REQUESTS

```
GET /api/user/customer/:id
        Expected Response should be :
        {
                “email” :  —> should be unique
                “history” : [123,232,341, etc. ] —> array of rideIDs
                “name” :
                “phone” :
        }

        Response in case of failure (404 Not found):
        {} <--- empty body

GET /api/user/driver/:id
        Expected Response should be :
        {
                “email” :  —> should be unique
                “history” : [123,232,341, etc. ] —> array of rideIDs
                “name” :
                “phone” :
                “service”: —> ex. “UberX”
                “car”:
                “customerRequest” :
                        {       “customerID” :  —> ID of the USER Going to Ride
                                “destinationLatitude” :
                                “destinationLongitude” :
                        }

        }
        Response in case of failure (404 Not found):
        {} <--- empty body

GET /api/available_drivers/
        Expected Response should be
        {
                “data” : [ {“driverID”:
                           “currLatDriver”:
                           “currLonDriver”:
                          } ]
        —> “data” is an Json Array containing json objects with data of all drivers
available         for duty
        }
        Response in case of failure (404 Not found):
```

{} ←--- empty body

GET /api/history/:rideid

Expected Response should be :

```
{
        “customerID”:
        “destination”:
        “distance”:
        “driverID” :
        “location” : {
                        “from” : {
                                        “lat” :
                                        “long” :
                                 }
                        “to” :    {
                                        “lat” :
                                        “long” :
                                 }
                   }
        “rating” :
        “timestamp” :
}
```

Response in case of failure (404 Not found):

{} ←--- empty body

POST /api/user/customer/register

Request will be of the form

```
{
        “email” :  —> should be unique
        “password” :
}
```

Response in case of failure (400 Bad request):

```
{
    “status”:“failure”,
    “reason”:“explanation”     ←--------- explanation can be any message
}
```

POST /api/user/driver/register

Request will be of the form

```
{
        “email” :  —> should be unique
        “password” :
        }
```

Response in case of failure (400 Bad request):

```
        {
            “status”:“failure”,
            “reason”:“explanation”     ←--------- explanation can be any
message
        }
```

POST /api/user/customer/login OR /api/user/customer/login

Request will be of the form:

```
{
        “email”:  ←-------- this is unique username
        “password”:
}
```

Response in case of success (200):

```
{
    “access_token”:“sdkfjlskjfs”     ←--------- sesssion token for User
}
```

Response in case of failure (404 (preferred) / 400):

```
{
    “status”:“failure”,
    “reason”:“explanation”     ←--------- explanation can be any message
}
```

POST /api/availabledrivers/add

Request will be of the form

```
        {
                “driverID”:
                “lat” :
                “lon” :
        }
```

Response in case of failure (400 Bad request):

```
        {
            “status”:“failure”,
            “reason”:“explanation”     ←--------- explanation can be any
message
        }
```

PUT /api/user/customer/updateDetails

Put Api should update the details of existing user. Request will be of the form containing one or more of the following fields:

```
{
        “name”:
```

```
        “phone”:
}
```

Response in case of success should be (with 200 OK response code):

```
{
   “status”:“success”
}
```

PUT /api/user/driver/updateDetails

Put Api should update the details of existing user. Request will be of the form containing one or more of the following fields:

```
{
        “name”:
        “phone”:
        “service”:
        “car”:

}
```

Response in case of success should be (with 200 OK response code):

```
{
   “status”:“success”
}
```