



---

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

---

# How to build a URL Shortener with Django ?

Last Updated : 24 Sep, 2024

---

Building a URL Shortener, Is one of the Best Beginner Project to Hone your Skills. In this article, we have shared the steps to build a URL shortener using Django Framework. To know more about Django visit – [Django Tutorial](#)

## Setup

We need some things setup before we start with our project. We will be using Virtual Environment for our project.

```
pip install virtualenv
virtualenv urlShort
source urlShort/bin/activate
```

Above Command will create, activate virtual environment named *urlShort*.

## Installing Important Packages

We need to Install some packages before hand,

```
pip install django
```

## Starting With Our Project

First of all, We need to create our project by,

```
django-admin startproject urlShort
cd urlShort
```

Above Command, Creates A Django Project and then cd into that directory. After that, We also need to create an app inside of our project. App is sort of a

container, where we will store our code. A project can have Multiple Apps and they can be interconnected

```
python manage.py startapp url
```

Above Command Creates an App named URL in our Project. Our File Structure Now will be —

```
urlShort
├── manage.py
├── url
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py
│   ├── tests.py
│   └── views.py
└── urlShort
    ├── asgi.py
    ├── __init__.py
    ├── __pycache__
    │   ├── __init__.cpython-37.pyc
    │   └── settings.cpython-37.pyc
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

## Checking If all is Great...

You can check if all is working by just typing this in Command Line. But cd into the main folder, here urlShort.

```
python manage.py runserver
```

*runserver* will run a local server where our website will load. Move to url

```
https://localhost:8000
```

Keep Your Console Window Open.

## How to Build a URL Shortener Project?

Building a URL shortener is a useful project for learning Django. If you're eager to develop more advanced Django apps, the [Django Web Development Course](#) is designed to help you tackle more complex projects.

Now, tighten your seat belts as we are starting to Code. First of all, we will play with *views.py*. *views.py* is basically used to connect our database, api with our Frontend. Open *views.py* and type

```
from django.http import HttpResponse
```

```
def index(request):  
    return HttpResponse("Hello World")
```

Save it and open localhost and check if it changes. It does not change because we have not mapped it to any route. Basically, if you write any function inside *views.py* it does not work but we need to map it inside *urls.py*. So, create a *urls.py* inside *url* folder.

```
from django.urls import path  
from . import views  
app_name = "url"  
urlpatterns = [  
    path("", views.index, name="home")  
]
```

Don't forget to add your app – "url" in *INSTALLED\_APPS* in *settings.py*

## Creating Django Models –

First of all, we need a Database to store our Shortened URL's. For that, we need to create a Schema for our Database Table in *models.py*.

*models.py*

```
from django.db import models  
class UrlData(models.Model):
```

```
url = models.CharField(max_length=200)
slug = models.CharField(max_length=15)
def __str__(self):
    return f"Short Url for: {self.url} is {self.slug}"
```

Above Code Creates a Table *UrlData* in our Database with Columns *url*, *slug*. We will use *url* column to store Original URL and *slug* to store 10-character string which will be used for shortening the URL.

*For Example,*

*Original URL* – <https://medium.com/satyam-kulkarni/>  
*Shorten Form* – <https://localhost:8000/sEqLKdsIUL>

URL's maximum length is 200 Characters and Slug's maximum length is 15 (Considering our Website's Address). After Creating Models for our Website, Let's create some Forms for taking input from User.

## Creating a Form

Create a *forms.py* in our Django App Folder.

*forms.py*

```
from django import forms
class Url(forms.Form):
    url = forms.CharField(label="URL")
```

We simply import *forms* from *django* and create a Class *Url* which we will use in *views.py* and render it in our HTML. *Url* form has only a *url* field to take input of Original URL.

## Creating Views

Now, We will create the Interface of our App using *views.py*. Let's divide this part in Functions.

***urlShort()***— This Function is where our *Main Algorithm* works. It takes a *url* from form after User submits it, then it generates a Random Slug which is then

stored in Database with Original Url. It is also the function which render *index.html* (entrypoint of our app)

*views.py urlShort()*

```
def urlShort(request):
    if request.method == 'POST':
        form = Url(request.POST)
        if form.is_valid():
            slug = ''.join(random.choice(string.ascii_letters)
                           for x in range(10))
            url = form.cleaned_data["url"]
            new_url = UrlData(url=url, slug=slug)
            new_url.save()
            request.user.urlshort.add(new_url)
            return redirect('/')
        else:
            form = Url()
            data = UrlData.objects.all()
            context = {
                'form': form,
                'data': data
            }
            return render(request, 'index.html', context)
```

*urlRedirect()* — This Function tracks the slug to Original URL and redirects it to Original URL.

*views.py urlRedirect()*

```
def urlRedirect(request, slugs):
    data = UrlData.objects.get(slug=slugs)
    return redirect(data.url)
```

## Creating Routes

Before Running This App, We need to specify URL paths in App's *urls.py*

*urls.py*

```

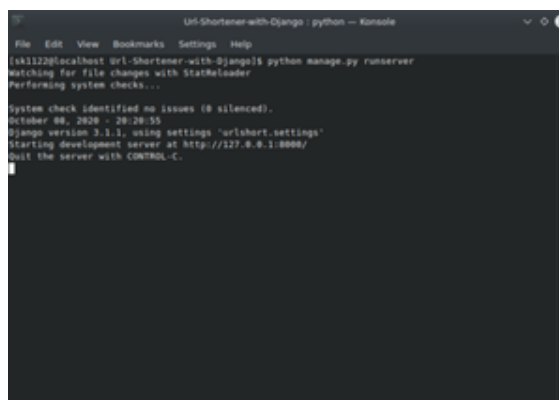
from django.urls import path
from . import views
app_name = "url"
urlpatterns = [
    path("", views.urlShort, name="home"),
    path("u/<str:slugs>", views.urlRedirect, name="redirect")
]

```

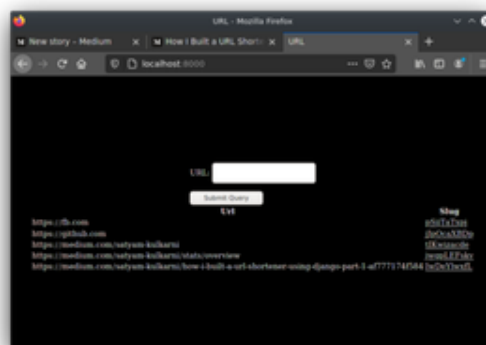
## Run the Project

Open Console in Main Project Directory.

```
python manage.py runserver
```



## Final Project Output –



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!

**S** sk1122

3

## Next Article

Build a URL Size Reduce App with Django

## Similar Reads

### How to Add URL Parameters to Django Template URL Tag?

When building a Django application, it's common to construct URLs dynamically, especially when dealing with views that require specific parameters. The Django...

4 min read

### Comparing path() and url() (Deprecated) in Django for URL Routing

When building web applications with Django, URL routing is a fundamental concept that allows us to direct incoming HTTP requests to the appropriate vie...

8 min read

### Build a URL Size Reduce App with Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. In this article, we will learn to build a...

5 min read

### Build an Application to extract URL and Metadata from a PDF using Python

The PDF (Portable Document Format) is the most common use platform-independent file format developed by Adobe to present documents. There are...

3 min read

## Django URL patterns | Python

Prerequisites: Views in Django In Django, views are Python functions which take a URL request as parameter and return an HTTP response or throw an exceptio...

2 min read

## url - Django Template Tag

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

3 min read

## URL fields in serializers - Django REST Framework

In Django REST Framework the very concept of Serializing is to convert DB data to a datatype that can be used by javascript. Every serializer comes with some...

5 min read

## How to use URL Validator in Django?

In Django, a popular Python web framework, URL validation can be easily implemented using built-in tools and libraries. In this article, we will explore ho...

3 min read

## Create URL Bookmark Manager Using Django

This article explains how to make a tool called a Bookmark Organizer with Django for a website. With this tool, we can add, create, update, delete, and edit...

5 min read

## Get the Current URL within a Django Template

Django, a high-level Python web framework, encourages rapid development and clean, pragmatic design. One common requirement when developing web...

3 min read

Article Tags : [Django](#) [Python](#) [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  
GFG Corporate Solution  
Placement Training Program  
GeeksforGeeks Community

## 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

## Languages

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

## 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

ReactJS  
NextJS  
Bootstrap  
Web Design

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

## System Design

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

## School Subjects

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar  
Commerce  
World GK

## 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  
Java  
C++  
Web Development  
Data Science  
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved