

Django commands

Start project

The `startproject <projectname>` command is a Django management command used to create a new Django project with the specified `<projectname>`. When you run this command in your terminal or command prompt, Django will set up the basic directory structure and configuration files for your new project.

Once, you open the project or directory, you use the following command to start up a project in Django:

```
django-admin startproject myproject
```

runserver

The `runserver` command is one of the most frequently used Django management commands. It allows you to start the development server, enabling you to run your Django application locally for testing and development purposes.

```
python manage.py runserver
```

Note: The `runserver` command is not intended for production use. For deploying your Django application to a production environment, you'll need to use a more robust web server like Apache or Nginx with a WSGI server like Gunicorn or uWSGI.

Command: startapp

Django projects are typically composed of multiple apps, each responsible for handling a distinct part of the application. An **app** is a self-contained component of a Django project that encapsulates specific functionality or features.

Once you are in the project's root directory, run the `startapp` command followed by the name you want to give to your app. For example, if you want to name your app "blog," the command would be:

```
python manage.py startapp blog
```

In Django, `migrate`, `makemigrations`, and `showmigrations` are management commands related to database migrations. They are used to manage and apply changes to the database schema as your models evolve over time.

Command: makemigrations

The `makemigrations` command is used to create new database migration files based on the changes you have made to your models. When you define new models, add fields, or modify existing fields in your Django app's `models.py` file, you need to create new migrations to capture these changes.

```
python manage.py makemigrations <appname>
```

Command: migrate

The `migrate` command is used to apply database migrations and synchronize the database schema with the current state of your models. It reads the migration files created by `makemigrations` and executes the necessary SQL queries to create or modify database tables, columns, and indices.

```
python manage.py migrate
```

Command: showmigrations

The `showmigrations` command displays a list of all the migrations for your project and indicates which ones have been applied and which ones are pending. It's useful for keeping track of the migration status of your app's models.

```
python manage.py showmigrations
```

Command: createsuperuser

The `createsuperuser` command is a Django management command used to create a superuser account for the Django admin interface. The superuser account has administrative privileges and can access and manage all aspects of the Django admin site, including managing users, groups, models, and other site content.

```
python manage.py createsuperuser
```

After running the command, Django will prompt you to enter the details for the superuser account. Once you have provided the required information, Django will create the superuser account and display a message indicating that the user has been created successfully.

Command: shell

The `shell` command in Django is a management command that opens up a Python interactive shell with the Django environment loaded. It allows you to interact with your Django project's models, database, and other components using Python code in an interactive manner.

```
python manage.py shell
```

To exit the shell, simply type `exit()` or press `Ctrl + D` (or `Ctrl + Z` on Windows).

Command: flush

The `flush` command in Django is a management command used to clear all data from the database associated with your Django project. It's useful during development and testing when you want to reset the database to a clean state without deleting or recreating the database schema.

```
python manage.py flush
```

After running the command, Django will prompt you to confirm whether you want to flush the database. If you confirm by typing yes, Django will delete all data from the database, and you'll receive a message indicating that the flush was successful.

Django commands table

Django Commands	Descripton
startproject	Creates a new Django project
startapp	Creates a new app within a Django project
runserver	Launches the development server for testing the application locally
makemigrations	Creates new database migration files based on model changes
migrate	Applies database migrations to update the database schema
showmigrations	Displays a list of all the migrations for your project along with their applied or pending status
createsuperuser	Creates a superuser account for the Django admin interface
shell	Opens an interactive Python shell with Django environment loaded
flush	Clears all data from the database (use with caution)