

temel mimari karmaşıklıklardan ziyade uygulama işlevselliğine daha fazla odaklanır. Bu kılavuz, bir Django projesini nasıl başlatacağınızı, verimli bir şekilde nasıl yapılandıracağınızı ve uygulamanızın gelişimi için nasıl güçlü bir temel oluşturacağınızı detaylandırmaktadır.

Adım 1: Geliştirme Ortamınızı Hazırlayın

Django projenizi başlatmadan önce, Django'nun geliştirme kurulumunuzda yüklü olduğundan emin olun. Django'yu henüz yüklememişseniz, Django belgelerindeki ayrıntılı yükleme talimatlarına bakın. Projenizin bağımlılıklarını düzgün bir şekilde yönetmek için sanal bir ortam kullanmanız tavsiye edilir.

Adım 2: Projeyi Başlatın

Yeni bir proje oluşturmak için `django-admin` aracını kullanın ve ardından `startproject` ile birlikte istediğiniz proje adını girin:

```
django-admin startproject mysite
```

Bu komut çalıştırıldığında `mysite` adında yeni bir dizin oluşturulur ve Django burada birkaç başlangıç dosyası düzenler:

- `manage.py`: Bu Django projesi ile etkileşime girmenizi sağlayan bir komut satırı yardımcı programı.
- **`mysite/`**: Projeniz için gerçek Python paketi.
 - **`__başlangıç__.py`**: Dizini bir Python paketi olarak işaretler.
 - **`settings.py`**: Bu Django projesi için ayarları/yapılandırmaları içerir.
 - **`urls.py`**: Projeniz için URL bildirimlerini barındırır - Django sitenizin bir "içindekiler tablosu".
 - **`asgi.py`** veya **`wsgi.py`**: Projenize hizmet edecek web sunucuları için giriş noktası.

Adım 3: Geliştirme Sunucusunu Başlatın

Django, yerel test ve geliştirme için yerleşik bir hafif geliştirme sunucusu içerir. Proje dizininizden (`manage.py` dosyasının bulunduğu yer), çalıştırın:

```
python manage.py runserver
```

Bu komut sunucuyu yerel olarak **http://127.0.0.1:8000/** adresinde başlatır. Bir web tarayıcısında bu URL'ye gittiğinizde Django'nun varsayılan karşılama sayfası görüntülenecek ve projenizin doğru şekilde kurulduğu ve çalışır durumda olduğu anlaşılabacaktır.

Adım 4: Veritabanı Yapılandırması

Django varsayılan olarak SQLite kullanır, bu da çoğu küçük ve orta ölçekli proje ve geliştirme amaçları için yeterlidir. Veritabanı yapılandırmaları settings.py dosyasında DATABASES ayarı altında belirtilir. PostgreSQL veya MySQL gibi farklı bir veritabanı kullanmak için, gerekli veritabanı paketlerini yükledikten sonra bu ayarı değiştirin.

Veritabanınızı şu şekilde başlatın:

```
python manage.py migrate
```

Bu komut, veritabanı şemanızı modelleriniz ve ayarlarınızla eşleştirecek şekilde ayarlayan geçişler uygular.

Adım 5: Bir Uygulama Oluşturma

Django'da bir proje, yapılandırma için genel bir kapsayıcı görevi görürken, uygulama adı verilen ayrı bileşenler belirli işlevleri yerine getirir. Projeniz içinde bir uygulama oluşturmak için:

```
python manage.py startapp polls
```

Bu, sırasıyla uygulamanızın veri modellerini ve iş mantığını tanımlayacağınız models.py ve views.py gibi dosyaları içeren, uygulama geliştirme için yapılandırılmış bir polls dizini oluşturur.

Adım 6: Geliştirme ve Keşif

Projeniz ve ilk uygulamanız hazır olduğunda, uygulamanızın özelliklerini geliştirmeye daha derinlemesine dalebilirsiniz. Models.py'de veri yapılarını, views.py'de istek işlemeyi ve urls.py'de URL eşlemelerini tanımlayın. Django belgeleri, uygulamanızın yeteneklerini genişletmek için kapsamlı kaynaklar ve öğreticiler sunar.

Sonuç

İlk Django projenizi başlatmak başlangıçta zorlu görünebilir, ancak bu ayrıntılı adımlara uyarak, iyi organize edilmiş bir kurabilirsiniz.

geliştirme ortamını keşfedin, yeni bir Django projesi başlatın ve sofistike web uygulamaları oluşturmak için hazırlanın. Django'nun sağlam çerçevesi, Python'un sezgisel sözdizimi ile birleştğinde, hem yeni başlayanlar hem de web geliştirme iş akışlarını optimize etmeyi amaçlayan deneyimli geliştiriciler için ideal olan verimli ve keyifli bir geliştirme deneyimi sağlar.

Proje Yapısını Anlama

Django'nun proje yapısını kavramak, web uygulamalarını etkin bir şekilde yönetmek ve optimize etmek isteyen geliştiriciler için çok önemlidir. Django'nun çerçevesi açık, mantıklı ve ölçeklenebilir bir kod organizasyonunu desteklemek için hazırlanmıştır ve uygulamalarında gezinmelerini ve geliştirmelerini kolaylaştırır. Bu kılavuz, Django'nun nasıl yapılandırıldığının daha iyi anlaşılmasını sağlamak için her birinin rollerini ve işlevlerini açıklayarak tipik bir Django projesinin çeşitli bileşenlerini incelemektedir.

Django Proje Yapısına Genel Bakış

Yeni bir Django projesi başlatıldığında (örneğin, `django-admin startproject myproject` çalıştırılarak), Django otomatik olarak standart bir dizin yapısı oluşturur. Bu yapının her bir parçasının ne işe yaradığını anlamak, çerçeve ile etkili bir şekilde çalışmanın anahtarıdır.

Birincil Dizin

Projenizin adını taşıyan ana dizin (bu örnekte `myproject`), projenin adını paylaşan bir alt dizinle birlikte birkaç önemli dosya içerir. İşte tipik içeriğe bir bakış:

```
myproject/  
  manage.py  
  myproject/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```

- **manage.py:** Bu betik, Django projenizle çeşitli şekillerde etkileşim kurmak için kullanabileceğiniz bir komut satırı yardımcı programıdır, örneğin