Exercise 1: Making REST Requests

You've been very patient watching videos and taking quizzes. By now, you should have a sense of how REST works. Now, let's get some practice making actual REST requests.

We are going to use a sample API that was created using a technology called Swagger. Swagger is an example of an automated documentation tool. Swagger combines documentation with the ability to try out the API right from within the documentation.

In this example API, you are making requests to modify a database for a pet store. You are going to create, retrieve, update, and delete a pet.

Note: The sample Swagger database can be unreliable. Just be patient and try it a few times.

There's a link to the answers that I came up with at the end of this lecture. But don't look until you've really tried to make it work!

Create a Pet

1. Open a browser window and go to http://petstore.swagger.io/

2. Click on pet to open up the pet operations.
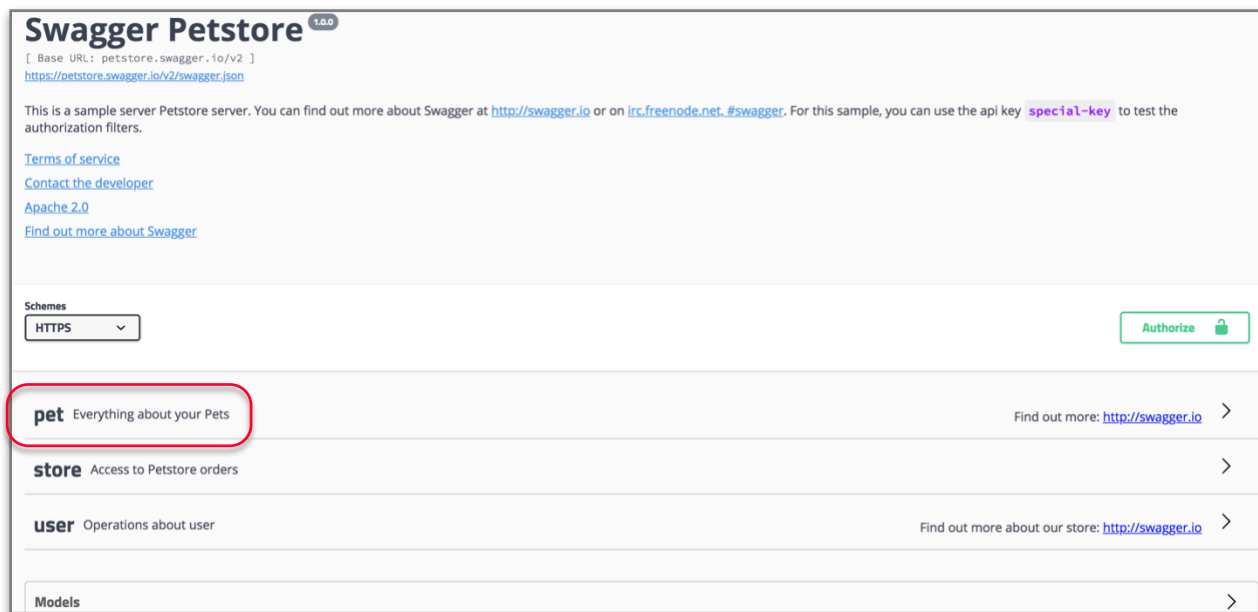
3. Remember, there are four types of operations: POST (create), GET (retrieve), PUT (update), and DELETE (remove). Since we want to create a pet, we want POST. Click on the first POST, which is used to create a new pet.
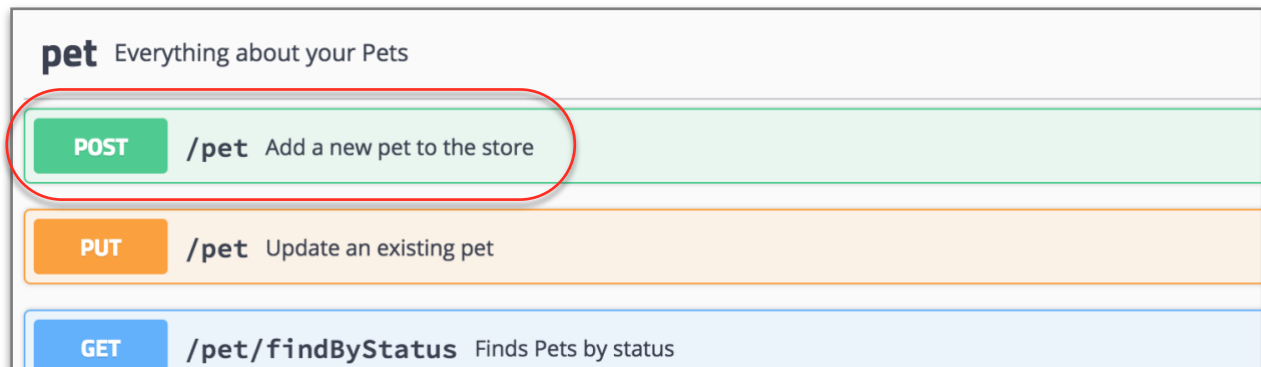
4. We need to pass it information on the new pet we are creating. This is done by setting the body of the POST to be JSON (JavaScript Object Notation). Look under Model | Model Schema, and you will see text that looks like this:

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

This describes the format of the JSON that the API is expecting.

5. Click on **Try it out!** This will give you place where you can edit your data.

6. Click on the text. Delete all fields but the id, name, and status. For the id, choose a 4-digit random number that's likely to be unique. Write down that number, since we will use it later. Give it a unique name and leave the status as "available".

```
{
  "id": 1036,
  "name": "Bela Bardog",
  "status": "available"
}
```

7. Click the big blue **Execute** button.
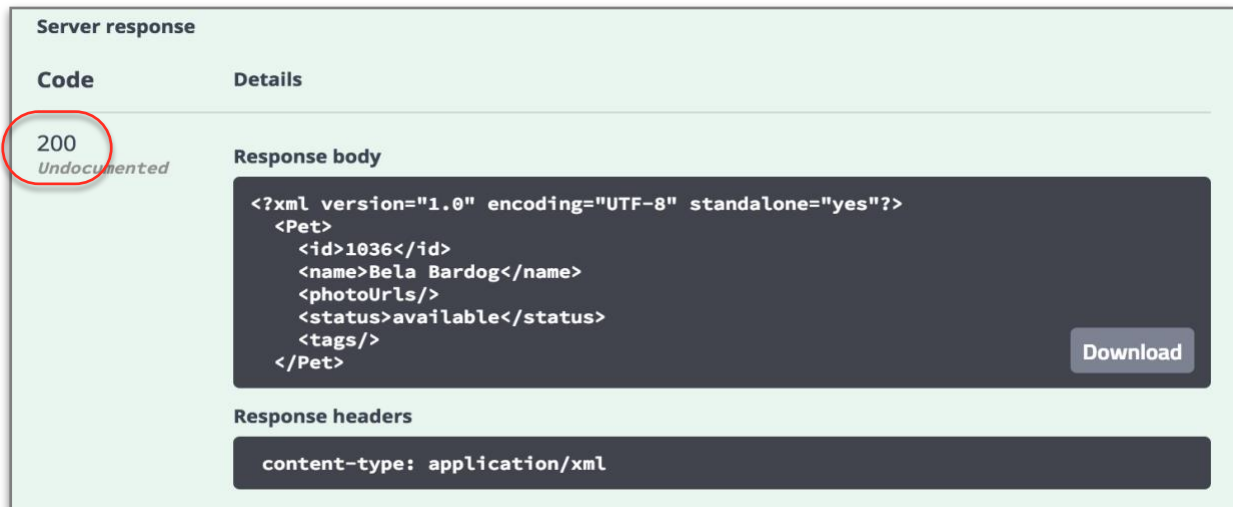
8. Scroll down to see what the pet store server sent back to you. The most important thing to look at is the Server response code. If you see a value of 200, that means that you have successfully added your pet.

Retrieve Your Pet

1. Let's see if your pet is in the database. See if you can figure out which call will return pet information for a pet with a certain ID. Hint: it will be a GET, since GET retrieves information. Click on it.

2. Click on **Try it out.**

3. Set the **Response Content Type** to application/json. This will make it return JSON.

4. There is a place to type the ID. First, type in an ID that people have probably not chosen already. Click **Try it out!** When you scroll down, you'll see that the response code is 404, which means that the pet was not found.

4. Now try it with the ID of the pet you created. You should get a response code of 200, which means success.

5. Look at the Request URL. It will have your ID at the end.

6. Look in the Response Body section. You should see all of the data that you entered when creating the pet.

Update Your Pet

You wouldn't actually use a web page like this to add a pet to the database. This is just a way of trying it out. What you would really do was create an application (a web app, a phone app, etc.) that does it for you. The app makes the HTTP call.

For example, let's say someone is using your app and wants to change the status from "available" to "sold". Figure out which HTTP call you would use to do that. (Hint: it won't say anything directly about availability. Think about the four HTTP operations and which one would be the best one to use.)

Use this HTTP request to change the status to "sold". You'll need to put in all of the information about the pet, not just the status.

Once you have successfully made this request, go back to the GET request and click **Try it out!** again. See if the new status is there.

Remove Your Pet

See if you can figure out how to delete your pet. Like before, in the exercise textbook, paste in the method, the Request URL and the Response Code into the exercise text box.

Make the GET request again. You should see a 404 (not found) error.

Answers

If you want to see the answers I came up with, they are here:

http://sdkbridge.com/ud/Exercise1Answers.pdf