Password Hashing Application

Agile Methodology

Grooming:

Password Hashing Application Specification

Details:

- When launched, the application should wait for http connections.
- It should answer on the PORT specified in the PORT environment variable
- It should support three endpoints:
    - A POST to /hash should accept a password. It should return a job identifier immediately. It should then wait 5 seconds and compute the password hash. The hashing algorithm should be SHA512
    - A GET to /hash should accept a job identifier. It should return the base64 encoded password hash for the corresponding POST request.
    - A GET to /stats should accept no data. It should return a JSON data structure for the total hash request since the server started and the average time of a hash request in milliseconds.
- The software should be able to process multiple connections simultaneously.
- The software should support a graceful shutdown request. Meaning, it should only allow any in-flight password hashing to complete, reject any new requests, respond with a 200 and shutdown.
- No additional password requests should be allowed with shutdown is pending.

Questions:

What is min/max length of password

I tested as though min and max were 8-12.

Are special characters allowed

I tested as though special characters are allowed including double byte

Is empty string allowed

I tested as though empty string is not allowed

Are there password minimum requirements

I tested as though there are no minimum requirements

Story Points:

As a team calibrate on how big the work is

Committed into sprint:

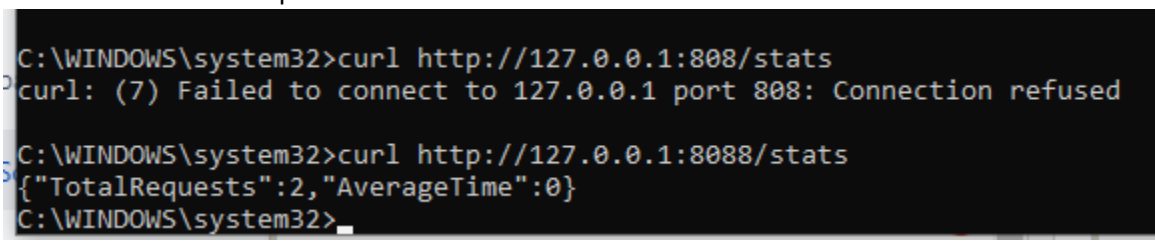
As Developer is writing application – QA writes test cases – and asks developer to approve test cases. – This is so developer knows what QA is looking for, and QA isn't missing anything. – This is also where I would communicate with DEV ask questions

- I wrote test cases to cover details given in the story
- I wrote my test cases to cover good and bad responses
- Added tests to test for length, empty string, special characters, all upper case, and all lower case, double byte characters, only numbers.
- Added test case to test if POST takes more than just "password"
- Added test case to test invalid identifier
- Test graceful shutdown

In Testing:

Execute approved test cases in test cycle

I used JIRA and made a story and linked tested cases using Zephyr Scale and executed.

1. Password Hashing Application – Correct Port
    a. PASS
    b. Validated have to have correct port
        i. 

2. Password Hashing Application – Post to /hash
    a. FAIL
    b. Job identifier is not returned immediately – it takes 5 seconds

i.

```
C:\Users\jbhan\Desktop>curl -X POST -H "application/json" -d "{\"password\":\"angrymonkey\"}" http://127
curl: (6) Could not resolve host: 127

C:\Users\jbhan\Desktop>curl -X POST -H "application/json" -d "{\"password\":\"angrymonkey\"}" -v --trace-time h
.0.0.1:8088/hash
Note: Unnecessary use of -X or --request, POST is already inferred.
13:44:05.375000 *   Trying 127.0.0.1...
13:44:05.390000 * TCP_NODELAY set
13:44:05.390000 * Connected to 127.0.0.1 (127.0.0.1) port 8088 (#0)
13:44:05.390000 > POST /hash HTTP/1.1
13:44:05.390000 > Host: 127.0.0.1:8088
13:44:05.390000 > User-Agent: curl/7.55.1
13:44:05.390000 > Accept: */*
13:44:05.390000 > Content-Length: 26
13:44:05.390000 > Content-Type: application/x-www-form-urlencoded
13:44:05.390000 >
13:44:05.406000 * upload completely sent off: 26 out of 26 bytes
13:44:10.390000 < HTTP/1.1 200 OK
13:44:10.406000 < Date: Thu, 25 Mar 2021 19:44:10 GMT
13:44:10.421000 < Content-Length: 1
13:44:10.437000 < Content-Type: text/plain; charset=utf-8
13:44:10.437000 <
413:44:10.453000 * Connection #0 to host 127.0.0.1 left intact

C:\Users\jbhan\Desktop>
```

3. Password Hashing Application – GET to /hash
    a. PASS
    b. Get command takes ID

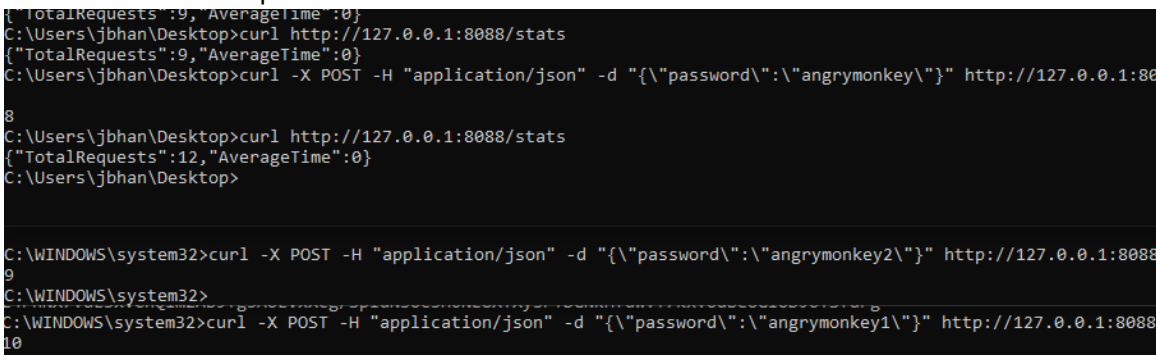
i.

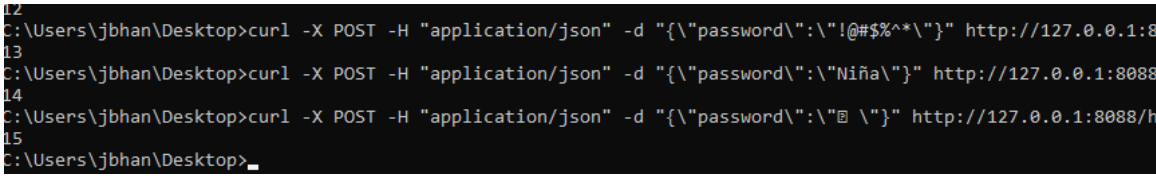
```
C:\Users\jbhan\Desktop>curl -X POST -H "application/json" -d "{\"password\":\"angrymonkey\"}" http://127.0.0.1:

6
C:\Users\jbhan\Desktop>curl -H "application/json" http://127.0.0.1:8088/hash/6
NN0PAKtieayiTY8/Qd53AeMzHkbvZDdwYYiDnwtDdv/FIWvcy1sKCb7qi7Nu8Q8Cd/MqjQeyCI0pWKDGp74A1g==
C:\Users\jbhan\Desktop>
```

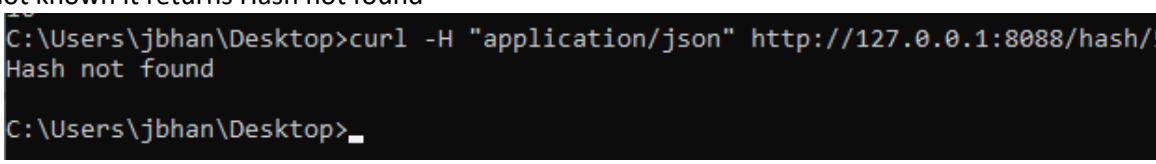
4. Password Hashing Application – GET stats
    a. FAIL
    b. It is not returning average time in milliseconds

i.

```
C:\Users\jbhan\Desktop>curl -v http://127.0.0.1:8088/stats
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8088 (#0)
> GET /stats HTTP/1.1
> Host: 127.0.0.1:8088
> User-Agent: curl/7.55.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Thu, 25 Mar 2021 19:48:15 GMT
< Content-Length: 35
< Content-Type: text/plain; charset=utf-8
<
{"TotalRequests":8,"AverageTime":0}* Connection #0 to host 127.0.0.1 left intact

C:\Users\jbhan\Desktop>curl -v --trace-time http://127.0.0.1:8088/stats
13:48:42.234000 *   Trying 127.0.0.1...
13:48:42.250000 * TCP_NODELAY set
13:48:42.250000 * Connected to 127.0.0.1 (127.0.0.1) port 8088 (#0)
13:48:42.250000 > GET /stats HTTP/1.1
13:48:42.250000 > Host: 127.0.0.1:8088
13:48:42.250000 > User-Agent: curl/7.55.1
13:48:42.250000 > Accept: */*
13:48:42.250000 >
13:48:42.265000 < HTTP/1.1 200 OK
13:48:42.265000 < Date: Thu, 25 Mar 2021 19:48:42 GMT
13:48:42.265000 < Content-Length: 35
13:48:42.265000 < Content-Type: text/plain; charset=utf-8
13:48:42.281000 <
{"TotalRequests":8,"AverageTime":0}13:48:42.281000 * Connection #0 to host 127.0.0.1 left intact

C:\Users\jbhan\Desktop>curl -X POST -H "application/json" -d "{\"password\":\"angrymonkey\"}" http://127.0.0.1:

7
C:\Users\jbhan\Desktop>curl http://127.0.0.1:8088/stats
{"TotalRequests":9,"AverageTime":0}
C:\Users\jbhan\Desktop>
```

5. Password Hashing Application – Multiple Connections
    a. PASS
    b. Able to send POST from multiple command windows
        i. 

6. Password Hashing Application – Post to /hash max length
    a. Marked as pass – this is assuming it was agreed and coded to be set to 15 and correct response was returned
7. Password Hashing Application – Post to /hash min length
    a. Marked as pass – this is assuming it was agreed and coded to be set to 5 and correct response was returned
8. Password Hashing Application – Post to /hash empty string
    a. Marked as pass – this is assuming it was agreed and coded to not accept empty strings and correct response was returned
9. Password Hashing Application – Post to /hash special characters
    a. Marked as pass – this is assuming it was agreed and coded to accept special characters and correct response was returned
        i. 

10. Password Hashing Application – Post to /hash upper characters
    a. Marked as pass – this is assuming it was agreed and coded to allow only upper characters and correct response was returned
11. Password Hashing Application – Post to /hash lower characters
    a. Marked as pass – this is assuming it was agreed and coded to allow only lower characters and correct response was returned
12. Password Hashing Application – Post to /hash numbers only
    a. Marked as pass – this is assuming it was agreed and coded to allow only numbers
13. Password Hashing Application – GET to /hash bad ID
    a. Pass
    b. If ID is not known it returns Hash not found
        i. 

14. Password Hashing Application—Post username to /hash

a. FAIL
b. It accepts more than password
i.

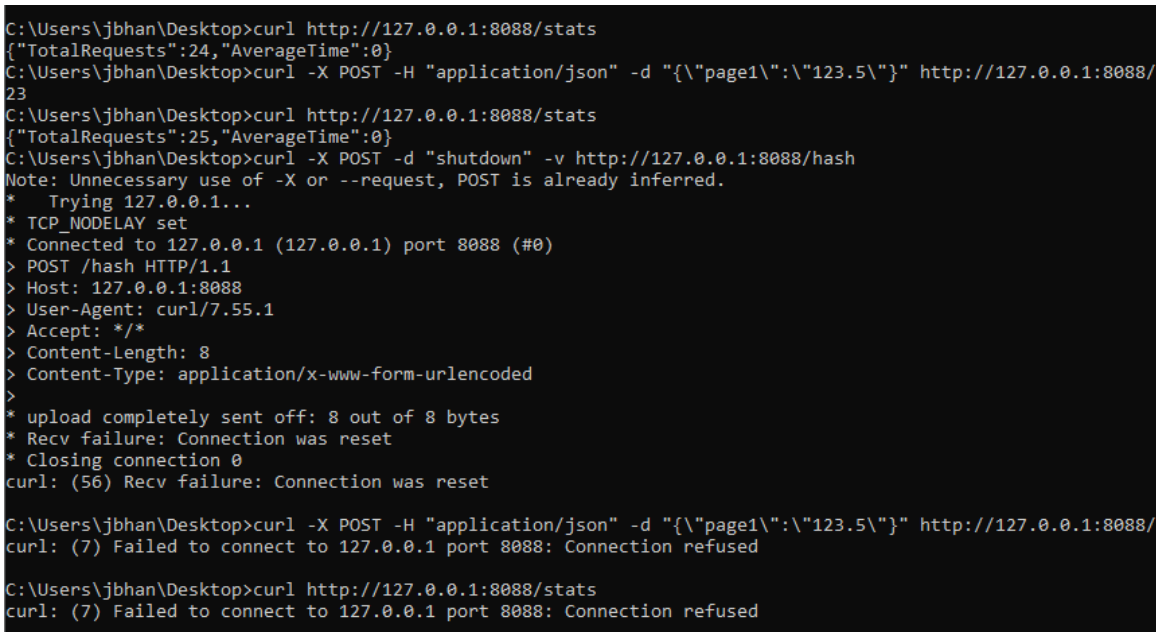
```
C:\Users\jbhan\Desktop>curl -X POST -H "application/json" -d "{\"UserName\":\"123.5\"}" http://127.0.0.1:8
17
C:\Users\jbhan\Desktop>
```

15. Password Hashing Application—GET to /hash/ID before 5 seconds
a. PASS
b. Not able to get hashed pw before 5 seconds
i.

```
C:\Users\jbhan\Desktop>curl -H "application/json" -v --trace-time http://127.0.0.1:8088/hash/22
20:19:16.843000 *   Trying 127.0.0.1...
20:19:16.859000 * TCP_NODELAY set
20:19:16.859000 * Connected to 127.0.0.1 (127.0.0.1) port 8088 (#0)
20:19:16.859000 > GET /hash/22 HTTP/1.1
20:19:16.859000 > Host: 127.0.0.1:8088
20:19:16.859000 > User-Agent: curl/7.55.1
20:19:16.859000 > Accept: */*
20:19:16.859000 >
20:19:16.875000 < HTTP/1.1 200 OK
20:19:16.875000 < Date: Fri, 26 Mar 2021 02:19:16 GMT
20:19:16.875000 < Content-Length: 88
20:19:16.890000 < Content-Type: text/plain; charset=utf-8
20:19:16.890000 <
5ZZ8AW/Zn/Vm9FnSdgrrcEtIOhFf5FwXDjxgRbx+o6OWfyPTWK8v91w1XHWlcohx5e91Q2cH6hnWQDVUCrCm/Q==20:19:16.890000 * Conn
to host 127.0.0.1 left intact

C:\WINDOWS\system32>curl -X POST -H "application/json" -d "{\"password\":\"angrymonkey2\"}" -v --trace-time htt
.0.1:8088/hash
Note: Unnecessary use of -X or --request, POST is already inferred.
20:19:10.656000 *   Trying 127.0.0.1...
20:19:10.656000 * TCP_NODELAY set
20:19:10.671000 * Connected to 127.0.0.1 (127.0.0.1) port 8088 (#0)
20:19:10.671000 > POST /hash HTTP/1.1
20:19:10.671000 > Host: 127.0.0.1:8088
20:19:10.671000 > User-Agent: curl/7.55.1
20:19:10.671000 > Accept: */*
20:19:10.671000 > Content-Length: 27
20:19:10.671000 > Content-Type: application/x-www-form-urlencoded
20:19:10.671000 >
20:19:10.687000 * upload completely sent off: 27 out of 27 bytes
20:19:15.671000 < HTTP/1.1 200 OK
20:19:15.687000 < Date: Fri, 26 Mar 2021 02:19:15 GMT
20:19:15.687000 < Content-Length: 2
20:19:15.703000 < Content-Type: text/plain; charset=utf-8
20:19:15.718000 <
2220:19:15.718000 * Connection #0 to host 127.0.0.1 left intact
```

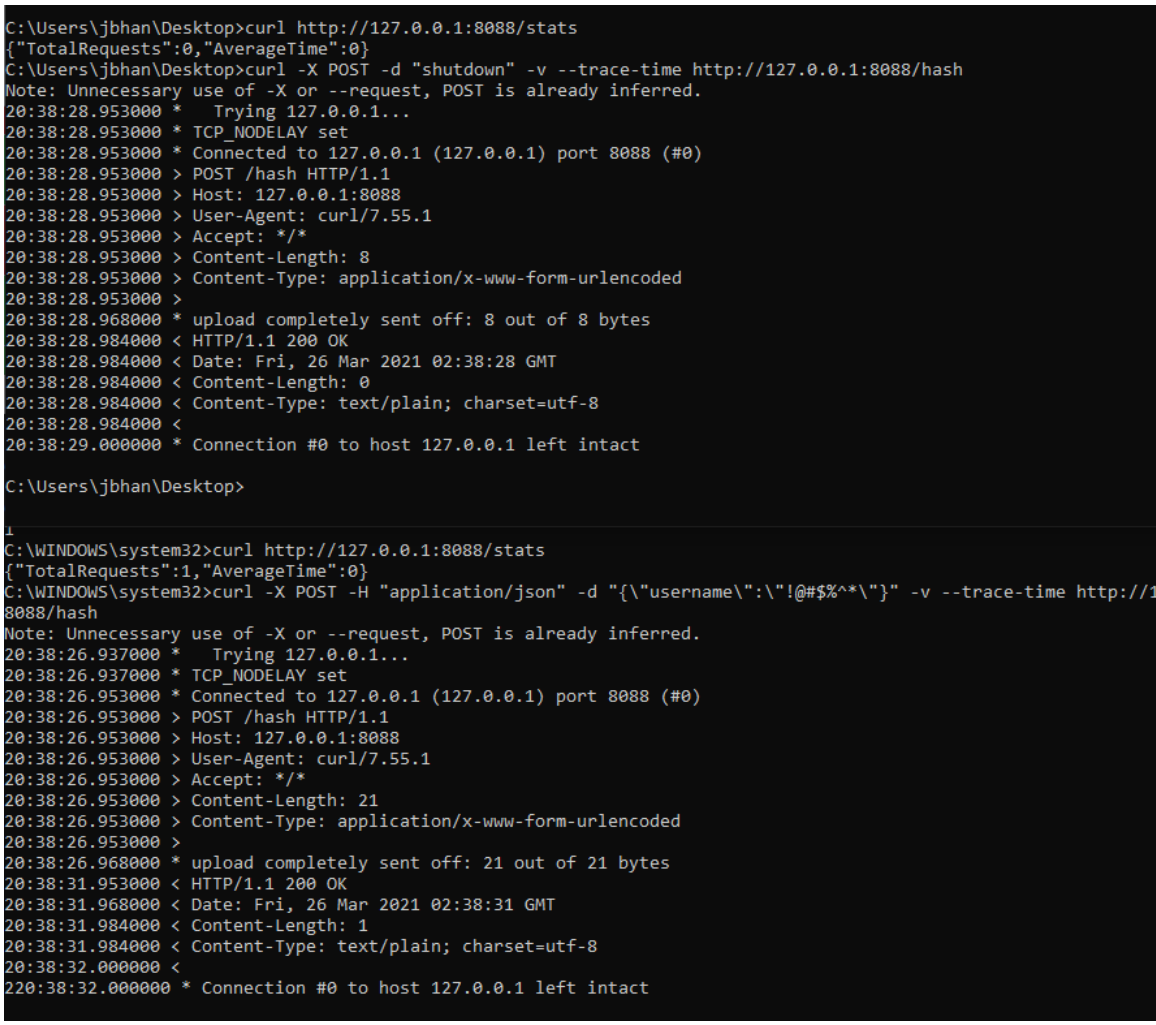
16. Password Hashing Application – Shutdown
a. FAIL
b. It does not send 200 valid response while shutting down

i.



17. Password Hashing Application – Shutdown – finish POST requests
    a. FAIL
    b. Have a active POST request being made – but it still shuts down and doesn't return ID

i.



In Testing:

While executing test cases, if there are failed items I will either make a task with in the story explaining my findings to be fixed immediately (does not meet acceptance criteria) Or if it is out of scope I will make stories in the backlog to be groomed and fixed at a later date. Stories made will have at minimum environment, test steps, credentials (if needed), screen shots, or screen capture video

Send to PO:

Wait for defects to be fixed and sent back to QA. Once all known defects are fixed and re-executed, I would finish up all of my documentation according to definition of done and send story to PO