

Test Plan

Product: Pet Store

Testing type: API Testing

API in scope:

1. **Manage Pet :**
 - a. Upload pet image: /pet/{petid}/uploadImage
 - b. Add a new pet record: /pet
 - c. Update a pet record: /pet
 - d. Get pet information by status: /pet/findBy Status
 - e. Get pet information by Id: /pet/{petid}
 - f. Delete pet record: /pet/{petid}

Resources/Tools:

1. Swagger API docker image
2. Npm mocha-chai test framework

Dependencies:

1. Docker
2. Swagger API is up and running

Risk Assesment:

1. Failure of adding pet could halt the business
2. Success of GET pet api is critical for business

Test Deliverable:

1. Test Cases
2. Test Automation Suite
3. Test Execution Report

Test Scenarios:

API: /pet

Method: POST

1. Add a new pet record
 - a. **Test Cases:**
 - i. Verify the 'POST' response payload follows JSON schema for '/pet' route
 - ii. Verify the 'POST' response code 200 for the '/pet' route
 1. Test data = [...]
 - iii. Verify the 'POST' response code 405 for the '/pet' route with 'invalid input'
 1. Test data = [...]

- iv. Verify the 'POST' response data for the '/pet' route with valid values
 - 1. Test data = [...]
- v. Verify the 'POST' response data type for '/pet' route
 - 1. Test data = [...]

API : /pet/{pet_id}

Method: GET

- 1. Find a pet record

a. Test Cases:

- i. Verify the 'GET' response payload follows json schema for '/pet' route
- ii. Verify the 'GET' response code 200 for '/pet/{pet_id}' route
 - 1. Test data = [...]
- iii. Verify the 'GET' response code 400 for '/pet/{pet_id}' for 'invalid id supplied'
 - 1. Test data = [...]
- iv. Verify the 'GET' response code 404 for '/pet/{pet_id}' route for 'pet not found'
 - 1. Test data = [...]
- v. Verify the 'GET' response data for '/pet/{pet_id}' route
 - 1. Test data = [...]
- vi. Verify the 'GET' response type for '/pet/{pet_id}' route
 - 1. Test data = [...]

API : /pet/

Method: PUT

- 2. Edit pet record

a. Test Cases:

- i. Verify the PUT response payload follows JSON schema for the '/pet' route
- ii. Verify the PUT response code 200 for the '/pet' route
 - 1. Test data = [...]
- iii. Verify the PUT response code 400 for the '/pet' route for the 'invalid ID supplied'
 - 1. Test data = [...]
- iv. Verify the PUT response code 404 for the '/pet' route for 'pet not found'
 - 1. Test data = [...]
- v. Verify the PUT response code 405 for the '/pet' route for 'validation exception'
 - 1. Test data = [...]

- vi. Verify the PUT response data with new data for the '/pet' route
 - 1. Test data = [...]
- vii. Verify the PUT response data for the '/pet/' route
 - 1. Test data = [...]
- viii. Verify the PUT response type for the '/pet/' route
 - 1. Test data = [...]