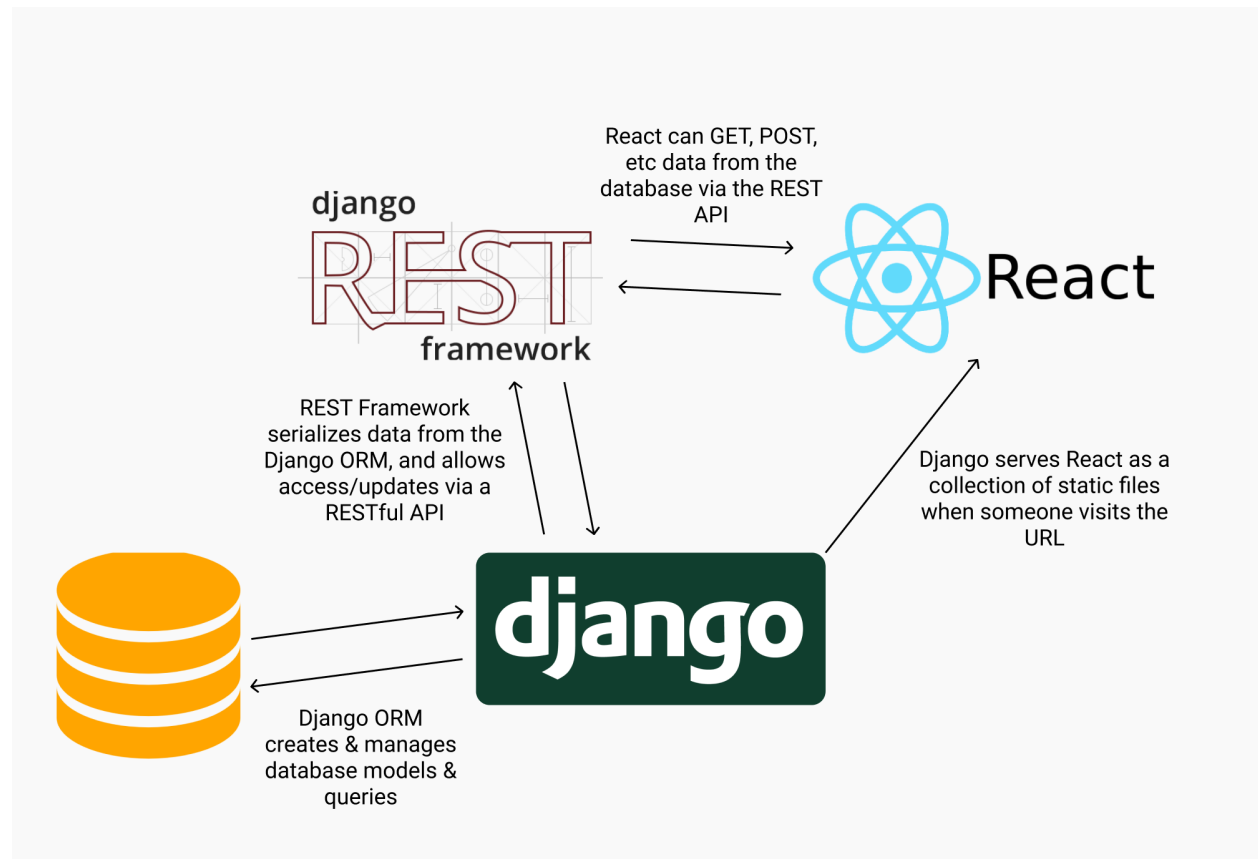


장고 기능

Web Server의 구성과 동작방식



특징

MVC 패턴 기반 MVT

- 장고는 Model-view-Controller 를 기반으로 한 프레임워크 이다.
- 장고에서 View를 Template
- Controller 를 View 라고 부른다.
- Model 은 DB에 액세스하는 컴포넌트 이다.

- Template 은 데이터를 사용자에게 보여주는 커포넌트

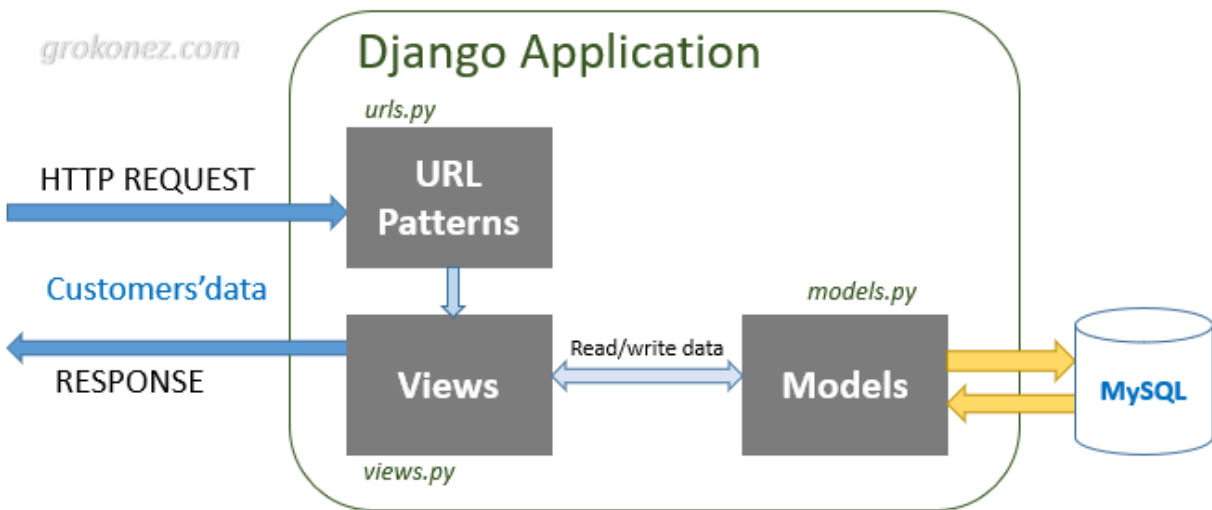
ORM

- 장고는 Object-Relational Mapping 방식을 이용해 DB에 데이터를 저장한다.
- 객체 관계 매핑 방식은 데이터 베이스 시스템과 데이터 모델 클래스를 연결시키는 다리 역할을 한다.
- SQL 문장을 사용하지 않고 테이블을 조작할수 있다.
- SQLite3, MySQL, PostgreSQL 등 엔진에 상관없이 사용할수 있다.

주요 기능

- Function Based Views : HTTP 요청을 함수로 처리한다.
- Class Based Views :Class 로 function 기반 Views 를 만들수 있따.
- Templates language : Views 에서 처리된 결과를 Templates Language로 변환하여 HTML 페이지로 보여진다.

전체적인 구조



- Client 로부터 Request 를 받으면 URLs.py 파일을 이용해 URL을 분석
- URLs.py 파일에 정의되어 요청을 처리할 Views 를 결정한다.
- Views 에 정의된 함수에 의해 로직이 처리되며 필요에 따라 Model을 참조한다.
- 처리된 결과를 Template로 전달하여 Template Language 에 의해 HTML페이지를 생성한다.
- HTML 파일을 Client로 전달하여 응답한다.

URLconf

- Client 로부터 요청을 받으면 장고는 가장먼저 요청에 들어있는 URL을 분석한다.
- URL이 urls.py 파일에 정의된 URL 패턴과 매칭되는지를 분석
 - settings.py 파일의 `ROOT_URLCONF` 항목을 통해 urls.py 파일의 위치가 정의된다.
 - `urlpatterns` 변수에 지정되어 있는 URL 리스트를 확인한다

```
path('pybo/<int:number>/', views.pybo_function),
```

- Request URL 이 **/pybo/3** 이면 View 함수를 **views.pybo_function(request, number=3)** 으로 호출한다.

<type: xxxx>

- <> 부분을 장고에서는 Path Converter 라고 명명한다.
- 사용 가능한 Type 은 다음과 같고 개발자가 추가로 등록 가능
 - str : / 를 제외한 모든 문자열과 매치된다.
 - int : 0 또는 양의 정수와 매치 된다.
 - 그외 slug, uuid, path
- re_path 함수를 이용해 정규표현식을 사용할수 있다.

```
from django.urls import re_path
urlpatterns = [re_path(정규표현식, views.functions)]
```

Views

```
# views.py

from django.http import HttpResponse

def function_name(request):
    html = <html 에 전달할 내용>
    return HttpResponse(html)
```

- View 함수는 첫 번째 인자로 HttpRequest 객체인 request 를 받는다.
- 그이후 필요한 처리 이후 HttpResponse 객체를 반환
- HttpResponse 이외 다른 객체들이 많이 존재 한다.

Template

- Client 에게 반환하는 최종 응답은 HTML 텍스트

- 템플릿 파일은 *.html 확장자를 가지며, 장고 템플릿 문법에 맞게 작성한다.\

```
# settings.py
TEMPLATES = [
{
'DIRS': [os.path.join(BASE_DIR, 'templates')]
}
]
```

장고 앱

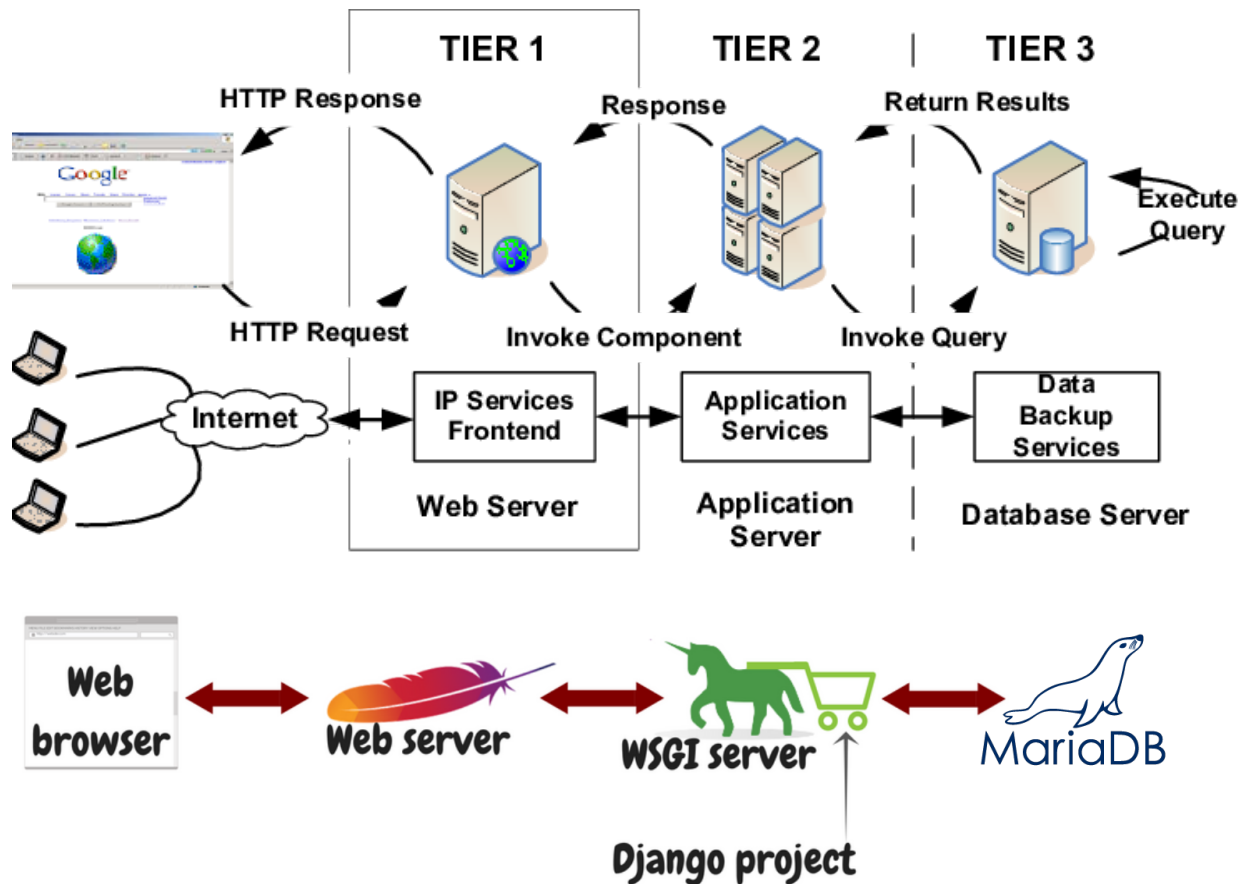
- 현재 프로젝트의 기능을 다른 프로젝트에서도 사용하려면?
- APP Template 형태로 제작하여 재사용 가능

```
python manage.py startapp <appName>
```

- 프로젝트 상에서 유일해야한다.
- settings.INSTALLED_APPS 에 등록시킨다.

Gole

- Web의 동작방식을 이해한다.
- 3 tier architecture를 이해하고 구현한다.



Django 디렉터리 구조

- `db.sqlite3` : SQLite3 데이터 베이스 파일
- `manage.py` : 장고 명령어 처리
- `urls.py` : 프로젝트 레벨/어플리케이션 레벨의 URL 패턴을 정의하는 URLconf 파일
- `wsgi.py` : 웹 서버와 WSGI 규격으로 연동하기 위한 파일
- `views.py` : 뷰 함수를 정의하는 파일. 함수형 뷰 및 클래스형 뷰 모두 이 파일에 정의한다.
- `templates` 디렉터리 : 템플릿 파일들이 들어있으며, 프로젝트/어플리케이션 레벨의 템플릿으로 구분된다.

- static 디렉터리 : CSS, Javascript 파일들이 들어있으며, 프로젝트/어플리케이션 레벨의 파일들로 구성되어있다.
- logs 디렉터리 : 로그 파일을 저장해 두는곳