|  |  |
| --- | --- |
| D:\PAL\Logo_4_21_15\Logo_4_21_15\Primary Logo\png_files\EPAM_LOGO_Full_Color_RGB.png | L&D Automated Testing Mentoring Program |

MODULE 12: Automated testing of Web Services with Java

## home task

if mentee's project has something to do with REST web-services, the tasks that are described below should be done using a mentee's REST web-service. Mentor is allowed to tailor given tasks to mentee’s REST web-service without losing a sense of the given tasks. Otherwise, it should be done using the provided REST web-service.

1. Create a test to verify an http status code:
   * Send the http request by using the GET method.
   * The URL is <https://jsonplaceholder.typicode.com/users>
   * Validation: status code of the obtained response is 200 OK
2. Create a test to verify an http response header:

* Send the http request by using the GET method.
* The URL is <https://jsonplaceholder.typicode.com/users>
* Validation: - the content-type header exists in the obtained response

- the value of the content-type header is application/json; charset=utf-8

1. Create a test to verify an http response body:

* Send the http request by using the GET method:
* The URL is <https://jsonplaceholder.typicode.com/users>
* Validation: the content of the response body is the array of 10 users

## Bonus TAsk

* Study what [swagger UI](https://swagger.io/) is.
* Launch a Bug Tracker Web Service locally by running the following command (mvn spring-boot:run) from the root folder of bug-tracker-service where pom.xml is placed. (bug-tracker-service archive should be unzipped prior to running mvn spring-boot:run)
* Examine Bug Tracker Web Service by using the documentation of Bug Tracker Web Service in shape of swagger UI located [here](http://localhost:8080/swagger-ui.html).
* Implement automated tests for one of the Bug Tracker Web Service resources such as application.
* Tests are created using any http client and serialization/deserialization libraries.

## acceptance criteria

1. Tests should be created using either Rest Assured or Spring Rest Template.
2. Tests have to include validations that are given.
3. Implemented tests should be readable.
4. Tests must be implemented so that they could be launched in parallel.
5. Naming and Code Conventions should be followed.
6. As for tests of the bonus task, they should be created to test CRUD operations of the given resource.