Revision Notes: Introduction to Django

Welcome to your revision guide on the basics of Django, based on the recent class session. The key topics covered are Django's framework, application setup, essential commands, and the workings of views, URLs, and virtual environments. Let's dive into each topic.

1. Introduction to Django

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

Prerequisites:

- **Python 3.x:** Ensure Python 3.x or above is installed on your system 【4:12†transcript.txt】.
- **Django Installation:** Install Django version 5.x or above using `pip install django` command 【4:0†transcript.txt】.

2. Setting Up a Django Project

Initial Setup:

- To create a new Django project, use the command:

  ```
  django-admin startproject <projectname>
  ```

  This command sets up a new project directory with a manage.py file and a subdirectory containing default settings 【4:15†transcript.txt】.

Starting an App:

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

This command generates a directory with several boilerplate files necessary for app development【4:16†transcript.txt】.

3. Django Structure Overview

Key Files:

- **manage.py:** This is the command-line utility to manage your Django project—it includes commands like starting the server, creating migrations, etc. 【4:17†transcript.txt】.
- **settings.py:** This configuration file contains all settings, including database configurations, middleware, templates, and installed apps 【4:16†transcript.txt】.
- **urls.py:** Manage and route different URLs of your app through this file 【4:3†transcript.txt】.

Virtual Environments:

- Utilize virtual environments to isolate dependencies between projects. This is crucial when different applications require different versions of libraries or the Django framework【4:4†transcript.txt】.
- For activation on Windows:

```
VENV\Scripts\activate
```

On macOS/Linux:

```
source VENV/bin/activate
```

4. Working with Apps, Views, and URLs

Apps:

- Apps are the core building blocks of a Django project, encapsulating specific functionality【4:13†transcript.txt】.

- Views in Django function as the Layer responsible for the business logic. They receive web requests, process data, and return responses. Views are linked to URLs via mappings defined in the urls.py file【4:8†transcript.txt】.

URLs:

- URLs route requests to the appropriate view within your Django app. They are defined using patterns in the urls.py file and connected using path functions【4:13†transcript.txt】.

Command to Start Server:

- Run your application using:

  ```
  python manage.py runserver
  ```

  By default, this runs your server on `localhost:8000`. You can specify a port with:

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

Configuring Views and Endpoints:

- After defining a view function in views.py, you must map it to a URL. Example:

  ```
  from django.http import HttpResponse

  def hello_world(request):
      return HttpResponse("Hello, world!")
  ```

- In urls.py:

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

  urlpatterns = [
  ```

Conclusion

This guide should provide a solid foundation for understanding how to start working with Django. You should now be familiar with creating projects and apps, navigating Django's directory structure, and setting up views and URLs to create basic web responses. Remember to utilize resources like Django's official documentation for more detailed explanations and advanced configurations. Happy coding!