

Django URL Pattern

- URL 이란
- Client 에서 URL Dispatcher 까지
- URL Dispatcher의 역할
- 정규 표현식
- views.py 함수 연결 및 HttpResponse
- urls.py 계층화

URL Pattern

1. URL 이란?

Uniform Resource Locator

`http://www.sample.com/example/?param=value`

Protocol

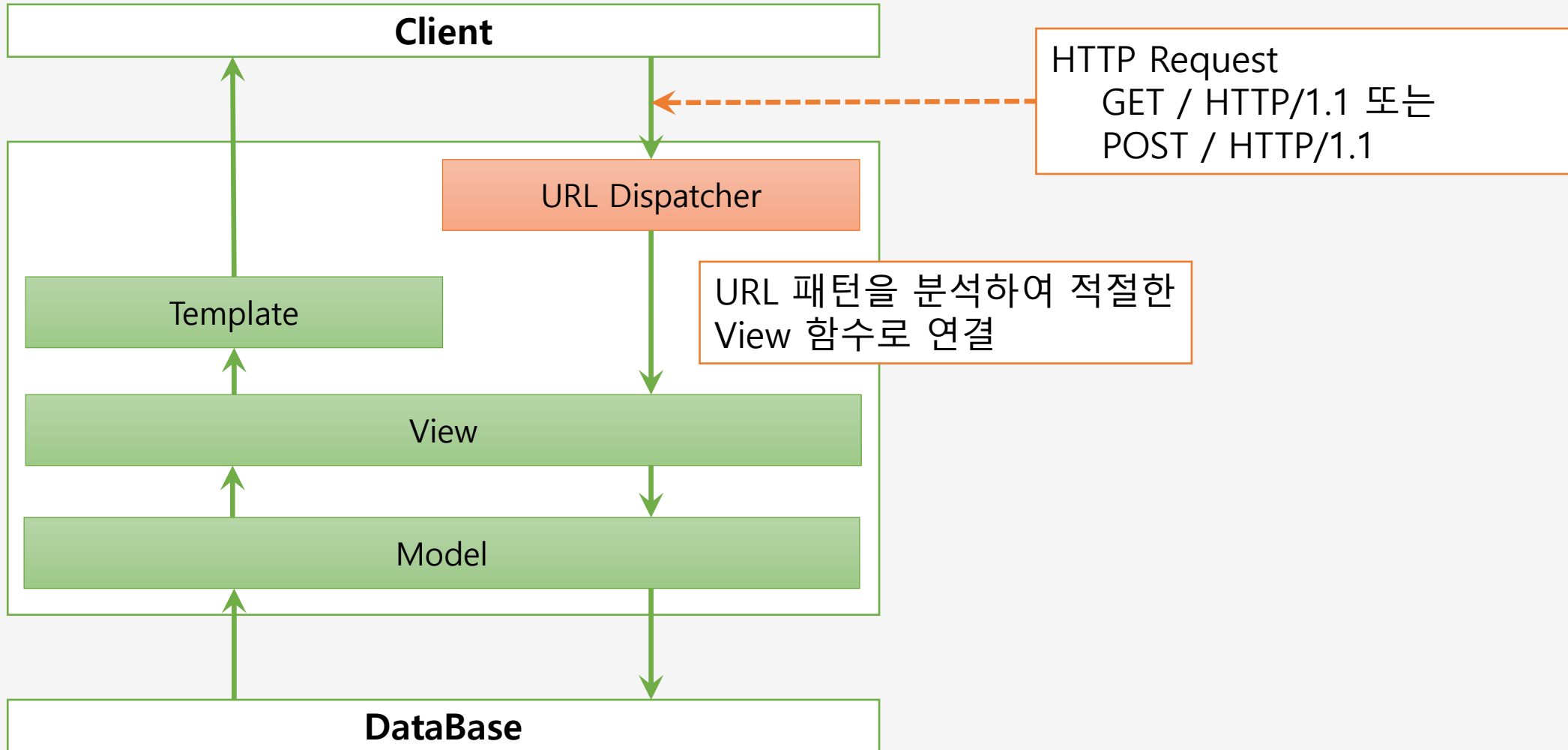
Domain

Path

Parameter

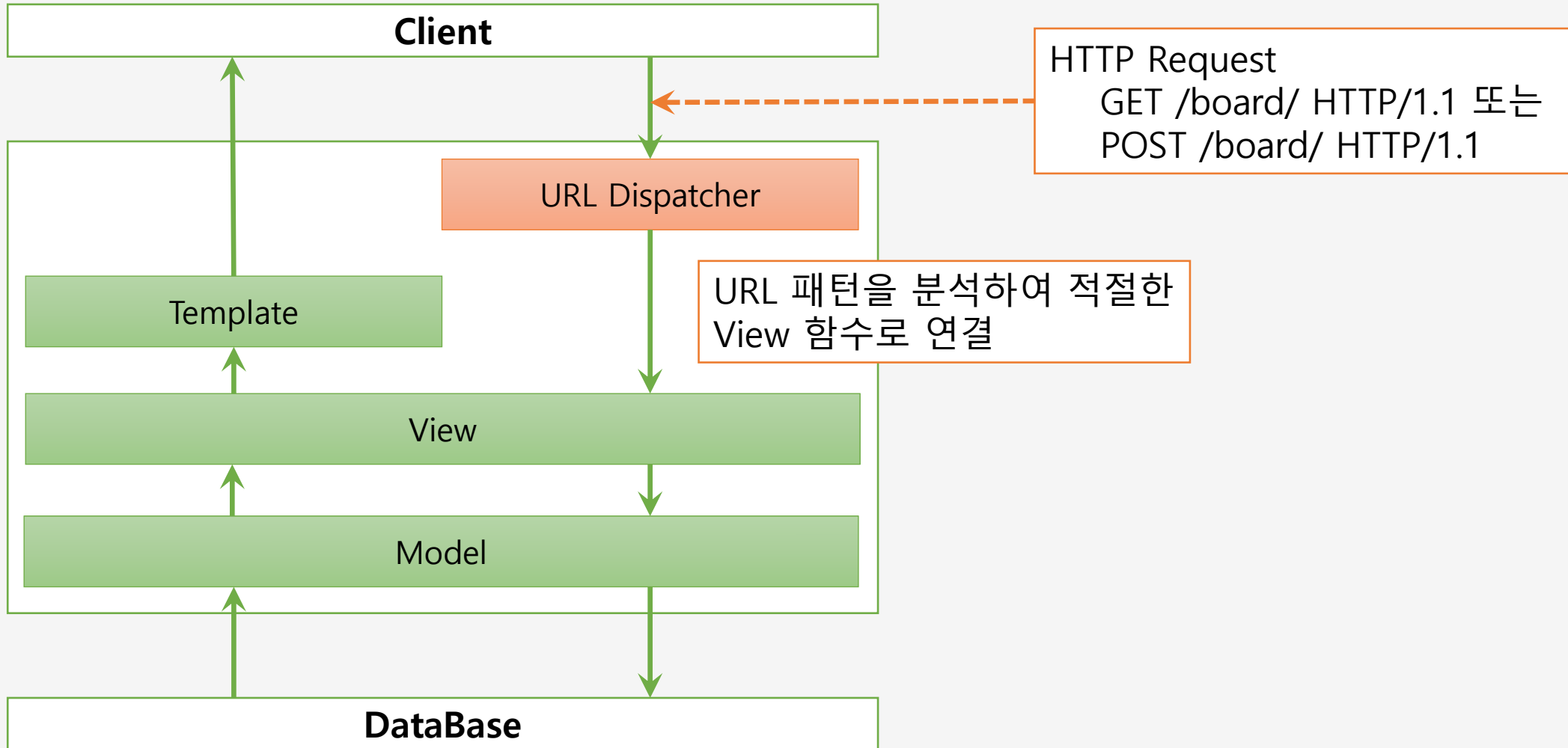
URL Pattern

2. Client에서 URL Dispatcher 까지 (1/2)



URL Pattern

2. Client에서 URL Dispatcher 까지 (2/2)



URL Pattern

3. URL Dispatcher의 역할 (1/2)

myWeb > urls.py

```
from django.conf.urls import include, url
from django.contrib import admin
```

```
urlpatterns = [
    # Examples:
    # url(r'^$', 'myWeb.views.home', name='home'),
    # url(r'^blog/', include('blog.urls')),
    url(r'^admin/', include(admin.site.urls)),
]
```

URL Pattern

3. URL Dispatcher의 역할 (2/2)

```
url(r'url 패턴과 정규표현식', '<app_name>.views.<function>', name='별칭')
```

```
url(r'url 패턴과 정규표현식', include('<app_name>.urls', namespace='이름공간'))
```

- # URL 패턴에 일치하는 요청에 대해 특정 동작을 수행 할 views.py의 함수로의 연결을 한다.
- # URL 패턴은 Domain 이후의 URL 정보(path)를 넣어주면 된다.
- # 항상 ^ 정규 표현식을 사용해야 한다.
- # include 함수를 사용하여 다른 App으로의 연결을 할 때는 \$ 정규 표현식을 사용하지 않는다.
하지만 그 외에는 항상 \$ 표현식을 가장 마지막에 사용한다.
- # 이름공간과 별칭은 차후 url 역순조회 할 때 사용 한다.
- # URL 패턴의 일부를 함수의 인자로 전달 할 수 있다.

URL Pattern

4. 정규 표현식 (1/4)

특정한 규칙을 지니고 있는 문자열의 집합을 표현하는데 사용하는 식이다.

특정 문자열 안에서 전화번호에 해당하는 문자열을 찾고자 할 때 또는 E-mail 주소에 해당하는 문자열을 찾고자 할 때 전화번호와 E-mail 주소의 특정 규칙을 정규 표현식으로 생성하여 찾아낼 수 있다.

URL Pattern

4. 정규 표현식 (2/4)

`r''` : 원시 문자열(raw String)로 사용 함을 알림, 이스케이프를 하지 않는 문자열

`^` : 패턴의 시작

`$` : 패턴의 끝

`+` : 한 자리 이상

`|` : or

`?` : 0 또는 1 자리

`[0-9]` : 0 ~ 9까지 한 자리

`[a-z]` : 소문자 a ~ z 까지 한 자리

`[A-Z]` : 대문자 A ~ Z 까지 한 자리

`{n}` : n 자리

`{n,m}` : n ~ m 자리까지

`(...)` : 소괄호 안의 패턴을 그룹으로 묶어 저장, 함수의 인자로 사용

`(?:...)` : 소괄호 안의 패턴을 그룹으로 묶지만 저장하지 않음, 함수의 인자로 사용하지 않음

`(P<이름>패턴)` : 소괄호 안의 패턴을 임의의 이름으로 그룹화하여 함수의 키워드 인자로 사용

URL Pattern

4. 정규 표현식 (3/4)

2 자리 숫자 00 ~ 99 까지의 정규 표현식

`[0-9][0-9]`

`[0-9]{2}`

4 자리 숫자 0000 ~ 9999 까지의 정규 표현식

`[0-9][0-9][0-9][0-9]`

`[0-9]{4}`

5 자리 소문자 정규 표현식

`[a-z]{5}`

python 또는 django 정규 표현식

`python|django`

URL Pattern

4. 정규 표현식 (4/4)

대소문자 구분 없는 6자리 정규 표현식

`[a-zA-Z]{6}`

1 ~ 4 자리 숫자 정규 표현식

`[0-9]{1,4}`

1 자리 이상 숫자 정규 표현식

`[0-9] +`

URL Pattern

5. views.py 함수 연결 및 HttpResponseRedirect (1/2)

myWeb > urls.py

```
from django.conf.urls import url

urlpatterns = [
    url(r'^sample/$', 'myWeb.views.func'),
]
```

myWeb > views.py

```
from django.http import HttpResponseRedirect

def func(request):
    return HttpResponseRedirect('Response Messages')
```

URL Pattern

5. views.py 함수 연결 및 HttpResponseRedirect (2/2)

myWeb > urls.py

```
from django.conf.urls import url
```

```
urlpatterns = [  
    url(r'^sample/([0-9]+)/$', 'myWeb.views.func'),  
]
```

myWeb > views.py

```
from django.http import HttpResponseRedirect
```

```
def func(request, number):  
    return HttpResponseRedirect('Server Get URL Value {}'.format(number))
```

URL Pattern

문제

다음의 규칙에 적합한 정규 표현식을 사용하여 각각의 view 함수에 연결 할 수 있는 코드를 작성 하시오.

1. 1990 ~ 2999 까지의 정규 표현식을 사용하여 myWeb의 viewOfYear 함수로 연결하십시오.
2. 01 ~ 12 까지의 정규 표현식을 사용하여 myWeb의 viewOfMonth 함수로 연결하십시오.
3. 1 ~ 31 까지의 정규 표현식을 사용하여 myWeb의 viewOfDay 함수로 연결하십시오.
4. -가 단어와 단어 사이에 포함되는 정규 표현식을 사용하여 myWeb의 viewOfPage 함수로 연결하십시오.
예) python-django 또는 django-url-pattern 또는 python

URL Pattern

6. urls.py 계층화 (1/3)

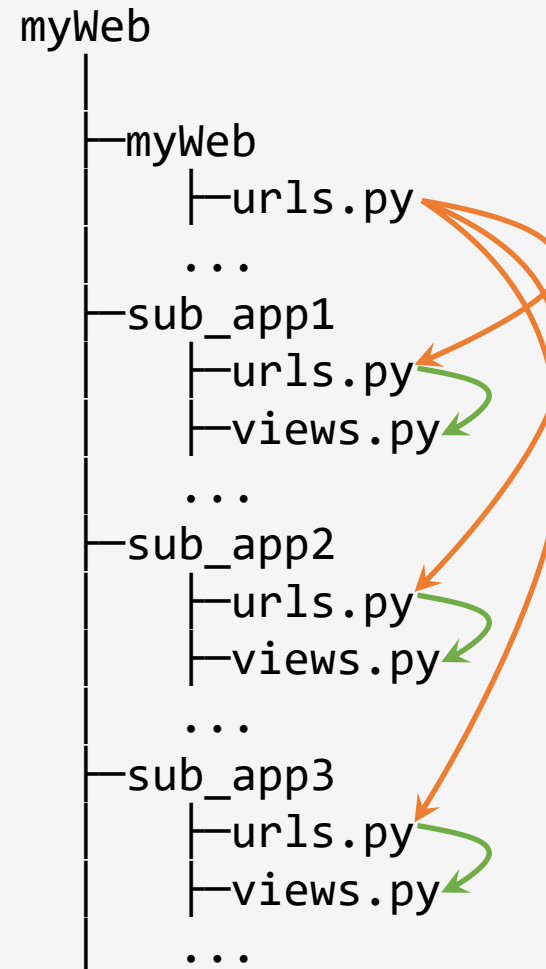
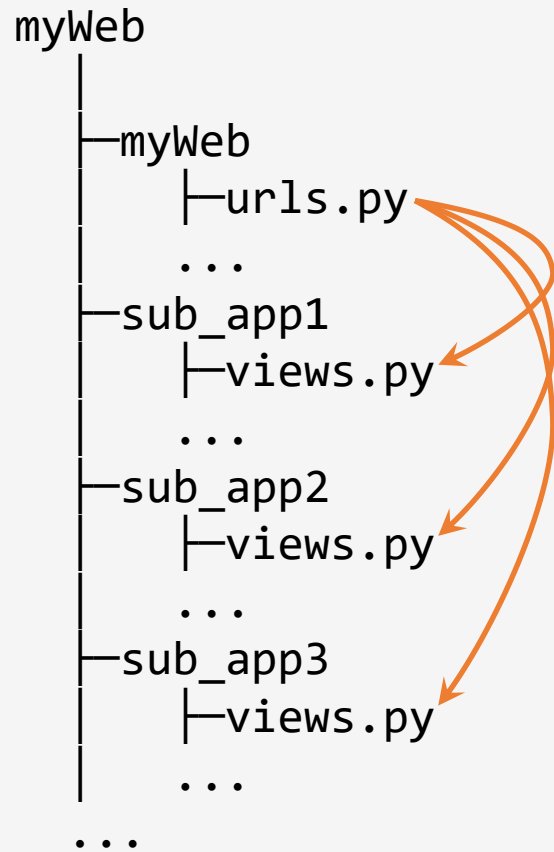
url 패턴을 분석하는 코드를 분리하여, 메인 App과 서브 App 간의 느슨한 연결을 통해 유연한 코드를 작성할 수 있게 한다.

프로젝트간 서브 App의 이식성이 높아진다.

동일 프로젝트 안에서의 URL 정보 수정이 쉬워진다. (관리가 편해진다.)

URL Pattern

6. urls.py 계층화 (2/3)



URL Pattern

6. urls.py 계층화 (3/3)

myWeb > urls.py

```
from django.conf.urls import url, include

urlpatterns = [
    url(r'^subapp/', include('sub_app.urls')),
]
```

sub_app > urls.py

```
from django.conf.urls import url

urlpatterns = [
    url(r'^$', 'sub_app.views.func'),
]
```