

# Defects for broken-hashserve V. 0c3d817.1

## Defect 1

### Description:

When shutting down the app, there is no response displayed on the terminal

### Prerequisites:

1. Start the app by typing broken-hashserve in the terminal

### Steps to reproduce:

1. Enter the following curl statement:  
`curl -x POST -d 'shutdown' http://127.0.0.1/8088/hash`

### Expected Behavior:

The terminal should output a message that the webserver has shutdown with a '200'

### Current Behavior:

No message is displayed

### Note:

There is some indication that the server was shutdown, albeit in a different session where the app is initially started.

## Defect 2

### Description:

There is no server-side validation for an empty string in the json payload. Server side validation should be coding standard for production.

### Prerequisites:

1. Start the app by typing broken-hashserve in the terminal

### Steps to reproduce:

1. Enter the following curl statement:  
`curl -X POST -H "application/json" -d '{"password":""}' http://127.0.0.1:8088/hash`

### Expected Behavior

Server should respond with an error message, a 400 status code and should not count towards the total count of requests made to the server.

### Current Behavior

A status code of 200 is returned, as well as, a job identifier

### Example:

```
→ broken-hashserve curl -X POST -H "application/json" -d '{"password":""}' http://127.0.0.1:8088/hash
18
→ broken-hashserve
```

## Defect 3

### Description:

Per the specification, the job identifier does not appear immediately when the curl command has been entered. The command is entered, a 5 second wait takes place and then the job identifier appears on the terminal.

### Prerequisites:

1. Start the app by typing broken-hashserve in the terminal

### Steps to reproduce:

1. Enter the command:  

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

### Current Behavior:

The job identifier appears after a 5 second wait. Documentation states that the job identifier should appear immediately.

### Expected Behavior:

The job identifier should appear immediately.

## Defect 4

### Description:

/stats endpoint does not count requests to the hash endpoint when the job Identifier is appended to the URL

### Prerequisites:

1. Start the app by typing broken-hashserve in the terminal

### Steps to reproduce:

1. Enter this command in the terminal an arbitrary number of times with variations on the password  

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

<http://127.0.0.1:8088/hash>
2. After multiple calls to that endpoint, let's say 5, run this next command to get the number of requests made to the server since uptime  

```
curl "application/json" http://127.0.0.1:8088/stats
```
3. Run this command (it's the same endpoint with the exception of a job identifier appended to the end of the URL)  

```
curl -H "application/json" http://127.0.0.1:8088/hash/any\_known\_job\_identifier
```
4. Run the command from step 2  

```
curl "application/json" http://127.0.0.1:8088/stats
```

### Current Behavior:

Note that the total requests remains the same and was not incremented after running the command to get base64 encoded password

### Expected Behavior:

The number of total requests should increment after calling the stats endpoint

## Improvement 1

### Description:

HTTPS is not supported. It is strongly suggested that support for https is added since this would be pushed to production servers. HTTPS, compared to HTTP, is much more secure since SSL encryption is used with HTTPS protocol.

## Improvement 2

### Description:

In the /hash endpoint, when a string or anything other than an integer is supplied, the server responds with a 400 status code with the message

For example:

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

*"strconv.Atoi: parsing "sdfa": invalid syntax"*

The improvement here would be to standardize the message with a meaningful message, similar to "Malformed input\n" in other 400 responses.

## Extended testing

### Description:

Testing for any possible path traversal attacks returns the following, which is fine and expected:

```
curl -H "application/json" -i  
http://127.0.0.1:8088/hash/../../../../../../root/.ssh/authorized_keys
```

```
HTTP/1.1 404 Not Found  
Content-Type: text/plain; charset=utf-8  
X-Content-Type-Options: nosniff  
Date: Thu, 24 Feb 2022 11:15:55 GMT  
Content-Length: 19
```

404 page not found

### Description:

Testing for any unexpected behaviors, with the following curl command, the server responds with multiple 400

```
curl -H "application/json" -i http://127.0.0.1:8088/hash/ WC COOL FREE NEW NG OK UP! ATM
```

```
→ broken-hashserve curl -H "application/json" -i http://127.0.0.1:8088/hash/ WC COOL FREE NEW NG OK UP! ATM  
HTTP/1.1 400 Bad Request  
Content-Type: text/plain; charset=utf-8  
X-Content-Type-Options: nosniff  
Date: Thu, 24 Feb 2022 11:24:29 GMT  
Content-Length: 53  
  
strconv.Atoi: parsing "%F0%9F%9A%BE": invalid syntax  
HTTP/1.1 400 Bad Request: malformed Host header  
Content-Type: text/plain; charset=utf-8 1  
Connection: close  
  
400 Bad Request: malformed Host headerHTTP/1.1 400 Bad Request: malformed Host header  
Content-Type: text/plain; charset=utf-8 2  
Connection: close  
  
400 Bad Request: malformed Host headerHTTP/1.1 400 Bad Request: malformed Host header  
Content-Type: text/plain; charset=utf-8 3  
Connection: close  
  
400 Bad Request: malformed Host headerHTTP/1.1 400 Bad Request: malformed Host header  
Content-Type: text/plain; charset=utf-8 4  
Connection: close  
  
400 Bad Request: malformed Host headerHTTP/1.1 400 Bad Request: malformed Host header  
Content-Type: text/plain; charset=utf-8 5  
Connection: close  
  
400 Bad Request: malformed Host headerHTTP/1.1 400 Bad Request: malformed Host header  
Content-Type: text/plain; charset=utf-8 6  
Connection: close  
  
400 Bad Request: malformed Host headerHTTP/1.1 400 Bad Request: malformed Host header  
Content-Type: text/plain; charset=utf-8 7  
Connection: close  
  
400 Bad Request: malformed Host header
```