": [
    {
      "id": 0,
      "name": ""
    }
  ],
  "status": "pending",
  "petType": "",
  "huntingSkill": 0
}
```

8.6.9. Model: Tag

Model Parameters

- Tag
  - id Id
  - name string

JSON


```
{
  "id": 0,
  "name": ""
}
```

8.6.10. Model: User

Model Parameters

User

- id `Id`
- pet `object` — One Of: Pet
- username `string`
- firstName `string`
- lastName `string`
- email `string<email>`
- password `string<password>`
- phone `string`
- userStatus `integer<int32>`
- image `string`
- addresses `string[]`

JSON


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