```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import sys
6  import os
7  import json
8  import datetime
9  import argparse
10
11 from sqlalchemy import Column
12 from sqlalchemy import ForeignKey
13 from sqlalchemy import Integer
14 from sqlalchemy import String
15 from sqlalchemy.orm import declarative_base
16 from sqlalchemy.orm import relationship
17 from sqlalchemy import create_engine
18
19 DEBUG = False # True
20
21 # 設定ファイルを読み込む
22 config_path = os.getenv('MYDATASERVER_LIB_CONFIG_PATH')
23 if config_path == None:
24     config_path = '.'
25 fp = open(config_path + '/config.json')
26 config = json.load(fp)
27 fp.close()
28 if DEBUG : print(config)
29
30 db_user = config['metadb']['USER']
31 db_pass = config['metadb']['PASS']
32 db_name = config['metadb']['DB']
33 db_host = config['metadb']['HOST']
34 db_port = config['metadb']['PORT']
35
36 # MySQL に接続
37 from sqlalchemy.engine.url import URL
38 db_url = URL.create(
39     drivername="mysql",
40     username=db_user,
```

```
41          password=db_pass,
42          host=db_host,
43          port=db_port,
44          database=db_name,
45          query = {"charset": 'utf8'},
46      )
47
48      engine = create_engine(db_url)
49
50
51      # データモデル
52      Base = declarative_base()
53      Base.metadata.bind = engine
54
55      ## channel
56      class DsChannel(Base):
57          __tablename__ = "ds_channel"
58          __table_args__ = {"autoload":True}
59
60      if DEBUG : print(DsChannel.__dict__)
61
62      ## InfluxDB
63      class DsInfluxDB(Base):
64          __tablename__ = "ds_influxdb"
65          __table_args__ = {"autoload":True}
66
67      if DEBUG : print(DsInfluxDB.__dict__)
68
69
70      ## Storage
71      class DsStorage(Base):
72          __tablename__ = "ds_storage"
73          __table_args__ = {"autoload":True}
74
75      if DEBUG : print(DsStorage.__dict__)
76
77      ## Ethereum
78      class DsEthereum(Base):
79          __tablename__ = "ds_ethereum"
80          __table_args__ = {"autoload":True}
```

```
81
82  if DEBUG : print(DsEthereum.__dict__)
83
84
85  ## function
86  class DsFunction(Base):
87      __tablename__ = "ds_function"
88      __table_args__ = {"autoload":True}
89
90  if DEBUG : print(DsFunction.__dict__)
91
92  ## event
93  class DsEvent(Base):
94      __tablename__ = "ds_event"
95      __table_args__ = {"autoload":True}
96
97  if DEBUG : print(DsEvent.__dict__)
98
99
100 class DB:
101     def __init__(self):
102         print("Constructor of DB")
103         # DB 接続
104         from sqlalchemy.orm import sessionmaker
105         SessionClass = sessionmaker(engine)
106         self.session = SessionClass()
107
108     def __del__(self):
109         print("Destructor of DB")
110         self.db_disconnect()
111
112     # DB 切断
113     def db_disconnect(self):
114         print("db_disconnect")
115         self.session.close()
116
117     # 全レコード取得
118     def list(self, tbl):
119         print("list", tbl)
120         r = self.session.query(tbl).all()
```

```
121             if DEBUG : print(r)
122             return r
123
124         # レコード追加
125         def insert(self, rec):
126             print("insert", rec)
127             try:
128                 self.session.add(rec)
129                 self.session.commit()
130             except Exception as e:
131                 print(e)
132             finally:
133                 pass
134
135         # レコード取得
136         def select(self, tbl, id):
137             print("select")
138             r   = self.session.query(tbl).filter(tbl.id == id).first()
139             if DEBUG : print(r)
140             return r
141
142         # レコード削除
143         def delete(self, rec):
144             print("delete", rec)
145             self.session.delete(rec)
146             self.session.commit()
147
148         # レコード更新
149         def update(self, rec):
150             print("update", rec)
151             self.session.commit()
152
153
```