Ryan Fleming

CS496

Cloud Final Project

The Zip File includes both the source code as well as the Postman **Collection** of tests, the **Results** of those tests, and a saved Postman **Environment** in 3 JSON files.

This API models a high end specialty car rental agency. There are two entities.

Car = {

“model”,

“manufacturer”,

“checkedIn”,

“id”,

“user”,

“vin”}

Customer = {

“name”,

“balance”,

“checked_out”,

“id”,

“user”,

“license”}

Every route requires an access token obtained through OAuth2. If an invalid or no token is included in the request header, the action does not take place and the response code is 401. If a valid token is used for an action but tries to use the ID of an entity that belongs to a different user, the action does not take place and the response code is 403.

Routes

1) Basic implementations for Car:

GET /cars

GET /cars/:car_id

DELETE /cars

DELETE /cars/:car_id

POST /cars

PUT /cars/:car_id

PATCH /cars/:car_id

2) Basic implementations for Customer:

GET /customers

GET /customers/:customer_id

DELETE /customers

DELETE /customers/:customer_id

POST /customers

PUT /customers/:customer_id

PATCH /customers/:customer_id

3) Check in/check out Car:

PUT /customers/:customer_id/cars/:car_id

DELETE /customers/:customer_id/cars/:car_id

4) Get Customer's list of checked out Cars:

GET /customer/:customer_id/cars

5) Filter Cars by checked out status:

GET /cars?checkedIn=:boolean

6) Delete all Cars and Customers:

DELETE /