**Django framework 만들기**

[**https://docs.djangoproject.com/ko/3.1/intro/tutorial02/**](https://docs.djangoproject.com/ko/3.1/intro/tutorial02/)

<Ubuntu prompt - MySQL 접속>

Sudo apt update

Sudo apt upgrade

Sudo service mysql start

Sudo service mysql status (확인)

<MySQL Workbench 연결 확인>

use tpch\_db; (ctrl+enter)

select \* from LINEITEM limit 10; (ctrl+enter)

<MySQL 연결 안될때>

ps -ef | grep mysql

ifconfig

mysql=>

edit connection : localhost->inet 172.22.118.45

<MySQL Workbench DB 만들기>

Ubuntu에서

Sudo mysql

create database django\_db default character set utf8;

show databases; ----> django\_db

권한 주기 :

grant all privileges on django\_db.\* to 'user1'@'%';

grant all privileges on django\_db.\* to 'user1'@'localhost';

flush privileges;

<MySQL Workbench connection test>

Schemas ‘Refresh’ => django\_db

<user 확인>

mysql> show users;

mysql> select user, host from user;

(USER NAME : user1 PSWD : 1234)

<E 에 django\_pycharm 폴더 만들기 >

<Anaconda>

conda create -n django-env python=3.7

conda env list

Conda activate django-env

<Pycharm 프로젝트 만들기>

1)가상환경을 물고 프로젝트 만들기

New project : Django\_pycharm₩BoardProject

Previously config interpreter - Add python interpreter - conda environment - interpreter - django-env

2)Conda activate django-env --> Conda install django

OR Pychamr - File-setting-project(board project)-python interpreter- (+ )- django(검색) + add package (install)

3)terminal : (django-env) E:\django\_pycharm\BoardProject> 되있어야

<django 프로젝트 만들기>

프로젝트 생성

Pycharmterminal : (django-env) E:\django\_pycharm\BoardProject>django-admin startproject board

Project - reload -> board(루트)-board(진짜 프로젝트) 생겼음

이름바꿔주기

refactor-rename-boardRoot (unclick 2개다)

\*장고 = 1 project = app + ()app + app

(django-env) E:\django\_pycharm\BoardProject>cd boardRoot

게시판용app생성

(django-env) E:\django\_pycharm\BoardProject\boardRoot>

python manage.py startapp noticeboard --> noticeboard 생겼음

**Board-settings.py**

(Debug true개발환경, ALLOWED\_HOSTS = [‘\*’]<---> Debug false 배포환경, ALLOWED\_HOSTS = [지정] )

ALLOWED\_HOSTS = [ ] -->localhost상태 (default)

INSTALLED\_APPS = ['noticeboard.apps.NoticeboardConfig',]

( File-setting-project(board project)-python interpreter- (+ )-mysqlclient (install) )

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'django\_db',

'USER': 'user1',

'PASSWORD': '1234',

'HOST': '127.0.0.1',

'PORT' : '3306',

}

}

TIME\_ZONE = 'Asia/Seoul'

USE\_TZ = False

(django-env) E:\django\_pycharm\BoardProject\boardRoot>

python manage.py migrate (mysql로 집어넣기)

==> The migrate command looks at the settings.py file INSTALLED\_APPS setting and creates any necessary database tables according to the database settings in your mysite/settings.py file

===> error (django & mysqlclient - version맞지 않음)

-----------------------------------------------------------------------------------------------------------------------------------------------------

<가상환경 새로 만들기-django 3 ver 직접 설치.>

conda create -n django3 python=3.7

conda activate django3

conda install django=3

New project : Django\_pycharm₩BoardProject2₩BoardProject

(django3) E:\django\_pycharm\BoardProject2\BoardProject>django-admin startproject board

(board) refactor-rename-boardRoot (unclick 2개다)

(django-env) E:\django\_pycharm\BoardProject>cd boardRoot

python manage.py startapp noticeboard

**Board-settings.py**

INSTALLED\_APPS = ['noticeboard.apps.NoticeboardConfig',]

TIME\_ZONE = 'Asia/Seoul'

USE\_TZ = False

( File-setting-project(board project)-python interpreter- (+ )-pymysql (install) )

Import pymysql

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'django\_db',

'USER': 'user1',

'PASSWORD': '1234',

'HOST': '127.0.0.1', /(에러시) ‘localhost’ / inet 172.22.118.45

'PORT': '3306',

}

}

pymysql.version\_info = (1, 4, 2, "final", 0)

pymysql.install\_as\_MySQLdb()

(django3) E:\django\_pycharm\BoardProject\boardRoot>

python manage.py migrate

마이그레이션은 매우 기능이 강력하여, 마치 프로젝트를 개발할 때처럼 데이터베이스나 테이블에 손대지 않고도

모델의 반복적인 변경을 가능하게 해줍니다. 동작 중인 데이터베이스를 자료 손실 없이 업그레이드 하는 데 최적화

-----------------------------------------------------------------------------------------------------------------------

python manage.py runserver

(에러시)**settings.py** 'HOST': ‘localhost’/ inet 172.22.118.45

<http://127.0.0.1:8000/> 클릭

<http://127.0.0.1:8000/admin>

\*포트 토워딩

windows : localhost 3306 (치면)

-----> ubuntu : 172 xxxxx (ifconfig : inet 맨날 바뀜)

port\_fw.ps1

파워쉘 관리자 권한으로 열기

cd e:

PowerShell.exe -ExecutionPolicy Bypass -File .\port\_fw.ps1

**settings.py** 'HOST': ‘localhost’/

---------------------------------------------------------------------------------------------------------------

**<관리 사이트에 로그인 할 수 있는 사용자를 생성>**

Ctrl + c (중지?)

(django3) E:\django\_pycharm\BoardProject2\BoardProject\boardRoot>

python manage.py createsuperuser

Username : user2

Password : 1234

python manage.py runserver

<http://127.0.0.1:8000/admin/> (로그인)

---------------------------------------------------------------------------------------------------------------

**ORM(Obj Relational Mapping)** = django default

Database 내 데이터를 (객체를 쓰는 것처럼) 추가,수정

**database table 추가 (게시판 만들기)**

**1.< models.py>:**

from django.db import models

from django.utils.timezone import now

class Notice(models.Model):

title = models.CharField(max\_length=1000)

content = models.CharField(max\_length=2000)

writeDate = models.DateTimeField(default=now, editable=False)

writeID = models.CharField(max\_length=50)

def \_\_str\_\_(self):

return self.title

**2.<admin .py>**

from django.contrib import admin

from noticeboard.models import Notice

(File-settings-project structure- board Root우클릭source)

admin.site.register(Notice)

1. **python manage.py makemigrations**

**4.python manage.py migrate**

**5.<mysql benchwork>**

use django\_db;

desc noticeboard\_notice; (ctrl+enter)

**다시 만들어야 할때 (꼬여서)**

**1.Noticeboard-migrations-0001\_initial 파일들 다 지워 (init빼고)**

**(mysql benchwork 에서)**

**2.Drop tables**

**3.select \* from django\_migrations; ==> noticeboard 날리기**

**delete from django\_migrations where app = ‘noticeboard’;**

python manage.py runserver ==> Notices 뜸

<http://127.0.0.1:8000/noticeboard/write/>

<Sample data 넣기>

Add + save (게시글 작성)

Mysql benchwork

select \* from noticeboard\_notice; (ctrl+enter) ==> 작성한 글 나옴

<client 요청?>

**board <urls.py>**

import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('noticeboard/', include('noticeboard.urls')),

]

**Noticeboard <urls.py>**

from django.urls import path

from . import views

app\_name = 'noticeboard'

urlpatterns = [  
 path('', views.index, name='index'),  
 path('write/', views.write\_article, name='write\_article'),  
 #path('add', views.add\_article, name='add\_article'),  
]

**<Settings.py>**

Templates =

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

New dir 만들기

Templates만들기-noticebaord만들기-index.html만들기

**<Views.py>**

-writeDate : (descending sort)

{‘key값’, value}

notice.save()   
#django OEM , data 저장

django {% %} = python (reverse())

def view\_article(request, article\_id):  
 notice = get\_object\_or\_404(Notice, pk=article\_id)

==> 있으면 page 보여주고, 없으면 404 보여줌

**<Index.html>**

a class="cls1" href="{% url 'noticeboard:write\_article' %}">

app\_name = 'noticeboard'

path(', name='write\_article'),

<textarea name....></textarea> = <text area name .... /> 같은거

**<writeArticle.html>**

- body- form

action="{% url 'noticeboard:add\_article' %}"

----------------------------------------------------------------------------------------------------------------------------------------

<http://127.0.0.1:8000/noticeboard/> (클릭)

[http://127.0.0.1:8000/noticeboard/ HYPERLINK "http://127.0.0.1:8000/noticeboard/write/"write/](http://127.0.0.1:8000/noticeboard/write/)

----------------------------------------------------------------------------------------------------------------------------------------

**redirect(일관성 있는 주소) vs forward(주소가 바꿈)**

client ------/write ----> web server

<----submit------

<----/noticeboard------ \*redirect\*

------------------->

------------------------------------------------------------------------------------------------------

**REST API**

URL :

URI (Uniform Interface) : 유닉하게 식별할 수 있으면 , direct하게 지정

-DELETE /members/1

---------------------------------------------------------------------------------------------------

fn enable 숨겨놓은거 보이게 ,

제목 내용 고칠수 있게 disabled 취소 시키기

---------------------------------------------------------------------------------------------------

**delete from noticeboard\_notice where id='13';**