Testing  
Here I will demonstrate that both the Web API and Discord+Twitch Bot function in their desired manner.  
Also that when presented with malicious or malformed data, no fault or breach of security occurs.

**Requestor –** A user that has performed a web request to the web api.  
Code – Indicates the highlighted text represents a small snippet of code  
**AccessToken** – Used to authenticate a **Login**  
**AuthToken** – Used to authenticate a **Bot  
SAMPLEDATA –** Indicates that this data resulted in a success

# Web API

To test functionality I will use the **cURL** command line utility. Which allows you to make web requests easily from a command line.  
A typical request looks like this curl -X GET 'https://owlcoin.co.uk/webapi/Viewer' -H 'ID: 1' –d ‘{}’  
the –X indicates the request method (GET or POST)  
the ‘https://….’ Is the url to poll  
-H ‘ID: 1’ adds a header with the Key as ID and value of 1  
-d ‘{}’ contains any JSON data that is to be sent with the request

For each key type of request, I will chose a suitable example and will demonstrate the following:

1. Will respond correctly when provided with fully correct data.
2. Will respond with suitable errors when provided with
   1. Non-existent data, like an ID that doesn’t match anything
   2. Boundary data, ie password too short or not strong enough
   3. Erroneous, ie utter gibberish that no sensible user would enter
   4. Lack of authorisation

How data flows through the Web API by using Visual Studios built in debbuger

|  |  |  |  |
| --- | --- | --- | --- |
| Command function | Command | Sample Data | Returned Data |
| This command should create a **Login** with the given  **UserName** / **Email** and **Password**.  Only when provided with a **UserName** / **Email** where it isn’t already attached to another **Login.**  And where the **Password** is longer than 8 characters and contains a Capital, Number and Special. | curl -X POST 'https://owlcoin.co.uk/webapi/signup' -H 'UserName: USERNAME'  -H 'Password: PASSWORD' -d '' | USERNAME = Test  PASSWORD = test | {  "Data": null,  "Code": 400,  "Message": "Bad Request, User already exists"  } |
| USERNAME = Test@ | {  "Data": null,  "Code": 400,  "Message": "Bad Request, Username is not AlphaNumeric"  } |
| USERNAME = TestAccount  PASSWORD = test | {  "Data": null,  "Code": 400,  "Message": "Bad Request, Password too short"  } |
| USERNAME = TestAccount  PASSWORD = testtesttest | {  "Data": null,  "Code": 400,  "Message": "Bad Request, Password requires at least 1 Capital, 1 Number, 1 Special"  } |
| USERNAME = TestAccount  PASSWORD = testtest1 | SAME AS ABOVE |
| USERNAME = TestAccount  PASSWORD = testtest1! | SAME AS ABOVE |
| USERNAME = TestAccount  PASSWORD = TestTest1! | {  "Data": null,  "Code": 200,  "Message": "The requested task was performed successfully"  } |
| curl -X POST 'https://owlcoin.co.uk/webapi/signup' -H 'Email: EMAIL'  -H 'Password: TestTest1!' -d '' | EMAIL = testmail | {  "Data": null,  "Code": 400,  "Message": "Bad Request, Email is not valid"  } |
| EMAIL = testmail@gmailcom | SAME AS ABOVE |
| EMAIL = testmailgmail.com | SAME AS ABOVE |
| EMAIL = testmail@gmail.com | {  "Data": null,  "Code": 200,  "Message": "The requested task was performed successfully"  } |
| This command allows us to obtain a **AccessToken** to used for authentication purposes.  To do this we must send the **UserName** or **Email** for the account. Along with the **Password.**  If the **UserName** / **Email** correspond to a **Login** and the **Password** matches, then we will receive said **AccessToken**.  If not we will receive an error, indicating what went wrong. | curl -X POST 'https://owlcoin.co.uk/webapi/login'  -H 'UserName: USERNAME'  -H 'Password: PASSWORD' -d '' | USERNAME = TestAccount1  PASSWORD = TestTest1! | {  "Data": null,  "Code": 400,  "Message": "Bad Request, UserName does not correspond to an existing user"  } |
| USERNAME = TestAccount  PASSWORD = TestTest1 | {  "Data": null,  "Code": 400,  "Message": "Bad Request, Password does not match"  } |
| USERNAME = TestAccount  PASSWORD = TestTest1! | {  "Data": {  "UserName": "TestAccount",  "HashedPassword": null,  "AccessToken": "ACCESSTOKEN",  "Email": "",  "LastLoginDateTime": "2019-04-02T18:23:26.8010768+01:00",  "ID": 39  },  "Code": 200,  "Message": "The requested task was performed successfully"  } |
| USERNAME = testaccount  PASSWORD = TestTest1! | SAME AS ABOVE |
|  | curl -X GET 'https://owlcoin.co.uk/webapi/login'  -H 'ID: ID' | ID = -1 | {  "Data": null,  "Code": 400,  "Message": "Bad Request, ID does not match an existing object"  } |
| ID = dd | {  "Data": null,  "Code": 400,  "Message": "Bad Request, Malformed ID"  } |
| ID = 39 | {  "Data": {  "UserName": "TestAccount",  "HashedPassword": null,  "AccessToken": null,  "Email": null,  "LastLoginDateTime": "2019-04-02T18:23:26",  "ID": 39  },  "Code": 200,  "Message": "The requested task was performed successfully"  } |
|  | curl -X GET 'https://owlcoin.co.uk/webapi/login'  -H 'ID: 39'  -H 'AccessToken:ACCESSTOKEN' | ACCESSTOKEN: thdfyui | {  "Data": {  "UserName": "TestAccount",  "HashedPassword": null,  "AccessToken": null,  "Email": null,  "LastLoginDateTime": "2019-04-02T18:23:26",  "ID": 39  },  "Code": 400,  "Message": "Bad Request, AccessToken doesnt match"  } |
| ACCESSTOKEN: (A Valid AccessToken for this **Login**) | {  "Data": {  "UserName": "TestAccount",  "HashedPassword": null,  "AccessToken": null,  "Email": "othertestmail@gmail.com",  "LastLoginDateTime": "2019-04-02T18:23:26",  "ID": 39  },  "Code": 200,  "Message": "The requested task was performed successfully"  } |