[Python Flask] 웹 페이지 만들기 06 - MySQL 연동하기(DB 클래스 생성)

2019. 2. 27. 12:00

#class #Database #Example #flask #MySQL #pymysql #Python #데이터베이스 #예제 #클래스 #플라스크

이전 포스트에서는 MySQL을 연동하는 방법을 알아보았습니다.

하지만 이 방법으로는 나중에 추가적인 기능을 개발하고, 또 다시 연동하는 데 있어서 매우 어려움이 있습니다..

즉, 계속해서 데이터베이스를 다시 접근하고 닫고를 반복해야 하는 이유 때문입니다.

따라서, DB를 따로 새로운 클래스로 구분하여 이를 사용하는 방법이 필요합니다.

이번에는 MySQL을 이용하기 위한 클래스 생성 및 연동하는 방법을 알아보도록 하겠습니다.

다른 플라스크 예제는 아래의 링크를 살펴봐주시기 바랍니다~

[Python Flask] 웹 페이지 만들기 01 - 환경 설정

[Python Flask] 웹 페이지 만들기 02 - 기본 파일 작성하기

[Python Flask] 웹 페이지 만들기 03 - 파이썬에서 웹으로 값 전달

[Python Flask] 웹 페이지 만들기 04 - Bootstrap 연동하기

[Python Flask] 웹 페이지 만들기 05 - MySQL 연동하기(pymysql)

01. 개발 디렉토리 설정 변경

먼저 개발을 수행하기 앞서, 프로젝트 내부에 있는 디렉토리를 추가적으로 생성할 필요가 있습니다.

우리가 추가할 디렉토리는 다음과 같이 나타나 있습니다.

```
[Project Name]
 ㄴ[app]
    ㄴ[test]
       ㄴ테스트를 위한 폴더
    ㄴ[module]
       ㄴ모듈로써 사용할 python 소스
    ㄴ[schema]
       ㄴMySQL에서 실행할 SQL 소스
    ㄴ[static]
       ㄴ자바스크립트, CSS, 이미지 등...
    ㄴ[templates]
       ㄴHTML 파일들(폴더별로 정리 가능)
       ㄴ[test]
          ㄴ테스트를 위한 폴더
 ㄴconfig.py
 ㄴrun.py
```

기존의 디렉토리에 module이라는 폴더를 새로 생성하고 그 아래에 우리가 사용할 module로써 사용할 python 소스를 작성할 것입니다.

여기서 우리는 dbModule.py라는 파일을 생성합니다.

02. Database 클래스 모듈 파일 작성

이 파일은 새로운 기능을 생성할 때마다 import 시켜 사용될 파일입니다.

dbModule.py 파일은 다음과 같습니다.

```
# file name : dbModule.py
# pwd : /project_name/app/module/dbModule.py

import pymysql

class Database():
    def __init__(self):
        self.db = pymysql.connect(host='localhost',
                                  user='root',
                                  password='your_password',
                                  db='your_dbname',
                                  charset='utf8')
        self.cursor = self.db.cursor(pymysql.cursors.DictCursor)

    def execute(self, query, args={}):
        self.cursor.execute(query, args)

    def executeOne(self, query, args={}):
        self.cursor.execute(query, args)
        row = self.cursor.fetchone()
        return row

    def executeAll(self, query, args={}):
        self.cursor.execute(query, args)
        row = self.cursor.fetchall()
        return row

    def commit():
        self.db.commit()
```

원래는 Database 클래스를 선언하고, execute함수를 사용한 후 fetch함수와 commit() 함수 등으로 DB를 이용해야 했습니다.

하지만 위와 같은 클래스를 이용하면 보다 편리하게 사용할 수 있게 됩니다.

그러면 이제 이 클래스를 사용해보도록 하겠습니다.

03. Database 클래스 사용 예제

이 예제는 Database를 사용하기 위한 예제입니다.

여기서 제공되는 예제는 sql 소스, python 소스, HTML 소스가 있습니다.

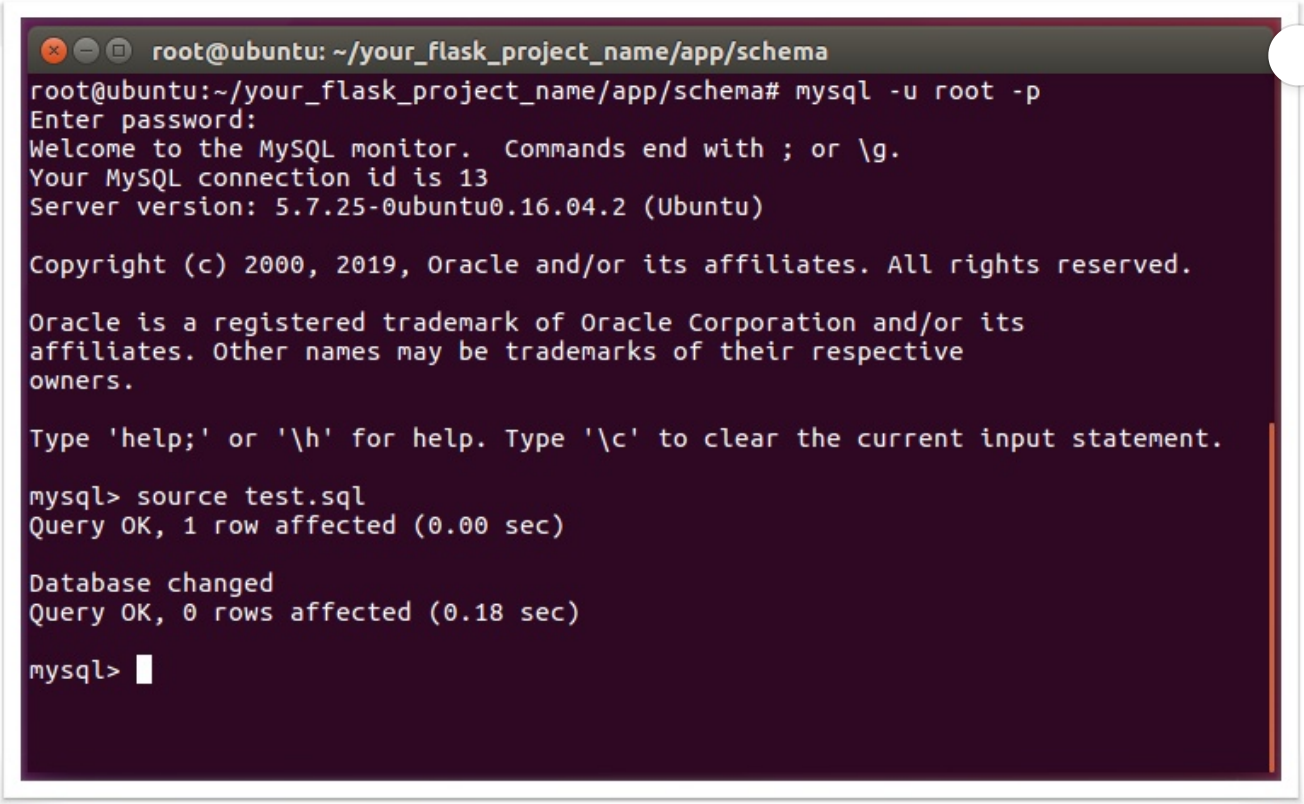
SQL 예제 소스

```sql
-- file name : test.sql
-- pwd : /project_name/app/schema/test.sql

CREATE DATABASE testDB default CHARACTER SET UTF8;

use testDB;

CREATE TABLE testTable(
    idx      INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    test     VARCHAR(256) NOT NULL
) CHARSET=utf8;
```

이제 이 예제를 이용하여 INSERT, SELECT, UPDATE를 모두 실습해보도록 합니다.

먼저 실습 파이썬 파일을 제작하기 전에 Ubuntu 16.04. 의 MySQL에 테이블을 생성하도록 합니다.

MySQL DB 생성 및 Table 생성은 다음 예제와 같습니다.



이제 DB 생성과 Table 생성이 모두 완료되었으니, INSERT 예제와 SELECT, UPDATE 예제를 살펴보시겠습니다.

HTML 소스 예제

먼저 HTML 소스를 통해 파이썬 소스를 실행할 버튼들을 만들어봅시다.

HTML 소스는 다음과 같습니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <h1> head Test </h1><br>
</head>

<body>

  <input type="button" value="INSERT 예제" onclick="location.href='/test
  <input type="button" value="SELECT 예제" onclick="location.href='/test
  <input type="button" value="UPDATE 예제" onclick="location.href='/test

  <br><br><br>
  <div>
      INSERT result : {{result}}
  </div>
  <br>
  <div>
```

```
        SELECT result idx  : {{resultData.idx}}<br>
        SELECT result test : {{resultData.test}}
    </div>
    <br>
    <div>
        UPDATE result idx  : {{resultUPDATE.idx}}<br>
        UPDATE result test : {{resultUPDATE.test}}
    </div>
    <br>

</body>
</html>
```

여기서 각각 INSERT 예제, SELECT 예제, UPDATE 예제를 실행시킬 버튼과 결과를 보여주는 영역을 작성했습니다.

이제 파이썬 소스를 보시겠습니다.

Python 소스 예제

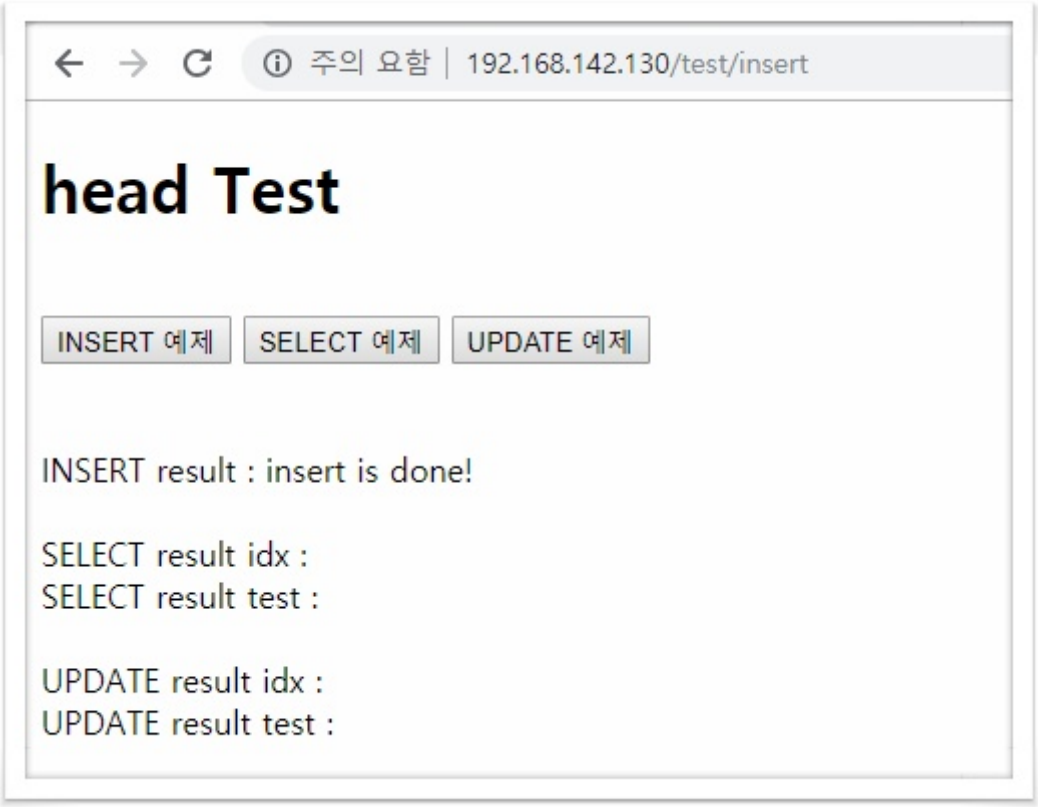
Python 소스는 위의 HTML 소스 예제와 마찬가지로 각각의 예제로 작성되었습니다.

```
# file name : test.py
# pwd : /project_name/app/test/test.py

from flask import Blueprint, request, render_template, flash, redirect,
from flask import current_app as current_app

from app.module import dbModule

test = Blueprint('test', __name__, url_prefix='/test')

@test.route('/', methods=['GET'])
def index():
    return render_template('/test/test.html',
                           result=None,
                           resultData=None,
                           resultUPDATE=None)



# INSERT 함수 예제
@test.route('/insert', methods=['GET'])
def insert():
    db_class = dbModule.Database()

    sql      = "INSERT INTO testDB.testTable(test) \
                VALUES('%s')" % ('testData')
    db_class.execute(sql)
    db_class.commit()

    return render_template('/test/test.html',
                           result='insert is done!',
```

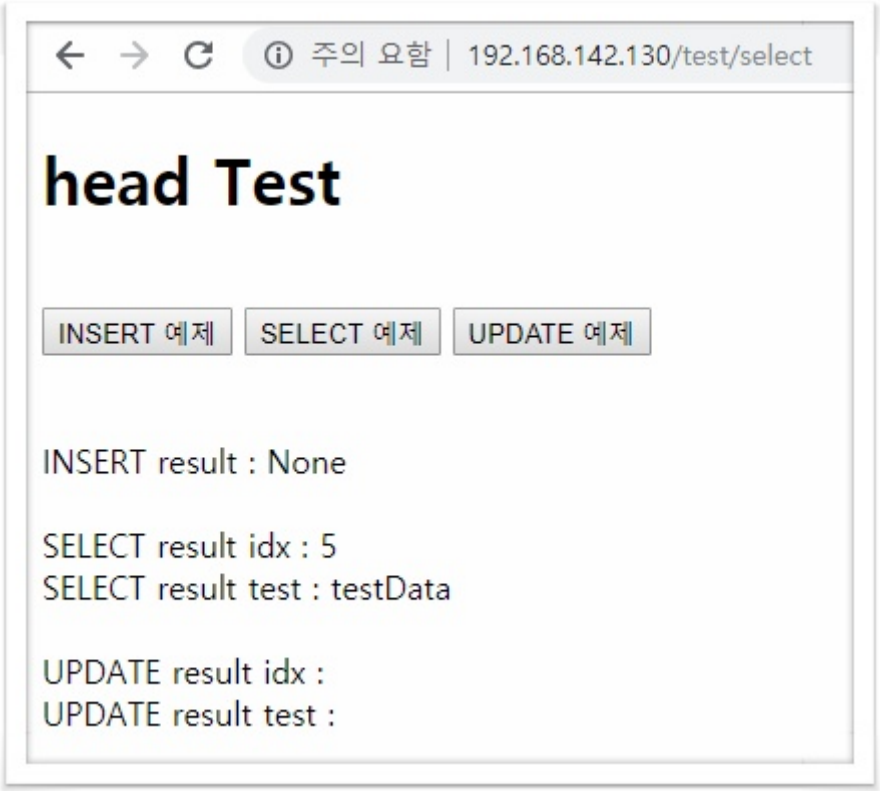
```
                                resultData=None,
                                resultUPDATE=None)



# SELECT 함수 예제
@test.route('/select', methods=['GET'])
def select():
    db_class = dbModule.Database()

    sql      = "SELECT idx, test \
                FROM testDB.testTable"
    row      = db_class.executeAll(sql)

    print(row)

    return render_template('/test/test.html',
                           result=None,
                           resultData=row[0],
                           resultUPDATE=None)



# UPDATE 함수 예제
@test.route('/update', methods=['GET'])
def update():
    db_class = dbModule.Database()

    sql      = "UPDATE testDB.testTable \
                SET test='%s' \
                WHERE test='testData'" % ('update_Data')
    db_class.execute(sql)
    db_class.commit()

    sql      = "SELECT idx, test \
                FROM testDB.testTable"
    row      = db_class.executeAll(sql)

    return render_template('/test/test.html',
                           result=None,
                           resultData=None,
                           resultUPDATE=row[0])
```

이제 이 파일을 실행시키게 되면 다음과 같은 모습으로 나타나게 됩니다.

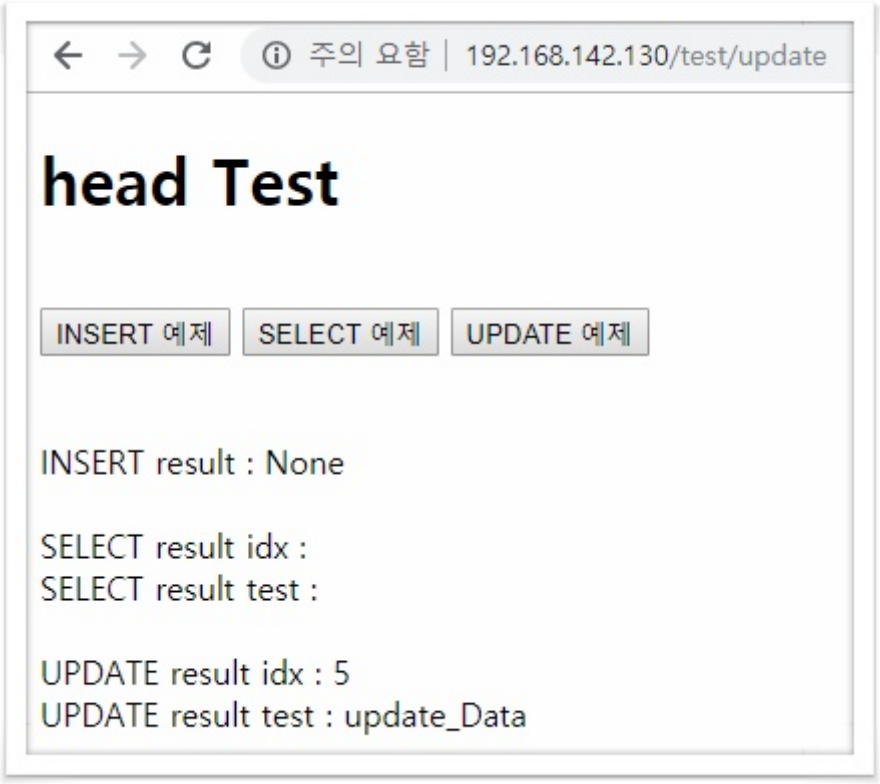
INSERT 예제 수행 결과



SELECT 예제 수행 결과



UPDATE 예제 수행 결과



2 구독하기

'PYTHON > Python Flask' 카테고리의 다른 글

[Python Flask] 웹 페이지 만들기 06 - MySQL 연동하기(DB 클래스 생성) (17)	2019.02.27
[Python Flask] 웹 페이지 만들기 05 - MySQL 연동하기(pymysql) (2)	2019.02.27
[Python Flask] 웹 페이지 만들기 04 - Bootstrap 연동하기 (1)	2019.02.26
[Python Flask] 웹 페이지 만들기 03 - 파이썬에서 웹으로 값 전달 (2)	2019.02.26
[Python Flask] 웹 페이지 만들기 02 - 기본 파일 작성하기 (39)	2019.02.25
[Python Flask] 웹 페이지 만들기 01 - 환경 설정 (1)	2019.02.20

NAME

PASSWORD

HOMEPAGE

http://

SECRE WRITE

yun 2019.07.05 09:45

포스트를 보다가 궁금한 점이 생겨서 댓글을 작성합니다. mysql을 매 뷰함수에다 연결을 하면 규모가 큰 Flask 웹 앱에서는 느려지지 않을까요..?

Delete Reply


깜이군 kkamikoon 2019.07.05 09:48 신고

흥미로운 댓글 감사합니당ㅎㅎ 한번 연구해봐야겠어요.

리서치 후 찾은 내용 있으면 추가 답글 달겠습니다~

Delete


깜이군 kkamikoon 2019.07.05 09:55 신고

혹시 DB 접근이라 함은 Class 생성 때문에 느려지는 걸 생각하신 건지, 아니면 MySQL에 계속 접근을 새로 하기 때문에 느려지는 걸 생각하신 건가요?

Delete

yun 2019.07.16 13:54

지금 댓글을 봤네요.. ㅎㅎ.. 저는 후자에 대한 의문점이 있어서 질문을 했습니다.

Delete

yun 2019.07.16 13:56

매 뷰함수마다 mysql 계속 접근(연결)을 할 때마다 느려지지 않을까..? 라는 의문점이 들어서 저 같은 경우에는 서버가 시작할 때마다 전역으로 연결을 시키고 시작하거든요..

Delete

---

kimyc 2019.08.04 20:11

혹시 낫파운드 에러가 떴을땐 어떻게 해결할 수 있나요?ㅠ

Delete Reply

 깜이군 kkamikoon 2019.08.04 20:14 신고

app 폴더 아래에 있는 __init__.py 에 해당 파이썬이 임포트 돼 있는지 보셔야 할 거 같네요

Delete

kimyc 2019.08.04 20:37

그.. 해결은했는데 인터널 에러(500)가 뜹니다. 혹시 TemplateNotFound(template) 에러가 뜨는데 혹시 test.html파일 위치와 관련이 있나요? 지금은 app/test 디렉토리에 위치해있습니다..

Delete

kimyc 2019.08.04 20:39

아 template/test 디렉토리를 만들고 거기에 넣으니 해결됐네요 감사합니다!

Delete

kimyc 2019.08.04 20:46

음.. 대신 인터널에러가 뜨는데 db모듈안 Database파일에 db 값에 무슨 값을 입력해야 하나요..? 저기엔 your_dbname이라고 되어있는데 뭘 넣어야 할지모르겠네요

Delete

 깜이군 kkamikoon 2019.08.04 20:51 신고

your_dbname에는 님께서 mysql에 만든 디비 이름을 넣어주시면 됩니다. 인터널 에러는 너무 다양한 가지 수가 있어서 파일위치, 선언 등의 에러가 있을 수 있습니다. 해당 에러는 아마 run.py 수행하실 때 해당

파일 안에 debug=True 옵션을 추가해주시고 실행해보시길 바랍니다

Delete

kimyc *2019.08.04 21:23*

아 이것도 해결됐습니다. testdb를 제가 언제 만든건지 모르겠는데 찾아보니 있더라고요. 여기서 또 테이블명이 testtalbe로 되있어서 실패했던거같습니다. 글 유익하게 잘 읽었습니다!

Delete

---

kimyc *2019.08.04 20:31*

혹시 낫파운드 에러가 뜰땐 어떻게 해야할까요..?

Delete Reply

01082539198 *2019.11.05 13:53*

안녕하세요 더이상 글을 올리지 않으시내요 잘 계시나요 ?무엇을 하고지내시나요

저는 파이썬으로 아동복 쇼핑몰을 만들어볼까하는데 ..

Delete Reply

 깜이군 kkamikoon *2019.11.05 13:54 신고*

이제 다시 쓰려구요ㅎㅅㅎ 꽤 크실 거 같은데... 개인이 운영하시는 건가요?

Delete

---

danimothman *2019.12.04 09:36*

위의 내용대로 진행해서 insert예제 버튼을 클릭했더니 (TypeError: commit() takes 0 positional arguments but 1 was given)오류가 나타나서 많이 해맸습니다.
맞는건진 모르겠지만 "dbModule.py" 파일에서 밑에서 두번째 줄에 [def commit():] -> [def commit(self):] 로 변경하니 되는군요;

참조레퍼런스 : https://wikidocs.net/1742

Delete Reply

danimothman *2019.12.04 09:42*

참고로 작업환경은
OS환경 : 우분투 16.04 server 버전
파이썬 : python3.6
편집기 : pycharm(Community)
DB : MySQL
입니다.

 # Delete

PREV 1 ··· 111 112 113 114 115 116 117 118 119 ··· 255 NEXT

+ Recent posts

2019.12.03. Webhacking.kr

[Vue.js] Vue-devtools 확장···

[Vue.js] package-lock.json…

[Vue.js] Vue Project 외부…