**Testing and Documentation – Register API**

**Endpoints**

https://rjohnston80.webhosting3.eeecs.qub.ac.uk/registerapi/

https://rjohnston80.webhosting3.eeecs.qub.ac.uk/registerapi/userapi.php

https://rjohnston80.webhosting3.eeecs.qub.ac.uk/registerapi/globalapi.php

**Endpoints –** The registration/users API is split up into three separate sub APIs, each with different access levels and security, a user one for customers accounts to use, an admin one for admin accounts to use and then a global one for new users to use for registration and verification

**Security –** The admin and user based registration APIs use API-Keys for added security so that users cannot view other users data if they have an API-Key. It also prevents them from performing potentially malicious activities such as changing usernames, passwords and verifying accounts.

In total there are 7 different functions associated with the register API. They are documented and tested below. Note, some of the same functions are used multiple times.

**Public Register API (userapi.php)**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Function** | **Request Type** | **Parameter** | **Pass** |
| Change a user’s password. | PUT | function = passchange | **Yes** |
| Change a user’s username | PUT | function = usernamechange | **Yes** |
| Get a specific user’s details from the system. | GET | function = user | **Yes** |

**Public Register API (globalapi.php)**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Function** | **Request Type** | **Parameter** | **Pass** |
| Create a new user account | POST | function = newaccount | **Yes** |
| Verify a new account | GET | function = verify | **Yes** |

**Admin Register API (index.php)**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Function** | **Request Type** | **Parameter** | **Pass** |
| Change a user’s password. | PUT | function = passchange | **Yes** |
| Get a specific user’s details from the system. | GET | function = user | **Yes** |
| Lock an account. | PUT | function = lockaccount | **Yes** |

**Function 1 –** Change a user’s password

A response informing us of the successful password change.

|  |  |
| --- | --- |
| Parameters | adminusername=<ADMIN USERNAME> |
| API-KEY=<YOUR API KEY> |
| function=passchange |

|  |  |
| --- | --- |
| **Expected Response** | **Actual Response** |

**Function 2 –** Change a user’s username

A response informing us of the successful username change.

|  |  |
| --- | --- |
| Parameters | adminusername=<ADMIN USERNAME> |
| API-KEY=<YOUR API KEY> |
| function=usernamechange |

|  |  |
| --- | --- |
| **Expected Response** | **Actual Response** |

**A picture containing logo

Description automatically generated**

**Function 3 –** Get user details from system for specific user.

|  |  |
| --- | --- |
| Parameters | adminusername=<ADMIN USERNAME> |
| API-KEY=<YOUR API KEY> |
| function=user |
| id=<USERNAME> |

|  |  |
| --- | --- |
| **Expected Response** | **Actual Response** |

Text

Description automatically generated

A JSON schema returning a users details by specifying their username.

userName - string

email – string

postcode - string

fname - string

lname – string

grandTotal - string

**Function 4 –** Add a new user account

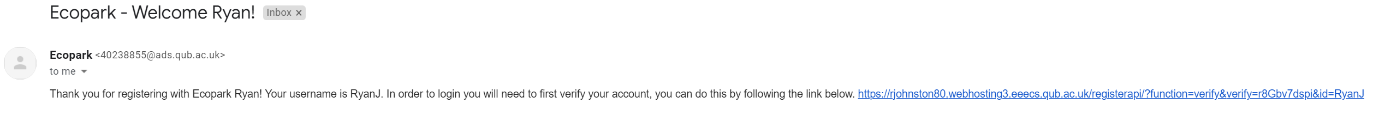
|  |  |
| --- | --- |
| Parameters | adminusername=<ADMIN USERNAME> |
| API-KEY=<YOUR API KEY> |
| email=<EMAIL> |
| id=<USERNAME> |
| fname=<FIRST NAME> |
| lname=<LAST NAME> |
| pcode=<POST CODE> |
| pass=<SHA256 ENCODED PASSWORD> |

|  |  |
| --- | --- |
| **Expected Response** | **Actual Response** |



A response informing us of the successful new account addition.

An email with verification link is also sent to the email associated with account.



**Function 5 –** Verify a user’s account

|  |  |
| --- | --- |
| Parameters | adminusername=<ADMIN USERNAME> |
| API-KEY=<YOUR API KEY> |
| function=verify |

|  |  |
| --- | --- |
| **Expected Response** | **Actual Response** |

Graphical user interface, application

Description automatically generated

A PHP Redirect to the verified account page if the verification code matches in the request.

**Function 6 –** Lock a user’s account

|  |  |
| --- | --- |
| Parameters | adminusername=<ADMIN USERNAME> |
| API-KEY=<YOUR API KEY> |
| function=lockaccount |
| id=<USERNAME> |

|  |  |
| --- | --- |
| **Expected Response** | **Actual Response** |



A response informing us of the successful account lock toggle.

**Error Scenarios**

In certain cases when incorrect functions are passed to the API the API will detect this and inform the user for debugging purposes.

The API will verify an API-Key against a user’s account or against the admin API Keys depending on which specific sub-API is used, if a wrong API Key is submitted it will return this response.



Incorrect function name will return this response from the API



In other cases where no admin username or API Key are passed at all the API will also detect this and inform the user.



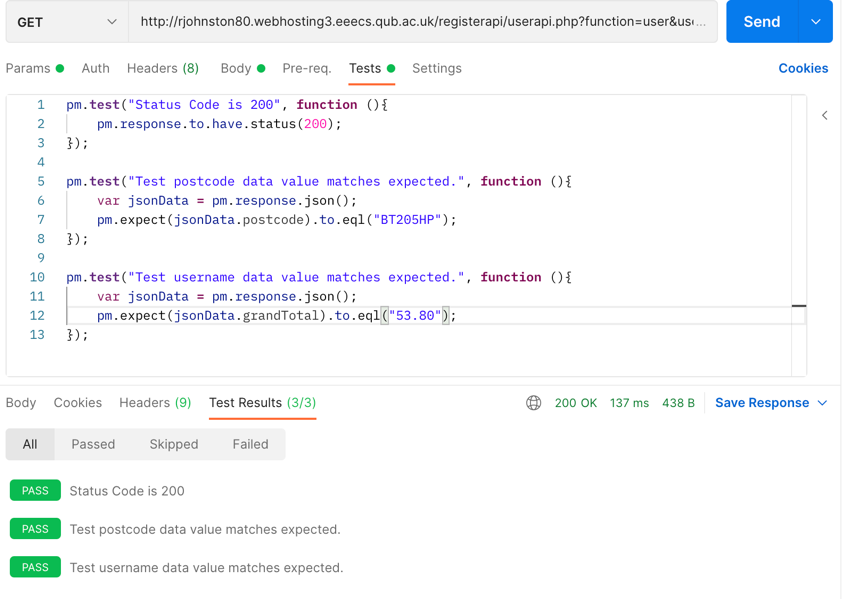
No function specified at all will return this response from the API



**Postman Tests**

We can also use postman and JS functions to verify our responses from the API to further troubleshoot issues using the Tests tab provided.

**Validating data types**



**Validating Error Responses**



Table

Description automatically generated with medium confidence