

Ethan Dunham

Cloud Only Final Project

### Usage instructions:

1. Navigate to <https://cloud-only-final.appspot.com/>
2. Use Google+ to login to your account
3. Copy your user ID provided on the next page
4. Open the Postman tests
5. Edit the Postman test environment to use your ID
  - a. You can run the tests with the default user id value once, but the account gets deleted at the end of testing, so it will only work for 1 test
  - b. For subsequent tests, repeat step 1 and 2. The user id will stay the same if you use the same Google+ account. So, editing the environment won't be needed.
6. Run the Postman tests

### Entities:

1. UserAccount
  - a. ID (private for security reasons)
  - b. User\_id
  - c. Fname
  - d. Lname
  - e. Email
2. Vehicle
  - a. Id (private for security reasons)
  - b. Owner\_id (private unless "none")
  - c. Vin
  - d. Vehicle\_make
  - e. Vehicle\_model
  - f. Purchase\_price

### Relationship:

- Vehicles usually have owners
- The owner\_id of the vehicle links to the private ID of the UserAccount
- Since the User does not have access to the ID of the UserAccount, the owner of the vehicles is protected
- A user can search for a vehicle by vin number. They will only see the car information if that car is unowned or owned by the user
- Unowned cars can be claimed by the user if they know the vin of the unowned car
- Owned cars can be deleted by the user

- Owned cars can be released by the user, where the owner\_id is set to “none”
- Users can have multiple vehicles, but vehicles cannot have multiple owners

### 3<sup>rd</sup> party service:

I used Google+ with OAuth2.0 to verify the user’s identity. They have to login to Google+, as in the OAuth2.0 assignment. Then, the program creates a user account based on their Google+ id if the account does not exist.

### Routes:

users/{user\_id}

- Get
  - Returns the user id, first name, last name, and email of the user.
  - Returns “401: user does not exist” if a user with that ID does not exist.
- Patch
  - Changes the first name, last name, or email of the user.
  - Requires a JSON with one of the following variables:
    - ‘fname’
    - ‘lname’
    - ‘email’.
  - Returns “402: Incorrect Information Sent” if the user puts too much information, no information, or incorrect information.
  - Returns the user’s first name, last name, and email if changed.
- Put
  - Changes the first name, last name, and email of the user.
  - Requires a JSON with all the following variables:
    - ‘fname’
    - ‘lname’
    - ‘email’.
  - Returns “402: Incorrect Information Sent” if the user puts too much information, not enough information, or incorrect information.
- Delete
  - Deletes the user account and sets all vehicles they own to ownerless. To reactivate the account, the user will have to relog into the homepage and go through the OAuth2.0 sign in again.
  - Returns “User no longer exists” if successful.
  - Returns “403: user was not deleted” if failed.

vehicles/{user\_id}

- Get
  - Returns a list of all the cars the user owns.

- Returns “401: user not found” if the user doesn’t exist.
- Post
  - Adds a new car that the user owns.
  - Requires a json with all the following information:
    - ‘vin’
    - ‘vehicle\_make’
    - ‘vehicle\_model’
    - ‘purchase\_price’
  - Returns the car object if successful.
  - Returns “402: Incorrect information sent” if the user did not send the correct information.
  - Returns “405: that vin is already owned by someone” if the vin matches another vehicle.
- Delete
  - Requires a json with the ‘vin’ of the car.
  - Deletes the car completely.
  - Returns “Car deleted” if successful.
  - Returns “401: Car not found” if it fails.

/user/{user\_id}/vin/{vin}

- Get
  - Returns the car with that vin if the user owns it or it is unowned.
  - Returns “401: user not found” if the user doesn’t exist.
  - Returns “401: car not found” if the car doesn’t exist.
- Post
  - Sets the owner of the car to the user if the car is unowned.
  - Returns “401: user not found” if the user doesn’t exist.
  - Returns “401: car not found” if the car doesn’t exist.
  - Returns “410: That car is already owned by someone” if it is already owned by someone
- Patch
  - Edits the information of a single value of a car.
  - Requires a json with one of the following variables:
    - ‘vin’
    - ‘vehicle\_make’
    - ‘vehicle\_model’
    - ‘purchase\_price’
  - Returns “402: Incorrect Information Sent” if the user puts too much information, no information, or incorrect information.
  - Returns “406: Vin already in use” if another car is using the ‘new\_vin’ already.
- Put
  - Edits all the information of a car.
  - Requires a json with all the following variables:
    - ‘vin’
    - ‘vehicle\_make’
    - ‘vehicle\_model’

- 'purchase\_price'
  - Returns "402: Incorrect Information Sent" if the user puts too much information, no information, or incorrect information.
  - Returns "406: Vin already in use" if another car is using that vin already.
- Delete
  - Sets the owner of the car to unowned.
  - Returns "401: user not found" if the user doesn't exist.
  - Returns "401: car not found" if the car doesn't exist.