



# FormView – Class Based Views Django

Last Updated : 25 Jul, 2022

---

FormView refers to a view (logic) to display and verify a Django Form. For example, a form to register users at Geeksforgeeks. Class-based views provide an alternative way to implement views as [Python](#) objects instead of functions. They do not replace function-based views, but have certain differences and advantages when compared to function-based views:

- The organization of code related to specific [HTTP methods](#) ([GET](#), [POST](#), etc.) can be addressed by separate methods instead of conditional branching.
- Object-oriented techniques such as mixins (multiple inheritances) can be used to factor code into reusable components.

Class-based views are simpler and more efficient to manage than function-based views. A function-based view with tons of lines of code can be converted into a class-based view with few lines only. This is where [Object Oriented Programming](#) comes into impact.

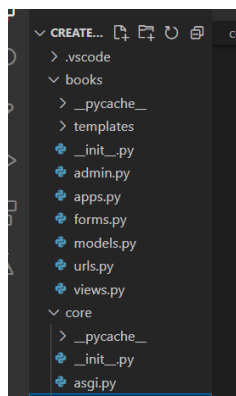
## [Django](#) FormView – Class-Based Views

Illustration of **How to create and use FormView** using an Example. Consider a project named Core having an app named books.

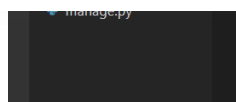
*Refer to the following articles to check how to create a project and an app in Django.*

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

## Folder Structure



Django Views Model Template Forms Jinja Python SQLite Flask Json Postman Interview Ques



## Stepwise Implementation to create Class-Based Views

### Step 1: Create a basic Project

Create a project by following the command

```
django-admin startproject core
```

- A New Folder with the name *project name* will be created.

### Step 2: Creating an App in Django

To create a basic app in your Django project you need to go to the directory containing `manage.py` and from there enter the command:

```
python manage.py startapp books
```

**Step 3:** Go to `core/URLs` and set the path for our app.

```
1  ✓ from django.contrib import admin
2    from django.urls import path, include
3
4  ✓ urlpatterns = [
5      path('admin/', admin.site.urls),
6      path('', include('books.urls', namespace='books'))
7  ]
8  |
```

**Step 4:** Go to core/setting and register our app name “books”.

```
# Application definition
✓ INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'books',
]
```

**Step 5:** Paste this to your books/admin.py

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

@admin.register(models.Books)
class AuthorAdmin(admin.ModelAdmin):
    prepopulated_fields = {'slug': ('title',), }
```

**Step 6:** Create a model in books/models.py.

```
from django.db import models
from django.template.defaultfilters import slugify

class Books(models.Model):
    # Example model only
    title = models.CharField(max_length=100)
    slug = models.SlugField(null=True)
    genre = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    isbn = models.IntegerField()
    count = models.IntegerField(null=True, default=0)
```

**Step 7:** After you have a model, Let's create a form in which we will be creating `CreateView`. In `books/forms.py`.

```
1  from django import forms
2  from .models import Books
3
4
5  class AddForm(forms.ModelForm):
6
7      class Meta:
8          model = Books
9          fields = ('title', 'genre', 'author', 'isbn')
10
11      widgets = {
12          'title': forms.TextInput(attrs={'class': 'form-control'}),
13          'genre': forms.TextInput(attrs={'class': 'form-control'}),
14          'author': forms.TextInput(attrs={'class': 'form-control'}),
15          'isbn': forms.TextInput(attrs={'class': 'form-control'}),
16      }
17
```

**Step 8:** After creating the form let's create `CreateView` in `geeks/views.py`,

```
from django.views.generic.edit import CreateView
from . forms import AddForm

class AddBookView(CreateView):
    # model = Books
    form_class = AddForm
    template_name = 'add.html'
    success_url = '/books/'
```

**Step 9:** Map a URL to this view in books/urls.py.

```
from django.urls import path
from .views import AddBookView

app_name = 'books'

urlpatterns = [
    path('', AddBookView.as_view(), name='add'),
]
```

**Step 10:** Set Urls for books/urls.py

```
from django.urls import path
from .views import AddBookView

app_name = 'books'

urlpatterns = [
    path('', AddBookView.as_view(), name='add'),
]
```

**Step 11:** Create a template for this view in books/add.html,

---

html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="container" style="max-width:600px">
    <div class="px-3 py-3 pt-md-5 pb-md-4 mx-auto text-center">
      <h1 class="display-4">Welcome to GFG Class Based Views Django</h1>

    </div>
    <div class="py-5">
      <div class="row">
        <div class="col-12">
          <form method="post">
            {% csrf_token %}
            {{ form.as_p }}
            <input type="submit">
          </form>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

**Step 12:** Now run your app by the following command.

```
python manage.py runserver
```

Now visit <http://127.0.0.1:8000/>,

[←](#) [↻](#) 127.0.0.1:8000

## Welcome to GFG Class Based Views Django

Title:

Genre:

Author:

Isbn:

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...](#)

6

### Previous Article

[UpdateView - Class Based Views Django](#)

### Next Article

## Similar Reads

### Class Based vs Function Based Views - Which One is Better to Use in Django?

Django...We all know the popularity of this python framework all over the world. This framework has made life easier for developers. It has become easier for...

7 min read

## Createview - Class Based Views Django

Create View refers to a view (logic) to create an instance of a table in the database. We have already discussed basics of Create View in Create View –...

3 min read

## ListView - Class Based Views Django

List View refers to a view (logic) to display multiple instances of a table in the database. We have already discussed the basics of List View in List View –...

4 min read

## UpdateView - Class Based Views Django

UpdateView refers to a view (logic) to update a particular instance of a table from the database with some extra details. It is used to update entries in the databas...

3 min read

## DetailView - Class Based Views Django

Detail View refers to a view (logic) to display one instances of a table in the database. We have already discussed basics of Detail View in Detail View –...

3 min read

## DeleteView - Class Based Views Django

Delete View refers to a view (logic) to delete a particular instance of a table from the database. It is used to delete entries in the database for example, deleting a...

3 min read

## Class based views - Django Rest Framework

Class-based views help in composing reusable bits of behavior. Django REST Framework provides several pre-built views that allow us to reuse common...

12 min read

## Django Class Based Views

Django is a Python-based web framework that allows you to quickly create web applications. It has a built-in admin interface which makes it easy to work with it...



10 min read

## How to Use permission\_required Decorators with Django Class-Based Views

In Django, permissions are used to control access to views and resources. When working with function-based views (FBVs), decorators like `@permission_require...`

4 min read

## Create View - Function based Views Django

Create View refers to a view (logic) to create an instance of a table in the database. It is just like taking an input from a user and storing it in a specified...

3 min read

Article Tags :

[Python](#)[Django-views](#)[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

[About Us](#)[Legal](#)[In Media](#)[Contact Us](#)[Advertise with us](#)

### Languages

[Python](#)[Java](#)[C++](#)[PHP](#)[GoLang](#)

GFG Corporate Solution  
Placement Training Program  
GeeksforGeeks Community

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

## Web Technologies

HTML  
CSS  
JavaScript  
TypeScript  
ReactJS  
NextJS  
Bootstrap  
Web Design

## Computer Science

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

## System Design

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns  
OOAD  
System Design Bootcamp  
Interview Questions

## School Subjects

Mathematics  
Physics

## Data Science & ML

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

## Python Tutorial

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

## DevOps

Git  
Linux  
AWS  
Docker  
Kubernetes  
Azure  
GCP  
DevOps Roadmap

## Interview Preparation

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

## GeeksforGeeks Videos

DSA  
Python

Chemistry	Java
Biology	C++
Social Science	Web Development
English Grammar	Data Science
Commerce	CS Subjects
World GK	

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved