Flask-SqlAlchemy

Flask SqlAlchemy 이란?

Flask SqlAlchemy

객체지향적 사고방식을 접목한 ORM(Object Relational Mapping) 방법

객체지향적으로 작성하는 모델들(Class) 들의 개념과
관계형 데이터베이스에 속하는 MySQL의 Relation 을 따로따로 보지않고 연결

```
from app import db

class User(db.Model):
__tablename__ = 'travel_user'
__table_args__ = {'mysql_collate': 'utf8_general_ci'}

user_id = db.Column(db.String(30), primary_key=True, unique=True)
user_name = db.Column(db.String(30))
profile_url = db.Column(db.String(100))
created = db.Column(db.DateTime)
```

Flask SqlAlchemy

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)

# app config
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://root:mysql-root@localhost/travel_mate?charset=utf8'
app.config['SQLALCHEMY_ECHO'] = True
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True
app.secret_key = 'manyrandombyte'

db = SQLAlchemy(app)

from app.models import *
from app.routes import *

db.create_all()
```

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///test.db'
```

해당 데이터베이스에 해당 테이블이 존재하면 넘어가고, 테이블이 없을 때 자동 생성해준다.
단, 객체 클래스를 바꾼다고 해서 기존 테이블이 Alter 되지 않는다.

Flask SqlAlchemy - select

```
def search_events_by_userid(user_id):
    """
    user_id 를 기준으로 event 찾기, all 로 찾음
    """
    return TravelEvent.query.filter_by(user_id=user_id).all()


def search_event_by_eventid(event_id):
    """
    event_id 를 기준으로 event 를 찾기. 한개만 리턴
    """
    return TravelEvent.query.filter_by(event_id=event_id).first()
```

TravelEvent 라는 모델 클래스를 이용해서 쿼리를 실행시킨다.
filter_by 가 where 문을 생성해주는 것이다.
여러조건일 경우 filter_by()안에 and_( ) 를 사용한다.
정렬 - order_by( )
limit – offset(), limit( )
모델클래스.query.filter_by().order_by().limit() 등

Flask SqlAlchemy - insert

```
def add_new_event(new_event_object, course_id):
new_event = TravelEvent(new_event_object['user_id'],
course_id,
new_event_object['title'],
new_event_object['description'],
new_event_object['max_tourist'],
new_event_object['start_time'],
new_event_object['end_time'],
new_event_object['event_end_time'],
new_event_object['hash_tag'])
db.session.add(new_event)
db.session.commit()
```

TravelEvent 라는 모델 클래스의 인스턴스를 생성
db.session.add() 함수에 해당 인스턴스를 넘긴뒤에
db.session.commit() 명령을 수행

Flask SqlAlchemy – update, delete

```
event = search_event_by_eventid(event_id)
current_tourist = event.current_tourist
if current_tourist + 1 <= event.max_tourist:
event.current_tourist=(event.current_tourist+1)
db.session.commit()
```

TravelEvent 라는 모델 클래스의 인스턴스중에서 특정 수정 대상이 되는 인스턴스를 찾는다
특정 인스턴스의 멤버변수 값을 수정한다.
db.session.commit() 명령을 수행

```
User.query.filter_by(id=123).delete()
db.session.commit()
```