

```
1 #-----  
-----  
2 # Name:          This script is for interacting front end with IOTA  
3 # Author:       Rejith Reghunathan  
4 # Created:      18/04/2022  
5 # Licence:      MIT  
6 #-----  
-----  
7  
8  
9 from flask import Flask, render_template, request, flash, redirect  
10 import config  
11 from web3 import Web3  
12 import ccxt, time  
13 from web3.middleware import geth_poa_middleware  
14  
15 from web3 import Web3, HTTPProvider  
16 import json  
17  
18 import os  
19 from dotenv import load_dotenv  
20  
21 load_dotenv()  
22  
23 PRIVATE_KEY = os.getenv('PRIVATE_KEY')  
24 OWNER_ADDRESS = os.getenv('OWNER_ADDRESS')  
25  
26  
27 # reading contract ABI data  
28 with open("brownie/build/contracts/IOT_Web3.json") as f:  
29     info_json = json.load(f)  
30 abi = info_json["abi"]  
31  
32  
33 # HTTP Provider  
34 #rinkeby_url =  
35 "https://rinkeby.infura.io/v3/6044b6684296420ca3436a6055ed50c3"  
36 #w3 = Web3(Web3.HTTPProvider(rinkeby_url))  
37  
38 # HTTP Provider  
39 iota_chain_url = "http://127.0.0.1:8545"  
40 w3 = Web3(Web3.HTTPProvider(iota_chain_url))  
41  
42 # Smart contract address - convert to checksum address  
43 address =  
44 w3.toChecksumAddress('0x4f362901b2a82f682207773d9e808AaFaEf57652')  
45  
46 # Initialize Smart contract  
47 contract = w3.eth.contract(address=address, abi=abi)  
48  
49  
50 # Set a default account to sign transactions - this account is unlocked  
with Ganache  
51 #w3.eth.defaultAccount = w3.eth.accounts[0]  
52  
53  
54
```

```

55
56 ##### FLASK #####
57 app = Flask(__name__)
58
59 @app.route("/", methods = ['POST', 'GET'])
60 def index():
61
62     value_from_ether_bc=contract.functions.getiot_datas().call(block_identifie
r='latest')
63     ## for displaying latest 4 datas
64     latest_data = []
65     temp_data =[]
66     vib_data =[]
67     index = 0
68
69     if(len(value_from_ether_bc)>5):
70         for data in range(len(value_from_ether_bc)-1,
len(value_from_ether_bc)-5, -1):
71             sensor_data_list = value_from_ether_bc[data]
72             latest_data.append(sensor_data_list)
73             if(latest_data[index][0] == "VIBR_SENSOR_002"):
74                 vib_data.append(sensor_data_list)
75             if(latest_data[index][0] == "TEMP_SENSOR_002"):
76                 temp_data.append(sensor_data_list)
77             index+=1
78
79
80
81
82     latest_vib_data = vib_data[0][2]
83     latest_temp_data = temp_data[0][2]
84
85     labels_vib = [row[1] for row in vib_data]
86     data_vib = [row[2] for row in vib_data]
87     labels_temp = [row[1] for row in temp_data]
88     data_temp = [row[2] for row in temp_data]
89
90     print(data_vib)
91     print(data_temp)
92
93     form_data =[]
94     service_data =[]
95
96     data_from_contract=contract.functions.getstatus().call(block_identifier='l
atest')
97
98     if(len(data_from_contract)>2):
99         for data in range(len(data_from_contract)-1,
len(data_from_contract)-2, -1):
100             data_temp = data_from_contract[data]
101             service_data.append(data_temp)
102             print(service_data)
103
104     if request.method == 'POST':
105         form_data = request.form
106         print(str(form_data["service_date"]))
107         nonce = w3.eth.getTransactionCount(OWNER_ADDRESS)
108         txn =
contract.functions.setstatus(str(form_data["service_date"]),str(form_data["

```

```
next_service")),str(form_data["healthy"]),str(form_data["company"]),str(form
m_data["engineer"])).buildTransaction(
108     {"gasPrice": w3.eth.gas_price, "from": OWNER_ADDRESS, "nonce":
nonce})
109
110
111     signed_txn = w3.eth.account.sign_transaction(txn,
private_key=PRIVATE_KEY)
112     txn_hash = w3.eth.sendRawTransaction(signed_txn.rawTransaction)
113
114     print(txn_hash)
115
116
117
118
119
120
121     return
render_template('index.html', latest_data=latest_data, latest_vib_data=latest
_vib_data, latest_temp_data=latest_temp_data, labels_vib=labels_vib,
labels_temp=labels_temp, data_temp=data_temp, data_vib=data_vib,
service_data= service_data)
122
123
124
125
126 #####
127 if __name__ == '__main__':
128     app.run(port=8001, host='0.0.0.0', debug=True)
129
130 #####
131
```