Test Cases for Broken Hash Serve Application

# 1.1 When launched, the application should wait for http connections.

Pre-requisites:

* Access to the hashserve application
* Specified PORT set as an environment variable

Steps for Testing:

1. Launch the hashserve application

Expected Results:

* Observe that when launched, the application runs without error and waits for http connections

# 1.2 It should answer on the ​PORT​ specified in the ​PORT​ environment variable.

Pre-requisites:

* The hashserve application is running on the specified port.

Steps for Testing:

1. Run sudo lsof -I -P -n | grep LISTEN
2. This will display the applications and ports

Expected Results:

* Observe that thehashserve app is listening on port 8088

# 1.3 It should support three endpoints:

Pre-requisites:

* The hashserve application is running

Steps for Testing: (note: Authentication Token needed?)

1. Open a terminal window
2. Enter “curl -X POST -H "application/json" -d '{"password":"angrymonkey"}' <http://127.0.0.1:8088/hash>”
3. Enter “curl -X POST -H "application/json" -d '{"password":"random gibberish"}' <http://127.0.0.1:8088/hash>”
4. Enter “curl -H "application/json" ​<http://127.0.0.1:8088/hash/1>”
5. Enter “curl ​http://127.0.0.1:8088/stats”

Expected Results:

* Observe that:
  + After entry for Step 2, the hashserve app will return a SHA512 hash.
  + After entry for Step 3, ??? (this is something I would confirm with dev as the expected result. I would assume an error, but without confirmation I will just note it in the test results)
  + After entry for Step 4, the hashserve app will return base64 encoded password hash.
  + After entry for step 5, the hashserve app will return JSON data for total hash requests and total amount of time.

# 1.4 The software should be able to process multiple connections simultaneously.

Pre-requisites:

* The hashserve application is running.

Steps for Testing:

1. Enter “curl -X POST -H "application/json" -d '{"password":"angrymonkey"}' <http://127.0.0.1:8088/hash>”
2. Immedietly after Step 1, before the process can be completed, Enter “curl ​<http://127.0.0.1:8088/stats>”

Expected Results:

* Observe that after Step 2, both requests to the hashserve app can be run and completed simultaneously without errors.

# 1.5 The software should support a graceful shutdown request, and no additional password requests should be allowed when shutdown is pending.

Pre-requisites:

* The hashserve application is running

Steps for Testing:

1. Open a terminal window
2. Enter “curl -X POST -H "application/json" -d '{"password":"angrymonkey"}' <http://127.0.0.1:8088/hash>”
3. Enter “curl -X POST -d ‘shutdown’ h​ttp://127.0.0.1:8088/hash”
4. Enter “curl -X POST -H "application/json" -d '{"password":"angrymonkey"}' <http://127.0.0.1:8088/hash>” a second time

Expected Results:

* Observe that:
  + After entering Step 3, verify that the process for Step 2 continues uninterrupted or without errors
  + After Step 3, verify that the process for Step 4 is not initiated, and does not disrupt the shutdown process
  + Verify that after Step 3 is completed, a 200 success message is delivered from the app server