

Praktek Django Forms Part II

By : Hendra Pasaribu

LATIHAN formset_factory

1. Buka web hosting anda, lalu buka Files – File Manager
2. Cari **PROJECT ANDA SENDIRI**, di sini saya menggunakan project dosen_latihan00.
3. Buka file web_app/models.py dan ketikkan perintah berikut :

```
class Contact(models.Model):
    name = models.CharField(max_length = 50)
    phone = models.CharField(max_length = 50)
    email = models.EmailField()

    def __str__(self):
        return str(self.name)
```

4. Buka file web_app/forms.py dan ketikkan perintah berikut:

```
from django import forms
from django.forms import ModelForm, formset_factory, modelformset_factory, inlineformset_factory
from .models import GeeksModel, Employee, Album, Song, Contact, Student

# Contoh formset_factory
class ContactForm(ModelForm):
    class Meta:
        model = Contact
        fields = '__all__'

ContactFormSet = formset_factory(ContactForm, extra = 3)
```

5. Buat method berikut pada web_app/views.py untuk menampilkan dan menambahkan data pada model Contact.

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import GeeksModel, Employee, Album, Song, Contact, Student
from .forms import GeeksForm, EmployeeForm, SongForm, ContactFormSet, StudentFormSet, AlbumForm, AlbumFormSet
from django.views.generic.list import ListView
from django.views.generic.detail import DetailView
from django.views.generic.edit import CreateView, UpdateView, DeleteView
from django.urls import reverse_lazy

def contact_list(request):
    contacts = Contact.objects.all().order_by('id')
    return render(request, 'contact_list.html', {'contacts':contacts})

def contact_add(request):
    formset = ContactFormSet()
    if request.method == 'POST':
        formset = ContactFormSet(request.POST)
        if formset.is_valid():
            for form in formset:
                form.save()
            return redirect('web_app:contact_list')
    else:
        formset = ContactFormSet()
    return render(request, 'contact_form.html', {'formset':formset})
```

6. Buat routing dari method yang anda buat di web_app/urls.py

```
from django.urls import path, include
from web_app import views
from .views import GeeksList, EmployeeList, EmployeeCreate, EmployeeUpdate, EmployeeDelete, EmployeeDetail
app_name = 'web_app'

urlpatterns = [
    path('', views.list_view, name = 'list_view'),
    path('geeks_create', views.create_view, name = 'create_view'),
    path('geeks_detail/<id>', views.detail_view, name = 'geeks_detail'),
    path('geeks_update/<id>', views.update_view, name = 'geeks_update'),
    path('geeks_delete/<id>', views.delete_view, name = 'geeks_delete'),
    #path('', GeeksList.as_view()),
    path('employee_create', EmployeeCreate.as_view(), name = 'employee_create'),
    path('employee_list', EmployeeList.as_view(), name = 'employee_list'),
    path('employee_detail/<pk>', EmployeeDetail.as_view(), name = 'employee_detail'),
    path('employee_update/<pk>', EmployeeUpdate.as_view(), name = 'employee_update'),
    path('employee_delete/<pk>', EmployeeDelete.as_view(), name = 'employee_delete'),
    path('employee_add', views.employee_add, name = 'employee_add'),
    path('song_add', views.song_add, name = 'song_add'),
    path('contact_list', views.contact_list, name = 'contact_list'),
    path('contact_add', views.contact_add, name = 'contact_add'),
    path('student_list', views.student_list, name = 'student_list'),
    path('student_add', views.student_add, name = 'student_add'),
    path('album_list', views.album_list, name = 'album_list'),
    path('album_add', views.album_add, name = 'album_add'),
    path('album_update/<id>', views.album_update, name = 'album_update'),
]
```

7. Buat template dengan nama contact_form.html dan ketikkan script berikut ini.

```
<form method="POST">
    {% csrf_token %}
    {{ formset.as_p }}
    <hr>
    <input type="submit" value="Submit">
</form>
<a href="{% url 'web_app:contact_list' %}">
    <input type="submit" value="Cancel">
</a>
```

8. Buat template dengan nama contact_list.html dan ketikkan script berikut ini.

```
<a href="{% url 'web_app:contact_add' %}">
    <input type="button" value="New Contact">
</a>
<div class='main'>
    <br/>
    {% for data in contacts %}
        #{{ data.id }}<br/>
        {{ data.name }}<br/>
        {{ data.phone }}<br/>
        {{ data.email }}<br/>
        <hr/>
        {% empty %}
        <p>No Object yet</p>
    {% endfor %}
</div>
```

9. Restart project anda dan hasil akhirnya seperti ini dan ketikkan url : http://dosen.tia6.xyz/contact_list

← → ↻ Not secure | dosen.tia6.xyz/contact_list

New Contact

#1
Dedi Santoso
1234
dedi@yahoo.com

#2
Susi
123456
susi@gmail.com

#3
Ryan
52342
ryan@hotmail.com

#13
Wawan
2121
wawan@yahoo.com

#14
Surya
231231
surya@hotmail.com

#15
Kurniawan
24232
awan@gmail.com

10. Lakukan penambahan data dengan menekan tombol New Contact.

← → ↻ Not secure | dosen.tia6.xyz/contact_add

Name:

Phone:

Email:

Name:

Phone:

Email:

Name:

Phone:

Email:

Submit

Cancel

Anda perhatikan bahwa untuk melakukan data, maka otomatis form Contact akan menampilkan 3 (tiga) buah form, hal ini dikarenakan `extra = 3` pada **`ContactFormSet = formset_factory(ContactForm, extra = 3)`**. Setiap data yang anda masukkan maka akan ditampilkan pada **`contact_list`**.

LATIHAN `modelformset_factory`

1. Buka web hosting anda, lalu buka Files – File Manager
2. Cari **PROJECT ANDA SENDIRI**, di sini saya menggunakan project `dosen_latihan00`.
3. Buka file `web_app/models.py` dan ketikkan perintah berikut :

```
class Student(models.Model):
    gender_choices = (
        ('Male', 'Male'),
        ('Female', 'Female'),
    )
    name = models.CharField(max_length = 100)
    gender = models.CharField(max_length = 20, choices = gender_choices, default = 'Male')
    age = models.IntegerField()

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

4. Buka file `web_app/forms.py` dan ketikkan perintah berikut:

```
from django import forms
from django.forms import ModelForm, formset_factory, modelformset_factory, inlineformset_factory
from .models import GeeksModel, Employee, Album, Song, Contact, Student

# Contoh modelformset_factory
StudentFormSet = modelformset_factory(Student, fields=['name', 'gender', 'age'], extra = 2, can_delete = True)
```

Anda dapat perhatikan bahwa pada saat membuat `modelformset_factory` bahwa yang dideklarasikan adalah model, bukan form, dan harus mendeklarasikan field-field yang akan anda gunakan pada formset tersebut.

5. Buat method berikut pada `web_app/views.py` untuk menampilkan data dan menambahkan data pada model Student.

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import GeeksModel, Employee, Album, Song, Contact, Student
from .forms import GeeksForm, EmployeeForm, SongForm, ContactFormSet, StudentFormSet, AlbumForm, AlbumFormSet
from django.views.generic.list import ListView
from django.views.generic.detail import DetailView
from django.views.generic.edit import CreateView, UpdateView, DeleteView
from django.urls import reverse_lazy

def student_list(request):
    students = Student.objects.all().order_by('id')
    return render(request, 'student_list.html', {'students':students})

def student_add(request):
    formset = StudentFormSet()
    if request.method == 'POST':
        formset = StudentFormSet(request.POST)
        if formset.is_valid():
            formset.save()
            # for form in formset:
            #     form.save()
            return redirect('web_app:student_list')
    else:
        formset = StudentFormSet()
        return render(request, 'student_form.html', {'formset':formset})
```

6. Buat penambahan routing pada file `web_app/urls.py` seperti pada gambar berikut ini.

```

from django.urls import path, include
from web_app import views
from .views import GeeksList, EmployeeList, EmployeeCreate, EmployeeUpdate, EmployeeDelete, EmployeeDetail
app_name = 'web_app'

urlpatterns = [
    path('', views.list_view, name = 'list_view'),
    path('geeks_create', views.create_view, name = 'create_view'),
    path('geeks_detail/<id>', views.detail_view, name = 'geeks_detail' ),
    path('geeks_update/<id>', views.update_view, name = 'geeks_update'),
    path('geeks_delete/<id>', views.delete_view, name = 'geeks_delete'),
    #path('', GeeksList.as_view()),
    path('employee_create', EmployeeCreate.as_view(), name = 'employee_create'),
    path('employee_list', EmployeeList.as_view(), name = 'employee_list'),
    path('employee_detail/<pk>', EmployeeDetail.as_view(), name = 'employee_detail'),
    path('employee_update/<pk>', EmployeeUpdate.as_view(), name = 'employee_update'),
    path('employee_delete/<pk>', EmployeeDelete.as_view(), name = 'employee_delete'),
    path('employee_add', views.employee_add, name = 'employee_add'),
    path('song_add', views.song_add, name = 'song_add'),
    path('contact_list', views.contact_list, name = 'contact_list'),
    path('contact_add', views.contact_add, name = 'contact_add'),
    path('student_list', views.student_list, name = 'student_list'),
    path('student_add', views.student_add, name = 'student_add'),
    path('album_list', views.album_list, name = 'album_list'),
    path('album_add', views.album_add, name = 'album_add'),
    path('album_update/<id>', views.album_update, name = 'album_update'),
]

```

7. Buat template dengan nama student_form.html dan ketikkan script berikut ini.

```

<form method="POST">
    {% csrf_token %}
    {{ formset.as_p }}
    <input type="submit" value="Submit">
</form>
<a href="{% url 'web_app:student_list' %}">
    <input type="submit" value="Cancel">
</a>

```

8. Buat template dengan nama student_list.html dan ketikkan script berikut ini.

```

<a href="{% url 'web_app:student_add' %}">
    <input type="button" value="New Student">
</a>
<div class='main'>
    <br/>
    {% for data in students %}
        #{{ data.id }}<br/>
        {{ data.name }}<br/>
        {{ data.gender }}<br/>
        {{ data.age }}<br/>
        <hr/>
        {% empty %}
        <p>No Object yet</p>
    {% endfor %}
</div>

```

9. Restart project anda dan hasil akhirnya seperti ini dan ketikkan url : http://dosen.tia6.xyz/student_list

← → 🔍 Not secure | dosen.tia6.xyz/student_list

New Student

#1	Rudy
	Male
	25
#2	Tuti
	Female
	20

11. Lakukan penambahan data dengan menekan tombol New Student.

← → ↻ ⓘ Not secure | dosen.tia6.xyz/student_add

Name: Rudy

Gender: Male ▾

Age: 25

Delete: ☐

Name: Tuti

Gender: Female ▾

Age: 20

Delete: ☐

Name:

Gender: Male ▾

Age:

Delete: ☐

Name:

Gender: Male ▾

Age:

Delete: ☐

Anda perhatikan bahwa untuk melakukan data, ketika anda menambahkan kembali data, maka data yang lama akan turut di tampilkan juga. Anda dapat melakukan penghapusan data dengan memberikan centang pada Delete dan menekan tombol Submit untuk melakukan eksekusi penghapusan data.

LATIHAN inlineformset_factory

1. Buka web hosting anda, lalu buka Files – File Manager
2. Cari **PROJECT ANDA SENDIRI**, di sini saya menggunakan project dosen_latihan00.
3. Pada bagian models, disini saya menggunakan models dari Album dan Song yang telah dikerjakan sebelumnya.
4. Buka file web_app/forms.py dan ketikkan perintah berikut:

```
from django import forms
from django.forms import ModelForm, formset_factory, modelformset_factory, inlineformset_factory
from .models import GeeksModel, Employee, Album, Song, Contact, Student

# Contoh inlineformset_factory
class AlbumForm(ModelForm):
    class Meta:
        model = Album
        fields = '__all__'

class SongsForm(ModelForm):
    class Meta:
        model = Song
        fields = '__all__'

AlbumFormSet = inlineformset_factory(Album, Song, form = SongsForm, can_delete = True, extra = 2)
```

Anda dapat perhatikan bahwa pada saat membuat inlineformset_factory, model Song terkait dengan model Album, dimana hal ini dapat dilihat dari field album pada model Song yang bersifat ForeignKey. Pada bagian ini juga, harus mendeklarasikan form yang akan digunakan sebagai formset yaitu : SongsForm, sedangkan **can_delete = True** menyatakan bahwa formset tersebut dapat melakukan penghapusan data.

5. Buat method berikut pada web_app/views.py untuk menampilkan data, menambahkan data dan memperbaharui data pada model Album dan Song.

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import GeeksModel, Employee, Album, Song, Contact, Student
from .forms import GeeksForm, EmployeeForm, SongForm, ContactFormSet, StudentFormSet, AlbumForm, AlbumFormSet
from django.views.generic.list import ListView
from django.views.generic.detail import DetailView
from django.views.generic.edit import CreateView, UpdateView, DeleteView
from django.urls import reverse_lazy
```

```

def album_list(request, template_name = 'album_list.html'):
    albums = Album.objects.all().order_by('id')
    context = {'albums':albums}
    return render(request, template_name, context)

def album_add(request, template_name = 'album_form.html'):
    tombol = 'Save'
    form = AlbumForm()
    formset = AlbumFormSet()
    if request.method == 'POST':
        form = AlbumForm(request.POST)
        formset = AlbumFormSet(request.POST)
        if form.is_valid() and formset.is_valid():
            album = form.save()
            formset.instance = album
            formset.save()
            return redirect('web_app:album_list')
    else:
        form = AlbumForm()
        formset = AlbumFormSet()
    context = {
        'form':form,
        'formset':formset,
        'tombol':tombol,
    }
    return render(request, template_name, context)

def album_update(request, id, template_name = 'album_form.html'):
    tombol = 'Update'
    albums = Album.objects.get(id = id)
    form = AlbumForm(instance = albums)
    formset = AlbumFormSet(instance = albums)
    if request.method == 'POST':
        form = AlbumForm(request.POST, instance = albums)
        formset = AlbumFormSet(request.POST, instance = albums)
        if form.is_valid() and formset.is_valid():
            album = form.save()
            formset.instance = album
            formset.save()
            return redirect('web_app:album_update', id = id)
    context = {
        'form':form,
        'formset':formset,
        'tombol':tombol,
    }
    return render(request, template_name, context)

```

Anda perhatikan bahwa ada suatu variabel yang bernama **tombol**. Variabel **tombol** digunakan untuk membuat penamaan nilai tombol akan disesuaikan ketika saat melakukan penambahan data baru maka nilai tombol menjadi **Save**, sedangkan saat melakukan pembaharuan data maka nilai tombol menjadi **Update**.

6. Buat penambahan routing pada file web_app/urls.py seperti pada gambar berikut ini.


```

from django.urls import path, include
from web_app import views
from .views import GeeksList, EmployeeList, EmployeeCreate, EmployeeUpdate, EmployeeDelete, EmployeeDetail
app_name = 'web_app'

urlpatterns = [
    path('', views.list_view, name = 'list_view'),
    path('geeks_create', views.create_view, name = 'create_view'),
    path('geeks_detail/<id>', views.detail_view, name = 'geeks_detail' ),
    path('geeks_update/<id>', views.update_view, name = 'geeks_update'),
    path('geeks_delete/<id>', views.delete_view, name = 'geeks_delete'),
    #path('', GeeksList.as_view()),
    path('employee_create', EmployeeCreate.as_view(), name = 'employee_create'),
    path('employee_list', EmployeeList.as_view(), name = 'employee_list'),
    path('employee_detail/<pk>', EmployeeDetail.as_view(), name = 'employee_detail'),
    path('employee_update/<pk>', EmployeeUpdate.as_view(), name = 'employee_update'),
    path('employee_delete/<pk>', EmployeeDelete.as_view(), name = 'employee_delete'),
    path('employee_add', views.employee_add, name = 'employee_add'),
    path('song_add', views.song_add, name = 'song_add'),
    path('contact_list', views.contact_list, name = 'contact_list'),
    path('contact_add', views.contact_add, name = 'contact_add'),
    path('student_list', views.student_list, name = 'student_list'),
    path('student_add', views.student_add, name = 'student_add'),
    path('album_list', views.album_list, name = 'album_list'),
    path('album_add', views.album_add, name = 'album_add'),
    path('album_update/<id>', views.album_update, name = 'album_update'),
]

```

7. Buat template dengan nama album_form.html dan ketikkan script berikut ini.

```

<form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    {{ formset.management_form }}
    <table class="table" id="id_forms_table">
        <thead class="text-center">
            <tr>
                <th>Title</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            {% for p in formset %}
                <tr>
                    <td>{{ p.id }}</td>
                    <td>{{ p.title }}</td>
                    <td>{{ p.DELETE }} Delete</td>
                </tr>
            {% endfor %}
        </tbody>
    </table>
    <input type="submit" value="{{ tombol }}">
</form>
<a href="{% url 'web_app:album_list' %}">
    <input type="submit" value="Cancel">
</a>

```

{{ formset.management_form }} : menyatakan bahwa form tsb berupa formset.

8. Buat template dengan nama album_list.html dan ketikkan script berikut ini.

```

<a href="{% url 'web_app:album_add' %}">
    <input type="button" value="New Album">
</a>
<div class='main'>
    <br/>
    {% for data in albums %}
        <div>
            <div>
                <div>{{ data.id }}</div>
                <div>{{ data.title }}</div>
                <div>{{ data.artist }}</div>
            </div>
            <a href="{% url 'web_app:album_update' data.id %}">
                <input type="button" value="Update">
            </a>
        </div>
    </div>
    <hr/>
    {% empty %}
    <p>No Object yet</p>
    {% endfor %}
</div>

```

9. Restart project anda dan hasil akhirnya seperti ini dan ketikkan url : http://dosen.tia6.xyz/album_list

← → ↻ ⓘ Not secure | dosen.tia6.xyz/album_list

New Album

#1
Pemrograman I
Kevin

Update

#2
Album 1
Regina

Update

12. Lakukan penambahan data dengan menekan tombol New Album.

← → ↻ ⓘ Not secure | dosen.tia6.xyz/album_add

Title:

Artist:

Title	Action
<input type="text"/>	<input type="checkbox"/> Delete
<input type="text"/>	<input type="checkbox"/> Delete

Save

Cancel

13. Berikut ini adalah pembaharuan data ketika anda menekan tombol Update.

← → ↻ ⓘ Not secure | dosen.tia6.xyz/album_update/2/

Title:

Artist:

Title	Action
<input type="text" value="Suka-suka"/>	<input type="checkbox"/> Delete
<input type="text" value="AA"/>	<input type="checkbox"/> Delete
<input type="text"/>	<input type="checkbox"/> Delete
<input type="text"/>	<input type="checkbox"/> Delete

Update

Cancel

Ketika pertama kali anda memasukkan data baru pada formset maka akan disediakan 2 (dua) buah form dan jika anda ingin menambahkan data baru kembali pada formset maka anda dapat melakukannya dengan menekan tombol **Update** pada album_list. Anda dapat melakukan penghapusan data dengan memberikan centang pada Delete dan menekan tombol Submit untuk melakukan eksekusi

penghapusan data. Penghapusan data bersifat permanen dan data tidak dapat dikembalikan lagi.