



Django Formsets

Last Updated : 09 Jan, 2020

Formsets in a Django is an advanced way of handling multiple forms on a single webpage. In other words, Formsets are a group of forms in Django. One might want to initialize multiple forms on a single page all of which may

[Django](#) [Views](#) [Model](#) [Template](#) [Forms](#) [Jinja](#) [Python SQLite](#) [Flask](#) [Json](#) [Postman](#) [Interview Ques](#)

```
from django import forms
class GeeksForm(forms.Form):
    title = forms.CharField()
    pub_date = forms.DateField()
```

Now one might want to permit the user to create articles at once, so if thought in a conventional manner one uses multiple forms and different names for each form to handle data on a single page but this would complicate the code as well as functionality. A formset is a layer of abstraction to work with multiple forms on the same page. It can be best compared to a data grid.

Now to create a formset of this **GeeksForm**,

```
from django.forms import formset_factory
GeeksFormSet = formset_factory(GeeksForm)
```

Creating and using Django Formsets

Illustration of **Rendering Django Forms Manually** using an Example. Consider a project named geeksforgeeks having an app named geeks.

Refer to the following articles to check how to create a project and an app in Django.

- [How to Create a Basic Project using MVT in Django?](#)
- [How to Create an App in Django ?](#)

In your geeks app make a new file called forms.py where you would be making all your forms. To create a Django form you need to use [Django Form Class](#).

Let's demonstrate how,

In your forms.py Enter the following,

```
from django import forms

# create a form
class GeeksForm(forms.Form):
    title = forms.CharField()
    description = forms.CharField()
```

Let's explain what exactly is happening, left side denotes the name of the field and to right of it, you define various functionalities of an input field correspondingly. A field's syntax is denoted as

Syntax :

```
Field_name = forms.FieldType(attributes)
```

Now to create a simple formset of this form, move to views.py and create a formset_view as below.

```
from django.shortcuts import render

# relative import of forms
from .forms import GeeksForm

# importing formset_factory
from django.forms import formset_factory

def formset_view(request):
    context = {}

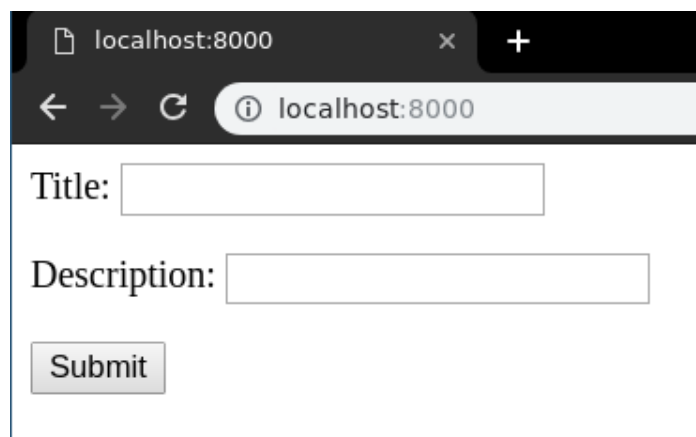
    # creating a formset
    GeeksFormSet = formset_factory(GeeksForm)
    formset = GeeksFormSet()

    # Add the formset to context dictionary
    context['formset'] = formset
    return render(request, "home.html", context)
```

To render the formset through HTML, create a html file “home.html”. Now let’s edit templates > home.html

```
<form method="POST" enctype="multipart/form-data">
    {% csrf_token %}
    {{ formset.as_p }}
    <input type="submit" value="Submit">
</form>
```

All set to check if our formset is working or not let’s visit <http://localhost:8000/>.



The screenshot shows a web browser window with the address bar set to localhost:8000. The page content includes a form with two text input fields. The first field is labeled 'Title:' and the second is labeled 'Description:'. Below these fields is a button labeled 'Submit'.

Our formset is working completely. Let’s learn how to modify this formset to use extra features of this formset.

How to create Multiple forms using Django Formsets

Django formsets are used to handle multiple instances of a form. One can create multiple forms easily using extra attribute of Django Formsets. In `geeks/views.py`,

```
from django.shortcuts import render

# relative import of forms
from .forms import GeeksForm

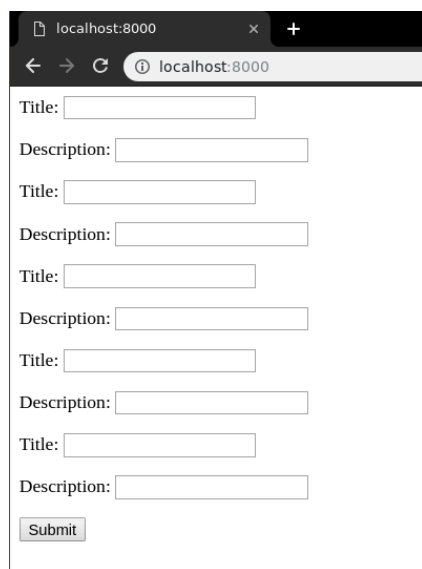
# importing formset_factory
from django.forms import formset_factory

def formset_view(request):
    context = {}
```

```
# creating a formset and 5 instances of GeeksForm
GeeksFormSet = formset_factory(GeeksForm, extra = 5)
formset = GeeksFormSet()

# Add the formset to context dictionary
context['formset'] = formset
return render(request, "home.html", context)
```

The keyword argument `extra` makes multiple copies of same form. If one wants to create 5 forms enter `extra = 5` and similarly for others. Visit <http://localhost:8000/> to check if 5 forms are created.



The screenshot shows a web browser window with the address bar set to 'localhost:8000'. The page displays five identical forms stacked vertically. Each form consists of a 'Title:' label followed by a text input field, and a 'Description:' label followed by a text input field. At the bottom of the forms is a 'Submit' button.

Handling Multiple forms using Django Formsets

Creating a form is much easier than handling the data entered into those fields at the back end. Let's try to demonstrate how one can easily use the data of a formset in a view. When trying to handle formset, Django formsets required one extra argument `{{ formset.management_data }}`. To know more about Management data, visit [Understanding the ManagementForm](#).

In `templates/home.html`,

```
<form method="POST" enctype="multipart/form-data">

    <!-- Management data of formset -->
    {{ formset.management_data }}

    <!-- Security token -->
    {% csrf_token %}
```

```
<!-- Using the formset -->
{{ formset.as_p }}

<input type="submit" value="Submit">
</form>
```

Now to check how and what type of data is being rendered edit `formset_view` to print the data. In `geeks/view.py`,

```
from django.shortcuts import render

# relative import of forms
from .forms import GeeksForm

# importing formset_factory
from django.forms import formset_factory

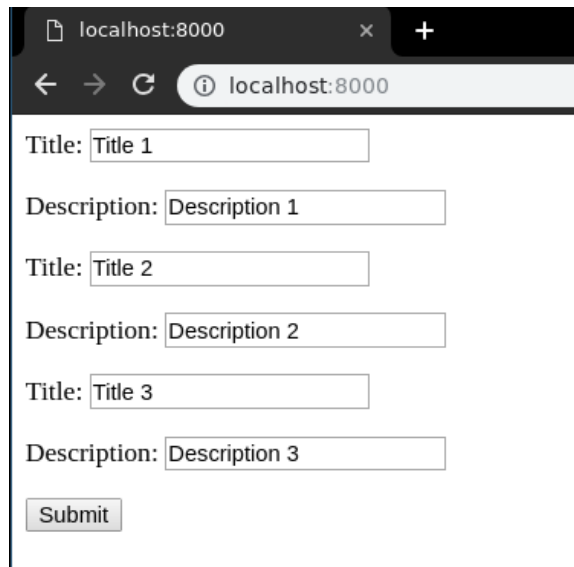
def formset_view(request):
    context = {}

    # creating a formset and 5 instances of GeeksForm
    GeeksFormSet = formset_factory(GeeksForm, extra = 3)
    formset = GeeksFormSet(request.POST or None)

    # print formset data if it is valid
    if formset.is_valid():
        for form in formset:
            print(form.cleaned_data)

    # Add the formset to context dictionary
    context['formset'] = formset
    return render(request, "home.html", context)
```

Now let's try to enter data in the formset through <http://localhost:8000/>



The screenshot shows a web browser window with the address bar set to localhost:8000. The page contains a form with three rows of input fields. Each row has a 'Title' field and a 'Description' field. The first row contains 'Title 1' and 'Description 1', the second row contains 'Title 2' and 'Description 2', and the third row contains 'Title 3' and 'Description 3'. At the bottom of the form is a 'Submit' button.

Hit submit and data will be display in command line where server is running.
One can use this data in any manner conveniently now.

```
System check identified no issues (0 silenced).
January 01, 2020 - 14:44:40
Django version 2.2.7, using settings 'geeksforgeeks.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[01/Jan/2020 14:44:43] "GET / HTTP/1.1" 200 1425
{'title': 'Title 1', 'description': 'Description 1'}
{'title': 'Title 2', 'description': 'Description 2'}
{'title': 'Title 3', 'description': 'Description 3'}
[01/Jan/2020 14:45:16] "POST / HTTP/1.1" 200 1539
```


Formset is advanced stuff which can be used to resolve a number of problems but should be used with correct syntax and field validations otherwise conflicts and errors will disrupt the normal functioning. To know more about Formsets, Visit [Official Documentation for Formsets](#).

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](#) on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you

to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

N

Nave...



16

Previous Article

Render Django Form Fields Manually

Next Article

Django Forms

Similar Reads

Adding Tags Using Django-Taggit in Django Project

Django-Taggit is a Django application which is used to add tags to blogs, articles etc. It makes very easy for us to make adding the tags functionality to our django...

2 min read

How to customize Django forms using Django Widget Tweaks ?

Django forms are a great feature to create usable forms with just few lines of code. But Django doesn't easily let us edit the form for good designs. Here, we...

3 min read

Styling Django Forms with django-crispy-forms

Django by default doesn't provide any Django form styling method due to which it takes a lot of effort and precious time to beautifully style a form. django-crispy...

1 min read

Integrating Django with Reactjs using Django REST Framework

In this article, we will learn the process of communicating between the Django Backend and React js frontend using the Django REST Framework. For the sake...

15+ min read

How to Fix Django "ImportError: Cannot Import Name 'six' from..."

After Django 3.0, django.utils.six was removed. The "ImportError: Cannot Import Name 'six' from 'django.utils'" error occurs because of the dependency issue. Thi...

3 min read

Handling Ajax request in Django

Introduction This tutorial explains how to carry out an ajax request in the Django web framework. We will create a simple post-liking app as a part of the exampl...

4 min read

Django Introduction | Set 2 (Creating a Project)

Note- This article is in continuation of Django introduction. Popularity of Django Django is used in many popular sites like as: Disqus, Instagram, Knight...

3 min read

now - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

2 min read

Python | User groups with Custom permissions in Django

Let's consider a trip booking service, how they work with different plans and packages. There is a list of product which subscriber gets on subscribing to...

4 min read

Python | Django Admin Interface

In this article, we delve into the capabilities and advantages of the Django Admin Interface, exploring how its customizable features and streamlined workflows...

4 min read

Article Tags : [Python](#) [Django-forms](#) [Python Django](#)

Practice Tags : [python](#)



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305



[Company](#)

[Languages](#)

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

System Design Bootcamp
Interview Questions

School Subjects

- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar
- Commerce
- World GK

GeeksforGeeks Videos

- DSA
- Python
- Java
- C++
- Web Development
- Data Science
- CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved