

Django Model View Template (MVT) deseninin temel kavramlarını anlamak için aşağıdaki başlıklar altında özetleyebiliriz:

MVT Deseni

- Django, Model-View-Controller (MVC) deseninin bir uygulaması olan Model-View-Template (MVT) desenini kullanır.
- MVT, uygulama mantığını üç bileşene ayırır: Model (veri yönetimi), View (veri sunumu) ve Template (veri sunum biçimi).

Model

- Django Model, verileri yönetir ve veritabanı ile iş mantığını işler.
- Model, veritabanı tablolarına karşılık gelen sınıflardır ve CRUD (Create, Read, Update, Delete) işlemleri için API'ler sunar.

View ve Template

- Django View, hangi verilerin sunulacağını belirler ve genellikle bir Python fonksiyonu olarak tanımlanır.
- Template, verilerin nasıl sunulacağını tanımlar ve dinamik web sayfaları oluşturmak için kullanılır.

Django Projesi Oluşturma

- Django projesi, Django uygulamaları ve ayarları için bir konteynerdir. `django-admin startproject mysite` komutu ile başlatılır.
- Proje, `manage.py`, `settings.py` ve `urls.py` gibi önemli dosyalar içerir; bu dosyalar projenin yönetimi ve yapılandırması için kullanılır.

Django Uygulaması Geliştirme

- Projeye bir Django uygulaması eklemek için `python manage.py startapp onlinecourse` komutu kullanılır. Bu, uygulama yapısını ve önemli dosyaları oluşturur.
- `models.py`, `views.py` ve `urls.py` gibi dosyalar, veri modelleri, görünüm fonksiyonları ve URL yönlendirmeleri için kullanılır.

Uygulamanın Yayınlanması

- Uygulama geliştirme tamamlandıktan sonra, `python manage.py runserver` komutu ile yerel bir geliştirme sunucusu başlatılır.
- Uygulama, belirli bir URL'ye yönlendirilerek çalıştırılır ve kullanıcı arayüzünde görüntülenir.

Django Admin Sitesi

- Django, içerik yöneticileri için güçlü bir yönetim sitesi sunar ve bu sayede kurslar, dersler ve sorular gibi içerikler kolayca yönetilebilir.

- Yönetici kullanıcı oluşturmak için `python manage.py createsuperuser` komutu kullanılır ve ardından yönetim paneline giriş yapılır.

Model Kaydı ve Özelleştirme

- Modellerin yönetim paneline kaydedilmesi için `admin.site.register()` kullanılır; örneğin, `Course` ve `Instructor` modelleri kaydedilir.
- Yönetim panelinde hangi alanların gösterileceğini belirlemek için model admin sınıfları oluşturulur ve `fields` listesi tanımlanır.

İlişkili Modellerin Yönetimi

- Bir kursun birçok dersi ve sorusu olabileceğinden, dersleri kursla birlikte eklemek için `admin.StackedInline` veya `admin.TabularInline` kullanılır.
- Yönetim panelinde arama ve filtreleme işlevselliği eklemek için `search_fields` ve `list_filter` kullanılır.

Django Görünümleri

- Django'da bir görünüm, HTTP GET, POST, DELETE veya UPDATE gibi web isteklerini alır ve bir web yanıtı döner.
- Görünüm, genellikle bir HTTP istek nesnesini ilk argüman olarak alır ve bu nesne, istekle ilgili okuma amaçlı meta verileri içerir.

Görünüm Oluşturma

- Görünüm fonksiyonu, iş mantığını işlemek için HTTP isteğini kullanır ve bir `HttpResponse` nesnesi döner.
- URL'leri görünüm fonksiyonlarına eşlemek için `urls.py` dosyasında bir yol nesnesi eklenir.

HTTP Yanıtları

- Her HTTP yanıtı, bir durum kodu ile ilişkilidir; örneğin, 200 kodu isteğin başarıyla işlendiğini gösterir.
- Eğer bir hata oluşursa, örneğin bir kaynak bulunamazsa, 404 Not Found gibi bir hata durumu dönebilir.

Django Şablonları

- Django şablonları, statik HTML öğeleri ile Django şablon etiketleri ve değişkenlerini birleştirerek dinamik içerik sunar.
- Şablon dosyaları, uygulama kök klasöründe `templates` adında bir klasör oluşturularak ve içinde `onlinecourse` adında bir alt klasör ile yapılandırılır.

Veri Sunumu ve Görselleştirme

- Şablonlar, bir bağlam (context) ile render edilir; bu bağlam, görünümlerden dönen bir sözlük benzeri nesnedir.

- if-else etiketleri ve for döngüleri kullanılarak, sunucudan dönen veri listeleri kontrol edilir ve görüntülenir.

CSS ile Geliştirme

- HTML dosyalarını daha etkileşimli ve stilize hale getirmek için CSS dosyaları eklenir.
- load static ve static etiketleri kullanılarak CSS dosyalarının yolu oluşturulur ve sayfanın görünümü iyileştirilir.