

---

# 실전 프로그램 개발 - Blog 앱

# bookmark 앱 URLConf 조정

---

## ❖ 라우팅의 분리 – urls 모듈 분리

- 애플리케이션 마다 자체 urls 모듈 운영
- `mysite/urls.py`에서 모듈 단위로 라우팅 연결

# bookmark 앱 URLConf 조정

## ❖ bookmark/urls.py

```
from django.urls import path
from bookmark.views import BookmarkLV, BookmarkDV

app_name = 'bookmark' # 해당 애플리케이션의 네임스페이스명

urlpatterns = [
    path('', BookmarkLV.as_view(), name='index'),
    path('<int:pk>/', BookmarkDV.as_view(), name='detail'),
]
```

## ❖ mysite/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),

    path('bookmark/', include('bookmark.urls')),
]
```

# bookmark 앱 URLConf 조정

## ❖ bookmark/templates/bookmark\_list.html

```
<div>
  <h1>Bookmark List</h1>
  <ul>
    {% for bookmark in object_list %}
      <li>
        <a href="{% url 'bookmark:detail' bookmark.id %}">
          {{bookmark}}</a>
        </li>
      {% endfor %}
    </ul>
  </div>
```

### ○ url 블록

- app\_name:경로이름 --> bookmark:detail

```
app_name = 'bookmark'    # 해당 애플리케이션의 네임스페이스명
path('<int:pk>/', BookmarkDV.as_view(), name='detail'),
```

# bookmark 앱 URLConf 조정

---

## ❖ bookmark/templates/bookmark\_detail.html

```
<div>
  <h1>Bookmark List</h1>
  <div>
    <h1>{{bookmark.title}}</h1>
    <ul>
      <li>URL : <a href="{{bookmark.url}}" target="_blank">
        {{bookmark.url}}</a></li>
    </ul>
  </div>
  <a href="{% url 'bookmark:index' %}">목록으로</a>
</div>
```

# blog 앱 개발하기

## ❖ 작업 순서

작업 순서	관련 명령/파일	필요한 작업 내용
빠대 만들기	startproject	(2장에서 이미 완료했으므로 생략)
	settings.py	
	migrate	
	createsuperuser	
	startapp	블로그 앱 생성
	settings.py	블로그 앱 등록
모델 코딩하기	models.py	모델(테이블) 정의
	admin.py	Admin 사이트에 모델 등록
	makemigrations	모델의 변경사항 추출
	migrate	변경사항을 데이터베이스에 반영
URLconf 코딩하기	urls.py	URL 정의
뷰 코딩하기	views.py	뷰 로직 작성
템플릿 코딩하기	templates 디렉터리	템플릿 파일 작성
그 외 코딩하기	-	(없음)

# blog 앱 개발하기

---

## ❖ 뼈대 만들기

```
$ python manage.py startapp blog
```

## ❖ mysite/settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'bookmark.apps.BookmarkConfig', # 추가  
    'blog.apps.BlogConfig', # 추가  
]
```

# blog 앱 개발하기

## ❖ 모델 정의

필드명	타입	제약 조건	설명
id	Integer	PK, Auto Increment	기본 키
title	CharField(50)		<u>포스트</u> 제목
slug	SlugField(50)	Unique	<u>포스트</u> 제목 별칭
description	CharField(100)	Blank	<u>포스트</u> 내용 한 줄 설명
content	TextField		<u>포스트</u> 내용 기록
create_date	DateTimeField	auto_now_add	<u>포스트</u> 를 생성한 날짜
modify_date	DateTimeField	auto_now	<u>포스트</u> 를 수정한 날짜



# blog 앱 개발하기

## ❖ 모델 정의: blog/models.py

```
from django.db import models
from django.urls import reverse

class Post(models.Model):
    title          = models.CharField(verbose_name='TITLE', max_length=50)
    slug           = models.SlugField('SLUG', unique=True, allow_unicode=True,
                                     help_text='one word for title alias.')
    description     = models.CharField('DESCRIPTION', max_length=100,
                                     blank=True, help_text='simple description text.')
    content        = models.TextField('CONTENT')
    create_dt      = models.DateTimeField('CREATE DATE', auto_now_add=True)
    modify_dt      = models.DateTimeField('MODIFY DATE', auto_now=True)

    class Meta:
        verbose_name = 'post'
        verbose_name_plural = 'posts'
        db_table = 'blog_posts'      # 테이블명 재정의
        ordering = ('-modify_dt',)  # orderby 절, -이면 내림차순
```

# blog 앱 개발하기

## ❖ 모델 정의: blog/models.py

```
def __str__(self):
    return self.title

def get_absolute_url(self):    # 현재 데이터의 절대 경로 추출
    return ''    # reverse('blog:post_detail', args=(self.slug,))

def get_previous(self):    # 이전 데이터 추출
    return self.get_previous_by_modify_dt()

def get_next(self):    # 다음 데이터 추출
    return self.get_next_by_modify_dt()
```

- `reverse('앱명칭:url명' [, args=(경로변수...)])`
  - 템플릿의 `{% url %}`과 같은 동작

# blog 앱 개발하기

---

## ❖ blog/admin.py

```
from django.contrib import admin
from blog.models import Post

@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('id', 'title', 'modify_dt')
    list_filter = ('modify_dt',)
    search_fields = ('title', 'content')
    prepopulated_fields = { 'slug' : ('title', )}
```

# blog 앱 개발하기

---

## ❖ 데이터베이스반영

```
$ python manage.py makemigrations blog
```

```
$ python manage.py migrate
```

# blog 앱 개발하기

---

## ❖ URLConf : mysite/urls.py

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

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

# blog 앱 개발하기

---

## ❖ URLConf : blog/urls.py (목록보기)

```
from django.urls import path, re_path
from blog.views import *

app_name = 'blog'

urlpatterns = [
    path('', PostLV.as_view(), name='index'),
]
```

# blog 앱 개발하기

---

## ❖ 뷰 정의 : blog/views.py (목록보기)

```
from django.views.generic import ListView, DetailView
from blog.models import Post
```

```
# ListView
```

```
class PostLV(ListView):
```

```
    model = Post
```

```
    template_name = 'blog/post_all.html' # 템플릿 파일명 변경
```

```
    context_object_name = 'posts' # 컨텍스트 객체 이름 변경(object_list)
```

```
    paginate_by = 2 # 페이지네이션, 페이지당 문서 건 수
```

```
# DetailView
```

```
class PostDV(DetailView):
```

```
    model = Post
```

# blog 앱 개발하기

---

## ❖ 페이지네이션 객체 `page_obj`

- 컨텍스트 변수로 자동 추가됨
- `page_obj.has_previous` : 이전 페이지 존재 여부
- `page_obj.has_next` : 이후 페이지 존재 여부
- `page_obj.previous_page_number` : 이전 페이지 번호
- `page_obj.next_page_number` : 다음 페이지 번호
- `page_obj.number` : 현재 페이지 번호
- `page_obj.paginator.num_pages` : 전체 페이지 수



# blog 앱 개발하기

## ❖ 템플릿 정의 : blog/templates/blog/post\_all.html (목록보기)

```
<h1>Blog List</h1>
<br>
{%for post in posts%}
    <h3><a href="{post.get_absolute_url}">{{post.title}}</a></h3>
    {{post.modify_dt|date:"N d, Y"}}
    <p>{{post.description}}</p>
{% endfor %}
<br>
<div>
    <span>
        {% if page_obj.has_previous %}
            <a href="?page={{page_obj.previous_page_number}}">Prev</a>
        {% endif %}

        Page {{ page_obj.number }} of {{ page_obj.paginator.num_pages }}

        {% if page_obj.has_next %}
            <a href="?page={{page_obj.next_page_number}}">Next</a>
        {% endif %}

    </span>
</div>
```

# blog 앱 개발하기

## ❖ URLConf : blog/urls.py (상세보기)

```
from django.urls import path, re_path
from blog.views import *

app_name = 'blog'

urlpatterns = [
    path('', PostLV.as_view(), name='index'),
    re_path(r'^(?P<slug>[-\w]+)/$', PostDV.as_view(), name='detail'),
]
```

- re\_path(정규식 패턴, view, name=)
  - 정규표현식 패턴에 대응하는 url과 view를 매핑
  - 모델에서 reverse() 함수로 전달한 인자(<인자명>)로 url을 재구성
    - reverse('blog:post\_detail', args=(self.slug,))

# blog 앱 개발하기

---

## ❖ 뷰 정의 : blog/views.py (상세보기)

:

```
# DetailView
class PostDV(DetailView):
    model = Post
```

# blog 앱 개발하기

## ❖ 템플릿 정의 : blog/templates/blog/post\_detail.html (상세보기)

```
<h2>{{object.title}}</h2>
```

```
<p>
    {%if object.get_previous %}
        <a href="{{object.get_previous.get_absolute_url}}"
            title="이전 글">prev</a>
    {%endif%}

    {%if object.get_next %}
        <a href="{{object.get_next.get_absolute_url}}"
            title="다음 글">next</a>
    {%endif%}
</p>
```

```
<p>{{object.modify_dt | date:"j F Y"}}</p>
<br>
```

```
<div>
{{object.content | linebreaks}}
</div>
```

# blog 앱 개발하기

## ❖ 모델 정의: blog/models.py

```
def __str__(self):
    return self.title

def get_absolute_url(self):    # 현재 데이터의 절대 경로 추출
    return reverse('blog:post_detail', args=(self.slug,))

def get_previous(self):      # 이전 데이터 추출
    return self.get_previous_by_modify_dt()

def get_next(self):          # 다음 데이터 추출
    return self.get_next_by_modify_dt()
```

- `reverse('앱명칭:url명' [, args=(경로변수...)])`
  - 템플릿의 `{% url %}`과 같은 동작

---

# 실전 프로그램 개발 - Blog 앱 ArchiveView

# ArchiveView

---

## ❖ ArchiveView

- 날짜 필드로 필터링
- 년도별/월별/일별/오늘 목록 얻을 때 사용
  - ArchiveIndexView      <model>\_archive.html
  - YearArchiveView      <model>\_archive\_year.html
  - MonthArchiveView      <model>\_archive\_month.html
  - DayArchiveView      <model>\_archive\_day.html
  - TodayArchiveView      <model>\_archive\_day.html
- model, date\_field 필드 정의 필요

# 개발 코딩하기 - 뷰

---

## ❖ bookmark/views.py

```
from django.views.generic import ListView, DetailView
from django.views.generic.dates import ArchiveIndexView, YearArchiveView,
MonthArchiveView, DayArchiveView, TodayArchiveView

:
# ArchiveView
class PostAV(ArchiveIndexView):
    model= Post
    date_field = 'modify_dt'

class PostYAV(YearArchiveView):
    model= Post
    date_field = 'modify_dt'
    make_object_list = True

class PostMAV(MonthArchiveView):
    model= Post
    date_field = 'modify_dt'
    month_format = '%m'
```



# 개발 코딩하기 - 뷰

---

## ❖ bookmark/views.py

```
class PostDAV(DayArchiveView):  
    model= Post  
    date_field = 'modify_dt'  
    month_format = '%m'  
  
class PostTAV(TodayArchiveView):  
    model= Post  
    date_field = 'modify_dt'  
    month_format = '%m'
```

# 개발 코딩하기 - URLConf

## ❖ blog/urls.py

```
:
urlpatterns = [
    :
    # Example: /blog/archive/
    path('archive/', views.PostAV.as_view(), name='post_archive'),

    # Example: /blog/archive/2019/
    path('archive/<int:year>', views.PostYAV.as_view(),
         name='post_year_archive'),

    # Example: /blog/archive/2019/nov/
    path('archive/<int:year>/<str:month>', views.PostMAV.as_view(),
         name='post_month_archive'),

    # Example: /blog/archive/2019/nov/10/
    path('archive/<int:year>/<str:month>/<int:day>', views.PostDAV.as_view(),
         name='post_day_archive'),

    # Example: /blog/archive/today/
    path('archive/today/', views.PostTAV.as_view(), name='post_today_archive'),
]
```

# 개발 코딩하기 - 템플릿

## ❖ date 템플릿 필터

문자	설명	예시
d	일 수에 대한 2자리 숫자	01~31
j	일 수에 대한 숫자	1~31
S	일 수에 대한 순서	st, nd, rd, th
m	월 수에 대한 2자리 숫자	01~12
n	월 수에 대한 숫자	1~12
b	월에 대한 영문 3자, 소문자	jan
M	월에 대한 영문 3자	Jan
F	월에 대한 영문	January
y	년도 2자리 숫자	19
Y	년도 4자리 숫자	2019
D	요일 3자리 문자	Fri
l	요일 문자	Friday

문자	설명	예시
G	시간 (24시 기준)	0~23
H	시간 2자리 숫자(24시 기준)	00~23
g	시간 (12시 기준)	1~12
h	시간 2자리 숫자(12시 기준)	01~12
a	오후, 오전(소문자)	a.m. , p.m.
A	오후, 오전(대문자)	AM, PM
i	분 2자리 숫자	00~59
s	초 2자리 숫자	00~59

❖ [bookmark/templates/bookmark/post\\_archive.html](#)

28

❖ **bookmark/templates/bookmark/post\_archive\_year.html**

29

❖ **bookmark/templates/bookmark/post\_archive\_month.html**[illegible]

❖ **bookmark/templates/bookmark/post\_archive\_day.html**

31

## 개발 코딩하기 - 템플릿

❖ **bookmark/templates/bookmark/post\_archive\_today.html**[illegible]