

Introduction to Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Ridiculously fast:

Django was designed to help developers take applications from concept to completion as quickly as possible.

Reassuringly secure:

Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable:

Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.

Why Django?

With Django, you can take web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Ridiculously fast.

Django was designed to help developers take applications from concept to completion as quickly as possible.

Fully loaded.

Django includes dozens of extras you can use to handle common web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box.

Reassuringly secure.

Django takes security seriously and helps developers

avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery, and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.

Exceedingly scalable.

Some of the busiest sites on the planet use Django's ability to quickly and flexibly scale to meet the heaviest traffic demands.

Incredibly versatile.

Companies, organizations, and governments have used Django to build all sorts of things—from content management systems to social networks to scientific computing platforms.

Install Django

Step 1: Install Python

Most Ubuntu systems come with Python pre-installed, but it's a good practice to install the latest version or the

version you prefer.

1. Update your package list:

```
sudo apt update
```

2. Install Python 3:

```
sudo apt install python3
```

3. Verify the installation:

```
python3 --version
```

Step 2: Install pipx

pipx is a tool designed to help you install and run Python applications in isolated environments. It simplifies the management of Python packages, especially those that are command-line applications, by creating a separate virtual environment for each application.

1. Install **pipx**:

```
sudo apt install pipx
```

2. Confirm the pipx installation

```
pipx --version
```

Step 3: Install Django

1. Install Django using `pipx`:

```
pipx install django
```

2. Show `pipx list` Confirm the installation

```
pipx list
```

Step 4. Always Activate the Virtual Environment on Startup

1. Open the configuration file with a text editor (e.g., for Bash):

```
nano ~/.bashrc
```

2. Add the line last on this file:

```
source /home/coderaktar/.local/share/pipx/venvs/  
django/bin/activate
```

```
# coderaktar => Your Ubuntu/OS username
```

3. Save the changes:

In `nano`, press `Ctrl + O`, then `Enter` to save, and `Ctrl + X` to exit.

4. Reload the configuration:

To apply the changes immediately, run:

```
source ~/.bashrc
```

Now Your Machine is ready for creating Django project.

Create django project

1. Go to the Directory where you create project.

2. Then open terminal on this Directory

3. Write in the terminal this command:

```
django-admin startproject My_First_Project
```

Here My_First_Project is your project name

5. After execute this command we can see a Directory named My_First_Project[Your give

project name]

6. Open the Directory where we see a file named manage.py and a Directory same as our project name

7. Open terminal again on this folder or write the previous terminal goes for My_First_Project Directory:

`cd My_First_Project/`

8. Navigate to your Django project directory and run:

`python manage.py runserver`

9. Now you see a server address `ctrl + click` to open it or copy it and open a browser past and hit the link

10. If all set up successfully completed you can see the page on this link: **Congratulations**



The install worked successfully! Congratulations!

View [release notes](#) for Django 5.1

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

django



Django Documentation
Topics, references, & how-to's



Tutorial: A Polling App
Get started with Django



Django Community
Connect, get help, or contribute

Run Our Application is Specific Port

All procedure are same. When we run application in the server open it specific port:

Syntax:

```
python manage.py runserver protnumber
```

Example:

```
python manage.py runserver 7000
```


