

QA Engineer Technical Task - Estefani Bouvier

Part 2: API Testing

ID	API_TC01
Title	Add pet - Add successfully a pet with available status
Description	The user should be able to successfully add a pet with available status when all fields are filled out.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre>{ "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 1, "name": "vaccinated" }], "status": "available" }</pre>
Expected result	http code: 200 body: <pre>{ "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "available" }</pre>

	}
ID	API_TC02
Title	Add pet - Empty name error
Description	Adding a new pet should fail when name data is empty.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre> { "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "available" } </pre>
Expected result	http response code is neither 200 nor 201
ID	API_TC03
Title	Add pet - Missing name attribute error
Description	Adding a new pet should fail when the name attribute is missing.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre> { "id": 1, "category": { "id": 1, "name": "DOG" } } </pre>

	<pre> }, "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "available" } </pre>
Expected result	http response code is neither 200 nor 201
ID	API_TC04
Title	Add pet - Null name attribute error
Description	Adding a new pet should fail when name attribute data is null.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	<p>Body:</p> <pre> { "id": 1, "category": { "id": 1, "name": "DOG" }, "name": null, "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "available" } </pre>
Expected result	http response code is neither 200 nor 201
ID	API_TC05

Title	Add pet - Empty photo error
Description	Adding a new pet should fail when photo data is empty.
Endpoint	POST /v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre>{ "id": 1, "category": { "id": 1, "name": "Ana" }, "name": "Ana", "photoUrls": [""], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "available" }</pre>
Expected result	http response code is neither 200 nor 201
ID	API_TC06
Title	Add pet - Missing photo attribute error
Description	Adding a new pet should fail when the photo attribute is missing.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre>{ "id": 1, "category": { "id": 1, "name": "Ana" }, "name": "Ana", "tags": [</pre>

	<pre> { "id": 0, "name": "vaccinated" }], "status": "available" } </pre>
Expected result	http response code is neither 200 nor 201
ID	API_TC07
Title	Add pet - Null photo attribute error
Description	Adding a new pet should fail when the photo attribute is null.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	<p>Body:</p> <pre> { "id": 1, "category": { "id": 1, "name": "Ana" }, "name": "Ana", "photoUrls": null, "tags": [{ "id": 0, "name": "vaccinated" }], "status": "available" } </pre>
Expected result	http response code is neither 200 nor 201
ID	API_TC08
Title	Add pet - Successfully add a pet without optional fields
Description	The user should be able to successfully add a pet when only required fields are provided, without sending optional fields.
Endpoint	POST https://petstore.swagger.io/v2/pet

Precondition	The pet ID does not exist in the system.
Request	Body: <pre>{ "name": "Ana", "photoUrls": ["photourl"] }</pre>
Expected result	http code: 200 body: <pre>{ "name": "Ana", "photoUrls": ["photourl"], "tags": [] }</pre> The response has an ID generated by the backend, it's expected but it can not be determined.
ID	API_TC09
Title	Add pet - Successfully add a pet with null optional fields
Description	The user should be able to successfully add a pet even if optional fields contain null values, as long as all required fields are filled out.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre>{ "id": null, "category": null, "name": "Ana", "photoUrls": ["photourl"], "tags": null, "status": null }</pre>
Expected result	http code: 200 body: <pre>{ "name": "Ana", "photoUrls": ["photourl"] }</pre>

	<pre>] }</pre> <p>The response has an ID generated by the backend, it's expected but it can not be determined.</p>
ID	API_TC10
Title	Add pet - Successfully add a pet with empty optional fields
Description	The user should be able to successfully add a pet even if optional fields are empty, as long as all required fields are filled out.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	<p>Body:</p> <pre>{ "id": 1, "category": {}, "name": "Ana", "photoUrls": ["photourl"], "tags": [], "status": "" }</pre>
Expected result	<p>http code: 200</p> <p>body:</p> <pre>{ "id": 1, "category": { "id": 0 }, "name": "Ana", "photoUrls": ["photourl"], "tags": [], "status": "" }</pre>
ID	API_TC11
Title	Add pet - Pet ID already exists error
Description	Adding a new pet should fail when the pet ID already exists.
Endpoint	POST https://petstore.swagger.io/v2/pet

Precondition	The pet ID 1 exists in the system.
Request	Body: <pre>{ "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "available" }</pre>
Expected result	http response code is neither 200 nor 201
ID	API_TC12
Title	Add pet - Add successfully a pet with pending status
Description	The user should be able to successfully add a pet with pending status when all fields are filled out.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre>{ "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], },</pre>

	<pre>"status": "pending" }</pre>
Expected result	<pre>http code: 200 body: { "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "pending" }</pre>
ID	API_TC13
Title	Add pet - Add successfully a pet with sold status
Description	The user should be able to successfully add a pet with sold status when all fields are filled out.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	<pre>Body: { "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "sold" }</pre>

	}
Expected result	http code: 200 body: <pre> { "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "sold" } </pre>
ID	API_TC14
Title	Add pet - Add a pet with not valid status
Description	Adding a new pet should fail when the status code is not valid.
Endpoint	POST https://petstore.swagger.io/v2/pet
Precondition	The pet ID does not exist in the system.
Request	Body: <pre> { "id": 1, "category": { "id": 1, "name": "DOG" }, "name": "Ana", "photoUrls": ["photourl"], "tags": [{ "id": 0, "name": "vaccinated" }], "status": "not valid status" } </pre>

Expected result	http response code is neither 200 nor 201
ID	API_TC15
Title	Get pet - Get successfully a pet
Description	The user should be able to successfully get a pet with an excited pet id.
Endpoint	GET https://petstore.swagger.io/v2/pet/1
Precondition	The pet ID exists in the system.
Request	-
Expected result	http code: 200 body: <pre> { "id": 1, "category": { "id": 0, "name": "DOG" }, "name": "Ana", "photoUrls": ["string"], "tags": [{ "id": 0, "name": "string" }], "status": "available" } </pre>
ID	API_TC16
Title	Get pet - Not found a pet error
Description	Getting a pet should fail when the id does not exist.
Endpoint	GET https://petstore.swagger.io/v2/pet/100
Precondition	The pet ID does not exist in the system.
Request	-
Expected result	http code: 404 <pre> { "code": 1, </pre>

	<pre>"type": "error", "message": "Pet not found" }</pre>
ID	API_TC17
Title	Get pet - Null parameter error
Description	Getting a pet should fail when the id parameter is null.
Endpoint	GET https://petstore.swagger.io/v2/pet/null
Precondition	-
Request	-
Expected result	<pre>http code: 404 { "code": 404, "type": "unknown", "message": "java.lang.NumberFormatException: For input string: \"null\"" }</pre>

I chose Playwright for this task because it is a versatile tool that supports both UI and API testing. Additionally, it provided a great opportunity to learn a new library.

For API testing, I prioritized automating the most critical and blocker test cases.

For the 'Add a Pet' functionality, I selected *API_TC01 - Successfully add a pet with available status* since it is a crucial flow, as adding pets is a core functionality of a pet store.

Another critical test case is *API_TC08 - Successfully add a pet without optional fields*, ensuring that users can add a pet using only the required data in case optional information is unavailable.

For the 'Get a Pet' functionality, I selected *API_TC15 - Get successfully a pet*, as it is a blocker test case. Accessing pet information is essential, and we must ensure users can access it.

Additionally, I included *API_TC16 - Not found a pet error* to verify that users receive an appropriate error message when searching for a non-existent pet ID.

Finally, I automated *API_TC17 - Null parameter error* to ensure the API does not return an HTTP 200 or 201 response when an invalid request is made.

For the automation, I used the 'toBe()' function to compare the received HTTP status code with the expected one. To validate the JSON response, I used 'toEqual()', and for cases where the ID should be ignored, I used 'toMatchObject()'.

For *API_TC16 - Not found a pet error*, I first performed a delete operation as a precondition to ensure that the pet ID does not exist in the database. After that, I called the GET endpoint to verify the expected behavior.