



Pet Store Management API Guide (Sample)

Author: Pritam Chakrabarti

Contents

An Overview of Pet Store Management API.....	3
API Reference.....	3
Creating a Pet.....	3
URL	3
HTTP Method	3
Header.....	3
Request	3
Example Request (Body)	4
Response	4
Response Code.....	5
Searching Pet by ID	6
URL	6
HTTP Method	6
Header.....	6
Request	6
Response	6
Example Response	7
Response Code.....	7
HTTP Response Codes.....	8
Screenshots of Tests - Creating Pet	9
Screenshots of Test - Searching Pet.....	10

An Overview of Pet Store Management API

Pet Store Management APIs allow you to utilize the Pet Store data of Swagger Sandbox. As a client, you can use the APIs in your application by means of various CRUD methods (POST, GET, PUT, and DELETE). Doing so, you will be able to integrate your application with Swagger Sandbox, and thereby manage the Pet, User, and Store data. The APIs are RESTful and supports both JSON and XML.

API Reference

In this document, we have explained only two methods:

- **POST** method to create a new Pet.
- **GET** method to search a Pet by Id.

Creating a Pet

This section describes about the POST method to create new Pet record. The Request part should contain details of the Pet you want to create. The method creates one record at a time.

Note: To view the screenshots of testing the POST method in Swagger and Postman, click [here](#).

URL

<https://petstore.swagger.io/v2/pet>

HTTP Method

POST

Header

Authentication/Authorization

NoAuth: Authentication and authorization are not required to establish connection to Swagger Sandbox (<https://editor.swagger.io/>)

Content Type

- application/JSON
- application/XML

Request

Parameter	Required	Datatype	Value	Description
id	Yes	Integer		A number that represents unique identifier of the new Pet record.
name	Yes	String		Name of the Pet.
photoURLs	Yes	Array (String)		Uniform Resource Locator (URL) of the Pet's image. Use this array to specify multiple URLs representing different images of the same Pet.
status	No	Array (String)	<ul style="list-style-type: none">• available• pending• sold	Status of a Pet in the pet store. By default, for any new Pet, the "status" is "Available".

category	It is an Object with key-value pairs for two elements, "id" and "name". Each unique combination of "id" and "name", represents a Pet Category, for example, Cat, Dog, Fish, Horse, Bird, etc.			
id		Integer		A number that represents unique identifier of the Pet category.
name		String		Name of a pet category.
tag	Array of Objects. Inside the Array, you can define each Object with key-value pairs for two elements, "id" and "name". Each unique combination of "id" and "name", represents a Tag. It helps in distinguishing a pet from a stray.			
id		Integer		A number that represents unique identifier of the tag linked to the Pet.
name		String		Name to describe a tag. For example, material, color, etc.

Example Request (Body)

```
{
  "id": 75298,
  "category": {
    "id": 752,
    "name": "Horse"
  },
  "name": "Green Arrow",
  "photoUrls": [
    "https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"
  ],
  "tags": [
    {
      "id": 52,
      "name": "GA"
    }
  ],
  "status": "available"
}
```

Response

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

```
    "name": "string"  
  },  
  ],  
  "status": "available"  
}
```

Note for the Reviewer:

In this case, while adding a new Pet in Swagger Sandbox database, the POST method returns the same information that we have sent through the Body of the Request. It is not adding any value to the developer. Instead, the successful execution of the method can be notified in the following format:

Successfully added a Pet, '[\$pet.name\$]' with ID [\$pet.id\$]

As example, "Successfully added the Pet, 'Green Arrow' with ID 75298"

Response Code

200 – OK.

This HTTP response code 200 indicates successful execution of the API. To know more about other HTTP response codes, click [here](#).

Searching Pet by ID

This section describes the method for searching Pet record. In this method, you need to send the Pet Id through the Path Parameter. The method retrieves details of the matching Pet Id.

Note: To view the screenshots of testing the GET method in Swagger and Postman, click [here](#).

URL

`https://petstore.swagger.io/v2/pet/{petId}`

HTTP Method

GET

Header

Authentication/Authorization

NoAuth: Authentication and authorization are not required to establish connection to Swagger Sandbox (<https://editor.swagger.io/>).

Content Type

- application/JSON
- application/XML

Request

Parameter	Required	Datatype	Description
Id	Yes	Integer	A number that represents unique identifier of an existing Pet, you want to search.

Response

Parameter	Datatype	Value	Description
id	Integer		A number that represents unique identifier of the new Pet record.
category	It is an Object with key-value pairs for two elements, "id" and "name". Each unique combination of "id" and "name", represents a Pet Category, for example, Cat, Dog, Fish, Horse, Bird, etc.		
id	Integer		A number that represents unique identifier of the Pet category.
name	String		Name of a pet category.
name	String		Name of the Pet
photoURLs	Array (String)		Uniform Resource Locator (URL) of the Pet's image. Use this array to specify multiple URLs representing different images of the same Pet.
tag	Array of Objects. Inside the Array, you can define each Object with key-value pairs for two elements, "id" and "name". Each unique combination of "id" and "name", represents a Tag. It helps in distinguishing a pet from a stray.		
id	Integer		A number that represents unique identifier of the tag linked to the Pet.
name	String		Name to describe a tag. For example, material, color, etc.

Parameter	Datatype	Value	Description
status	Array (String)	<ul style="list-style-type: none">• available• pending• sold	Status of a Pet in the pet store. By default, for any new Pet, the “status” is “Available”.

Example Response

```
{
  "id": 75298,
  "category": {
    "id": 752,
    "name": "Horse"
  },
  "name": "Green Arrow",
  "photoUrls": [
    "https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"
  ],
  "tags": [
    {
      "id": 52,
      "name": "GA"
    }
  ],
  "status": "available"
}
```

Response Code

200 – OK.

This HTTP response code 200 indicates successful execution of the API. To know more about other HTTP response codes, click [here](#).

HTTP Response Codes

Use the HTTP Response Codes to indicate various status of API response:

- 1xx (100 to 199): If the response is informational
- 2xx (200 to 299): If the response is success
- 3xx (300 to 399): If the response redirects
- 4xx (400 to 499): If the response indicates any Client-side error
- 5xx (500 to 599): If the response indicates any Server-side error

Screenshots of Tests - Creating Pet

Request	Response
<p>POST /pet Use this POST method to create a new Pet</p> <p>This method supports creating only one record at a time</p> <p>Parameters</p> <p>Name Description</p> <p>body ^{required} Pet object that needs to be added to the store</p> <p>Edit Value Model</p> <pre>{ "id": 75298, "category": { "id": 752, "name": "Horse" }, "name": "Green Arrow", "photoUrls": ["https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"], "tags": [{ "id": 52, "name": "GA" }], "status": "available" }</pre> <p>Parameter content type: application/json</p> <p>Execute Clear</p>	<p>Responses</p> <p>Response content type: application/json</p> <p>Curl</p> <pre>curl -X POST \ https://petstore.swagger.io/v2/pet \ -H 'accept: application/json' \ -H 'content-type: application/json' \ -d '{ "id": 75298, "category": { "id": 752, "name": "Horse" }, "name": "Green Arrow", "photoUrls": ["https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"], "tags": [{ "id": 52, "name": "GA" }], "status": "available" }'</pre> <p>Request URL: https://petstore.swagger.io/v2/pet</p> <p>Server response</p> <p>Code Details</p> <p>200</p> <p>Subsequent</p> <pre>{ "id": 75298, "category": { "id": 752, "name": "Horse" }, "name": "Green Arrow", "photoUrls": ["https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"], "tags": [{ "id": 52, "name": "GA" }], "status": "available" }</pre> <p>Response headers</p> <p>content-type: application/json</p>

POST Create a Pet GET Search Pet by ID No Environment

FinalAssessment / Create a Pet

Save

POST https://petstore.swagger.io/v2/pet

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```
1 {
2   "id": 75298,
3   "category": {
4     "id": 752,
5     "name": "Horse"
6   },
7   "name": "Green Arrow",
8   "photoUrls": [
9     "https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"
10  ],
11  "tags": [
12    {
13      "id": 52,
14      "name": "GA"
15    }
16  ],
17  "status": "available"
18 }
```

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 290 ms Size: 527 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 75298,
3   "category": {
4     "id": 752,
5     "name": "Horse"
6   },
7   "name": "Green Arrow",
8   "photoUrls": [
9     "https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"
10  ],
11  "tags": [
12    {
13      "id": 52,
14      "name": "GA"
15    }
16  ],
17  "status": "available"
18 }
```

Screenshots of Test - Searching Pet

The screenshot displays the Swagger UI interface for testing an API endpoint. The top section is divided into 'Request' and 'Response' panels.

Request Panel:

- Method: GET
- URL: /pet/{petId} (Find pet by ID)
- Parameters: petId (required, Integer (int64), path) with value 75298.
- Buttons: Execute, Clear.

Response Panel:

- Response content type: application/json
- Request URL: https://petstore.swagger.io/v2/pet/75298
- Server response: 200
- Response body (JSON):

```
{  "id": 75298,  "category": {    "id": 752,    "name": "Horse"  },  "name": "Green Arrow",  "photoUrls": [    "https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"  ],  "tags": [    {      "id": 52,      "name": "GA"    }  ],  "status": "available"}
```
- Response headers: content-type: application/json

Bottom Section:

- Tab: Search Pet by ID
- Method: GET
- URL: https://petstore.swagger.io/v2/pet/75298
- Buttons: Send, Save, Bulk Edit
- Query Params table:

KEY	VALUE	DESCRIPTION
Key	Value	Description
- Body tab selected, showing the JSON response in Pretty format:

```
1  {
2    "id": 75298,
3    "category": {
4      "id": 752,
5      "name": "Horse"
6    },
7    "name": "Green Arrow",
8    "photoUrls": [
9      "https://live.staticflickr.com/3003/2963453697_c0f73fcd7b_b.jpg"
10   ],
11   "tags": [
12     {
13       "id": 52,
14       "name": "GA"
15     }
16   ],
17   "status": "available"
18 }
```