I- **http://someUrl.com/getAssets**

Method: **get**

Input: -

machine fields = [id , name , code , phone]

Response: Table of all machines information

{

    machineID1 : { machine fields } ,

    machineID2 : { machine fields } ,

    … ,

    machineIDn : { machine fields } ,

}

Away

II-   ws://someUrl.com/update

Description: Just connect to the channel and receive updating data as following:

Must send the Input below on the channel after each time you receive the new packet of data. (websocket.send("repeat"))

Input: 'repeat'

Response: List of the **latest GPS** and **Status** data for **each** Machine. Layout:

```
{
        machID1: {
                                'gps':       all fields of GPS data table,
                                'status':    all fields of status data table,
                } ,
        machID2: {
                                'gps':       all fields of GPS data table,
                                'status':    all fields of status data table,
                } ,
        …
}
```

III-   http://someUrl.com/call/<int:machID>

Method: **get**

Input: -

Response: 1 as OK, -1 as notOK

IV-   http://someUrl.com/history/raw

Method: **get**

Input: -

Response:

```
{
        raw data of all records for loadings and activities
}
```

V- **http://someUrl.com/loadHistory**

Method: **post**

Input: {

         'machID':   machID or All   # whether an ID is given or 'all' keyword

         'intervalType':   intervalType,   # [daily, weekly, monthly]

         'count':   countOfIntervals # integer value

     }

Response: Requested Data from Loading History. Layout:

{

     machID1: {

         '*interval$_1$*':   Loading Counts,

         '*interval$_2$*':   Loading Counts,

         …

         '*interval$_n$*':   Loading Counts,

     } ,

     machID2: {

         '*interval$_1$*':   Loading Counts,

         '*interval$_2$*':   Loading Counts,

         …

         '*interval$_n$*':   Loading Counts,

} ,

     …

}

hint : n is the number of intervals.

VI- http://someUrl.com/activityHistory

Method: **post**

Input: {

        'machID':   machID or All   # whether an ID is given or 'all' keyword

        'intervalType':   intervalType,   # [daily, weekly, monthly]

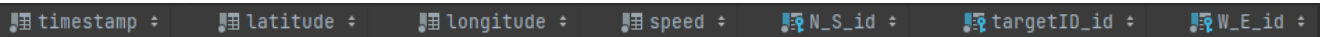   'count':   countOfIntervals # integer value

        }

Response: Requested Data from Loading History. Layout:

    {

      machID1: {

               '*interval$_1$*':   Loading Counts,

               '*interval$_2$*':   Loading Counts,

               …

               '*interval$_n$*':   Loading Counts,

          } ,

      machID2: {

               '*interval$_1$*':   Loading Counts,

               '*interval$_2$*':   Loading Counts,

               …

               '*interval$_n$*':   Loading Counts,

          } ,

      …

    }

hint : n is the number of intervals.

P.S:

gpsData Fields:

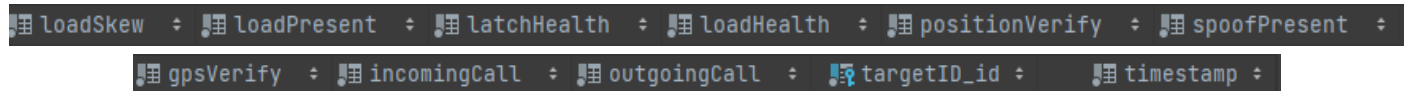| timestamp	latitude	longitude	speed	N_S_id	targetID_id	W_E_id

N_S : North_South

W_E : West_East

target : The Module

statusData Fields:

*All fields are Boolean Type, but timestamp and targetID

loadSkew	loadPresent	latchHealth	loadHealth	positionVerify	spoofPresent
gpsVerify	incomingCall	outgoingCall	targetID_id	timestamp	

laodSkew   :   کجی محل بار

loadPresent   :   حضور بار

latchHealth   :   سلامت قفل

loadHealth   :   سلامت بار

positionVerify :   احراز موقعیت

spoofPresent   :   وجود حمله

gpsVerify   :   احراز جی‌پی‌اس

incomingCall   :   تماس ورودی

outgoingCall   :   تماس خروجی

target   :   The Module Id