

Під час створення колекції помітив що більшість валідації на тестовій АПІ не працює, ID в суб'єкта який створюється не є унікальним - відповідно оновлення даних через PUT також працює неправильно але я вирішив все одно не прибирати цей метод. Також помітив що під час рану запитів вертаються час від часу різні відповіді - підозрюю що сервер може обробляти з затримкою запити(тому я ставив ділей в налаштуваннях тест рану щоб хоч якось зменшити частоту таких випадків) або це така особливість демо-апі:)

З файлів присутні: сама колекція, енвайронмент зі змінними, та репорт тест-рану.

Summary: TS01 - Create New Pet and retrieval of its data
ID: TS01

Type: Functional Test

Priority: High

Section: Pets Section

Description

Test checks creating of a new pet in the Pet store via API (/pet endpoint) and retrieval of its data after.

Preconditions

Working Test API Collection and test server or environment

Configurations:

API Doc: <https://petstore.swagger.io/#/>

Steps:

1. Select POST Create pet method
2. Input data in body of request and make sure its valid
3. Send request
4. Pay attention to the response status code
5. Select GET Get Pet method
6. Input Pet ID of the created Pet into URL
7. Send request
8. Pay attention to the response status code

Expected Result: Responses Codes are 200/201, New Pet is created correctly and all the sent data is present inside body of GET response.

Execution Status:

Comments/Notes:

Summary: TS02 - Get non-existent Pet

ID: TS02

Type: Functional Test

Priority: Medium

Section: Pets Section

Description

Test checks negative scenario: retrieval of non-existent pet in the Pet store via API (/pet endpoint).

Preconditions

Working Test API Collection and test server or environment

Configurations:

API Doc: <https://petstore.swagger.io/#/>

Steps:

1. Select GET Get pet method
2. Input NON-EXISTENT Pet ID into URL
3. Send request
4. Pay attention to the response status code

Expected Result: Response Code is 404 "Pet not found", Pet with non-existent ID isn't present in the store

Execution Status:

Comments/Notes:

Summary: TS03 - Create Pet with invalid and missing data

ID: TS03

Type: Functional Test

Priority: Medium

Section: Pets Section

Description

Test checks negative scenario: creating pet with invalid and missing data in the Pet store via API (/pet endpoint).

Preconditions

Working Test API Collection and test server or environment

Configurations:

API Doc: <https://petstore.swagger.io/#/>

Steps:

1. Select POST Create Pet method
2. Add invalid data (e.g. wrong data type) to one of the fields(e.g. "name": 123)
3. Delete one of the keys and its value from the body (e.g. "status")
4. Send Request
5. Pay attention to the response status code

Expected Result: Response Code is 400 "Bad Request/ "Invalid input", Pet with invalid and missing data isn't created

Execution Status:

Comments/Notes:

Summary: TS04 - Update Pet data

ID: TS04

Type: Functional Test

Priority: High

Section: Pets Section

Description

Test checks updating pet data in the Pet store via API (/pet endpoint).

Preconditions

Working Test API Collection and test server or environment;
Created Pet, its petID and data

Configurations:

API Doc: <https://petstore.swagger.io/#/>

Steps:

1. Select PUT Update Pet method
2. Insert prepared data and petID inside body
3. Update data in any of the fields (e.g. "name", "category")
4. Send Request
5. Pay attention to the response status code
6. Select GET Get Pet method
7. Input Pet ID of the updated Pet into URL
8. Send request
9. Pay attention to the updated data and response status code

Expected Result: Response Code is 200, Pet data is successfully updated and matches the updated input.

Execution Status:

Comments/Notes:

Summary: TS05 - Update Pet with missing data

ID: TS05

Type: Functional Test

Priority: Medium

Section: Pets Section

Description

Test checks negative scenario: updating pet with missing body data in the Pet store via API (/pet endpoint).

Preconditions

Working Test API Collection and test server or environment;
Created Pet, its petID

Configurations:

API Doc: <https://petstore.swagger.io/#/>

Steps:

1. Select PUT Update Pet method
2. Insert only the petID into the request body, omitting other required fields
3. Send Request
4. Pay attention to the response status code

Expected Result: Response Code is 400 "Bad Request"/"Validation Exception", Pet data isn't updated.

Execution Status:

Comments/Notes:

Summary: TS06 - Create a pet with an already existing petID

ID: TS06

Type: Functional Test

Priority: Medium

Section: Pets Section

Description

Test checks negative scenario: creating pet with existing petID in the Pet store via API (/pet endpoint).

Preconditions

Working Test API Collection and test server or environment;
Created Pet, its petID

Configurations:

API Doc: <https://petstore.swagger.io/#/>

Steps:

1. Select POST Create Pet method
2. Insert prepared petID and required valid data inside body
3. Send Request
4. Pay attention to the response status code

Expected Result: Response Code is 400/409 “Bad Request” or “Conflict”, Pet isn’t created.

Execution Status:

Comments/Notes:

Summary: TS07 - Delete Pet

ID: TS07

Type: Functional Test

Priority: High

Section: Pets Section

Description

Test checks deleting existing pet in the Pet store via API (/pet endpoint).

Preconditions

Working Test API Collection and test server or environment;
Created Pet, its petID

Configurations:

API Doc: <https://petstore.swagger.io/#/>

Steps:

1. Select DELETE Delete Pet method
2. Insert prepared petID into URL
3. Send Request
4. Pay attention to the response status code -> 200 OK
5. Select GET Get Pet method
6. Input Pet ID of the created Pet into URL
7. Send request
8. Pay attention to the response status code

Expected Result: Response Code is 404 "Not Found" , Pet is missing after deleting.

Execution Status:

Comments/Notes: