1. **http://someUrl.com/Login** ***Not Written Yet.***

*Method*: **post**

*Input*: {

‘username’: username

‘password’: password

}

*Response*: a TOKEN for accessing all APIs. Layout:

{

‘token’: TOKEN

}

This TOKEN must be passed through the Headers of each API request. As Follows:

*Headers*:

{

… ,

'Authorization': “Token [TOKEN variable]”,

… ,

}

“Token [TOKEN variable]”: **String** with the following format :

“Token” **+** space **+**  TOKEN\_Variable **=** “Token [TOKEN variable]”

1. **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 } ,

]

1. **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,

} ,

…

}

1. **http://someUrl.com/call/<int:machID>**

*Method*: **get**

*Input*: -

*Response*: 1 as OK, -1 as notOK

1. **http://someUrl.com/history/raw**

*Method*: **get**

*Input*: -

*Response*:

{

raw data of all records for loadings and activities

}

1. **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*: {

‘*interval1’*: Loading Counts,

‘*interval2’*: Loading Counts,

…

‘*intervaln’*: Loading Counts,

} ,

*machID2*: {

‘*interval1’*: Loading Counts,

‘*interval2’*: Loading Counts,

…

‘*intervaln’*: Loading Counts,

} ,

…

}

***hint : n is the number of intervals.***

1. **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*: {

‘*interval1’*: Loading Counts,

‘*interval2’*: Loading Counts,

…

‘*intervaln’*: Loading Counts,

} ,

*machID2*: {

‘*interval1’*: Loading Counts,

‘*interval2’*: Loading Counts,

…

‘*intervaln’*: Loading Counts,

} ,

…

}

***hint : n is the number of intervals.***

***P.S:***

gpsData Fields:



N\_S : North\_South

W\_E : West\_East

target : The Module

statusData Fields:

\*All fields are Boolean Type, but timestamp and targetID

  


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