

Goal

The goal of this assignment is to show your testing skills and what is your general approach to quality assurance. We value new ideas so next to the original requirement feel free to improve/add/extend.

We evaluate the assignment depending on your level of seniority regarding coverage, strategy and presentation.

The assignment

A new portal for WAES heroes is under construction, and the initial functionalities (sign up and login) are being developed. For now there is a REST **backend** implementation that deals with:

- creating accounts;
- logging in;
- updating accounts;
- deleting accounts;
- authorization (basic admin rights).

Your assignment is to create a separate scalable project to regression test the **functionality** of the running backend initiative. This must be a coded solution (**no** API testing tools like Postman or Katalon).

Make assumptions in the implementation explicit, choices are good but need to be communicated.

Assignment for Testers



The running application

A running project is provided by cloning the below repository (instructions on how to run it and examples of usage are provided on the README):

```
https://bitbucket.org/waesworks/backend-for-testers.git
```

Once running, the application provides the following:

- 1 http endpoint to retrieve **user information**
GET `<host>/waesheroes/api/v1/users/details?username=<username>`
- 1 http endpoint to retrieve information from **all users** (requires Basic Auth)
GET `<host>/waesheroes/api/v1/users/all`
- 1 http endpoint to **log in** (requires Basic Auth)
GET `<host>/waesheroes/api/v1/users/access`
- 1 http endpoint to **sign-up**
POST `<host>/waesheroes/api/v1/users`
- 1 http endpoint to **update** user information (requires Basic Auth)
PUT `<host>/waesheroes/api/v1/users`
- 1 http endpoint to **delete** user (requires Basic Auth)
DELETE `<host>/waesheroes/api/v1/users`
- A fake runtime database with 3 registered users (for testing).

Example scenario: GET user details with valid username

Given the username 'dev' was registered in the database

When a client app attempts to request user 'dev' details

Then the response should be a JSON object equal to

```
{
  "id": 2,
  "name": "Zuper Dooper Dev",
  "username": "dev",
  "email": "zd.dev@wearewaes.com",
  "superpower": "Debug a repellent factory storage.",
  "dateOfBirth": "1999-10-10",
  "isAdmin": false
}
```

- Validate and comment the efficiency and **functional** test coverage of the unit tests within the provided project (note that we do not want a code coverage report, only a short text file with comments about the unit tests is enough - do not change the original application).
- Your project should run while the provided application is running locally (your test runner should be independent of this implementation, only requiring the server to be running).

Must haves

- Coverage of functional scenarios, including corner cases.
- A report with test results is automatically generated.
- Short analysis (in text) of the unit tests in the running application.
- Documentation in code.
- Clear README containing:
 - How to run the test suite (assume the reader does not know your coding language).
 - How to access the generated report.

Nice to haves

- Flexibility regarding the server that the test project is running against.
- If defects are found, a report with all relevant information.
- Suggestions for improvements.

Please upload the assignment to your personal GitHub/Bitbucket account once finished and send the link to the responsible Tech Sourcer before deadline.