Overview

Are you looking for a furry family member to complete your family? My Pet is designed to choose your furry friend based on your choice. We have a database of pets and their details stored in our system. It's just one click away from you. All you have to do is, visit our website and search for your forever furry pet.
Pet store owners can use our database for their daily organization or pet-handling tasks. Pet stores can easily integrate their systems through our API endpoints and fulfill the required task without any difficulty.
Moreover, if you wish to give away or sell your pet. This is the place. You can add the details of your pet to our system and let their new owner find them online.

My Pet API Facilities

Our API helps users to search, view, add, delete, and modify details about pets. Also, it enables creating new orders for pets, order searching, deleting, and checking inventory statuses. The API contains functionalities to manage users as well. Pet store owners can integrate the required endpoints from our API with your system and proceed with the functionalities. Our system contains the data collection and the API endpoints to manage those data. You can use your own programming language and technology stack to handle the data.

API Authentication

You will be using API_key authorization mechanism for this API. Please use below key to test the authorization filters.

- api_key = special-key

Base URL

Use this URL before every endpoint. This completes the web request of the API.

- base_url = https://petstore.swagger.io/v2/

API Endpoint Descriptions

This endpoint enables adding a new pet to the store.

1. Add a Pet to the Store

Request Type: POST

Request URL: /pet

Request Content Types

- application/json
- application/xml

Request URL

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

Request Body

Use the below parameters to build the request body for a pet object.

Parameter Name	Data Type	Format	Description		Required/Optional
id	Integer	int64	ID number for the pet	25	Optional
category	array		List of categories for the pet		Optional
category : id	integer	int64	Category id	1	Optional
category : name	string		Category name	Dogs	Optional
name	string		Name of the pet	doggie	Required
photoUrl	array		List of URLs of the photo of the pet		Required
tags	array		Tags for pet		Optional
tag : type	sting	int64	URL of the photo of the pet		Optional
status	string	Enum [available, pending, solid]	pet status in the store		Optional

Request Example

This a Curl request example:

```
curl -X 'POST'   'https://petstore.swagger.io/v2/pet'   -H 'accept: application/json'   -H
'Content-Type: application/json'  -d '{ "id": 2334,   "category": {  "id": 0, "name": "string" },
"name": "doggie", "photoUrls": [ "string" ], "tags": [ { "id": 0, "name": "string" } ], "status":
"available" }'
```

Response

Response code : 200

Error Details

Code	Description
405	Invalid Input

Response Example

```
{
  "id": 9223372036854378000,
  "category": {
            "id": 70,
            "name": "Husky"
},
  "name": "Cookie",
  "photoUrls": [
            "string"
],
  "tags": [
{
            "id": 0,
            "name": "pet"
}
  ],
  "status": "available"
}
```

2. Delete a Pet from the Store

This endpoint enables deleting an existing pet from the store. It will use petId to identify the pet.

Request Type: DELETE

Request URL: /pet/{petId}

Request Parameters

Use the below parameters to send the delete request.

Parameter Name	Data Type	Format	Description	Required/Optional	Category
api_key	string	Header Parameter	Api_key provided by the system	Optional	Header Parameter
petId	integer	Int64	Id number of the pet	Required	Path Parameter

Request Content Types

- application/json
- application/xml

Request URL

https://petstore.swagger.io/v2/pet/456

Request Example

This a Curl request example:

```
curl -X 'DELETE' \
  'https://petstore.swagger.io/v2/pet/456' \
  -H 'accept: application/json'
```

Response

Response code : 200

Response Example

```
{
  "code": 200,
  "type": "unknown",
  "message": "456"
}
```

Error Details

Code	Description
400	Invalid ID supplied
404	Pet not found

3. Fetching a Pet from the Store

This endpoint enables retrieving a single pet from the store. It will use petId to identify the pet.

Request Type: GET

Request URL: /pet/{petId}

Request Parameters

Use the below parameters to send the fetch request.

Parameter Name	Data Type	Format	Required/Optional	Description	Category
petId	integer	Int64	Required	Id number of the pet	Path

Request Content Types

- application/json
- application/xml

Request URL

https://petstore.swagger.io/v2/pet/456

Request Example

This a Curl request example:

```
curl -X 'GET' \
  'https://petstore.swagger.io/v2/pet/456' \
  -H 'accept: application/json'
```

Response

Response code : 200

Response Example

```
{
  "id": 457,
  "category": {
            "id": 70,
            "name": "Husky"
},
  "name": "Cookie",
  "photoUrls": [
            "string"
],
  "tags": [
{
            "id": 0,
            "name": "pet"
}
  ],
  "status": "available"
}
```

Error Details

Code	Description
400	Invalid ID supplied
404	Pet not found

Integrating with the Petstore API and Getting Results

Use this code to integrate the Pet store’s ‘Find by Status’ endpoint with your system. Java language is used to write the program here and this will output a list of petId and name collection.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.*;
import org.json.simple.*;
import org.json.simple.parser.*;


public class PetList {

public static void main(String[] args) throws IOException {

    private static final String GET_URL =
"https://petstore.swagger.io/v2/pet/findByStatus?status=available";
    getPetList();
}

  private static void getPetList() throws IOException {

                URL obj = new URL(GET_URL);

                JSONObject jsonObject = obj.toJSON();

                HttpURLConnection con = (HttpURLConnection) obj.openConnection();
                con.setRequestMethod("GET");
                int responseCode = con.getResponseCode();

                if (responseCode == HttpURLConnection.HTTP_OK) {
                     Integer id = (Integer)jsonObject.get("id");
                     String name = (String)jsonObject.get("name");
                     System.out.println("id: " + id + “ ”);
                     System.out.println("name: " + name);
                } else {
                        System.out.println("GET request not worked");
                }

        }
  }
}
```

Note:
I found last question needs more time, especially to do further research on Google. Besides I was unable to find a proper JVM based system to test. When I tried with online Java compiler, I got several import errors and was unable to address all those within the given time period.