

Problem-3

Task description

- Your team is at the very beginning of the process of creating a new application. At this early stage, a backend application is already running with a few completed endpoints. Your task is to create five basic API tests to cover cases defined in the team's Test Plan. You're going to use the Javascript language and already installed modules: SuperTest used for API calls and assertions and MochaJS test runner.

Task

1. The first test, named API Test returns JSON with a list of users, should check whether, for a GET request sent to the endpoint /users, the application returns correct data. It should return valid JSON with a list of users and their name and id key/value pairs. Make sure that the returned list includes mocked up users: john doe with id 1 and anna boe with id 2.
2. The second test, called API Test endpoint /new returns expected text, should verify that the GET request endpoint /new returns correct JSON with a key named text and "welcome to the new page" value.
3. The third test, API Test endpoint /nonexisting returns 404 status, should check that the backend application properly handles nonexistent endpoints. To verify that, send a GET request to the endpoint named /nonexisting and verify that it returns a 404 status code.
4. The fourth test, API Test root path returns redirects, should verify whether the backend application properly handles redirects on a root / path. Therefore, on sending a GET request to the root path, the returned status code should be 301.
5. The last test, named API Test adding new user to the list, should confirm whether, on sending a POST request to the endpoint /users, the new user is added to the existing users list. The POST request should include the parameter name set to new user's user name. For this test you can use the user name from the NEW_USER_NAME constant, defined at the top of your solution file. On success, the endpoint should respond with a message: <USER_NAME> has been added to the users list, where <USER_NAME> is the sent value.

Disclaimer

- Your solution will be run against multiple correct and incorrect versions of the implemented application. If your tests fail to pass the correct implementation, your task will be evaluated as a 0% score.

Available Packages and libraries

- Mocha version 7.2.0
- SuperTest version 4.0.2