Class: DAAA/FT/1B/01

Name: Tan Wen Tao Bryan

Student ID: p2214449

**BED CA2 APIs Explanation**

**Model Layer:**

loginStaff: (Login Page)

* Select all data from staff table that matches email and password
* If query returns a result, code creates a JWT token using jwt.sign() function
* Token created with staff members’ staff\_id and last\_name as payload, signed using the config object and set to expire in 24 hours
* Password is then removed from the result object and the token, along with the result (without the password), returned through callback
* If query returns no result, Code statusCode is set to 403 and creates an error.

getFilms: (Home Page)

* Displays all the films search results (default) in home page
* Select film\_id, film title, film release\_year and film rating from film table

getCategory: (Home Page)

* Select categories names and ids by category table in ascending order
* Displays all possible categories in dropdown of the search bar

getSearchResults: (Home Page)

* Get films related to search results by category, title and those below max rental price
* Select film\_id, film title, film release\_year and film rating from film, category and film\_category table while joining fc.film\_id and f.film\_id and c.category\_id and fc.category\_id
* Coded for different case scenarios like when category and title contain input but price do not contain input

getDetails: (Details Page)

* Get film\_id, title, rating, category, release\_year, description, actor’s first\_name and last\_name from film, category, film\_category, film\_actor and actor table
* film\_id is used as parameter

insertActor: (Add A New Actor Page)

* Insert first\_name and last\_name values into actor table

getStores: (Add A New Customer Page)

* Select addresses and store\_id from address and store tables where address.address\_id and store.address\_id are joined

insertCustomer: (Add A New Customer Page)

* Insert information as stated into address table
* Store id of result which is address\_id
* Insert information as stated into customer table
* When there is an error inserting into customer table, deletes information from address table to make sure insertion is consistent across both tables

insertRating: (Ratings Page)

* Get the customer\_id based on the customer first\_name and last\_name
* If customer\_id exists, store it in a variable. Else, returns an error message
* Get the film\_id based on the film title
* If film\_id exists, store it in a variable. Else, returns an error message
* Inserts stated information into ratings table

getReviews: (Details Page)

* Retrieve necessary information and format ratings.last\_update as (Year, Month, Day) from film, customer, ratings table based on the film\_id parameter

**Controller Layer:**

/login (Login Page)

* Handles login requests, if email or password input is empty, returns res.status 400
* Queries database for staff member with provided email and password
* If query returns a result, JWT token is generated for the staff member, the password value is deleted from the result and callback function is called with the token and results as arguments
* If query do not return a result, it means that email or password do not match
* Status code set to 201 on success, 403 on invalid email/password or 500 on server error

/films (Home Page)

* Status code set to 200 on success or 500 on server error

/film\_categories/category (Home Page)

* Status code set to 200 on success or 500 on server error

/films/results (Home Page)

* If category, title or maxRental parameters are missing, it will return a status code (400)
* Status code set to 200 on success or 500 on server error

/details/:film\_id (Details Page)

* film\_id is requested as parameter
* Status code set to 200 on success or 500 on server error

/actors (Add A New Actor Page)

* Request body contains the request for first\_name and last\_name
* verifyToken is used to verify the authenticity of the token
* If first\_name or last\_name parameters, or one of the parameters’ value is missing, returns a status code (400)
* Status code set to 201 on success or 500 on server error

/add\_customers/address (Add A New Customer Page)

* Status code set to 200 on success or 500 on server error

/add\_customers (Add A New Customer Page)

* Verifies token for authentication first
* Customer data is all in the request body and checks if all the necessary information are filled up, else returns a status code (400)
* Status code set to 201 on success and 500 on server error and 409 on duplicate entry for email

/add\_ratings (Ratings Page)

* Request body contains the request for first\_name, last\_name, title, score and comments
* If any of the necessary information is not filled up, returns a status code (400)
* If the score is not between 0 and 5, returns a status code (400)
* Status code set to 201 on success and 500 on server error

/details/ratings/:film\_id (Details Page)

* Take film\_id as parameter
* Status code set to 200 on success and 500 on server error