



RoboTaxi

Controller API

Guillermo Morán

Revised: 17 April 2018

Note: All examples are written in Swift (for simplicity's sake)

Endpoint

<https://gmoran.create.stedwards.edu/RoboTaxi/Controller.php>

We are only POSTing information to the controller, therefore we only submit POST requests.

PUT/DELETE/GET requests are handled by the private APIs.

Request Types

In order for the controller to detect what kind of request the client is making, a “Request type” has been implemented. As of now, these are the request types available to the client:

ORDER	Submits a request for an order and vehicle
AUTHENTICATE	Requests server authentication (user/pass)
PING	Pings the server - checks for a connection
UPDATE_VEHICLE	Requests the specified vehicle's current information
VEHICLE_LIST	Requests a list of available vehicles in an area
ROUTE_VEHICLE	Begin routing a specific vehicle to the given coordinate(s)

--	--

“ORDER” Request

The ORDER request is used when the client is requesting an order (vehicle and route). In order to construct a valid request, the following parameters must be POSTed to the server*:

POST Key	Description
user_name	Valid username for authentication
user_pass	Valid password for authentication
user_longitude	Client’s current longitude location
user_latitude	Client’s current latitude location
dest_long	Intended destination longitude
dest_lat	Intended destination latitude
date	Today’s date
request_type	request_type must be “ORDER”

*The user_name and user_pass must be a valid username/password combination in order to request a vehicle.

Failure to meet any of these requirements will result in the JSON response:

```
{status : Failure}
```

A valid request will yield the following JSON response (example):

```
{
  "status": Success,
  "order_id": 893,
  "end_lat": 1,
  "orderDate": Today,
  "user_id": 0,
  "start_lat": 1,
  "end_long": 1,
  "start_long": 1

  "vehicle": {
```

```

        capacity = 4;
        currentLatitude = "30.22828600";
        currentLongitude = "-97.75381000";
        inService = 1;
        inUse = 1;
        ownerID = 1;
        vehicleID = 1;
    },
}

```

The JSON Response will return a trip route information and a vehicle object (with information on such vehicle)

“AUTHENTICATE” Request

The AUTHENTICATE request is used to authenticate a username/password combination. In order to construct a valid request, the following parameters must be POSTed to the server:

POST Key	Description
user_name	Valid username for authentication
user_pass	Valid password for authentication
request_type	request_type must be “AUTHENTICATE”

Successful authentication will yield the following JSON response:

```
{“status”: Authenticated}
```

Otherwise:

```
{“status”: Authentication Failure}
```

“PING” Request

The PING request is used to verify that all components of the server are working as intended. In order to construct a valid request, the following parameters are required:

POST Key	Description
request_type	request_type must be “PING”

Successfully pinging the server will yield the following JSON response:

```
{“status”: PONG}
```

If one or more servers are unavailable:

```
{“status”: Server Unavailable}
```

“UPDATE_VEHICLE” Request

The PING request is used to authenticate a username/password combination. In order to construct a valid request, the following parameters must be POSTed to the server:

POST Key	Description
request_type	request_type must be “UPDATE_VEHICLE”
vehicleID	A valid vehicle id number

Successfully requesting a vehicle would yield the following JSON response:

```
{
  capacity = 4;
  currentLatitude = "30.22828600";
  currentLongitude = "-97.75381000";
  inService = 1;
  inUse = 1;
  ownerID = 1;
  vehicleID = 1;
}
```

“VEHICLE_LIST” Request

The VEHICLE_LIST request is used to return a list of all vehicles on the map. In order to construct a valid request, the following parameters must be POSTed to the server:

POST Key	Description
request_type	request_type must be “VEHICLE_LIST”
user_longitude	User’s current longitude coordinate
user_latitude	User’s current latitude coordinate

Returns a JSON dictionary with other JSON vehicle objects.

“ROUTE_VEHICLE” Request

The ROUTE_VEHICLE request is used to begin routing a vehicle:

POST Key	Description
request_type	request_type must be “ROUTE_VEHICLE”
vehicleID	ID of vehicle we want to move
route_coordinates	String of coordinate pairs (i.e: 1 -1 1 -2)

If routing was successful, the following response is returned:

```
{status : Success}
```