

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 its 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:

bpjs.jmx (C:\Users\guygo\Desktop\bpjs.jmx) - Apache JMeter (5.3)

File Edit Search Run Options Tools Help

00:00:58 0 0/50

Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

Sample #	Start Time	Thread Name	Label	Sample Time...	Status ↑	Bytes	Sent Bytes	Latency	Connect Time...
1	01:11:49.504	Thread Group...	http://132.72.2...	44239	✓	3882671	4672	370	121
2	01:11:49.503	Thread Group...	Test	44239	✓	3882671	4672	370	121
3	01:11:49.952	Thread Group...	http://132.72.2...	49531	✓	3882671	4672	843	105
4	01:11:49.951	Thread Group...	Test	49531	✓	3882671	4672	843	105
5	01:11:50.043	Thread Group...	http://132.72.2...	50542	✓	3882671	4672	800	91
6	01:11:50.043	Thread Group...	Test	50542	✓	3882671	4672	800	91
7	01:11:49.723	Thread Group...	http://132.72.2...	51398	✓	3882671	4672	653	44
8	01:11:49.723	Thread Group...	Test	51398	✓	3882671	4672	653	44
11	01:11:49.383	Thread Group...	http://132.72.2...	52668	✓	3882671	4672	127	30
12	01:11:49.382	Thread Group...	Test	52668	✓	3882671	4672	127	30
15	01:11:49.683	Thread Group...	http://132.72.2...	53240	✓	3882671	4672	660	60
16	01:11:49.682	Thread Group...	Test	53240	✓	3882671	4672	660	60
17	01:11:49.542	Thread Group...	http://132.72.2...	53507	✓	3882671	4672	926	94
18	01:11:49.542	Thread Group...	Test	53507	✓	3882671	4672	926	94
19	01:11:49.443	Thread Group...	http://132.72.2...	53927	✓	3882671	4672	220	49
20	01:11:49.443	Thread Group...	Test	53927	✓	3882671	4672	220	49
21	01:11:50.145	Thread Group...	http://132.72.2...	53253	✓	3882671	4672	1762	81
22	01:11:50.144	Thread Group...	Test	53253	✓	3882671	4672	1762	81
25	01:11:49.703	Thread Group...	http://132.72.2...	53907	✓	3882671	4672	672	50
26	01:11:49.703	Thread Group...	Test	53907	✓	3882671	4672	672	50

☐ Scroll automatically? ☐ Child samples? No of Samples: 100 Latest Sample: 58589 Average: 33418 Deviation: 2814

155 [2020-06-18 01:12:47,876 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-18
156 [2020-06-18 01:12:47,945 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-42
157 [2020-06-18 01:12:47,945 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-42
158 [2020-06-18 01:12:48,252 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-21
159 [2020-06-18 01:12:48,252 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-21
160 [2020-06-18 01:12:48,252 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test
161 [2020-06-18 01:12:48,253 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)
162

11:12 AM 6/18/2020

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.

bpjs.jmx (C:\Users\guygo\Desktop\bpjs.jmx) - Apache JMeter (5.3)

File Edit Search Run Options Tools Help

00:03:24 0 0/100

Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

bpjs

- HTTP Header manager
- User Defined Variables
- HTTP Request Defaults
- DNS Cache Manager
- HTTP Authorization Manager
- HTTP Cookie Manager
- HTTP Cache Manager
- Thread Group
 - Test
- View Results in Table

Sample #	Start Time	Thread Name	Label	Sample Time...	Status ↑	Bytes	Sent Bytes	Latency	Connect Time...
15	01:03:09.950	Thread Group...	http://132.72.2...	69759	✓	3882671	4672	7659	590
16	01:03:09.949	Thread Group...	Test	69759	✓	3882671	4672	7659	590
31	01:03:09.239	Thread Group...	http://132.72.2...	73903	✓	3882671	4672	922	92
32	01:03:09.238	Thread Group...	Test	73903	✓	3882671	4672	922	92
39	01:03:09.511	Thread Group...	http://132.72.2...	74423	✓	3882671	4672	1931	862
40	01:03:09.510	Thread Group...	Test	74423	✓	3882671	4672	1931	862
43	01:03:09.544	Thread Group...	http://132.72.2...	75153	✓	3882671	4672	1937	844
44	01:03:09.544	Thread Group...	Test	75153	✓	3882671	4672	1937	844
45	01:03:09.811	Thread Group...	http://132.72.2...	75238	✓	3882671	4672	14125	741
46	01:03:09.810	Thread Group...	Test	75238	✓	3882671	4672	14125	741
47	01:03:09.662	Thread Group...	http://132.72.2...	75492	✓	3882671	4672	1887	741
48	01:03:09.662	Thread Group...	Test	75492	✓	3882671	4672	1887	741
51	01:03:09.700	Thread Group...	http://132.72.2...	75750	✓	3882671	4672	1872	719
52	01:03:09.700	Thread Group...	Test	75750	✓	3882671	4672	1872	719
71	01:03:10.030	Thread Group...	http://132.72.2...	76706	✓	3882671	4672	7599	509
72	01:03:10.029	Thread Group...	Test	76706	✓	3882671	4672	7599	509
73	01:03:09.683	Thread Group...	http://132.72.2...	77757	✓	3882671	4672	1907	736
74	01:03:09.682	Thread Group...	Test	77757	✓	3882671	4672	1907	736
77	01:03:09.084	Thread Group...	http://132.72.2...	78850	✓	3882671	4672	120	26
78	01:03:09.083	Thread Group...	Test	78850	✓	3882671	4672	120	26

☐ Scroll automatically? ☐ Child samples? No of Samples 200 Latest Sample 1611 Average 74721 Deviation 18119

```

417 at org.apache.jmeter.threads.JMeterThread.run(JMeterThread.java:256) [ApacheJMeter_core.jar:5.3]
418 at java.lang.Thread.run(Thread.java:834) [?:?]
419 2020-06-18 01:06:32,818 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-96
420 2020-06-18 01:06:32,818 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-96
421 2020-06-18 01:06:32,819 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test
422 2020-06-18 01:06:32,820 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)
423

```

1:09 AM 6/18/2020