

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import sys
6  import json
7
8  from LIB import db
9  from LIB import ethereum
10
11  DEBUG = False # True
12
13
14  class Event:
15      # コンストラクタ
16      def __init__(self):
17          print("Constructor of Event")
18          self.event_list = {}
19          self.eth_rec = None
20
21          s = db.DB()
22          # Ethereum への接続設定
23          eth_rec = s.select(db.DsEthereum, "Ethereum")
24          if DEBUG : print("eth_rec=",eth_rec)
25
26          url = "http://" + eth_rec.address + ":" + str(eth_rec.port)
27          account = eth_rec.account
28          password = eth_rec.password
29          private_key = eth_rec.private_key
30          self.eth = ethereum.Ethereum(url, account, password, private_key)
31
32      # イベント登録
33      self.events = s.list(db.DsEvent)
34      for rec in self.events:
35          if DEBUG : print(rec.id, rec.name, rec.func_id)
36
37          rec.key = rec.method + rec.url
38          rec.call = self.call_contract
39
40          # Function 情報の取得
```

```

41         rec.func = s.select(db.DsFunction, rec.func_id)
42         if DEBUG : print(rec.func)
43         if rec.func:
44             # コントラクトの実行準備
45             self.eth.set_event_contract(rec.func_id, rec.func.address, rec.func.abi)
46
47         if DEBUG:
48             for rec in self.events:
49                 print(rec.id,rec.method,rec.url,rec.call,rec.func)
50
51
52     # デストラクタ
53     def __del__(self):
54         print("Destructor of Event")
55
56
57     # コントラクト実行
58     def call_contract(self, evt, request, response):
59         print("call_contract", evt, request, response)
60         print("event = ", evt.id, evt.name)
61
62         account = None
63         if evt.func:
64             params = {}
65             params['name'] = evt.id
66             params['url'] = request.url.path
67             params['method'] = request.method
68             params['user'] = request.client.host
69
70             if 'user' in request.query_params:
71                 params['user'] = request.query_params['user']
72             if 'account' in request.query_params:
73                 params['account'] = request.query_params['account'].lower()
74                 account = params['account']
75             if 'need_token' in request.query_params:
76                 params['need_token'] = request.query_params['need_token']
77
78             print("PARAM for eth.put_event_contract:", evt.func_id, params)
79             self.eth.put_event_contract(evt.func_id, account, params)
80

```

```
81         res = {}
82
83         return res
84
85
86     # イベント実行チェック
87     def check_event(self, request, check_point):
88         # リクエストと情報取り出し
89         method = request.method
90         path = request.url.path
91         user = request.client.host
92         print(method, path, user)
93
94         # 実行するイベントを探す
95         event_list = []
96         for evt in self.events:
97             if DEBUG : print(evt.id, evt.method, evt.url, method, path, check_point)
98             if check_point.lower() == 'pre' and evt.act_pre != True:
99                 continue
100             elif check_point.lower() == 'post' and evt.act_post != True:
101                 continue
102
103             if evt.method != '*' and evt.method != method:
104                 continue
105             if evt.url != '*' and path.startswith(evt.url) != True:
106                 continue
107
108             if DEBUG : print(evt.id, "in event list")
109
110             event_list.append(evt)
111
112         return event_list
113
114
115     # 処理前イベント
116     def pre_event(self, request):
117         if DEBUG : print("pre_event", request)
118
119         # 実行イベントを探す
120         event_list = self.check_event(request, 'pre')
```

```
121         if DEBUG : print(event_list)
122
123         # イベント実行
124         res = {}
125         for evt in event_list:
126             r = evt.call(evt, request, None)
127             if r : res[evt.id] = r
128
129         return res
130
131
132     # 処理後イベント
133     def post_event(self, request, response):
134         if DEBUG : print("post_event", request, response)
135
136         # 実行イベントを探す
137         event_list = self.check_event(request, 'post')
138         if DEBUG : print(event_list)
139
140         # イベント実行
141         res = {}
142         for evt in event_list:
143             r = evt.call(evt, request, response)
144             if r : res[evt.id] = r
145
146         return res
```