Blog

Getting Started with Django: Build Your First Web Hosted App in 10 Minutes

Cristi Miloiu
Dec 12, 2024

Share on:

Introduction

Hi, I'm Cristi Miloiu, and in this tutorial, I'll guide you through building

your first web application using Django, one of the most popular Python web frameworks. By the end of this article, you'll have a simple, fully functional web app running on your local machine.

What is Django?

Django is a high-level Python web framework that simplifies web development by automating complex backend tasks. It enables developers to focus on building features instead of reinventing the wheel. With built-in tools like an authentication system, database integration, and support for CRUD operations, Django is designed to promote reusability and follows the DRY (Don't Repeat Yourself) principle.

Built on an MVT (Model-View-Template) architecture, Django separates concerns such as data, logic, and user interface, ensuring clean, maintainable code. It's particularly suited for database-driven websites and is known for its ability to scale as your project grows.

Django is an excellent choice for beginners due to its robust features and ease of use. Ready? Let's get started!

Want to skip the setup and jump straight to the code? **click here** .

What You'll Learn

1. Setting up a Python environment.
2. Installing Django and creating your first project and app.
3. Running your app locally.
4. Deploying your app to the cloud.

Step-by-Step Guide

Before you begin, make sure you have `Python` and `pip` installed on your machine.

Step 1: Create Your Project Directory

Run the following commands to initialize a new project directory:

```
mkdir django-app
cd django-app
```

Step 2: Install Django and Required Libraries

First, let's set up a virtual environment and install the necessary dependencies:

```
python3 -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activat
pip3 install Django
pip3 freeze > requirements.txt
```

Explanation:

- `venv` creates an isolated Python environment for the project.
- `requirements.txt` ensures consistent dependencies when deploying the app.

Step 3: Create Your Django Project

Initialize a new Django project:

```
django-admin startproject myproject .
```

This will create the `myproject` directory with all the necessary configuration files.

Step 4: Create a Django App

Create a new app inside your project:

```
python manage.py startapp myapp
```

Step 5: Configure URLs

Update the `myproject/urls.py` file to include your app's URLs:

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

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

Next, create a `myapp/urls.py` file

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

urlpatterns = [
    path('', index, name='index'),
]
```

Step 6: Create Your First View

Add the following code to the `myapp/views.py` file.

```
from django.http import HttpResponse

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

Step 7: Run the Development Server

Run the server using the following command:

```
python manage.py runserver
```

Open a browser and navigate to `http://localhost:8000`. You should see “Hello, World!” displayed on the screen.

Deployment Guide

Step 1: Configure Allowed Hosts

Add the Genezio domain .genez.io to the ALLOWED_HOSTS list in your settings.py file:

```
ALLOWED_HOSTS = [
    '127.0.0.1',
    '.genez.io',
    'localhost',
]
```

Step 2: Install the Genezio CLI

Install Genezio for deployment:

```
npm install –g genezio
```

Step 3: Analyze Your Project

Run the following command to generate the genezio.yaml configuration file:

```
genezio analyze
```

You’ll be prompted to provide the project name and region.

Step 4: Test Locally

Test your app locally with Genezio:

```
genezio local
```

Step 5: Deploy Your App

Deploy your app to the cloud with a single command:

```
genezio deploy
```

Your app will be live at a custom subdomain, such as `https://your-app-name.app.genez.io`.

Known limitations

Genezio does not support SQLite databases, because SQLite is not suitable for serverless applications. You need to use a different database like PostgreSQL or MongoDB. You can create it at this **link** in the Genezio dashboard.

Conclusion

Congratulations! 🎉You've built and deployed your first web app using Django in just 10 minutes.

If you have any questions or feedback, feel free to reach out to me at **cristi@genezio.com**. The codebase for this tutorial is open-source and available on **GitHub** .

Happy coding! 🚀

Subscribe to our newsletter

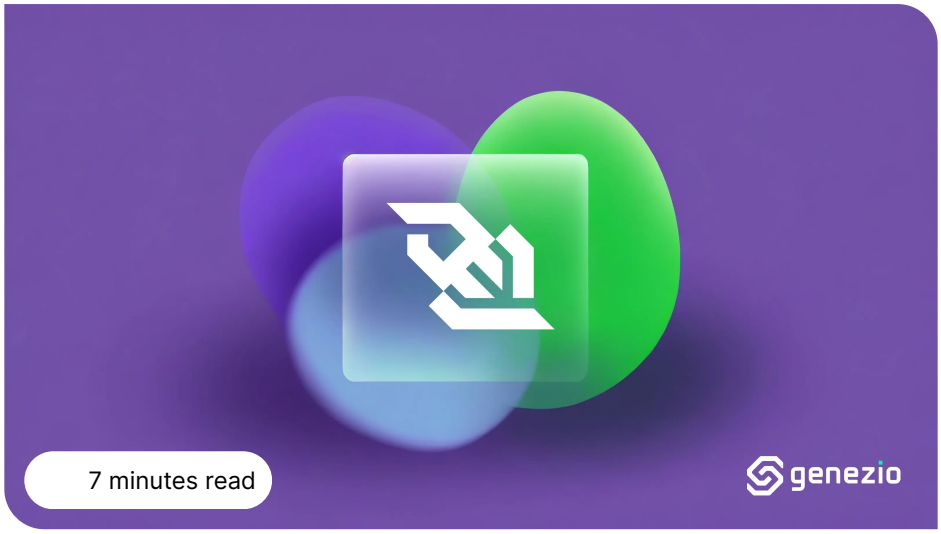
Genezio is a serverless platform for building full-stack web and mobile applications in a scalable and cost-efficient way.

Enter your email

Subscribe

Run Scalable Real-Time WebSocket Applications on Genezio

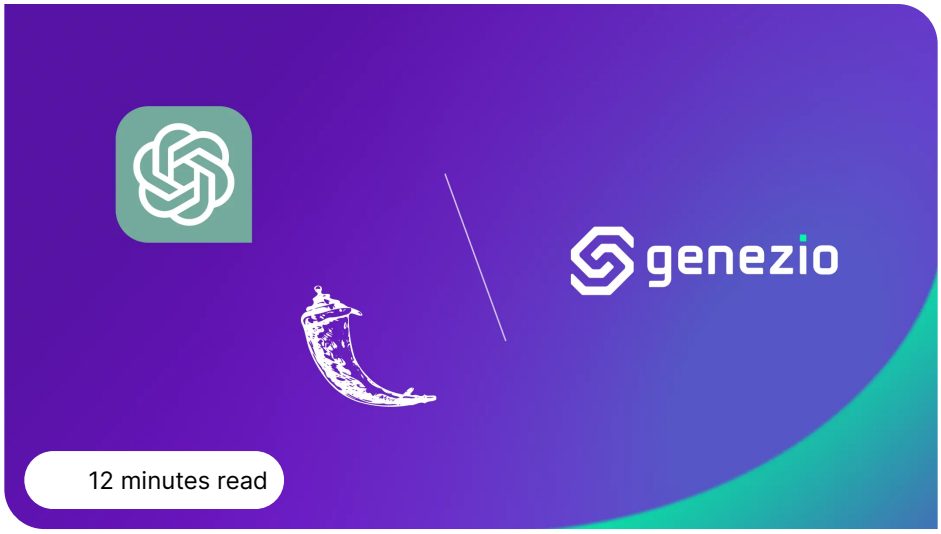
Related articles



Run Scalable Real-Time WebSocket Applications ...

At Genezio, we've solved one of the biggest challenges in serverless technology: deploying auto-scalable WebSocket applicat...

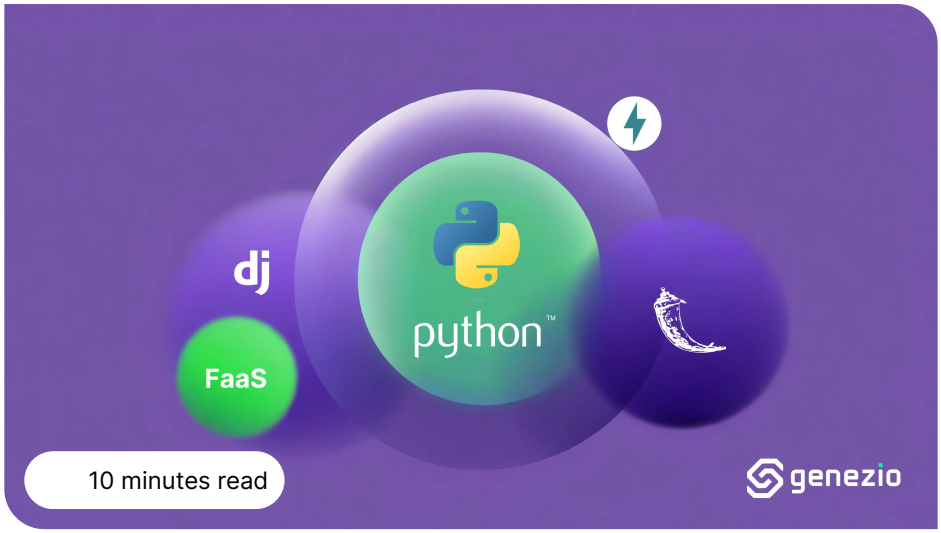
Bogdan Vlad
Dec 11, 2024



Build a ChatGPT Chatbot App with Flask: A Step...

Hi! I'm Cristi Miloiu, and in this article, I'll walk you through building a ChatGPT-powered chat app using Flask, Python, ...

Cristi Miloiu
Dec 10, 2024



Genezio Brings Serverless Python Support: Djan...

We're thrilled to announce that Genezio now fully supports deploying Python popular frameworks like Django, Flask, FastAPI....

Cristi Miloiu

Dec 09, 2024

More from Tutorials

How to Use Genezio with Express for Easy and Error-Free API Management

Bogdan Vlad

Apr 10, 2024

Upgrade Your Express.js App: Add a Frontend Seamlessly with Genezio

Radu Dumitrescu

Sep 27, 2024

Implement a newsletter on static websites with Mailchimp or HubSpot

Tudor Anghelescu

Mar 21, 2024

Now Available: Upstash Redis

Andreia Ocanoaia

Nov 22, 2023

How to add a MongoDB to your genezio project

Radu Dumitrescu

Jul 06, 2023

Mastering Automation: A Step-by-Step Guide to Creating a WhatsApp Bot

Cristi Miloiu

May 07, 2024

Services

Genezio Typesafe

Authentication

Databases

Cron Jobs

Frameworks

Express.js

Next.js

Resources

Blog

Docs

Github

Support

Contact Us

Privacy Policy

EULA

Brand Assets

Status

Careers

Subscribe to our newsletter

Genezio is a serverless platform for building full-stack web and mobile applications in a scalable and cost-efficient way.

Enter your email

Subscribe