
실전 프로그램 개발

- 콘텐츠 편집 기능 -

(Photo 앱)

애플리케이션 설계하기

❖ 화면 설계

Album-Photo Create/Update - shkim
album_form.html 화면

create or update album contents

NAME: 국가별-수정

One Line Description: 국가별로 그 나라를 대표하는 사진들

create or update photo contents

IMAGE: Currently: photo/2019/08/brazil-festival_H5845Xo.jpg
Change: Choose File No file chosen

TITLE: 브라질 축제-수정

Photo Description: 브라질 상파울로에서 매년 열리는 삼바 축제.

UpdateView 내부적으로 만든 모델폼 사용

PhotoInlineFormSet 사용

애플리케이션 설계하기

❖ 화면 설계

The screenshot shows a web application titled "Django - Python Web Programming" with a navigation bar containing links for Home, Bookmark, Blog, Photo, Add, Change, Util, and a global search bar. The main heading is "Photo Create/Update - shkim". Below this is a form for creating or updating a photo. The form includes fields for Album (a dropdown menu), TITLE, IMAGE (with a "Choose File" button and "No file chosen" text), and Photo Description. A "Submit" button is at the bottom. Annotations in Korean are present: "photo_form.html 화면" points to the form title; "photo 메뉴 클릭" points to the "Photo" option in the "Add" dropdown menu; "앨범을 생성하기 위한 버튼" points to the "Add Album" button; and "CreateView 내부적으로 만든 모델폼 사용" points to the form fields.

Django - Python Web Programming

Home Bookmark Blog Photo Add Change Util globa

Photo Create/Update - shkim

photo_form.html 화면 your photo.

Album: Add Album

TITLE:

IMAGE: Choose File No file chosen

Photo Description:

Submit

photo 메뉴 클릭

앨범을 생성하기 위한 버튼

CreateView 내부적으로 만든 모델폼 사용

애플리케이션 설계하기

❖ 테이블 설계

○ Album 모델 클래스

| 필드명 | 타입 | 제약 조건 | 설명 |
|-------------|------------------|--------------------|--------|
| id | Integer | PK, Auto Increment | 기본 키 |
| name | CharField(50) | | 앨범 이름 |
| description | CharField(100) | Blank | 앨범 설명 |
| owner | ForeignKey(User) | Null | 앨범 소유자 |

○ Photo 모델 클래스

| 필드명 | 타입 | 제약 조건 | 설명 |
|-------------|---------------------|--------------------|-------------|
| id | Integer | PK, Auto Increment | 기본 키 |
| album | ForeignKey(Album) | | 사진이 소속된 앨범 |
| title | CharField(50) | | 사진 제목 |
| image | ThumbnailImageField | | 이미지 파일 |
| description | TextField | Blank, NotNull | 사진 설명 |
| upload_date | DateTimeField | auto_now_add | 사진을 업로드한 시각 |
| owner | ForeignKey(User) | Null | 사진 소유자 |

애플리케이션 설계하기

❖ URL 설계

| URL 패턴 | 뷰 이름 | 템플릿 파일명 |
|--------------------------------------|-----------------------------|---------------------------|
| /photo/ | AlbumLV(ListView) | album_list.html |
| /photo/album/ | AlbumLV(ListView) | album_list.html |
| /photo/album/99/ | AlbumDV(DetailView) | album_detail.html |
| /photo/photo/99/ | PhotoDV(DetailView) | photo_detail.html |
| /photo/album/add/ ¹ | AlbumPhotoCV(CreateView) | album_form.html |
| /photo/album/change/ | AlbumChangeLV(ListView) | album_change_list.html |
| /photo/album/99/update/ ¹ | AlbumPhotoUV(UpdateView) | album_form.html |
| /photo/album/99/delete/ | AlbumDeleteView>DeleteView) | album_confirm_delete.html |
| /photo/photo/add/ ² | PhotoCreateView(CreateView) | photo_form.html |
| /photo/photo/change/ | PhotoChangeLV(ListView) | photo_change_list.html |
| /photo/photo/99/update/ ² | PhotoUpdateView(UpdateView) | photo_form.html |
| /photo/photo/99/delete/ | PhotoDeleteView>DeleteView) | photo_confirm_delete.html |

애플리케이션 설계하기

❖ 작업 절차

| 작업 순서 | 관련 명령/파일 | 필요한 작업 내용 |
|--------------|-----------------|---------------------|
| 뼈대 만들기 | startproject | (2장에서 이미 완료했으므로 생략) |
| | settings.py | |
| | migrate | |
| | createsuperuser | |
| | startapp | (변경 사항 없음) |
| | settings.py | |
| 모델 코딩하기 | models.py | owner 필드 추가 |
| | admin.py | (변경 사항 없음) |
| | makemigrations | 변경 사항을 데이터베이스에 반영 |
| | migrate | |
| URLconf 코딩하기 | urls.py | URL 정의 |
| 뷰 코딩하기 | forms.py | 인라인 폼셋 정의 |
| | views.py | 뷰 로직 작성 |
| 템플릿 코딩하기 | templates 디렉터리 | 템플릿 파일 작성 |
| 그 외 코딩하기 | templates 디렉터리 | home.html 수정 |

개발 코딩하기

❖ photo/models.py

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

from photo.fields import ThumbnailImageField

class Album(models.Model):
    name = models.CharField('NAME', max_length=30)
    description = models.CharField('One Line Description', max_length=100,
                                   blank=True)
    owner = models.ForeignKey('auth.User', on_delete=models.CASCADE,
                              verbose_name='OWNER', blank=True, null=True)

    class Meta:
        ordering = ('name',)

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return reverse('photo:album_detail', args=(self.id,))
```

개발 코딩하기

❖ photo/models.py

```
class Photo(models.Model):
    album = models.ForeignKey(Album, on_delete=models.CASCADE)
    title = models.CharField('TITLE', max_length=30)
    description = models.TextField('Photo Description', blank=True)
    image = ThumbnailImageField('IMAGE', upload_to='photo/%Y/%m')
    upload_dt = models.DateTimeField('UPLOAD DATE', auto_now_add=True)
    owner = models.ForeignKey('auth.User', on_delete=models.CASCADE,
                              verbose_name='OWNER', blank=True, null=True)

    class Meta:
        ordering = ('title',)

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        return reverse('photo:photo_detail', args=(self.id,))
```


개발 코딩하기

❖ photo/forms.py

```
from django.forms import inlineformset_factory
from photo.models import Album, Photo

PhotoInlineFormSet = inlineformset_factory(Album, Photo,
    fields = ['image', 'title', 'description'],
    extra = 2)
```

개발 코딩하기

❖ photo/views.py

```
from django.views.generic import ListView, DetailView
from photo.models import Album, Photo

from django.contrib.auth.mixins import LoginRequiredMixin
from django.shortcuts import redirect
from django.urls import reverse_lazy
from django.views.generic import CreateView, UpdateView, DeleteView
from mysite.views import OwnerOnlyMixin
from photo.forms import PhotoInlineFormSet


class AlbumLV(ListView):
    model = Album


class AlbumDV(DetailView):
    model = Album


class PhotoDV(DetailView):
    model = Photo
```

개발 코딩하기

❖ photo/views.py

```
#--- Create/Change-list/Update/Delete for Photo
class PhotoCV(LoginRequiredMixin, CreateView):
    model = Photo
    fields = ('album', 'title', 'image', 'description')
    success_url = reverse_lazy('photo:index')

    def form_valid(self, form):
        form.instance.owner = self.request.user
        return super().form_valid(form)

class PhotoChangeLV(LoginRequiredMixin, ListView):
    model = Photo
    template_name = 'photo/photo_change_list.html'

    def get_queryset(self):
        return Photo.objects.filter(owner=self.request.user)
```

개발 코딩하기

❖ photo/views.py

```
class PhotoUV(OwnerOnlyMixin, UpdateView):
    model = Photo
    fields = ('album', 'title', 'image', 'description')
    success_url = reverse_lazy('photo:index')

class PhotoDelV(OwnerOnlyMixin, DeleteView):
    model = Photo
    success_url = reverse_lazy('photo:index')
```

개발 코딩하기

❖ photo/views.py

```
#-- Change-list/Delete for Album
class AlbumChangeLV(LoginRequiredMixin, ListView):
    model = Album
    template_name = 'photo/album_change_list.html'

    def get_queryset(self):
        return Album.objects.filter(owner=self.request.user)

class AlbumDelV(OwnerOnlyMixin, DeleteView):
    model = Album
    success_url = reverse_lazy('photo:index')
```

개발 코딩하기

❖ photo/views.py

```
#--- (InlineFormSet) Create/Update for Album
class AlbumPhotoCV(LoginRequiredMixin, CreateView):
    model = Album
    fields = ('name', 'description')
    success_url = reverse_lazy('photo:index')

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        if self.request.POST:
            context['formset'] = PhotoInlineFormSet(self.request.POST,
                                                    self.request.FILES)
        else:
            context['formset'] = PhotoInlineFormSet()
        return context
```

개발 코딩하기

❖ photo/views.py

```
def form_valid(self, form):
    form.instance.owner = self.request.user
    context = self.get_context_data()
    formset = context['formset']
    for photoform in formset:
        photoform.instance.owner = self.request.user
    if formset.is_valid():
        self.object = form.save()
        formset.instance = self.object
        formset.save()
        return redirect(self.get_success_url())
    else:
        return self.render_to_response(self.get_context_data(form=form))
```

개발 코딩하기

❖ photo/views.py

```
class AlbumPhotoUV(OwnerOnlyMixin, UpdateView):
    model = Album
    fields = ('name', 'description')
    success_url = reverse_lazy('photo:index')

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        if self.request.POST:
            context['formset'] = PhotoInlineFormSet(self.request.POST,
                                                    self.request.FILES, instance=self.object)
        else:
            context['formset'] = PhotoInlineFormSet(instance=self.object)
        return context
```


개발 코딩하기

❖ photo/views.py

```
def form_valid(self, form):
    context = self.get_context_data()
    formset = context['formset']
    if formset.is_valid():
        self.object = form.save()
        formset.instance = self.object
        formset.save()
        return redirect(self.get_success_url())
    else:
        return self.render_to_response(self.get_context_data(form=form))
```

개발 코딩하기

❖ photo/urls.py

```
from django.urls import path

from photo import views

app_name = 'photo'
urlpatterns = [
    :
    # Example: /photo/album/add/
    path('album/add/', views.AlbumPhotoCV.as_view(), name='album_add'),

    # Example: /photo/album/change/
    path('album/change/', views.AlbumChangeLV.as_view(), name='album_change'),

    # Example: /photo/album/99/update/
    path('album/<int:pk>/update/', views.AlbumPhotoUV.as_view(),
         name='album_update'),

    # Example: /photo/album/99/delete/
    path('album/<int:pk>/delete/', views.AlbumDelV.as_view(),
         name='album_delete'),
```

개발 코딩하기

❖ photo/urls.py

```
# Example: /photo/photo/add/  
path('photo/add/', views.PhotoCV.as_view(), name='photo_add'),  
  
# Example: /photo/photo/change/  
path('photo/change/', views.PhotoChangeLV.as_view(), name='photo_change'),  
  
# Example: /photo/photo/99/update/  
path('photo/<int:pk>/update/', views.PhotoUV.as_view(),  
      name='photo_update'),  
  
# Example: /photo/photo/99/delete/  
path('photo/<int:pk>/delete/', views.PhotoDelV.as_view(),  
      name='photo_delete'),
```

```
]
```

개발 코딩하기

❖ templates/base.html

```
<li class="nav-item dropdown mx-1 btn btn-primary">
  <a class="nav-link dropdown-toggle text-white" href="#"
    data-toggle="dropdown">Change</a>
  <div class="dropdown-menu">
    <a class="dropdown-item"
      href="{% url 'bookmark:change' %}">Bookmark</a>
    <a class="dropdown-item" href="{% url 'blog:change' %}">Post</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item"
      href="{% url 'photo:album_change' %}">Album</a>
    <a class="dropdown-item"
      href="{% url 'photo:photo_change' %}">Photo</a>
  </div>
</li>
```

개발 코딩하기

❖ photo/templates/photo/photo_form.html

```
{% extends "base.html" %}
{% load widget_tweaks %}

{% block title %}photo_form.html{% endblock %}

{% block content %}
    <h1>Photo Create/Update - {{user}}</h1>
    <p class="font-italic">This is a creation or update form for your photo.</p>

    {% if form.errors %}
    <div class="alert alert-danger">
        <div class="font-weight-bold">
            Wrong! Please correct the error(s) below.</div>
        {{ form.errors }}
    </div>
    {% endif %}
</div>
```

개발 코딩하기

❖ photo/templates/photo/photo_form.html

```
{% if form.is_multipart %}
<form enctype="multipart/form-data" action="" method="post"
    class="card pt-3">
{% else %}
<form action="." method="post" class="card pt-3">
{% endif %}
{% csrf_token %}

<div class="form-group row">
    {{ form.album
        |add_label_class:"col-form-label col-sm-2 ml-3 font-weight-bold" }}
    <div class="col-sm-2">
        {{ form.album|add_class:"form-control" }}
    </div>
    <div class="col-sm-2 my-auto">
        <a href="{% url 'photo:album_add' %}"
            class="btn btn-outline-primary btn-sm">Add Album</a>
    </div>
</div>
```

개발 코딩하기

❖ photo/templates/photo/photo_form.html

```
<div class="form-group row">
    {{ form.title|
        add_label_class:"col-form-label col-sm-2 ml-3 font-weight-bold" }}
    <div class="col-sm-5">
        {{ form.title|add_class:"form-control"|attr:"autofocus" }}
    </div>
</div>
```

```
<div class="form-group row">
    {{ form.image|
        add_label_class:"col-form-label col-sm-2 ml-3 font-weight-bold" }}
    <div class="col-sm-5">
        {{ form.image|add_class:"form-control-file" }}
    </div>
</div>
```

```
<div class="form-group row">
    {{ form.description|
        add_label_class:"col-form-label col-sm-2 ml-3 font-weight-bold" }}
    <div class="col-sm-8">
        {{ form.description|add_class:"form-control"|attr:"rows:3" }}
    </div>
</div>
```

개발 코딩하기

❖ photo/templates/photo/photo_form.html

```
<div class="form-group">
  <div class="offset-sm-2 col-sm-5">
    <input type="submit" value="Submit" class="btn btn-info"/>
  </div>
</div>

</form>

{% endblock %}
```


개발 코딩하기

❖ photo/templates/photo/photo_change_list.html

```
{% extends "base.html" %}

{% block title %}photo_change_list.html{% endblock %}

{% block content %}

    <h1>Photo Change - {{user}}</h1>

    <table class="table table-striped table-bordered table-condensed">

        <thead>
        <tr class="table-info">
            <th>Album</th>
            <th>Title</th>
            <th>Description</th>
            <th>Owner</th>
            <th>Update</th>
            <th>Delete</th>
        </tr>
        </thead>
```

개발 코딩하기

❖ photo/templates/photo/photo_change_list.html

```
<tbody>
{% for item in object_list %}
<tr>
    <td>{{ item.album }}</td>
    <td>{{ item.title }}</td>
    <td>{{ item.description }}</td>
    <td>{{ item.owner }}</td>
    <td><a href="{% url 'photo:photo_update' item.id %}">Update</a></td>
    <td><a href="{% url 'photo:photo_delete' item.id %}">Delete</a></td>
</tr>
{% endfor %}
</tbody>

</table>

{% endblock %}
```

개발 코딩하기

❖ photo/templates/photo/photo_confirm_delete.html

```
{% extends "base.html" %}

{% block title %}photo_confirm_delete.html{% endblock %}

{% block content %}

    <h1>Photo Delete</h1>
    <br>

    <form action="." method="post">{% csrf_token %}
        <p>Are you sure you want to delete "{{ object }}" ?</p>
        <input type="submit" value="Confirm" class="btn btn-danger btn-sm" />
    </form>

{% endblock %}
```

개발 코딩하기

❖ photo/templates/photo/album_form.html

```
{% extends "base.html" %}
{% load widget_tweaks %}

{% block title %}album_form.html{% endblock %}

{% block content %}
    <h1>Album-Photo Create/Update - {{user}}</h1>
    <p class="font-italic">
        This is a creation or update form for your album using PhotoInlineFormSet.
    </p>

    {% if form.errors %}
    <div class="alert alert-danger">
        <div class="font-weight-bold">
            Wrong! Please correct the form error(s) below.</div>
        {{ form.errors }}
    </div>
    {% endif %}
{% endblock %}
```

개발 코딩하기

❖ photo/templates/photo/album_form.html

```
{% if formset.errors %}
<div class="alert alert-warning">
  <div class="font-weight-bold">
    Wrong! Please correct the formset error(s) below.
  </div>
  {% for formerrors in formset.errors %}
    {{ formerrors }}
  {% endfor %}
</div>
{% endif %}
```

개발 코딩하기

❖ photo/templates/photo/album_form.html

```
{% if form.is_multipart or formset.is_multipart %}
<form enctype="multipart/form-data" action="" method="post">
{% else %}
<form action="." method="post">
{% endif %}
    {% csrf_token %}
    <h4>create or update album contents</h4>
    <fieldset class="card pt-3">
        <div class="form-group row">
            {{ form.name
                |add_label_class:"col-form-label col-sm-3 ml-3 font-weight-bold" }}
            <div class="col-sm-5">
                {{ form.name|add_class:"form-control"|attr:"autofocus" }}
            </div>
        </div>
        <div class="form-group row">
            {{ form.description
                |add_label_class:"col-form-label col-sm-3 ml-3 font-weight-bold" }}
            <div class="col-sm-5">
                {{ form.description|add_class:"form-control" }}
            </div>
        </div>
    </fieldset>
```

개발 코딩하기

❖ photo/templates/photo/album_form.html

```
<br>
<h4>create or update photo contents</h4>

<fieldset class="card pt-3">
    {{ formset.management_form }}
    {% for form in formset %}
        {{ form.id }}
        {# form.album #}
        <div class="form-group row">
            {{ form.image|
                add_label_class:"col-form-label col-sm-3 ml-3 font-weight-bold" }}
            <div class="col-sm-5">
                {{ form.image|add_class:"form-control-file" }}
            </div>
        </div>
        <div class="form-group row">
            {{ form.title|
                add_label_class:"col-form-label col-sm-3 ml-3 font-weight-bold" }}
            <div class="col-sm-5">
                {{ form.title|add_class:"form-control" }}
            </div>
        </div>
    </div>
```

개발 코딩하기

❖ photo/templates/photo/album_form.html

```
<div class="form-group row">
    {{ form.description|
        add_label_class:"col-form-label col-sm-3 ml-3 font-weight-bold" }}
    <div class="col-sm-8">
        {{ form.description|add_class:"form-control"|attr:"rows:3" }}
    </div>
</div>
<hr>
{% endfor %}
</fieldset>

<div class="form-group card py-1">
    <div class="offset-sm-3 col-sm-5">
        <input type="submit" value="Submit" class="btn btn-info"/>
    </div>
</div>

</form>
{% endblock %}
```


개발 코딩하기

❖ photo/templates/photo/album_change_list.html

```
{% extends "base.html" %}

{% block title %}album_change_list.html{% endblock %}

{% block content %}

    <h1>Album Change - {{user}}</h1>

    <table class="table table-striped table-bordered table-condensed">

        <thead>
        <tr class="table-success">
            <th>Name</th>
            <th>Description</th>
            <th>Owner</th>
            <th>Update</th>
            <th>Delete</th>
        </tr>
        </thead>
```

개발 코딩하기

❖ photo/templates/photo/album_change_list.html

```
<tbody>
{% for item in object_list %}
<tr>
    <td>{{ item.name }}</td>
    <td>{{ item.description }}</td>
    <td>{{ item.owner }}</td>
    <td><a href="{% url 'photo:album_update' item.id %}">Update</a></td>
    <td><a href="{% url 'photo:album_delete' item.id %}">Delete</a></td>
</tr>
{% endfor %}
</tbody>

</table>

{% endblock %}
```

개발 코딩하기

❖ photo/templates/photo/album_confirm_delete.html

```
{% extends "base.html" %}

{% block title %}album_confirm_delete.html{% endblock %}

{% block content %}

    <h1>Album Delete <small class="font-italic">
        (including related photos)</small></h1>
    <br>

    <form action="." method="post">{% csrf_token %}
        <p>Are you sure you want to delete "{{ object }}" ?</p>
        <input type="submit" value="Confirm" class="btn btn-danger btn-sm" />
    </form>

{% endblock %}
```