Templates and Static Content Management

This concept page introduces Django templates and the management of static content within Django projects. It covers the creation and utilization of templates for rendering dynamic web pages, as well as the handling of static files such as CSS, JavaScript, and images.

Concept Overview

Topics

* Django Templates
* Template Language
* Template Inheritance
* Static Files Management

Learning Objectives

* Understand the role of templates in generating dynamic web content
* Learn the syntax and features of Django’s template language
* Explore template inheritance and its benefits for code reuse and organization
* Master the management of static files in Django projects, including CSS, JavaScript, and images

Django Templates

In Django, templates are text files that define the structure and presentation of data for web pages. They separate the presentation logic from the application logic, allowing developers to create reusable and maintainable code. Templates provide a way to generate dynamic HTML by interpolating data from the application’s views and models.

Template Language

Django’s template language provides a set of tags, filters, and variables for manipulating and displaying dynamic data within templates. Templates can access and display data passed from views using variable interpolation ({{ variable }}) and can execute loops and conditional statements using template tags ({% tag %}).

Tags such as **{% for %}**, **{% if %}**, and **{% include %}** allow for control flow and template inclusion, while filters like **{{ value|date }}** and **{{ value|truncatechars:30 }}** modify the displayed data.

<!-- book\_list.html -->

<h1>Book List</h1>

<ul>

{% for book in book\_list %}

<li>{{ book.title }} by {{ book.author }}</li>

{% endfor %}

</ul>

Template Inheritance

Template inheritance is a powerful feature in Django that allows you to create a base template with common elements and extend it with child templates for specific pages. This promotes code reuse and consistency across your web application.

<!-- base.html -->

<html>

<head>

<title>{% block title %}My Site{% endblock %}</title>

</head>

<body>

{% block content %}{% endblock %}

</body>

</html>

<!-- book\_list.html -->

{% extends 'base.html' %}

{% block title %}Book List{% endblock %}

{% block content %}

<h1>Book List</h1>

<ul>

{% for book in book\_list %}

<li>{{ book.title }} by {{ book.author }}</li>

{% endfor %}

</ul>

{% endblock %}

Template Tags and Filters

Django provides a rich set of built-in template tags and filters for common tasks like looping, conditional rendering, URL generation, and string formatting. You can also create custom template tags and filters to extend the fun

<a href="{% url 'book-detail' book.id %}">{{ book.title|truncatechars:30 }}</a>

Static Files Management

Django provides built-in tools for managing static files such as CSS, JavaScript, and images. Static files are typically stored in the **static** directory within Django apps and are served directly by the web server in production for improved performance.

<!-- base.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>{% block title %}My Site{% endblock %}</title>

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

<!-- Content -->

<script src="{% static 'js/script.js' %}"></script>

</body>

</html>

In this example, **{% static %}** template tag is used to include static files like CSS and JavaScript in the HTML template. These files are stored in the **static** directory of the app and are referenced using the **{% static %}** tag.

Practice Exercises

* Create a base template with common elements like header, footer, and navigation menu
* Extend the base template to create specific templates for different pages of your website
* Utilize template tags and filters to display dynamic data fetched from the backend
* Experiment with template inheritance to understand its benefits in code organization and reuse
* Manage static files such as CSS, JavaScript, and images in your Django project
* Serve static files locally during development

Additional Resources

* [Django Templates](https://savanna.alxafrica.com/rltoken/uL-p4zKRwl9dtKiqLQUiaw)
* [Template Syntax](https://savanna.alxafrica.com/rltoken/SGBlqoiKYFOkChKrJYRuqA)
* [Template Inheritance](https://savanna.alxafrica.com/rltoken/vGfRv48yrP-QZAo8SWUmdQ)
* [Built-in Template Tags and Filters](https://savanna.alxafrica.com/rltoken/-gUZN8yFhP6k5_r3ypAa9Q)
* [Managing Static Files](https://savanna.alxafrica.com/rltoken/I1_1cYOOLnJ2S06ZdJPf-A)