API testing with Postman

Assignment

Stefania Pruteanu

- Create a collection of requests for each of the methods of the specified API (using variables and at least 2 tests for each request) — so that the data from the web service response for one request is used in subsequent requests.
  The collection must be launched via the Collection runner with positive passing of all tests.

I'll save some snippets here that I used in Postman

// Initial user - name:Stefania, 27, Constanta

```
pm.collectionVariables.set("NAME-v1", "Stefania");
pm.collectionVariables.set("AGE-v1", 27);
pm.collectionVariables.set("CITY-v1","Constanta");
```

// Saving the created ID to my variable 'stefID'

```
pm.test("User Stefania received an ID.", function () {
// Saving dynamic ID to variable, for next requests
    var responseBody = pm.response.json();
    var id = responseBody.id;
    pm.collectionVariables.set("stefID", id);
    pm.expect(pm.response.text("")).to.include("Stefania", id);
});
```

// Body includes the saved ID, from previous request assigned to name 'Stefania'

```
pm.test("Found user 'Stefania' by ID", function () {
    pm.expect(pm.response.text()).to.include(pm.collectionVariables.get("stefID"),"Stefania");
});
```

```
// Saving new name to variable NAME-v2
var responseBody = pm.response.json();
var name = responseBody.name;
pm.collectionVariables.set("NAME-v2", name);

// Saving new age to variable AGE-v2
var responseBody = pm.response.json();
var age = responseBody.age;
pm.collectionVariables.set("AGE-v2", age);

// Saving new city to variable AGE-v2
var responseBody = pm.response.json();
var city = responseBody.name;
pm.collectionVariables.set("CITY-v2", city);

// Testing the PUT method for response with the new values for each variable above
pm.test("UPDATED - with new information: city, name, age", function () {
    pm.expect(pm.response.text()).to.include(city,name,age);
});

// Checking with GET method, that the user has indeed the new updated values for name,city,age
pm.test("User changed their Name - Age - City, successfully", function () {
    pm.expect(pm.response.text()).to.include(pm.collectionVariables.get("NAME-v2","AGE-v2","CITY-v2"));
});

// Calling DELETE method with the saved ID for same user
pm.test("User has deleted their account", function () {
    pm.expect(pm.response.text()).to.include("User is deleted");
});

// Finally checking that the user no longer exists with that ID
pm.test("User was successfully deleted - not found", function () {
    pm.expect(pm.response.text()).to.include("No users found"); });
```