



Django Web atısı

Mete Alpaslan Katirciođlu

Emre Yüce

Portakal Teknoloji

mete.alpaslan@portakalteknoloji.com

emre.yuce@portakalteknoloji.com

Akademik Bilişim 2008

anakkale, 30 Ocak 2008



Sunum Planı

- Django Nedir?
- Django projesi oluşturmak
- Basit bir blog uygulaması
URL tasarımı, model ve görünüm oluşturmak
- Şablon Kullanımı
- Diğer özellikleri
Çoklu dil desteği, hazır bileşenler vs.

Django: Karmaşık internet uygulamalarını hayata geçirme imkanı tanıyan bir web çatısıdır.

- Yeniden kullanılabilirlik,
- Modülerlik,
- Hızlı geliştirilme,
- DRY (Do not Repeat Yourself) prensiplerini sonuna kadar kullanma politikasına sahip bir yapıda tasarlanır.



Django Bileşenleri Özellikleri:

- Modellerin yaratılması için nesne ile ilişkisel eşleme (PostgreSQL, MySQL, SQLite ve Microsoft® SQL veritabanlarına kolay entegrasyon),
- Son kullanıcılar için tasarlanmış hoş görünümlü yönetici arabirimi,
- Zarif URL tasarımı,
- Tasarımı kolay şablon dili,
- Önbelleğe alma sisteminden oluşur.



Django Mimarisi:

- MTV (Model-Template-View) yapısını kullanır.
- Geliştirme sürecinde verinin tanımlanmasını sağlar.
- Verinin erişilmesi için yazılan kodun istek yönlendirme mantığından yani kullanıcı arayüzünden ayrılmasını sağlayan bir yazılım geliştirme yöntemi tanımlar.



Django Projesi Oluşturmak

ÖRNEK:

```
$ django-admin.py startproject ab08
```

```
$ ls ab08/
```

```
__init__.py
```

```
manage.py
```

```
settings.py
```

```
urls.py
```

dosyalarını oluşturur.



Django Projesi Oluşturmak

__init__.py

- Uygulamanın Python paketi olduğunu gösterir
- Paketin ilk çalıştırılan dosyasıdır.

manage.py

- Yeni uygulamalar yaratırken ya da komut verirken kullanılmak üzere python betiğini içerir.

urls.py

- İstek yapılan adresin görünümle eşleştirir.

settings.py

- Yapılandırma ayarları



Django Uygulaması Yaratmak

ÖRNEK:

```
$ python manage.py startapp blog
```

```
$ ls -l ab08/blog
```

```
__init__.py
```

```
models.py
```

```
views.py
```

dosyalarını oluşturur.



Django ile Model Oluşturmak

models.py

- Veritabanı tablolarının tanımlandığı Python sınıfları içerir.
- Tekrarlayan SQL ifadelerinden kurtarır.



Django ile Model Oluşturmak

ÖRNEK: *models.py*

```
class Blog(models.Model):  
    title = models.CharField(max_length=100)  
    is_featured = models.BooleanField(default=False)
```

```
class Entry(models.Model):  
    title = models.CharField(max_length=100)  
    body = models.TextField()  
    blog = models.ForeignKey(Blog)
```



Django ile Model Oluşturmak

ÖRNEK: *settings.py*

```
DATABASE_ENGINE = 'sqlite3'
```

```
DATABASE_NAME = '/path/to/ab08/database.db'
```

```
DATABASE_USER = ''
```

```
DATABASE_PASSWORD = ''
```

```
DATABASE_HOST = ''
```

```
DATABASE_PORT = ''
```



Django ile Model Oluşturmak

- *sql komutu kullanılarak kurulacak modelin sql kodu görülebilir.*

```
$ python manage.py sql blog
```

- *syncdb komutu kullanılarak model kurulur ve başlatılır.*

```
$ python manage.py syncdb
```



Django ile Görünümleri Oluşturmak

views.py

- Bir URL istendiğinde çağrılan Python yöntemlerini içerir.

ÖRNEK:

```
def entry_list(request, blog_id):  
    blog = Blog.objects.get(id=blog_id)  
    return render_to_response('blog/entry_list.html',  
                              {'entries': blog.entry_set.all()})
```



Django ile URL Yönetimi

urls.py

- URL dizisi desenlerini görünümle views.py dosyasındaki Python yöntemleriyle eşler.
- URL yönetimini temel koddan ayırır, denetim ve esneklik sağlar.
- Düzenli İfadeler (Regular Expression) ile güzel adresler :-)

ÖRNEK:

```
urlpatterns = patterns('',
    (r'^admin/', include('django.contrib.admin.urls.admin')),
    (r'^blog/', include('ab08.blog.urls')),
    (r'^blog/(?P<blog_id>d+)/$', 'blog.views.entry_list'),
)
```



Django Uygulaması Yaratmak

- Yarattılan uygulama **settings.py** dosyasındaki **INSTALLED_APPS** bölümüne eklenir.

ÖRNEK:

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'ab08.blog',  
)
```



Django'da Şablon Kullanımı

- Basit bir şablon dili
- Değişkenler: `{{ variables }}`
- Etiketler: `{% tags %}`
- Kolay filtreleme

Örnek: `{{ entry.body|truncatewords:50 }}`

- Template inheritance (blok etiketli bir iskelet ya da temel belge)

ÖRNEK: `{% extends "base.html" %}`

- *settings.py* dosyasında yol belirtilir.

ÖRNEK: `TEMPLATE_DIRS = ('path/to/ab08/templates',)`



Django'da Şablon Kullanımı

ÖRNEK:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
  <head>
    <title>{% block title %}Akademik Bilişim Konferansı{% endblock
    %}</title>
    {% block extrahead %}{% endblock %}
  </head>
  <body>
    {% block content %}    {% endblock %}
  </body>
</html>
```



Django'da Şablon Kullanımı

ÖRNEK: *entries listesi içinde döngüye girerek girdilerin başlıklarını ve yazıların ilk 50 karakterini ekrana basar.*

```
{% for entry in entries %}  
    {{ entry.blog.title }}: {{ entry.title }}  
  
    {{ entry.body|truncatewords:50 }}  
{% endfor %}
```



Admin (Yönetici) aracı

- Kontrol panelini kullanıma açmak için `django.contrib.admin` satırı `INSTALLED_APPS` ayarına eklenir.

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'ab08.blog',  
    'django.contrib.admin',  
)
```



Admin (Yönetici) aracı

- Admin paneline web arayüzünden ulaşmak için **urls.py** dosyası düzenlenmelidir.

ÖRNEK:

```
urlpatterns = patterns('',  
    (r'^admin/',  
    include('django.contrib.admin.urls.admin'))),  
)
```

```
$ python manage.py syncdb
```

- Komutu ile veritabanınızı güncelleyerek kullanıcı oluşturabilirsiniz.



Diğer Özellikler

- Değiştirilebilir hata sayfaları, DEBUG modunun kullanılması

ÖRNEK: *404.html , 500.html, settings.py DEBUG = TRUE*

- Çoklu dil desteği (multilingual)

ÖRNEK: *Şablon sistemine kolay uygulanabilirlik {% trans('Home') %}*

- Cache desteği

ÖRNEK: *Dilerseniz tek tek sayfaları, dilerseniz yüklü bir sorguyu kaydederek kullanabilirsiniz.*

- Django'nun hazır bileşenleri

ÖRNEK: *FreeComment blog uygulamalarına hazır yorum eklentisi. Zangetsu'da kullanılmaktadır.*



Sunum Özeti

- Yeniden kullanılabilirlik, modülerlik, hızlı geliştirilme,
- DRY (Do not Repeat Yourself) prensipleri
- Zarif URL tasarımı, kolay veritabanı entegrasyonu
- Tasarımı kolay şablon dili,
- Hoş görünümlü ve kolay kullanılabilir yönetici arayüzü
- Çoklu dil desteği, hata sayfaları, cache desteği hazır bileşenler.



Teşekkürler,

Sorularınız ve Görüşleriniz

Django internet sitesi: www.djangoproject.com/

Python Kaynağı: www.diveintopython.org/

Django, Rails ve Php: [wiki.rubyonrails.com/rails/pages/Framework
+ Performance](http://wiki.rubyonrails.com/rails/pages/Framework+Performance)