

Test Cases

TC 1: Verify that when the application is launched, it waits for http connections on the specified port.

Pre-conditions: A PORT environment variable must be set before executing the application.

```
$ export PORT=8088
```

Test Steps:

1. Launch the application `./broken-hashserve_linux`
2. Verify that it's open to http connections on the port specified in the PORT environment variable

```
$ curl http://127.0.0.1:8088/stats
```

Expected Result: `{"TotalRequests":0,"AverageTime":0}`

Status: PASS

TC 2: Verify a POST to /hash accepts a password and returns a job identifier immediately.

Test Steps:

1.

```
$ curl -X POST -H "application/json" -d '{"password":"angrymonkey"}'
```



```
http://127.0.0.1:8088/hash
```

Expected Result: Returns a job identifier immediately
> 1

Status: FAIL

QA Notes: Immediately is vague. The job identifier is returned in approximately 6 seconds, therefore, I am creating a low priority bug for this issue.

TC 3: Verify a GET to /hash with an invalid job identifier is handled properly.

Test Steps:

1.

```
$ curl -H "application/json" http://127.0.0.1:8088/hash/500
```

Expected Result: Returns hash not found.

Status: PASS

TC 4: Verify a GET to /hash with a valid job identifier returns the base64 encoded password hash for the corresponding POST request.

Test Steps:

1. `$ curl -H "application/json" http://127.0.0.1:8088/hash/1`

Expected Result:

```
> zHkbvZDdwYYiDnwtDdv/FIWvcy1sKCb7qi7Nu8Q8Cd/MqjQeyCI0pWKDGp74A1g==
```

Status: PASS

TC 5: Verify a GET to /stats returns a JSON data structure for the total hash requests since the server started.

Test Steps:

1. `$ curl http://127.0.0.1:8088/stats`

Expected Result:

```
> {"TotalRequests":3,"AverageTime":5004625}
```

Status: Needs PM input

QA Notes: Does count include both GET and POST hash requests? It is only counting the POST requests currently.

TC 6: Verify the average time of a hash request in /stats response is in milliseconds.

Test Steps:

1. `$ curl http://127.0.0.1:8088/stats`

Expected Result:

```
> {"TotalRequests":3,"AverageTime":5004625}
```

Status: FAIL

QA Notes: Calculation seems to be incorrect. My responses averaged 6-7 seconds but I am seeing 67800 in the response which converts to 67.8 seconds.

```
{"TotalRequests":1,"AverageTime":67800}
```

TC 7: Verify the application can process multiple connections simultaneously.

Test Steps:

1. Send multiple requests in parallel where X is a different password
`$ curl -X POST -H "application/json" -d '{"password":"X"}' http://127.0.0.1:8088/hash`

Expected Result:

All requests are processed correctly.

Status: PASS

TC 8: Verify shutdown request empty response and status code.

Test Steps:

1. \$ curl -X POST -d 'shutdown' <http://127.0.0.1:8088/hash>

Expected Result:

> 200 Empty Response

Status: PASS

QA Notes: Did not verify status code

TC 9: Verify shutdown request allows in-flight password hashing jobs to finish.

Test Steps:

1. \$ curl -X POST -H "application/json" -d '{"password":"angrymonkey"}' <http://127.0.0.1:8088/hash>
2. \$ curl -X POST -d 'shutdown' <http://127.0.0.1:8088/hash>

Expected Result:

Verify job finishes and returns a job identifier.

Status: PASS

TC 10: Verify new requests are rejected while the shutdown request is in progress.

Test Steps:

```
$ curl -X POST -d 'shutdown' http://127.0.0.1:8088/hash  
$ curl -X POST -H "application/json" -d '{"password":"angrymonkey"}' http://127.0.0.1:8088/hash
```

Expected Result:

Connection refused?

Status: FAIL and Needs PM input

QA Notes: The requirements don't state what the response should be. Additionally, I had unreliable results with this test case. I received a "Server unavailable" response but was also able to send a new request.

Bugs

Bug 01: A POST to /hash does not return a job identifier immediately.

Priority: Low

Severity: Low Risk

Steps to Reproduce:

1. `$ curl -X POST -H "application/json" -d '{"password":"angrymonkey"}'`
<http://127.0.0.1:8088/hash>
2. Verify that a job identifier is returned immediately.

Expected Results: A job identifier is returned within 1-2 seconds of request.

Actual Result: A job identifier is returned in approximately 6-7 seconds. It's possible that we return the identifier once the hash is computed.

Bug 02: Average time of a hash request in /stats response is incorrect.

Priority: Low

Severity: Low Risk

Steps to Reproduce:

```
$ curl -X POST -H "application/json" -d '{"password":"angrymonkey"}'
```

<http://127.0.0.1:8088/hash>

Take note of time of the request (approx 6 seconds)

```
$ curl http://127.0.0.1:8088/stats
```

Expected Results: "Average Time" in the response is in milliseconds which converts to 6780.

Actual Result: {"TotalRequests":1,"AverageTime":67800}

Bug 03: New requests are allowed while shutdown pending.

Priority: High

Severity: Medium Risk

Steps to Reproduce:

1. `$ curl -X POST -H "application/json" -d '{"password":"angrymonkey1"}'`
<http://127.0.0.1:8088/hash>
2. `$ curl -X POST -d 'shutdown'` <http://127.0.0.1:8088/hash>
3. `$ curl -X POST -H "application/json" -d '{"password":"angrymonkey2"}'`
<http://127.0.0.1:8088/hash>
4. Receive Service Unavailable response

5. `$ curl -X POST -H "application/json" -d '{"password":"angrymonkey2"}'`
<http://127.0.0.1:8088/hash>

Expected Results: The new request should be rejected with a relevant response.

Actual Result: The new request was processed.