API DOCUMENTATION

FOR PERFORMING REST API REQUESTS ON LAPTOP STORE DATABASE V1.0

Prepared by

Tadeo Bennett

University of Belize

Prepared for

Mr. Lester Pech

University of Belize

CMPS4191: Advanced Web-Technologies

October 1, 2023

Table of Contents

Introduction

This API enables developers to perform operations such as GET, POST, PUT, and DELETE on the users of the laptopstore database. Its primary purpose is to enable developers to understand the requests' makeup and perform operations that yield expected results from the database. In addition, any error during the process is communicated to the developer to allow easy troubleshooting. This API is designed for any developer with knowledge of REST APIs, PHP, and MySQL.

These request examples can also be found in the online documentation.

Getting Started

If permitted, you can clone the repo from GitHub. In the extras directory, the file laptopstore_sql_dump must be used to create the laptopstore database. First, you'll need to create the "laptopstore" database and then run the file's contents in that database. Include the necessary login credentials for your mysql database in the index.php file. If Apache is used, I recommend having the root directory set up as the parent directory of the repo. It should look like the following:

root/CMPS4191_adv_web_RESTAPI/…

If required, ensure that your port is provided in the URI. It should look like the following:

127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/…

Endpoints

GET Get All Users

http://127.0.0.1/CMPS4191_adv_web_RESTAPI/users

This request returns all information for all users in the laptopstore database.

A successful request for all users will result in an *HTTP 200* status code.

Example Request:

curl --location '127.0.0.1/CMPS4191_adv_web_RESTAPI/users'

Example Response

See Appendix Page 9

GET Get Users by ID

http://127.0.0.1/CMPS4191_adv_web_RESTAPI/users/{id}

This request returns all information for a user with a certain ID in the laptopstore database.

A successful request for a user will result in an *HTTP 200* status code.

Example Request:

curl --location '127.0.0.1/CMPS4191_adv_web_RESTAPI/users/3'

Example Response

See Appendix Page 10

GET Get Employees

http://127.0.0.1/CMPS4191_adv_web_RESTAPI/users/employees

The request returns all information for employees in the laptopstore database.

A successful request for all employees will result in an *HTTP 200* status code.

Example Request:

curl --location '127.0.0.1/CMPS4191_adv_web_RESTAPI/users/employees'

Example Response

See Appendix Page 10

GET Get Customers

http://127.0.0.1/CMPS4191_adv_web_RESTAPI/users/customers

The request returns all information for customers in the laptopstore database.

A successful request for all customers will result in an *HTTP 200* status code.

Example Request:

curl --location '127.0.0.1/CMPS4191_adv_web_RESTAPI/users/customers'

Example Response

See Appendix Page 11

POST Add New User

http://127.0.0.1/CMPS4191_adv_web_RESTAPI/users

The request creates a new user in the laptopstore database with the details sent in the request body.

A successful request to create a user using the provided details will result in an *HTTP 200* status code.

Example Request:

```
curl --location '127.0.0.1/CMPS4191_adv_web_RESTAPI/users' \
--form 'firstname="Mary"' \
--form 'lastname="Jane"' \
--form 'age="25"' \
--form 'email="mary@gmail.com"' \
--form 'username="MJane"' \
--form 'member="1"'
```

Example Response

See Appendix Page 11

PUT Update User By ID

`http://127.0.0.1/CMPS4191_adv_web_RESTAPI/users/{id}`

The request updates an existing user record using an existing user's ID provided in the URI, and details to update in the request body.

A successful request to update a user using the provided ID will result in an *HTTP 200* status code.

Example Request

```
curl --location --request PUT '127.0.0.1/CMPS4191_adv_web_RESTAPI/users/5' \
--form 'firstname="Mary"' \
--form 'lastname="Jane"' \
--form 'age="25"' \
--form 'member="1"' \
--form 'email="mary@gmail.com"'
```

Example Response

See Appendix Page 12

DELETE Delete a User By ID

`http://127.0.0.1/CMPS4191_adv_web_RESTAPI/users/{id}`

The request deletes an existing user record using an existing user's ID provided in the URI. A successful request to delete a user using the provided ID will result in an *HTTP 200* status code.

Example Request

```
curl --location --request DELETE '127.0.0.1/CMPS4191_adv_web_RESTAPI/users/5'
```

Example Response

See Appendix Page 12

Response Format

API Structure

HTTP Response Codes Usage

Two response codes are used in the current version of the API:

HTTP Response Code 200

- 200 OK: The request was successful.

HTTP Response Code 400

- The client's request is malformed.

Response Codes and Their Meanings

Common Errors

-1: Invalid Request. Unknown resource Requested; expecting “users”.

-2: Unsupported Request Method. Could not find the request method GET, POST, PUT, or DELETE.

-3: No Database connection; Database connection was lost

GET Requests Errors

-11: Default; Invalid GET Request

-12: Default; Users’ details not found

-13: STMT Execution Error when getting all users

-14: Default; User details not found with the ID provided

-15: STMT Execution Error when getting user with the ID provided

-16: Default; Employees’ details not found

-17: STMT Execution Error when getting all employees

-18: Default; Customers’ details not found

-19: STMT Execution Error when getting all customers

POST Request Errors

-20: Default; Invalid POST Request

-21: STMT Execution Error when creating a new user

PUT Request Errors

-22: Default; Invalid PUT Request

-23: STMT Execution Error when updating an existing user’s details

DELETE Request Errors

-24: Default; Invalid DELETE Request

-25: STMT Execution Error when deleting a user

Success Response Codes

50: Successful GET request for all users’ details

51: Successful GET request for user with the provided ID

52: Successful GET request for all employees’ details

53: Successful GET request for all customers’ details

54: Successful POST request for creating a new user

55: Successful PUT request for updating a user with provided ID

56: Successful DELETE request for deleting a user with provided ID

100: Test function is working.

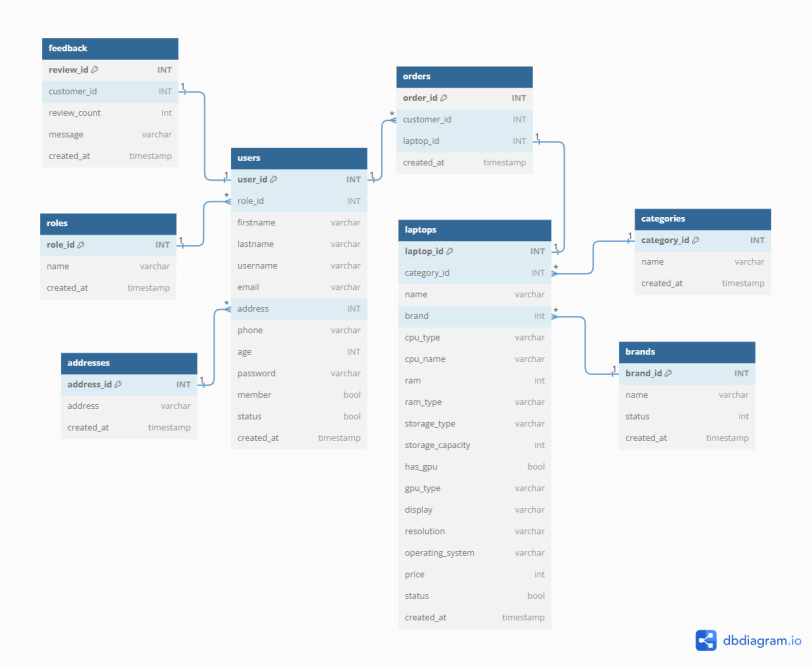
Support

If there are unhandled errors, kindly email me at 2021154344@ub.edu.bz. Ensure you have proper descriptions of the encountered problem and the events leading up to it. If you can, include pictures to demonstrate further.

Changelog

API updates, new features, and bug fixes are communicated in the commits of the git repository.

Appendix



Database Entity Relationship Diagram (ERD)

GET - Get All Users Example Response


```
{
    "rc": 50,
    "message": "Success",
    "data": [
        {
            "user_id": 1,
            "role_id": 1,
            "firstname": "Tadeo",
            "lastname": "Bennett",
            "username": "TBennett",
            "email": "tadeo@gmail.com",
            "address": 4,
            "phone": "343-8371",
            "age": 21,
            "password": "Tadeo2002",
            "member": 0,
            "status": 1,
            "created_at": "2023-09-30 17:30:43"
        },
        {
            "user_id": 2,
            "role_id": 2,
            "firstname": "William",
            "lastname": "Locario",
            "username": "WLocario",
            "email": "william@gmail.com",
            "address": 6,
            "phone": "837-3421",
            "age": 22,
            "password": "William2002",
            "member": 1,
            "status": 1,
            "created_at": "2023-09-30 17:30:43"
        },
        {
            "user_id": 3,
            "role_id": 2,
            "firstname": "Victor",
            "lastname": "Castillo",
            "username": "VCastillo",
            "email": "victor@gmail.com",
            "address": 2,
            "phone": "998-3234",
            "age": 19,
            "password": "Victor2002",
            "member": 0,
            "status": 1,
            "created_at": "2023-09-30 17:30:43"
        }
    ]
}
```

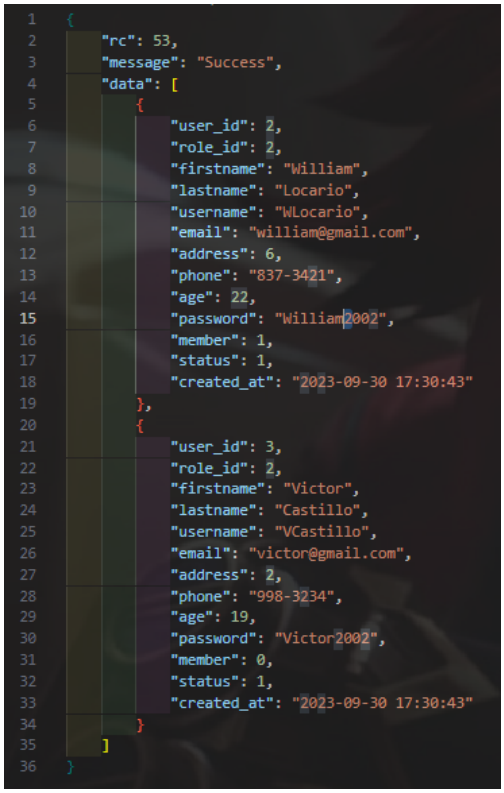
GET - Get Users by ID Example Response


```
{
    "rc": 51,
    "message": "Success",
    "data": [
        {
            "user_id": 3,
            "role_id": 2,
            "firstname": "Victor",
            "lastname": "Castillo",
            "username": "VCastillo",
            "email": "victor@gmail.com",
            "address": 2,
            "phone": "998-3234",
            "age": 19,
            "password": "Victor2002",
            "member": 0,
            "status": 1,
            "created_at": "2023-09-30 17:30:43"
        }
    ]
}
```


GET - Get Employees Example Response

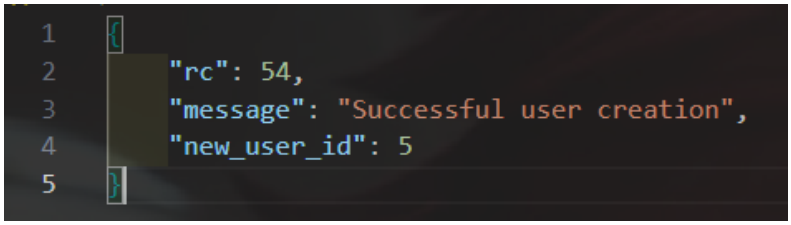

```
{
    "rc": 52,
    "message": "Success",
    "data": [
        {
            "user_id": 1,
            "role_id": 1,
            "firstname": "Tadeo",
            "lastname": "Bennett",
            "username": "TBennett",
            "email": "tadeo@gmail.com",
            "address": 4,
            "phone": "343-8371",
            "age": 21,
            "password": "Tadeo2002",
            "member": 0,
            "status": 1,
            "created_at": "2023-09-30 17:30:43"
        }
    ]
}
```

GET - Get Customers Example Response



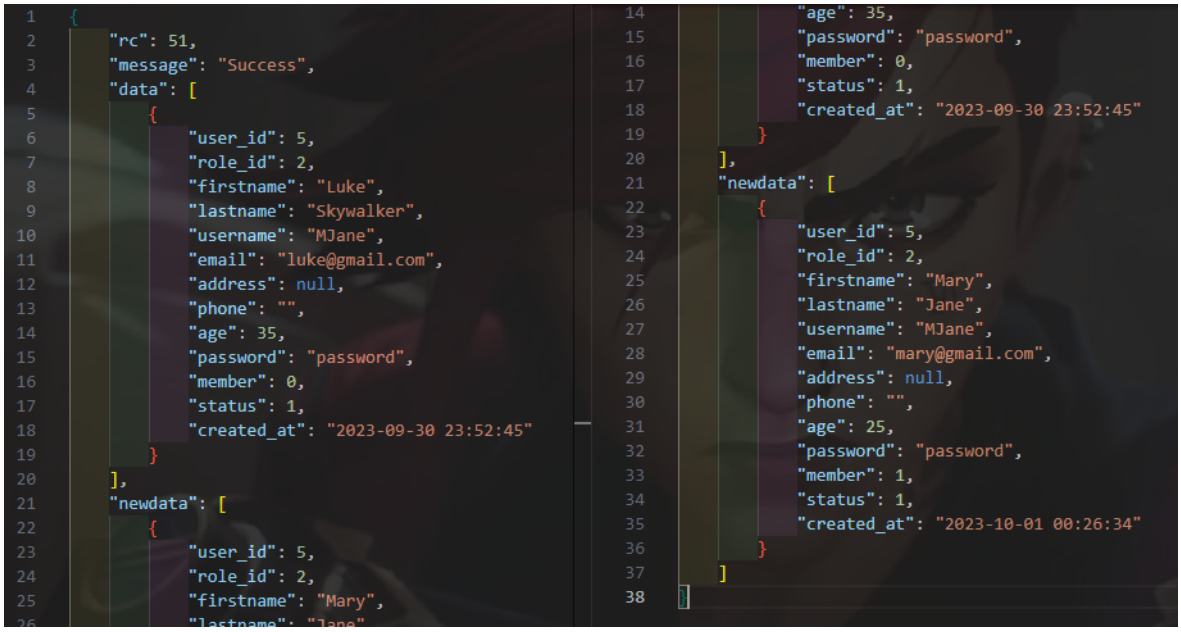

```
{
    "rc": 53,
    "message": "Success",
    "data": [
        {
            "user_id": 2,
            "role_id": 2,
            "firstname": "William",
            "lastname": "Locario",
            "username": "WLocario",
            "email": "william@gmail.com",
            "address": 6,
            "phone": "837-3421",
            "age": 22,
            "password": "William2002",
            "member": 1,
            "status": 1,
            "created_at": "2023-09-30 17:30:43"
        },
        {
            "user_id": 3,
            "role_id": 2,
            "firstname": "Victor",
            "lastname": "Castillo",
            "username": "VCastillo",
            "email": "victor@gmail.com",
            "address": 2,
            "phone": "998-3234",
            "age": 19,
            "password": "Victor2002",
            "member": 0,
            "status": 1,
            "created_at": "2023-09-30 17:30:43"
        }
    ]
}
```


POST - Add New User Example Response




```
{
    "rc": 54,
    "message": "Successful user creation",
    "new_user_id": 5
}
```

PUT - Update User Record by ID Example Response



```
{
    "rc": 51,
    "message": "Success",
    "data": [
        {
            "user_id": 5,
            "role_id": 2,
            "firstname": "Luke",
            "lastname": "Skywalker",
            "username": "MJane",
            "email": "luke@gmail.com",
            "address": null,
            "phone": "",
            "age": 35,
            "password": "password",
            "member": 0,
            "status": 1,
            "created_at": "2023-09-30 23:52:45"
        }
    ],
    "newdata": [
        {
            "user_id": 5,
            "role_id": 2,
            "firstname": "Mary",
            "lastname": "Jane",
            "username": "MJane",
            "email": "mary@gmail.com",
            "address": null,
            "phone": "",
            "age": 25,
            "password": "password",
            "member": 1,
            "status": 1,
            "created_at": "2023-10-01 00:26:34"
        }
    ]
}
```


DELETE - Delete User Record by ID Example Response

```
{
    "rc": 2,
    "message": "Success. User deleted with id 5",
    "data": [
        {
            "user_id": 5,
            "role_id": 2,
            "firstname": "Mary",
            "lastname": "Jane",
            "username": "MJane",
            "email": "mary@gmail.com",
            "address": null,
            "phone": "",
            "age": 25,
            "password": "password",
            "member": 1,
            "status": 1,
            "created_at": "2023-10-01 00:26:34"
        }
    ]
}
```