Test Plan and Bugs Report

Sprint ****

Description

As test Specialist I want to test the api Swagger PetStore.

Scope

For that have the following proposed scenarios:

pet Everything about your Pets

- Verify that when consuming the "GET Find by status" service, it returns all the statuses in Available.
- Validate that when consuming the "GET Find by status" service, it returns all the statuses in Pending.
- Verify that when consuming the "GET Find by status" service, it returns all the statuses in Sold.
- Validate that when consuming the "GET Find by status" service, can not search by unknown status for example bad, Good, etc.
- Verify that when consuming the “POST add a new pet to the Store” service, it return the name that was sent in te request.

Scenario to automate in Gherkin Language:
GetFindByStatus

Available Status:

```
@FindByStatusAvailable
Scenario: validate that the user can search by the available status
  Given the user search with status available
  Then the user validate status 200 and response status Available
```

Pending Status:

```
@FindByStatusPending
Scenario: validate that the user can search by the Pending status
  Given the user search with status Pending
  Then The user Validate Response status pending
```

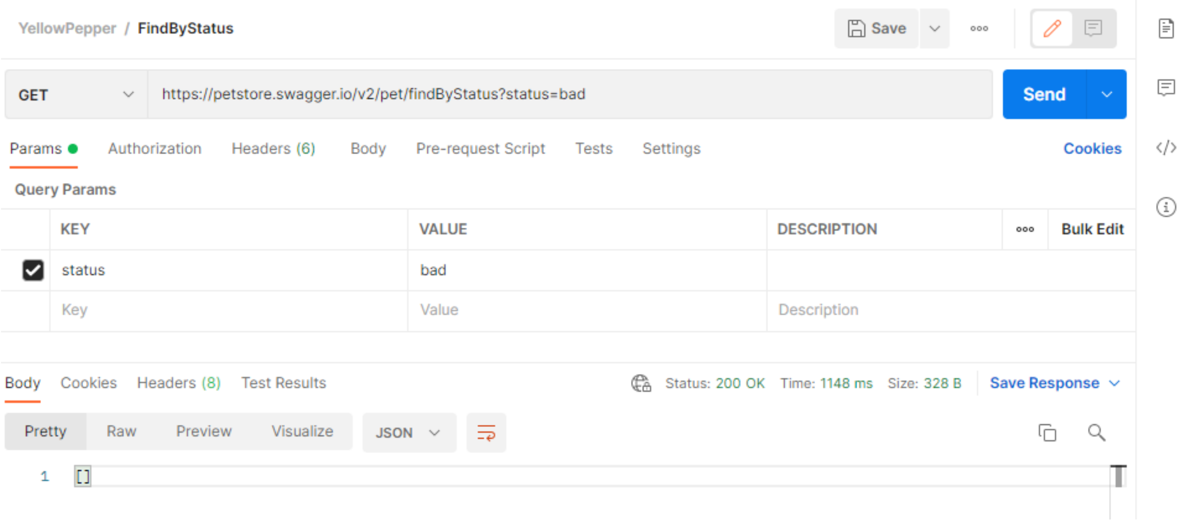
Sold Status:

```
@FindByStatusSold
Scenario: validate that the user can search by the Sold status
  Given the user search with status Sold
  Then The user Validate Response and status
```

Unknown Status:

```
Scenario: validate that the user can not search by the unknown status
  Given the user search with status unknown
  Then The user Validate Response and status 400 Bad request
```

Note: Unknown Status i found a bug because it allows searching for any status. For example Bad Status, as shown in te following image:



YellowPepper / FindByStatus

GET https://petstore.swagger.io/v2/pet/findByStatus?status=bad

Query Params

	KEY	VALUE	DESCRIPTION
☑	status	bad	
	Key	Value	Description

Status: 200 OK Time: 1148 ms Size: 328 B

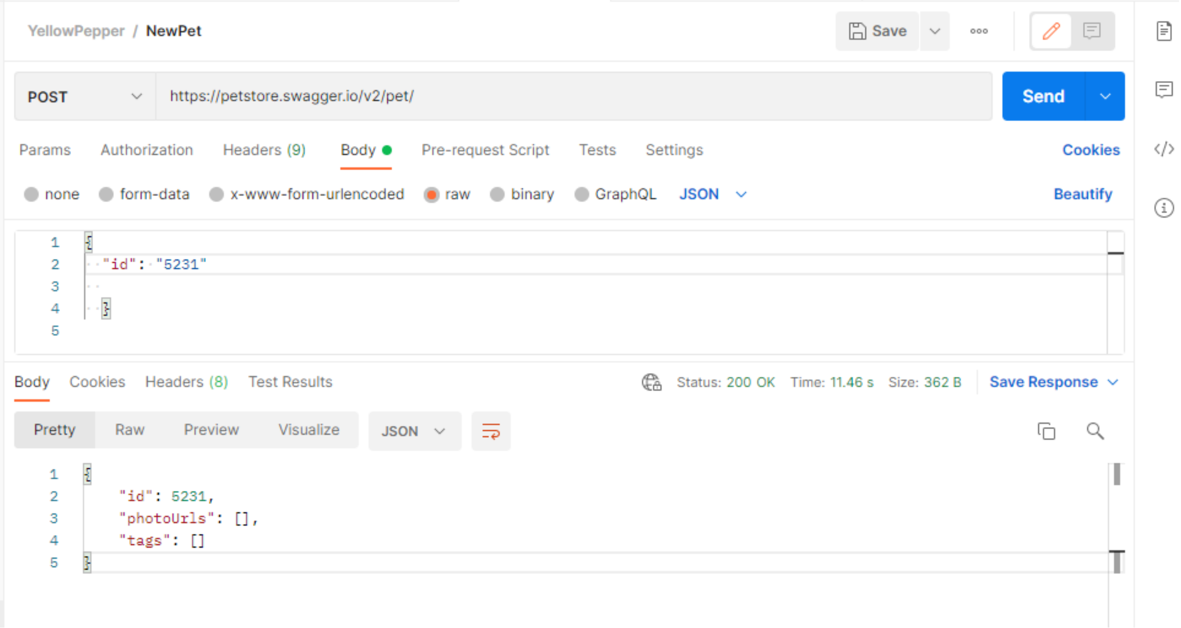
```
1  []
```

PostNewPetToTheStore

New Pet:

```
@NewPet
Scenario: Validate that the new pet is on the store
  Given the user send a information
  Then Validate status and name
```

Note: I think this service could have an improvement, put mandatory fields since everything you send is not validated.



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

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
  "id": "5231"
  
  }
```

Body — Status: 200 OK Time: 11.46 s Size: 362 B

JSON

```
{
    "id": 5231,
    "photoUrls": [],
    "tags": []
}
```

Store Access to Petstore orders

- Validate that when when consuming the “POST Order Store Pet” service, it return status code 200 and id
- Verify when taking the ID that the “POST Order Store Pet” returns, this order is eliminated with “Delete pet Order” service

Scenario to automate in Gherkin Language:

Post Store Order

```
@SendOrder
Scenario: Validate that the user send order with status code and response ID
  Given the user send order to the store
  Then validate status code and id response
```

Delete Store Order

```
@DeleteOrder
Scenario: Validate the correct delete the order
  When  create order and Delete the order
  Then the user status an response ok of delete
```

User Operations about user

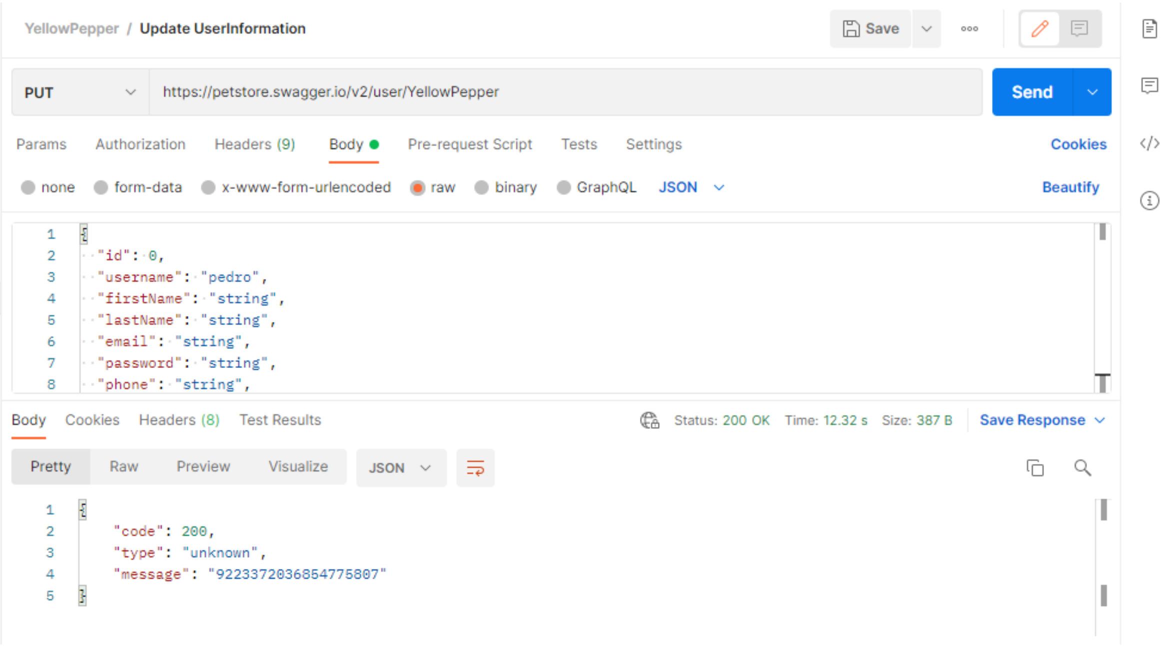
Post UserInformation

```
@SendDataUser
Scenario: Validate the send of data user
  Given Send request with data
  Then Validate status code ok
```

Update UserInformation

```
@UpdateUserInformation
Scenario: Validate the update of user information
  Given Send request with data and send update
  Then Validate satus and response of update
```

Note: The update service UserInformation receives any parameter even if it has not been created with the Post UserInformation service. For example:



PUT https://petstore.swagger.io/v2/user/YellowPepper

Body: ● raw JSON

```
{
  "id": 0,
  "username": "pedro",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "password": "string",
  "phone": "string",
```

Status: 200 OK Time: 12.32 s Size: 387 B

```
{
    "code": 200,
    "type": "unknown",
    "message": "9223372036854775807"
}
```

Strategy

The QA team will make the proposed validations in the business rules.

Automated tests will be run for the scenarios already proposed above and the Git Repository on Resources/features.

To supposes

Updated environment QA with the last versión.

we have all the tools for testing

Work Team

PO: ********

Development: ************

QA: Juan Pablo Navas Arango