#### BPjs Online IDE Amir Dimri | Guy Shuster | Ofek Ashkenazi

# **Tests Document**

### **Functional Requirements:**

We created a Google Sheets document that contains a mapping of all tests by the type of the test:

- Unit Test
- Integration Test
- End to End (E2E) Tests

For each unit test and integration test, the following data is listed:

Req id Req desc func Test id	Test desc Is done?
------------------------------	--------------------

For each end-2-end test, the following data is listed:

Test Description	Actions	Is done?
------------------	---------	----------

## **Link To The Full Test Tables**

#### **User Interface:**

We tested the UI using Jasmine and Karma.

Using unit tests and integration on the client side tests we tested things like: correlation between buttons and functions they should call, that the display loads correctly when the system is turned on, that the display changes after pressing the relevant button, the link between the components used for the display, etc.

We ran into some issues when testing that the UI is responsive, because the Clarity header wasn't fitting our constraints. All the other components of our system react well to resizing, and we will replace and check the header in the future.

#### **Coverage - Client:**

Statements: 70.09% (402/567)
Branches: 52.46% (96/183)
Functions: 74.4% (154/207)
Lines: 71.74% (391/545)

The code editor component has really low coverage (below 30%) because most of its code belongs to the Ace Editor plugin which we incorporated in our project, and we trust that it works correctly (they have their own tests that prove that).

Besides, half of the breakpoint functionality (next breakpoint and previous breakpoint) and all of it's parts can't be tested in this version because it is not implemented yet.

We plan to update and test this functionality and all other not tested functionalities in the next and last version of the project.

## Coverage - Server:

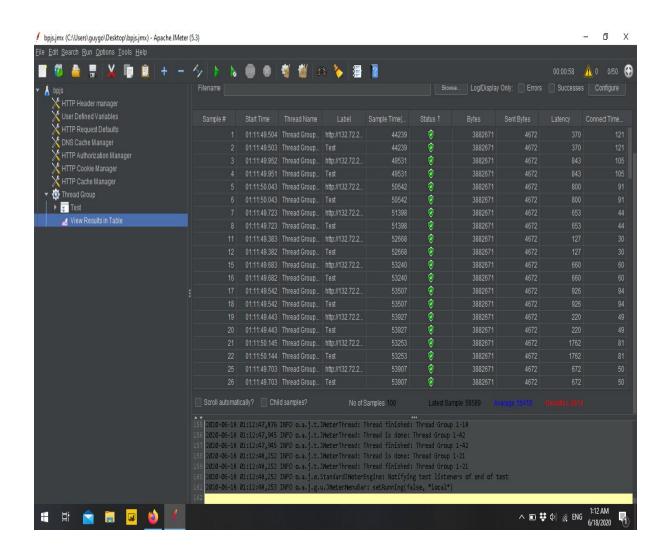
The functionality we added to bpjs itself is minimal, and we tested pretty much all of it, and the coverage was 73%. The coverage of BPjs is 84% (stated in the project's github), and we can safely assume that the coverage of our server is around that percentage as well, since the main functionality is not ours to begin with.

#### **Load Testing:**

In order to test our system in extreme conditions we used apache JMeter. We loaded a recording into the system of a user entering our site and running a basic "Hello World" example which is loaded automatically, which is a pretty realistic scenario for most users that see BPjs for the first time.

Our system currently runs on a BGU server (a linux system with 8GB of RAM), which has an Apache Tomcat server running on it - not a lot of computing power but enough to prove that our project is deployable and can withstand a concurrent number of users.

In the first test where 50 threads were loaded in 1 second, all of them finished successfully in 58 seconds:



In the second test where 100 threads were loaded in 1 second, all of them also finished successfully, but only within 3:24 minutes - a much larger time.

