

Django 웹 프레임워크

- To-Do-List

Mini-Project

장고 프로젝트

◦ ToDoList

To-Do

← → ↻ ⓘ 127.0.0.1:8000

To-do List with Django

메모할 내용을 적어주세요 메모하기!

오늘의 할일 완료

내일의 할일 완료

데이터엔지니어링 완료

데이터사이언스 완료

장고에서의 애플리케이션 개발 방식

MVT 코딩 순서

- **프로젝트 뼈대 만들기** : 프로젝트 및 앱 개발에 필요한 디렉토리와 파일 생성
- **모델 코딩하기** : 테이블 관련 사항을 개발(models.py, admin.py 파일)
- **URLconf 코딩하기** : URL 및 뷰 매핑 관계를 정의(urls.py 파일)
- **템플릿 코딩하기** : 화면 UI 개발(templates/ 디렉토리 하위의 *.html 파일들)
- **뷰 코딩하기** : 애플리케이션 로직 개발(views.py 파일)

프로젝트 뼈대 만들기

프로젝트 뼈대 만들기 순서 명령

- > `django-admin startproject ToDoList` // ToDoList라는 프로젝트를 생성
- > `python manage.py startapp my_to_do_app` // my_to_do_app라는 애플리케이션을 생성
- > `notepad settings.py` //설정 파일을 확인 및 수정
- > `python manage.py migrate` //데이터베이스에 기본 테이블을 생성
- > `python manage.py runserver` //현재까지 작업을 개발용 웹 서버로 확인

프로젝트 뼈대 만들기

프로젝트 생성

- ToDoList라는 프로젝트 만듦
- (base) C:\MyTest>django-admin startproject ToDoList

```
(base) C:\MyTest>django-admin startproject ToDoList  
(base) C:\MyTest>
```

프로젝트 뼈대 만들기

프로젝트 생성

◦ ToDoList 구성

```
(base) C:\MyTest>cd ToDoList

(base) C:\MyTest\ToDoList>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest\ToDoList 디렉터리

2021-08-25 오후 07:02 <DIR> .
2021-08-25 오후 07:02 <DIR> ..
2021-08-25 오후 07:02      686 manage.py
2021-08-25 오후 07:02 <DIR>   ToDoList
                1개 파일              686 바이트
                3개 디렉터리 18,365,161,472 바이트 남음
```

프로젝트 뼈대 만들기

애플리케이션 생성

- 프로젝트 루트 디렉토리 ToDoList으로 이동해서 my_to_do_app라는 애플리케이션을 만드는 명령을 실행
- (base) C:\WMyTest>cd ToDoList
- (base) C:\WMyTest\ToDoList>python manage.py startapp my_to_do_app

```
(base) C:\WMyTest\ToDoList>python manage.py startapp my_to_do_app  
(base) C:\WMyTest\ToDoList>
```

프로젝트 뼈대 만들기

애플리케이션 생성

- my_to_do_app라는 애플리케이션 폴더 확인
- (base) C:\MyTest\ToDoList\my_to_do_app>dir

C:\MyTest\ToDoList 디렉터리

```
2021-08-25 오후 07:04 <DIR> .
2021-08-25 오후 07:04 <DIR> ..
2021-08-25 오후 07:02      686 manage.py
2021-08-25 오후 07:04 <DIR> my_to_do_app
2021-08-25 오후 07:04 <DIR> ToDoList
                1개 파일      686 바이트
                4개 디렉터리 18,368,880,640 바이트 남음
```

C:\MyTest\ToDoList\my_to_do_app 디렉터리

```
2021-08-25 오후 07:04 <DIR> .
2021-08-25 오후 07:04 <DIR> ..
2021-08-25 오후 07:04      66 admin.py
2021-08-25 오후 07:04     159 apps.py
2021-08-25 오후 07:04 <DIR> migrations
2021-08-25 오후 07:04      60 models.py
2021-08-25 오후 07:04      63 tests.py
2021-08-25 오후 07:04      66 views.py
2021-08-25 오후 07:04       0 __init__.py
                6개 파일      414 바이트
                3개 디렉터리 18,368,434,176 바이트 남음
```


프로젝트 뼈대 만들기

프로젝트 설정 파일 변경

- 프로젝트에 필요한 설정 값들은 **settings.py** 파일에 지정
- settings.py 파일은 프로젝트의 전반적인 사항들을 설정해주는 곳으로, 루트 디렉토리를 포함한 각종 디렉토리의 위치, 로그의 형식, 프로젝트에 포함된 애플리케이션의 이름 등이 지정되어 있음

```
(base) C:\MyTest\ToDoList>cd ToDoList
```

```
(base) C:\MyTest\ToDoList\ToDoList>notepad settings.py
```

프로젝트 뼈대 만들기

프로젝트 설정 파일 변경

- ALLOWED_HOSTS 항목을 적절하게 지정
 - `ALLOWED_HOSTS = ['192.168.35.61', 'localhost', '127.0.0.1']`
- 애플리케이션들은 모두 설정 파일에 등록
(my_to_do_app 애플리케이션도 등록)
 - `INSTALLED_APPS = [~, ' my_to_do_app ',]`
- 프로젝트에 사용할 데이터베이스 엔진
 - 장고는 디폴트로 SQLite3 데이터베이스 엔진을 사용하도록 설정
- 타임존 지정(기본은 세계표준시(UTC)로 되어 있음. 한국시간을 변경)
 - `#TIME_ZONE = 'UTC'`
 - `TIME_ZONE = 'Asia/Seoul'`

프로젝트 뼈대 만들기

기본 테이블 생성

- migrate 명령은 데이터베이스에 변경사항이 있을 때 반영해주는 명령
- (base) C:\MyTest\ToDoList>python manage.py migrate

```
(base) C:\MyTest\ToDoList>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

프로젝트 뼈대 만들기

기본 테이블 생성

- o (base) C:\MyTest\ToDoList>dir

```
(base) C:\MyTest\ToDoList>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest\ToDoList 디렉터리

2021-08-25 오후 07:21 <DIR> .
2021-08-25 오후 07:21 <DIR> ..
2021-08-25 오후 07:21      131,072 db.sqlite3
2021-08-25 오후 07:02       686 manage.py
2021-08-25 오후 07:21 <DIR> my_to_do_app
2021-08-25 오후 07:04 <DIR> ToDoList
                2개 파일              131,758 바이트
                4개 디렉터리  18,366,414,848 바이트 남음
```

프로젝트 뼈대 만들기

작업 확인하기

- 프로젝트의 뼈대에 해당하는 프로젝트 디렉토리, 애플리케이션 디렉토리를 비롯해 관련 파일들 그리고 사용자 및 그룹 테이블을 만들었음
- 확인을 위해서 웹 서버를 실행하고, 그 웹 서버에 접속
- 장고에서는 개발 과정 도중에 현재 상태를 확인해볼 수 있도록 **runserver**라고 하는 간단한 테스트용 웹 서버를 제공

프로젝트 뼈대 만들기

작업 확인하기

- 웹 서버를 실행하기 위해 하나의 창에서 작업해도 되지만 runserver 용으로 별도의 cmd창을 열어 사용하는 것이 편리
- (base) C:\MyTest\ToDoList>python manage.py runserver 127.0.0.1:8000
- (base) C:\MyTest\ToDoList >python manage.py runserver
- (base) C:\MyTest\ToDoList >python manage.py runserver 0:8000

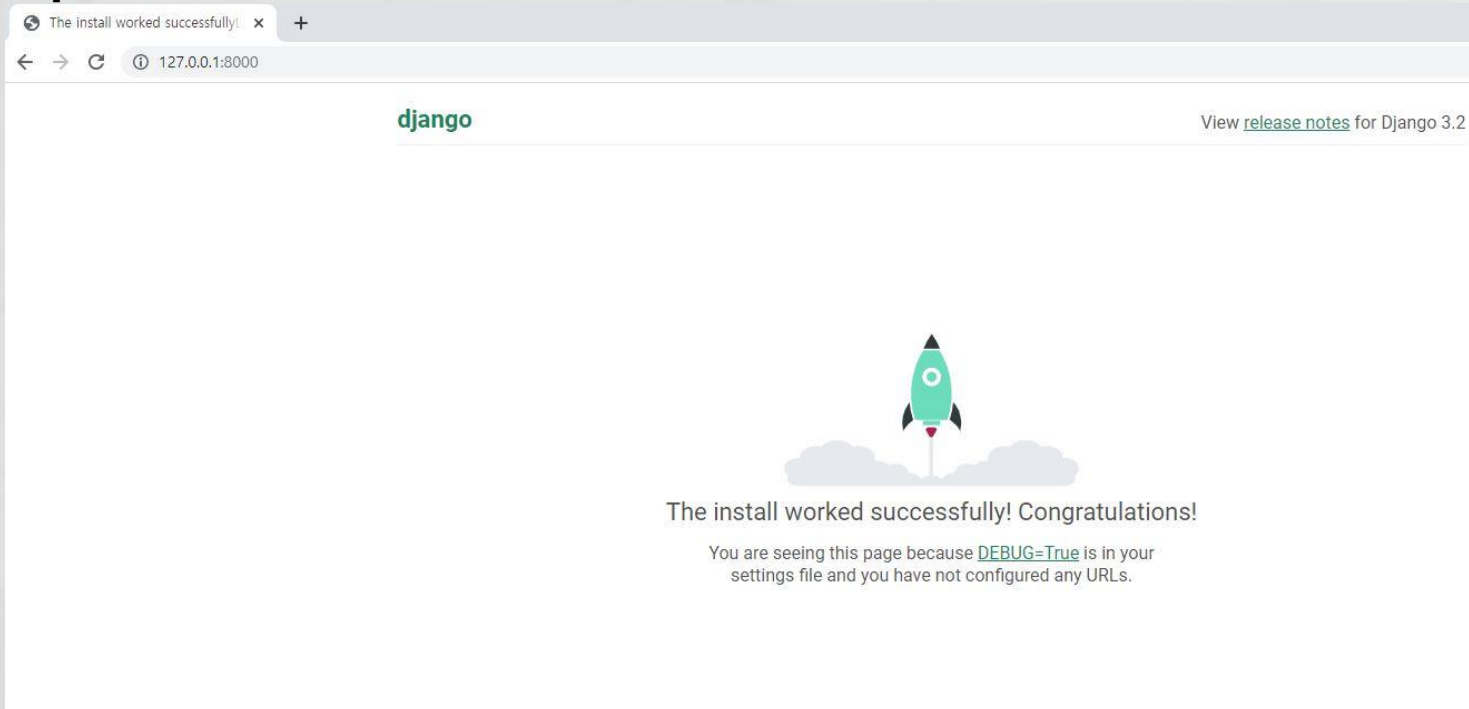
```
(base) C:\MyTest\ToDoList>python manage.py runserver 127.0.0.1:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 25, 2021 - 19:25:32
Django version 3.2.6, using settings 'ToDoList.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

프로젝트 뼈대 만들기

작업 확인하기

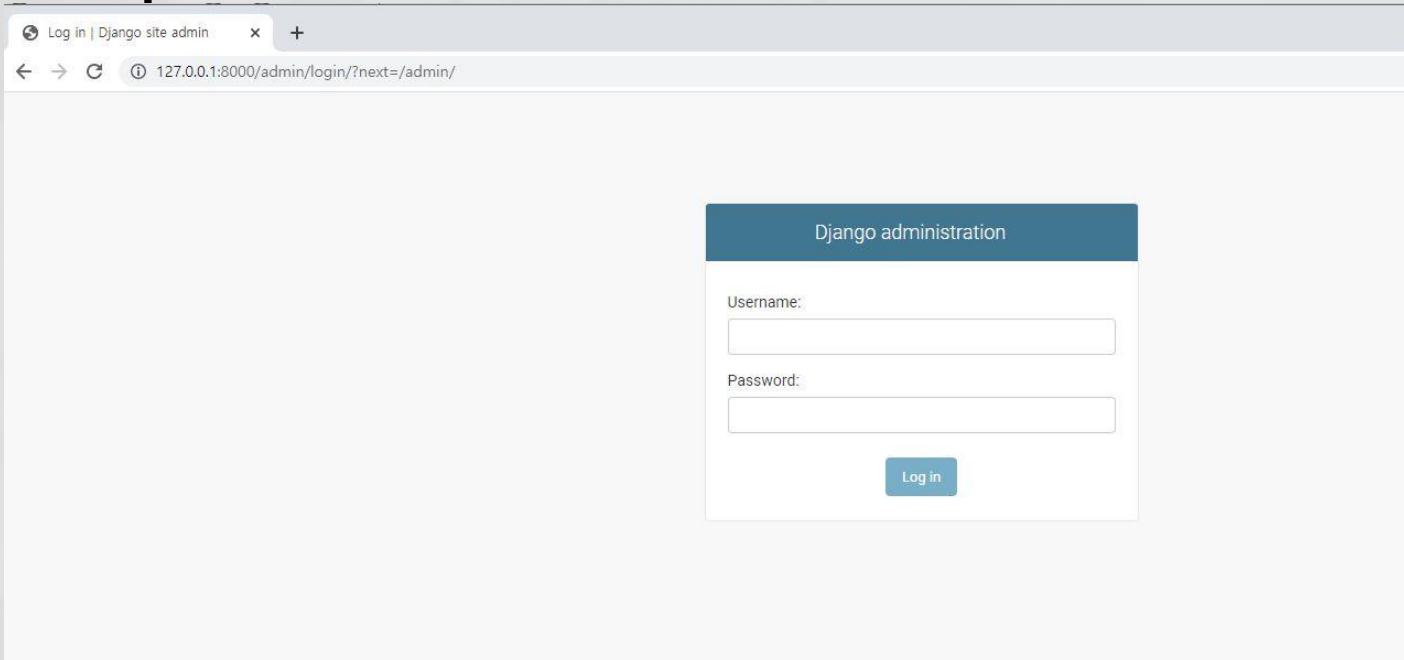
- runserver가 정상적으로 실행되었다면, 웹 브라우저를 열고 IP 주소는 runserver가 동작하는 서버 IP주소를 입력
- `http://127.0.0.1:8000/`



프로젝트 뼈대 만들기

Admin 사이트 접속

- 기본적으로 제공하는 Admin 사이트에 접속해서 테이블이 생성 확인
- 브라우저의 주소창에 IP 주소와 포트번호 동일, URL경로만 /admin 추가
- `http://127.0.0.1:8000/admin`



프로젝트 뼈대 만들기

Admin 사이트 접속

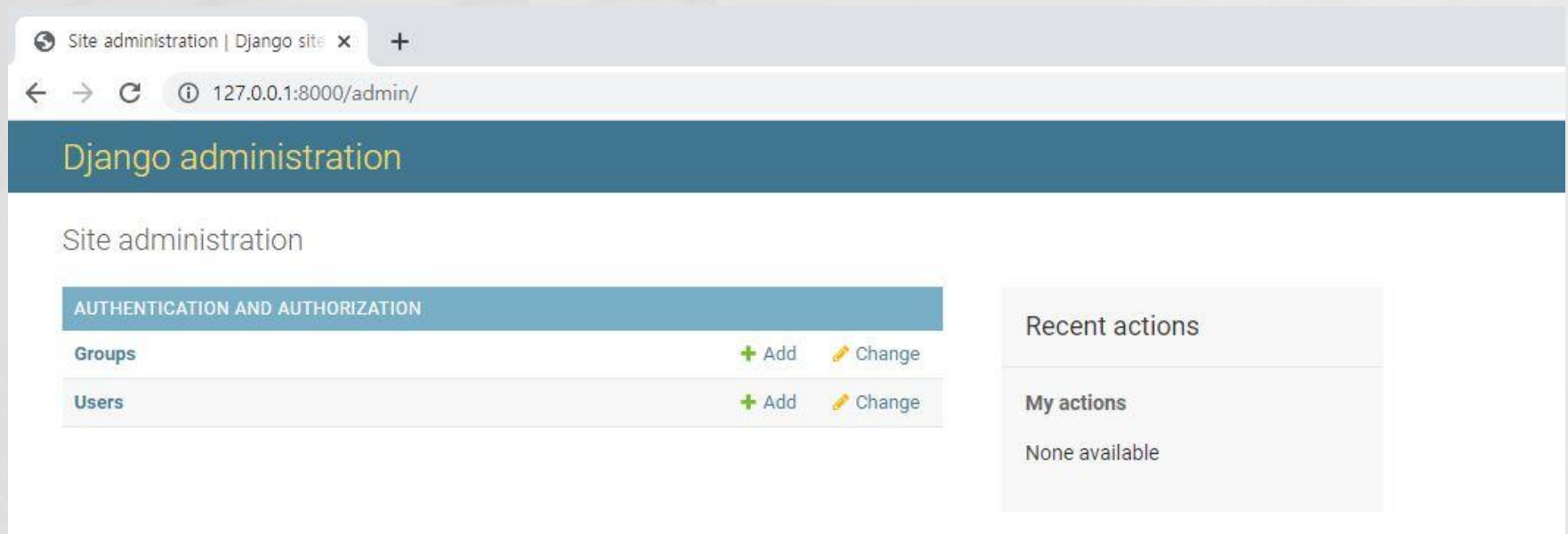
- 로그인하려면 username, password를 넣어야 되는데 아직 생성하지 않았음
- Admin 사이트에 로그인하기 위한 관리자(슈퍼유저)를 생성
- (base) C:\MyTest\ToDoList>python manage.py createsuperuser

```
(base) C:\MyTest\ToDoList>python manage.py createsuperuser
Username (leave blank to use 'bbobbee'): smcpacs
Email address: smcpacs@naver.com
Password:
Password (again):
Superuser created successfully.
```

프로젝트 뼈대 만들기

Admin 사이트 접속

- `http://127.0.0.1:8000/admin`
- 로그인 창에서 `username`, `password` 입력



프로젝트 뼈대 만들기

Admin 사이트 접속

○ 사용자 확인

Select user to change | Django

127.0.0.1:8000/admin/auth/user/

Django administration

Home > Authentication and Authorization > Users

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

Select user to change

Search

Action: [-----] Go 0 of 1 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	smcpacs	smcpacs@naver.com			✓

1 user

프로젝트 뼈대 만들기

Admin 사이트 접속

- Admin 사이트에서 Users와 Groups 테이블을 포함하여 새롭게 만들 테이블에 대한 데이터의 입력, 변경, 삭제 등의 작업을 할 수 있음
- Admin 화면에서 기본적으로 Users와 Groups 테이블이 보이는 것은 이미 settings.py 파일에 django.contrib.auth 애플리케이션이 등록되어 있기 때문
- 즉 장고에서 기본으로 제공하는 auth 앱에 Users와 Groups 테이블이 미리 정의

프로젝트 뼈대 만들기

골격 생성

- 애플리케이션을 MVT 패턴 방식으로 개발할 수 있도록 골격 생성
- (base) C:\MyTest>tree /F ToDoList

```
(base) C:\MyTest>tree /F ToDoList
폴더 PATH의 목록입니다.
블록 일련 번호가 000000C6 421A:9EBA입니다.
C:\MYTEST\ToDoList
db.sqlite3
manage.py

my_to_do_app
  admin.py
  apps.py
  models.py
  tests.py
  views.py
  __init__.py

migrations
  __init__.py
  __pycache__
    __init__.cpython-38.pyc

__pycache__
  admin.cpython-38.pyc
  apps.cpython-38.pyc
  models.cpython-38.pyc
  __init__.cpython-38.pyc

ToDoList
  asgi.py
  settings.py
  urls.py
  wsgi.py
  __init__.py

__pycache__
  settings.cpython-38.pyc
  urls.cpython-38.pyc
  wsgi.cpython-38.pyc
  __init__.cpython-38.pyc
```

애플리케이션 개발하기 – Model 코딩

Model 코딩

- 모델 작업은 데이터베이스에 테이블을 생성하는 작업
- `>notepad models.py` // 테이블을 정의함
- `>notepad admins.py` // 정의된 테이블이 Admin 화면에 보이게 함
- `>python manage.py makemigrations` // 데이터베이스에 변경이 필요한 사항을 추출함
- `>python manage.py migrate` // 데이터베이스에 변경사항을 반영함
- `>python manage.py runserver` // 현재까지 작업을 개발용 웹 서버로 확인함

애플리케이션 개발하기 – Model 코딩

테이블 정의

- my_to_do_app 애플리케이션은 Todo 테이블이 필요
- 테이블은 **models.py** 파일에 정의

테이블 컬러명	컬럼 타입
Content	Varchar(255)
isDone	Boolean

애플리케이션 개발하기 – Model 코딩

테이블 정의

- 테이블은 **models.py** 파일에 정의

```
from django.db import models

# Create your models here.
class Todo(models.Model):
    content = models.CharField(max_length = 255)
    isDone = models.BooleanField(default=False) #완료버튼에 사용
```


애플리케이션 개발하기 – Model 코딩

Admin 사이트에 테이블 반영

- models.py 파일에서 정의한 테이블도 admin 사이트에 보이도록 등록
- admin.py 파일에 등록

```
from django.contrib import admin

# Register your models here.
from my_to_do_app.models import Todo

admin.site.register(Todo)
```

- models.py 모듈에서 정의한 toDo클래스를 임포트하고 , admin.site.register() 함수를 사용하여 임포트 한 클래스를 Admin 사이트에 등록

애플리케이션 개발하기 – Model 코딩

데이터베이스 변경사항 반영

- 테이블의 신규 생성, 테이블의 정의 변경 등 데이터베이스에 변경이 필요한 사항이 있으면, 이를 데이터베이스에 실제로 반영해주는 작업
- 변경사항을 데이터베이스에 반영
- (base) C:\MyTest\ToDoList>python manage.py makemigrations

```
(base) C:\MyTest\ToDoList>python manage.py makemigrations
Migrations for 'my_to_do_app':
  my_to_do_app\migrations\0001_initial.py
    - Create model Todo
```

애플리케이션 개발하기 – Model 코딩

데이터베이스 변경사항 반영

- makemigrations 명령에 의해 my_to_do_app/migrations 디렉토리 하위에 마이그레이션 파일들이 생김

```
C:\MyTest\ToDoList\my_to_do_app 디렉토리
2021-08-25 오후 07:21 <DIR> .
2021-08-25 오후 07:21 <DIR> ..
2021-08-25 오후 07:54 133 admin.py
2021-08-25 오후 07:04 159 apps.py
2021-08-25 오후 07:21 <DIR> migrations
2021-08-25 오후 07:50 212 models.py
2021-08-25 오후 07:04 63 tests.py
2021-08-25 오후 07:04 66 views.py
2021-08-25 오후 07:04 0 __init__.py
2021-08-25 오후 07:54 <DIR> __pycache__
6개 파일 633 바이트
4개 디렉토리 18,346,868,736 바이트 남음
```

애플리케이션 개발하기 – Model 코딩

데이터베이스 변경사항 반영

- 마이그레이션 파일들을 이용해 migrate 명령으로 데이터베이스에 테이블을 생성
- (base) C:\MyTest\ToDoList>python manage.py migrate

```
(base) C:\MyTest\ToDoList\my_to_do_app>cd ..  
(base) C:\MyTest\ToDoList>python manage.py migrate  
Operations to perform:  
  Apply all migrations: admin, auth, contenttypes, my_to_do_app, sessions  
Running migrations:  
  Applying my_to_do_app.0001_initial... OK
```

프로젝트 뼈대 만들기

작업 확인하기

- 웹 서버를 실행하기 위해 하나의 창에서 작업해도 되지만 runserver 용으로 별도의 cmd창을 열어 사용하는 것이 편리
- (base) C:\MyTest\ToDoList>python manage.py runserver 127.0.0.1:8000
- (base) C:\MyTest\ToDoList >python manage.py runserver
- (base) C:\MyTest\ToDoList >python manage.py runserver 0:8000

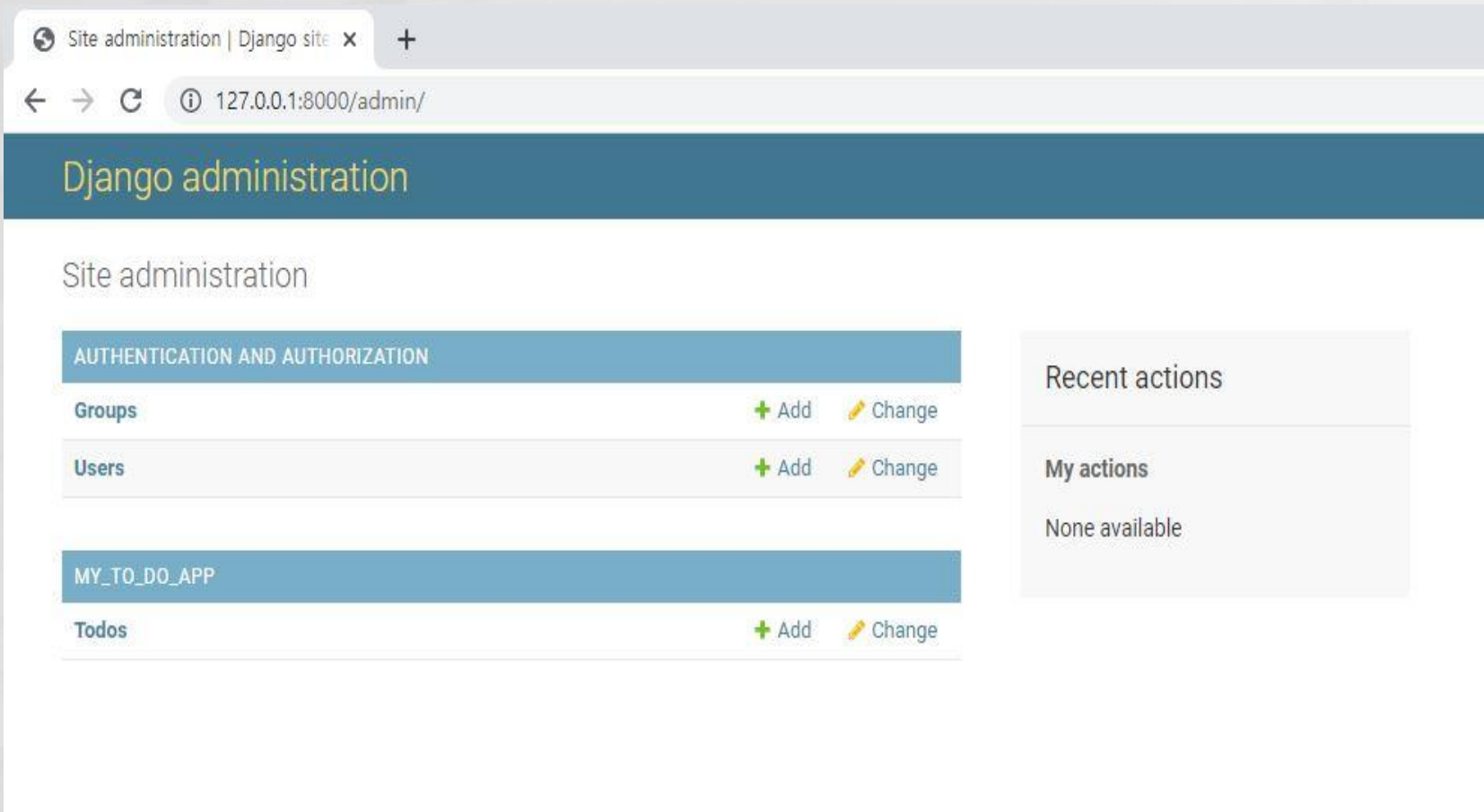
```
(base) C:\MyTest\ToDoList>python manage.py runserver 127.0.0.1:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 25, 2021 - 19:25:32
Django version 3.2.6, using settings 'ToDoList.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

애플리케이션 개발하기 – Model 코딩

작업 확인

- Todos테이블을 만든 모델이 정상적으로 등록



애플리케이션 개발하기 –View 및 Template

URLconf 코딩

- URLconf 설계 따르면, Admin사이트까지 포함해서 URL과 뷰가 필요
- `urls.py(C:\MyTest\ToDoList\ToDoList)` 파일에 코딩
- `urlpatterns = [~ path('', include('my_to_do_app.urls')), ~]`

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('', include('my_to_do_app.urls')),
    path('admin/', admin.site.urls),
]
```


애플리케이션 개발하기 –View 및 Template

URLconf 코딩

- urls.py(C:\MyTest\ToDoList\my_to_do_app) 파일에 코딩

```
# my_to_do_app > urls.py
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
    path('createTodo/', views.createTodo, name='createTodo'),
    path('doneTodo/', views.doneTodo, name='doneTodo')
]
```


애플리케이션 개발하기 –View 및 Template

뷰 함수 index() 및 템플릿 작성

- 뷰함수와 템플릿은 서로에게 영향을 미치기 때문에 보통 같이 작업
- 내용을 구현하기 위해 템플릿 파일 index.html

```
<body>
  <div class="container">
    <div class="header">
      <div class="page-header">
        <h1>To-do List <small>with Django</small> </h1>
      </div>
    </div>
    <div class="content">
      <div class="messageDiv">
        <form action="/.createTodo/" method="POST">{% csrf_token %}
          <div class="input-group">
            <input id="todoContent" name="todoContent" type="text" class="form-control" placeholder="메모할 내용"
            <span class="input-group-btn">
              <button class="btn btn-default" type="submit">메모하기!</button>
            </span>
          </div>
        </form>
```

애플리케이션 개발하기 –View 및 Template

뷰 함수 index() 및 템플릿 작성

- 뷰 코딩을 위해 views.py에 index() 함수 정의

```
def index(request):
    todos = Todo.objects.all() # Todo 클래스 이용해 db에 저장된 모든 레코드 select
    content = {'todos': todos} #content dict 생성
    return render(request, 'my_to_do_app/index.html', content) #index.html 파일로 전달
# # return HttpResponse("my_to_do_app first page")
# return render(request, 'my_to_do_app/index.html')
```

애플리케이션 개발하기 –View 및 Template

뷰 코딩을 위한 기능 작성

- 뷰 코딩을 위해 `views.py`에 기능 함수 정의

```
def createTodo(request):
    user_input_str = request.POST['todoContent']
    new_todo = Todo(content=user_input_str)
    new_todo.save()
    return HttpResponseRedirect(reverse('index'))
    # return HttpResponse("DB에 저장되었어요 =>" + user_input_str)

def doneTodo(request):
    done_todo_id = request.GET['todoNum']
    print("완료한 todo의 id", done_todo_id)
    todo = Todo.objects.get(id = done_todo_id)
    todo.isDone = True
    todo.save()
    return HttpResponseRedirect(reverse('index'))
```

애플리케이션 개발하기 –View 및 Template

작업 확인하기

- 웹 서버를 실행하기 위해 하나의 창에서 작업해도 되지만 runserver 용으로 별도의 cmd창을 열어 사용하는 것이 편리
- (base) C:\MyTest\ToDoList>python manage.py runserver 127.0.0.1:8000
- (base) C:\MyTest\ ToDoList >python manage.py runserver
- (base) C:\MyTest\ ToDoList >python manage.py runserver 0:8000

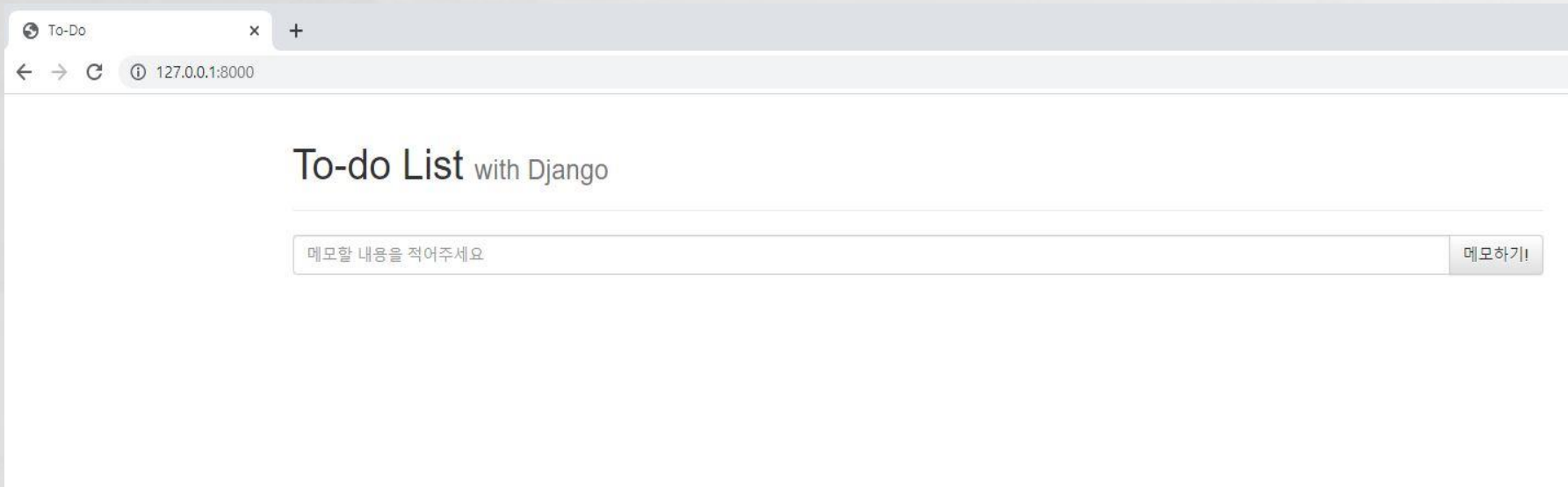
```
(base) C:\MyTest\ToDoList>python manage.py runserver 127.0.0.1:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 25, 2021 - 19:25:32
Django version 3.2.6, using settings 'ToDoList.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

애플리케이션 개발하기 –View 및 Template

작업 확인하기

- 웹 브라우저에서 `http://127.0.0.1:8000`



애플리케이션 개발하기 –View 및 Template

작업 확인하기

◦ 데이터 입력

The screenshot shows a web browser window with the title 'To-Do' and the address bar displaying '127.0.0.1:8000'. The page content is titled 'To-do List with Django'. It features a form for adding new tasks with a text input field containing the placeholder '메모할 내용을 적어주세요' and a red button labeled '완료'. Below this, there is a list of existing tasks, each with a text input field and a red '완료' button. The tasks listed are '오늘의 할일', '내일의 할일', '데이터엔지니어링', and '데이터사이언스'.

Task	Status
메모할 내용을 적어주세요	완료
오늘의 할일	완료
내일의 할일	완료
데이터엔지니어링	완료
데이터사이언스	완료

애플리케이션 개발하기 –View 및 Template

작업 확인하기

- DB에서 데이터 확인
- (base) C:\MyTest\ToDoList>python manage.py dbshell

```
(base) C:\MyTest\ToDoList>python manage.py dbshell
SQLite version 3.33.0 2020-08-14 13:23:32
Enter ".help" for usage hints.
sqlite> .tables
auth_group                                django_admin_log
auth_group_permissions                   django_content_type
auth_permission                          django_migrations
auth_user                                django_session
auth_user_groups                         my_to_do_app_todo
auth_user_user_permissions
sqlite> select * from my_to_do_app_todo;
1|오늘의 할일|0
2|내일의 할일|0
3|데이터엔지니어링|0
4|데이터사이언스|0
5|저녁 약속|1
sqlite>
```

```
sqlite> PRAGMA table_info(my_to_do_app_todo);
```

정리

정리

- 프로젝트 뼈대 만들기
- 애플리케이션 개발하기 - Model 코딩
- 애플리케이션 개발하기 - View 및 Template