



리액트와 함께 장고 시작하기 / 장고 Views

URL Dispatcher와 정규 표현식

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

URL Dispatcher

"특정 URL 패턴 → View"의 List

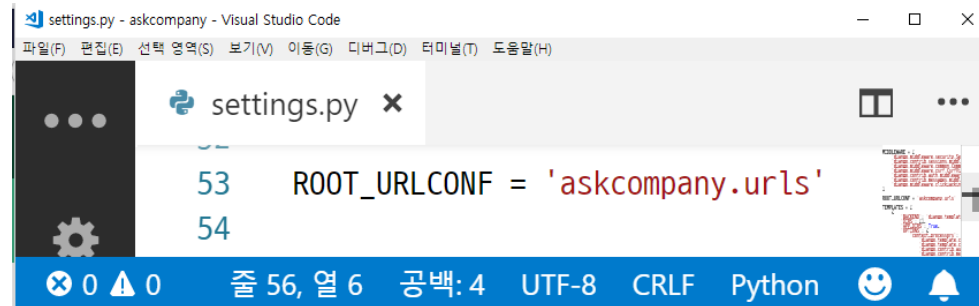
프로젝트/settings.py에서 최상위 URLConf 모듈을 지정

최초의 urlpatterns로부터 include를 통해, TREE구조로 확장

HTTP 요청이 들어올 때마다, 등록된 urlpatterns 상의 매핑 리스트를 처음부터 순차적으로 훑으며 URL 매칭을 시도

매칭이 되는 URL Rule이 다수 존재하더라도, 처음 Rule만을 사용

매칭되는 URL Rule이 없을 경우, 404 Page Not Found 응답을 발생



urlpatterns 예시

shop/urls.py

```
from django.urls import path, re_path
from shop import views
```

```
urlpatterns = [
    path('', views.item_list, name='item_list'),           # Item 목록
    path('new/', views.item_new, name='item_new'),         # 새 Item

    path('<int:id>/', views.item_detail, name='item_detail'), # Item 보기
    re_path(r'^(?P<id>\d+)/$', views.item_detail, name='item_detail'), # 혹은 re_path 활용

    path('<int:id>/edit/', views.item_edit, name='item_edit'), # Item 수정
    path('<int:id>/delete/', views.item_delete, name='item_delete'), # Item 삭제

    path('<int:id>/reviews/', views.review_list, name='review_list'), # 리뷰 목록
    path('<int:item_id>/reviews/<int:id>/edit/$', views.review_edit, # 리뷰 수정
        name='review_edit'),
    path('<int:item_id>/reviews/<int:id>/delete/$', views.review_delete, # 리뷰 삭제
        name='review_delete'),
]
```

path()와 re_path()의 등장

장고 1.x에서의 Django.conf.urls.url() 사용이 2가지로 분리

django.urls.re_path()

django.conf.urls.url()과 동일

django.urls.path()

기본 지원되는 Path converters를 통해 정규표현식 기입이 간소화 ➔ 만능은 아닙니다.

자주 사용하는 패턴을 Converter로 등록하면, 재사용면에서 편리

```
from django.conf.urls import url      # django 1.x 스타일
from django.urls import path, re_path # django 2.x ~ 스타일

urlpatterns = [
    # 장고 1.x에서의 다음 코드를
    url(r'^articles/(?P<year>[0-9]{4})/$', views.year_archive),

    # 다음과 같이 간소화 가능
    path('articles/<int:year>/', views.year_archive),

    # 물론 다음과 같이 동일하게 쓸 수 있습니다.
    re_path(r'^articles/(?P<year>[0-9]{4})/$', views.year_archive),
]
```

기본 제공되는 Path Converters

IntConverter → `r"[0-9]+"`

StringConverter → `r"[^/]+"`

UUIDConverter → `r"[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}"`

SlugConverter (StringConverter 상속) → `r"[-a-zA-Z0-9_]+"`

PathConverter (StringConverter 상속) → `r".+"`

<https://github.com/django/django/blob/3.0.2/django/urls/converters.py>

정규 표현식

Regular Expression

거의 모든 프로그래밍 언어에서 지원

문자열의 패턴, 규칙, Rule을 정의하는 방법

문자열 검색이나 치환작업을 간편하게 처리

장고 [URL Dispatcher](#)에서는 정규표현식을 통한 URL 매칭

문법

1글자에 대한 패턴 + 연속된 출연 횟수 지정

대괄호 내에 1글자에 대한 후보 글자들을 나열

다양한 정규 표현식 패턴 예시

1자리 숫자 → "[0123456789]" 혹은 "[0-9]" 혹은 r"[\d]" 혹은 r"\d"

2자리 숫자 → "[0123456789][0123456789]" 혹은 "[0-9][0-9]" 혹은 "\d\d"

3자리 숫자 → r"\d\d\d" 혹은 r"\d{3}"

2자리~4자리 숫자 → r"\d{2,4}"

휴대폰 번호 → r"010[1-9]\d{7}"

알파벳 소문자 1글자 → "[abcdefghijklmnopqrstuvwxyz]" 혹은 "[a-z]"

알파벳 대문자 1글자 → "[ABCDEFGHIJKLMNOPQRSTUVWXYZ]" 혹은 "[A-Z]"

반복횟수 지정 문법

$r^*\backslash d$: 별도 횟수 지정 없음 ➔ 1회 반복

$r^*\backslash d\{1\}$: 1회 반복

$r^*\backslash d\{2\}$: 2회 반복

$r^*\backslash d\{2,4\}$: 2회~4회 반복

$r^*\backslash d?$: 0회 혹은 1회 반복

$r^*\backslash d^*$: 0회 이상 반복

$r^*\backslash d^+$: 1회 이상 반복

주의 : 정규표현식은 띄워쓰기 하나에도 민감합니다. 가독성을 이유로 띄워쓰기를 하지 마세요.

커스텀 Path Converter

앱이름/converters.py

```
class FourDigitYearConverter:
```

```
    regex = '\d{4}'
```

```
    def to_python(self, value): # url로부터 추출한 문자열을 뷰에 넘겨주기 전에 변환  
        return int(value)
```

```
    def to_url(self, value): # url reverse 시에 호출  
        return "%04d" % value
```

앱이름/urls.py

```
from django.urls import register_converter
```

```
register_converter(FourDigitYearConverter, 'yyyy')
```

```
urlpatterns = [  
    path('articles/<yyyy:year>/', views.year_archive),  
]
```

커스텀 Converter 예시 : Slug Unicode

정규 표현식

```
slug_unicode_re = [-\w]+
```

Converter

```
from django.urls.converters import StringConverter
```

```
class SlugUnicodeConverter(StringConverter):  
    regex = r"[-\w]+"
```

<https://github.com/django/django/blob/3.0.2/django/core/validators.py#L244>

새로운 장고 앱을 생성할 때, 추천 작업

앱 내 `urls.py` 를 생성하고 등록해주세요.

1. 앱 생성
2. 앱이름/`urls.py` 파일 생성
3. 프로젝트/`urls.py`에 `include` 적용
4. 프로젝트/`settings.py`의 `INSTALLED_APPS`에 앱 이름 등록

```
# 프로젝트/settings.py
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    '앱이름',
]
```

```
# 앱/urls.py
from django.urls import path
from 앱이름 import views
```

```
app_name = '앱이름'
```

```
urlpatterns = [
]
```

```
# 프로젝트/urls.py
from django.contrib import admin
from django.urls import include, path
```

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

Life is short.
You need Python and Django.

I will be your pacemaker.

