Introduction

This documentation provides a detailed explanation of the "IPL Matches" project, which involves web scraping match data from ESPN and displaying it on a webpage using Django.

1. Web Scraping Process

1.1. Web Scraping Implementation

URL Selection: The web scraping process starts with selecting the target URL, which in this case is https://www.espn.in/cricket/scores/series/8048/season/2024/indian-premier-league.

Sending HTTP Request: Using the requests library in Python, an HTTP GET request is sent to the selected URL with custom headers to mimic a real browser request.

Parsing HTML Content: The HTML content of the response is parsed using BeautifulSoup, which provides a structured way to navigate and extract data from the HTML.

Finding Match Elements: Within the parsed HTML content, specific match-related elements are identified using class selectors (e.g., cscore\_link--button, cscore\_info-overview, cscore\_notes\_game) to extract match details such as match number, teams, and result.

Iterating through Matches: The scraper iterates over each match element, extracts relevant details (e.g., match number, teams, result), and stores them in a structured format (e.g., list of dictionaries).

2. Django Project Setup and Rendering

2.1. Project Structure

Directory Structure: The project directory consists of the main project folder (espn) and an app folder (app) for Django app-specific components.

Settings Configuration: Django settings (settings.py) are configured to include necessary apps, middleware, and database settings.

2.2. Django App Components

URL Routing: URL routing is defined in urls.py to map URLs to views within the app.

Views and Data Rendering: The index view within views.py retrieves match data (either from web scraping or a hardcoded list) and renders it using an HTML template.

HTML Template Rendering: The index.html template uses Jinja templating to display match data in a structured table format on the webpage.

Styling: Basic CSS styling is applied to the HTML template to enhance visual presentation (e.g., font, colors, table layout).

3. Project Setup and Execution

Dependencies Installation: Ensure Python and Django are installed on your system. Install project dependencies (e.g., requests, beautifulsoup4) using pip.

Run Django Development Server: Navigate to the project directory (espn) and start the Django development server using python manage.py runserver.

Access Webpage: Open a web browser and go to http://localhost:8000/ to view the IPL Matches webpage, which dynamically displays match data retrieved either from web scraping or hardcoded list.