



# Django CSV Output

Last Updated : 17 Oct, 2023

Generating CSV files in a Django web application can be achieved easily with the right libraries and a structured approach. By following the steps outlined in this article, you can provide users with the ability to download data in these popular formats, enhancing the usability and functionality of your Django application.

## Step to Create CSV Output In Django

You can [install Django](#) by following these steps.

**Step 1:** First make a project by using the command :

```
django-admin startproject csv_downloader
cd csv_downloader
```

**Step 2:** Create an application named 'gfg' by using the command

```
python3 manage.py startapp gfg
```

The next step is to register your app in setting.py

```
# Application definition

INSTALLED_APPS = [
```

[Python Basics](#) [Interview Questions](#) [Python Quiz](#) [Popular Packages](#) [Python Projects](#) [Practice Python](#) [AI Wit](#)

```
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

**Step 2:** Create all the necessary files a mentioned below.

**gfg/view.py:** The “home” view handles HTTP requests, checking for POST submissions. If valid, it saves book data to the database and redirects to the home page. For GET requests, it displays an empty book input form via an HTML template ('myapp/create\_user\_profile.html') with the form as context data.

The generate\_csv view generates a CSV file as an HTTP response, sets the content type to 'text/csv' with the filename “book\_catalog.csv,” writes column names as the header, fetches all books from the database, writes book details to the CSV, and serves the file for user download upon accessing the view.

---

## Python3

```
from django.http import FileResponse
from reportlab.pdfgen import canvas
from .models import Book
from django.shortcuts import redirect, render
from .forms import BookForm

def home(request):
    if request.method == 'POST':
        form = BookForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('home') # Redirect to PDF generation after adding a l
    else:
        form = BookForm()
    return render(request, 'myapp/create_user_profile.html', {'form': form})

# csvapp/views.py
import csv
from django.http import HttpResponse
from .models import Book

def generate_csv(request):
    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = 'attachment; filename="book_catalog.csv"'

    writer = csv.writer(response)
    writer.writerow(['Title', 'Author', 'Publication Year'])

    books = Book.objects.all()
    for book in books:
        writer.writerow([book.title, book.author, book.publication_year])
```

**return** response

**gfg/forms.py:** The provided code defines a Django form called BookForm using the forms.ModelForm class. It's designed to work with the Book model in a Django application. The form includes fields for 'title', 'author', and 'publication\_year', which are derived from the corresponding model fields.

---

## Python3

```
from django import forms
from .models import Book

class BookForm(forms.ModelForm):
    class Meta:
        model = Book
        fields = ['title', 'author', 'publication_year']
```

**gfg/model.py:** This code defines a Django model named Book with three fields: title (for the book title), author (for the author's name), and publication\_year (for the publication year of the book). It also includes a \_\_str\_\_ method to display the book's title as its string representation.

---

## Python3

```
from django.db import models

class Book(models.Model):
    title = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    publication_year = models.PositiveIntegerField()

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

**gfg/urls.py:** This is the URL file inside the test app .The first path is used to direct to the home page and the second is used to generate the CSV file.

## Python3

```
# pdfapp/urls.py
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('generate-csv/', views.generate_csv, name='generate_csv'),
]
```

**urls.py:** This is the main url file. Here we have included the app URLs by the help of “include”

---

## Python3

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

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

## Setting up GUI

**Html file:** This file is used to Collect the book information from the user and download the csv file after collecting all the data from the user.

---

## Python3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Generate PDF</title>
</head>
<body>
```

```

<h1>Generate and Download CSV</h1>
<h2>Add a New Book</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Add Book</button>
</form>
<!-- <p><a href="{% url 'home' %}" download>Download Book Catalog (PDF)</a></p>
<p><a href="{% url 'generate_csv' %}" download>Download Book Catalog (CSV)</a>
<br>

</body>
</html>

```

## Deployment of the Project

Run these commands to apply the migrations:

```
python3 manage.py makemigrations
python3 manage.py migrate
```

Run the server with the help of following command:

```
python3 manage.py runserver
```

## Output

← → ↻ ⓘ 127.0.0.1:8000

## Generate and Download CSV

### Add a New Book

Title:

Author:

Publication year:

[Download Book Catalog \(CSV\)](#)

	Standard	Standard	Standard
1	Title	Author	Publication Year
2	GeeksforGeeks	Puja	2023
3	GeeksforGeeks	Puja kr	2023
4	hello python	Misti	2012
5	youtube	Harsh	2012
6	The User	Garima	2022

Looking to dive into the world of programming or sharpen your Python skills? Our [Master Python: Complete Beginner to Advanced Course](#) is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!

P prakr...



Previous Article

Next Article

Merge multiple CSV files using R

Similar Reads

## How to create multiple CSV files from existing CSV file using Pandas ?

In this article, we will learn how to create multiple CSV files from existing CSV file using Pandas. When we enter our code into production, we will need to deal...

3 min read

## How to create CSV output in Flask?

In this article, we will see how to create a CSV output using Flask. Python and Flask Web Frameworks provide powerful tools to generate a report, export data...

3 min read

## 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

## How to resolve a UnicodeDecodeError for a CSV file in Python?

Several errors can arise when an attempt to decode a byte string from a certain coding scheme is made. The reason is the inability of some encoding schemes to...

5 min read

## Python - Read CSV Column into List without header

Prerequisites: Reading and Writing data in CSV CSV files are parsed in python with the help of csv library. The csv library contains objects that are used to rea...

2 min read

## Using csv module to read the data in Pandas

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. There were various formats ...

3 min read

Article Tags :

[Geeks Premier League](#)

[Project](#)

[Python](#)

[Geeks Premier League 2023](#)

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

Inteivew 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