

Web Uygulama Anaçatıları

Tahir Emre KALAYCI

May 5, 2009

Gündem

1 Web Uygulama Anaçatıları

- Nedir?
- Spring, Ruby On Rails, ZK, CakePHP, Flex, GWT, Dojo

2 İçerik Yönetim Sistemleri

- Nedir?
- Drupal, Joomla!, Plone, Moodle, WordPress

3 Django

Web Uygulama Anaçatısı Nedir?

- Web uygulamaları geliřtirmek için kullanılan yazılım çerçevesidir
- Dinamik web siteleri, web servisleri ve web uygulamaları geliřtirmek için kullanılır
- Web uygulaması geliřtirme sürecini kolaylařtırmak ve geliřtiricilere araçlar sunmak amacına sahiptir
- Örneğın çoğuu veritabanı erişimini, şablonlama desteğii ve oturum yönetim desteğii sunarlar.
- Temel olarak kodun tekrar kullanımı (“code reuse”) pratiğini desteklerler

Anaçatı Mimarileri

- Model View Controller (MVC) tasarım deseni
- Push-based: Çoğu MVC çerçevesi “push-based“ yöntemi kullanır. Bunlar işlem yapmak için eylemleri kullanır, daha sonra veriyi görünüm katmanına “iterler”. (Struts, Django, Ruby On Rails, Spring MVC)
- Pull-based: Bileşen temelli adı da alırlar. Bu çerçeveler görünüm katmanıya başlarlar, bu katman verileri gerektiğinde bir çok denetleyiciden “çekerler”. Bu tip mimaride bir görünümün birden fazla denetleyicisi olabilir. (Struts2, Tapestry, JBoss Seam, Wicket)
- İçerik Yönetim Sistemleri: Sunumun ilerisinde anlatacağım içerik yönetim sistemleri modüler yapıyı desteklemeleri ve genişletilebilmeleri nedeniyle farklı bir mimaride de olsa web uygulama çerçevesi haline gelmişlerdir.

Özellikler

- Güvenlik: Bazı çerçeveler kimlik denetleme, doğrulama ve yetkilendirme için destek sunarlar. Sayfalara erişimi rol temelli bir şekilde kısıtlarlar.
- Veritabanı erişimi ve eşleme: Çoğu çerçeve veritabanına erişmek için veritabanı bağımsız uygulama geliştirme arayüzü (API) sunar. Geliştiricilerin veritabanı değiştirdiğinde kod değiştirmeden, daha yüksek soyutlama ile çalışmasına olanak sağlarlar.
- URL eşleme: Kullanıcı dostu URL adresi desteği sunarlar.
- Web şablon sistemi: Dinamik web sayfaları genellikle bir kısmı statik web sayfalarından oluşan yapılar içerir (sol tarafta değişmeyen menüler gibi), şablonlama sistemi yardımıyla bu şekilde tüm sayfalarda ortak olan alanların tanımlanması sağlanır.

Özellikler

- Önbellekleme: Bant genişliğini daha verimli kullanmak için web belgelerinin önbelleklenmesi desteği sunarlar.
- Ajax desteği: AJAX mimarisini desteklerler
- Otomatik yapılandırma: Bazı çerçeveler uygulama yapılandırması işlemini otomatikleştirerek yapılması gereken işi azaltır.
- Web servisleri desteği
- Açık standart mimarisi: Genel kabul görmüş açık standart ve teknolojileri desteklerler

- Bugüne kadar gördüklerimiz : Struts, JavaServer Faces
- Spring: Açık kaynak uygulama çerçevesi, Java ve .NET platformlarını destekliyor.
- ZK Framework: Javascript bilgisi gerektirmeyen AJAX anaçatısı, RIA, Açık kaynak, Standartlara dayanıyor, Genişletilebilir
- CakePHP: Açık kaynak hızlı uygulama geliştirme PHP anaçatısı. MVC'ye dayanıyor.
- Flex: Adobe tarafından geliştirilen ve Actionscript ve Flash tabanlı bir RIA anaçatısı.
- GWT: Google tarafından geliştirilen AJAX web uygulama geliştirme araç kutusu
- Dojo: Javascript tabanlı, AJAX ve DHTML araç kutusu

İçerik Yönetim Sistemi Nedir?

- İçerik yönetim sistemi adından anlaşılacağı gibi içerik yaratma ve düzenlemeyi kolaylaştıran sistemlerdir.
- İçerik? (haber makaleleri, günlük yazıları, kullanım klavuzları, rehberler, teknik belgeler, satış rehberleri, pazarlama broşürleri,. bilgisayar dosyaları, görüntü, ses dosyaları, elektronik belgeler,....)
- Tamamen içeriklere dayalı web sitelerinin (haber siteleri, günlük siteleri, ...) oluşturulmasında kullanılır.
- Katılımcılığı destekler, kullanıcılar ve rolleri tanımlanabilir, işbirliğine dayanan üretimleri ve ilgili iş akışını destekler
- Otomatik şablonlar, kolay düzenlenebilen içerik, ölçeklenebilir özellik kümeleri, web standartları güncelleme desteği, iş akışı yönetimi, kullanıcı ve rol yönetimi, belge yönetimi, içerik sanallaştırma

Drupal, Joomla!, Plone, Moodle, WordPress

- Bizi şu an ilgilendiren Web içerik yönetim sistemleri: Geniş, dinamik web materyalini yönetmek ve kontrol etmek için kullanılır.
- Drupal: PHP, Açık kaynak, modüler yapı, özelleştirilebilir, tüm içeriği arama, tema sistemi, rol tabanlı izin sistemi, kayıt ve raporlama, önbellekleme, optimizasyon
- Joomla!: PHP, Açık kaynak, eklentiler, bileşenler,
- Plone: Zope web sunucusu üstünde çalışır (Python), GPL lisansı,
- Moodle: PHP, Açık kaynak uzaktan eğitim sistemi,
- WordPress: PHP, Açık kaynak yayım (publishing) sistemi (günlük, kitap, ...)

Django

- Python programlama diliyle hazırlanmıştır
- Yeniden kullanılabilirlik, modülerlik, hızlı uygulama geliştirme, Kendini Tekrar Etme ("Don't Repeat Yourself") kavramlarını sonuna kadar kullanır
- Ek olarak yönetim ekranı sunar (CRUD özelliği olan)
- Veritabanı güdümlü web sitelerinin yaratılmasında kullanılır
- MVC tasarım desenine yakın bir yöntemi kullanır (Model-Template-View)

Özellikler

- Geliştirme ve sınamaya için hafif web sunucusu
- HTML form verileri ve veritabanı verilerine dönüşüm için form serileştirme ve doğrulama
- Önbellekleme
- Uluslararasılaştırma
- XML, JSON desteği
- Şablon sistemi
- Slogan: “The Web Framework for perfectionists with deadlines”

İçerdiği Uygulamalar

- Genişletilebilir kimlik denetleme
- Dinamik yönetim arayüzü
- RSS ve Atom besleme araçları
- Esnek yorum sistemi
- Tek Django kurulumunda birden fazla site barındırma desteği
- Google Sitemaps üretim araçları
- Şablon kütüphaneleri
- CBS (GIS) uygulamaları yaratmak için destek

MVT?

- Model: Veritabanı soyutlaması için kullanılan veri sınıflarını barındıran katman (Django ORM)
- Görünüm (View): MVC'deki denetleyicinin görevlerini yapan kısım, kullanıcının ne göreceğini ayarlar. Fonksiyonlar.
- Şablon (Template): Kullanıcıların görmesi gerekeni ayarlar (Django Template Engine)

Python?

- Nesne yönelimli
- Yüksek seviyeli dinamik veri tipleri
- Modüler ve etkileşimli
- Girintilere dayalı, sade sözdizimi
- Yorumlayıcı (Interpreter) kullanıyor
- İstisna işleme, betik olarak kullanılabilme, çok sayıda kütüphane, öğrenim kolaylığı, özgür yazılım, yoğun kullanım (Django, Zope, GIMP, Inkscape, Scribus, Paint Shop Pro, Ubuntu, Debian, Pardus, Google, NASA, CERN, YouTube, Industrial Light & Magic,....)

```
#A is height B is radius
```

```
def cone (a, b):  
    formula = (3.14 * .33 *a)*(b * b)  
    return formula
```

Terimler

- Proje: Django'nun bir örneği, genellikle tek bir web sitesiyle ilişkilendirilebilir (blog sitesi)
- Uygulama: Django içerisinde yazılmış ve web sitesinin bir işlevselliğini gerçekleştirilmiş kod kümesi (Bir blog sitesinde blog ve yorumlar iki ayrı uygulama olarak ele alınabilir)
- Uygulamalar tekrar kullanılabilir.

Model örneği

```
1 from django.db import models;
2
3 class Kisi(models.Model):
4     ad = models.CharField(max_length=300)
5     soyad = models.CharField(max_length=300)
6
7     def __unicode__(self):
8         return ("%s %s") % (self.ad, self.soyad)
9
10    class Meta:
11        ordering = ["soyad"]
12        verbose_name = "Kişi"
13        verbose_name_plural = "Kişiler"
14 -----
15 İLİŞKİLER:
16 class Yazı(models.Model):
17     baslik = models.CharField(max_length=300)
18     tarih = models.DateTimeField()
19     yazar = models.ForeignKey(Kisi, related_name="yazilar")
20 -----
21 KULLANIM:
22 k = Kisi(
23     ad = "Mehmet"
24     soyad = "Akan"
25 )
26 k.save()
27 -----
28 Kisi.objects.all()
29 Kisi.objects.get(id=10)
30 Kisi.objects.filter(ad="Mehmet")
31 Kisi.objects.exclude(ad_contains="Meh")
32 Kisi.objects.all().order_by("ad")
33 Yazı.objects.filter(yazar__ad="Mehmet")
```


Görünüm (View) örneği

```
1 from django.http import HttpResponseRedirect
2
3 def homepage(request):
4     return HttpResponseRedirect("Anasayfaya Hoşgeldiniz!")
5
6 -----
7 from django.shortcuts import render_to_response
8 from myproject.myapp import Kisi
9
10 def kisi_listesi(request):
11     kisiler = Kisi.objects.all()
12     return render_to_response("kisi/list.html", {"kisi_listesi":kisiler})
13 -----
14 from django.shortcuts import render_to_response, get_object_or_404
15 from myproject.myapp import Kisi
16
17 def kisi_detay(request, soyad):
18     kisi = get_object_or_404(Kisi, soyad=soyad)
19     return render_to_response("kisi/detay.html", {"kisi":kisi})
```

Şablon örneği

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
2 <html lang="tr">
3   <head><title>Kişiler</title></head>
4   <body>
5     <h1>Toplam ({{kisi_listesi|length}}) Kişi</h1>
6     <ul> {% for k in kisi_listesi %}
7       <li> < a href="{{k.soyad}}"> {{k.ad}} {{k.soyad}} </a> </li>
8       {% endfor %}
9     </ul>
10  </body>
11 </html>
```

Sonuç yerine: Web Uygulama Anaçatılarının Avantajları, Dezavantajları

■ Avantajlar

- Tekrarları önler: Programcı zamanı olarak daha verimlidir
- Otomatik olarak üretilmiş kod daha güvenilir ve yapısal olarak daha tekbiçimli
- DRY yaklaşımı destekleniyor

■ Dezavantajlar

- Otomatik üretilmiş kodun anlaşılması zor olabilir, değiştirmesi zor olabilir
- Sistemler genellikle büyük ve yavaş olabilir.
- Bkz: “Why I hate frameworks?”
(<http://discuss.joelonsoftware.com/default.asp?joel.3.219431.121>)

Kaynaklar

- Tony C Shan; Winnie W Hua, "Taxonomy of Java Web Application Frameworks," e-Business Engineering, 2006. ICEBE '06. IEEE International Conference on , vol., no., pp.378-385, Oct. 2006
- Wikipedia "Web application framework" başlığı
- Wikipedia "Comparison of web application frameworks" başlığı
- Wikipedia "Web content management system" başlığı
- <http://www.springsource.org>
- <http://www.rubyonrails.org>
- <http://www.zkoss.org/product/zk.dsp>
- <http://cakephp.org/>
- <http://www.adobe.com/products/flex/>
- <http://www.drupal.org>
- <http://www.drupaldersleri.org>
- <http://www.joomla.org/>

Kaynaklar

- <http://plone.org/>
- <http://moodle.org/>
- <http://wordpress.org/>
- <http://www.djangoproject.com/>
- <http://e-bergi.com/2008/Ocak/Django>
- Wikipedia "Python (programlama dili)" başlığı
- <http://www.belgeler.org/uygulamalar/python-tutorial.html>
- <http://docs.djangoproject.com/en/dev/>