Pytest documentation for Steamy app

# Main.py

Routing tests for main.py. Database fetches will always be tested with total failure.

## /register

(I am making the assumption we wont check for @ etc.. )

Crud operations: get\_user\_by\_email, create\_user

Call /register can return:

* Success (User object created and returned)
* Error (User object already exists 🡪 HTTP 400)
* Error (Database failure 🡪 Exception)

Email validity can be [Valid/Non-Exist, Invalid/Exist]

Database fetch can be [No match, Match, Error]

|  |  |  |
| --- | --- | --- |
| Email | Database fetch | Expected outcome |
| Non-existing | No match | Success, User created |
| Existing | Match | HTTP 400 Error |
| - | Error | Raise Exception |

## /login

Crud operations: authenticate\_user

Call /login can return:

* Success (Auth user returned)
* Error (Invalid credentials 🡪 HTTP 400)
* Error (Database failure 🡪 Exception)

Credentials can be [Valid, Invalid]

Database fetch [Match, No match, Error]

( Valid & Match mutually exclusive )

|  |  |  |
| --- | --- | --- |
| Credentials | Database fetch | Expected outcome |
| Valid | Match exists | Success, auth user returned |
| Invalid | No match | HTTP 400 Error |
| - | Error | Raise Exception |

## /updateProfile

Crud operations: get\_user\_by\_id, update\_user

Call /updateprofile can return:

* Success (Updated user object)
* Error (User not found 🡪 HTTP 400)
* Error (Database failure 🡪 Exception)

User ID can be [Valid, Invalid]

Database fetch [Match, No match, Error]

( Valid & Match mutually exclusive )

|  |  |  |
| --- | --- | --- |
| User ID | Database fetch | Expected outcome |
| Valid | Match exists | Success, user updated and returned |
| - | No match | HTTP 400 Error |
| - | Error | Raise Exception |

## /sendMessage

Crud operations: send\_message

Call /sendMessage can return:

* Success (Message sent)
* Error (Users not matched 🡪 HTTP 400)
* Error (Database failure 🡪 Exception)

Sender and receiver ID can be [Valid, Invalid]

Database fetch [Match, No match, Error]

( Valid & Match mutually exclusive )

|  |  |  |
| --- | --- | --- |
| IDs | Database fetch | Expected outcome |
| Valid | Match exists | Success, Message sent and returned. |
| Valid | No match | HTTP 400 Error |
| Invalid | - | Val. Error |
| - | Error | Raise Exception |

## /match

Crud operations: create\_or\_update\_match

Call /match can return:

* Success (Match created or updated)
* Error (Users not matched 🡪 HTTP 400)
* Error (Database failure 🡪 Exception)

User IDs can be [Valid, Invalid]

Database fetch [Match, No match, Error]

( Valid & Match mutually exclusive )

|  |  |  |
| --- | --- | --- |
| IDs | Database fetch | Expected outcome |
| Valid | No match | Success, Message sent and returned. |
| Valid | Match exists | HTTP 400 Error |
| Invalid | - | Val. Error |
| - | Error | Raise Exception |