Açık Kaynaklı J2EE Geliştirme Araçları

Ahmet Işık

İdeal Teknoloji A.Ş.

Giriş

- Açık kaynaklı yazılım geliştirme araçları ile baştan sona web tabanlı bir kurumsal uygulama geliştirmek ve çalıştırmak mümkündür.
- Java'nın platform bağımsızlığı sayesinde bu araçları Java Runtime Environment desteği olan herhangi bir platform'da kullanmak mümkündür.
- Böylece hem geliştirme safhasında hem de çalıştırma safhasında hiçbir platforma bağımlı kalınmamış olur.
- Bu sunum kapsamında çeşitli uygulama katmanlarında kullanılabilecek araçlar, framework'ler ve bunların nasıl entegre bir şekilde kullanılabileceği anlatılacaktır.

Uygulama Gereksinimleri

- Orta Katman Servisleri
  - Kaynak Yönetimi, Kimlik Kontrolü, Yetki Kontrolü, Mesajlaşma, Yük Dağıtımı, Fail-Safety, vb…
- Uygulama Framework'ü
  - Uygulama gerçekleştirimi sırasında kodun modüler ve birbirinden mümkün olduğunca bağımsız bileşenler halinde yazılmasını sağlar.
  - Farklı geliştiriciler tarafından gerçekleştirilen bir uygulamada standartlaşma sağlar.
  - Bazı orta katman ve sunum servislerini destekler. Örnek: Kimlik Kontrolü, Yetki Kontrolü, Ekran düzeni, Sunum araçları (Tag Kütüphaneleri)
- Kalıcı Bilgi Saklama Mekanizması
  - Durum bilgisinin saklanması gereken nesneler için kullanılır.
  - Veri tabanları, Dosya Sistemleri, RAM gibi saklama ortamları kullanılabilir.
- Sunum Servisleri
  - Oluşturulan nesnelerin kullanıcıya gösterilmesi, uygulama akışının gidebileceği yönlerin kullanıcıya sunulması, fonksiyonların erişilebilir hale getirilmesi
- Make Aracı
  - Uygulama çalışma zamanı dizin yapısının oluşturulması
  - Uygulama kaynak kodunun derlenmesi
  - Gerekli dosyaların (kütüphaneler, konfigürasyon dosyaları, destek dosyaları, vb.) uygun konumlara kopyalanması
  - Otomatik kod ve konfigürasyon dosyası üretimi

Gündem

- Web Sunucusu: Apache Tomcat
- Uygulama Framework'ü: Jakarta Struts
- Make Aracı: Apache Ant
- Persistency: Hibernate
- Otomatik Kod Üretme: XDoclet

Apache Tomcat

Tomcat

- Apache Software Foundation tarafından açık kaynaklı olarak Jakarta projesi altında geliştirilmektedir.
- jakarta.apache.org/tomcat
- Uygulama mimarisinde Web Sunucusu katmanını gerçekleştirir.
- En son versiyonu Servlet 2.4 ve JSP 2.0 belirtimlerini destekler.
- Tomcat'in sunduğu servisler:
  - Kimlik denetimi: Veritabanı, LDAP gibi standart mekanizmalar halihazırda gelmektedir.
  - Genişletilebilir bir kimlik denetim mekanizmasına sahiptir.
- Bunların yanında SSL desteği, Kümeleme ve Yük dağıtımı imkanı sunar.

Jakarta Struts

Jakarta Struts

- Apache Software Foundation tarafından Jakarta projesi kapsamında geliştirilmektedir.
- jakarta.apache.org/struts
- Java Web uygulamaları için açık kaynak kodlu bir framework
- Uygulama mimarisini Model-2 yaklaşımına göre tasarlamayı kolaylaştırır.

Tasarım Kalıbı – Model 1

- Model-1 Web uygulaması
- Birbirinden bağımsız olarak çalışan JSP sayfalarından oluşur
- Her sayfa kontrolü direkt olarak bir sonraki sayfaya yönlendirir
- Ayrı bir akış kontrolcüsü yoktur

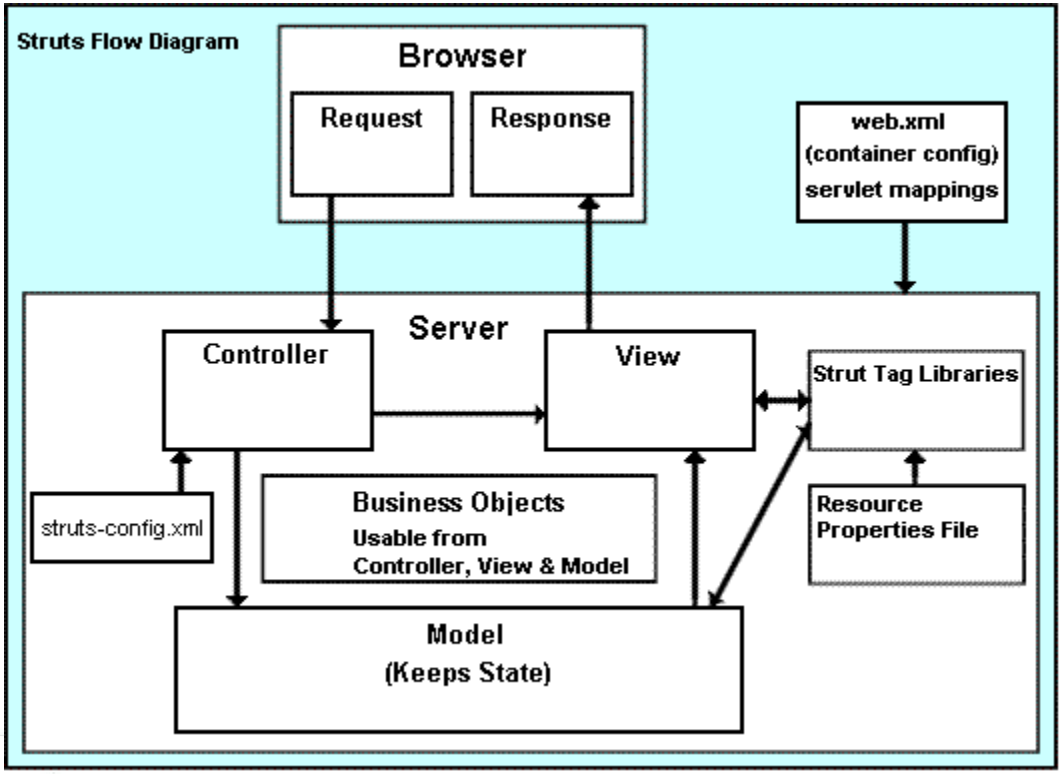
Model-1 Ana Sorunlar

- Basit bir değişiklik birçok farklı sayfada basamaklı olarak tahmin edilemeyecek etkilere neden olur.
- Karmaşıklık çabuk artar
- İlk bakışta basit gözüken bir iş parçalar eklendikçe büyük bir karmaşaya dönüşebilir.

Tasarım Kalıbı Model-2

- Model-View-Controller tasarım kalıbını temel alır
- Model-2’de her istem kontrolcüden geçer
- Kontrolcü bir sonraki sayfayı seçer

Jakarta Struts Mimarisi



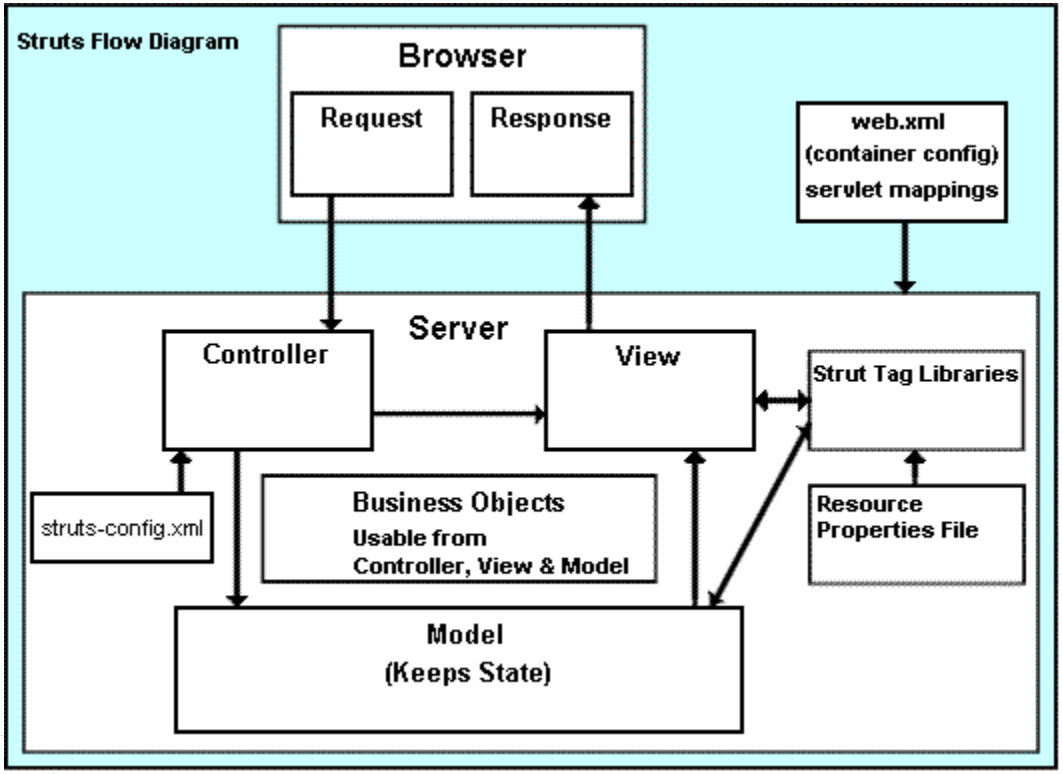
Controller

```
<web-app>
 <servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
  ……
  <init-param>
   <param-name>config</param-name>
   <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
   ……
 </servlet>

 <servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
 </servlet-mapping>
……
 </web-app>
```

Controller (devam…)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE struts-config PUBLIC
        "-//Apache Software Foundation//DTD Struts Configuration 1.0//EN"
        "http://jakarta.apache.org/struts/dtds/struts-config_1_0.dtd">


<struts-config>
 <!-- ========== Form Bean Definitions ======== -->
 <form-beans>
     <form-bean      name="ArrayComboCreationForm"
                type="edu.harvard.mgh.molbio.microarray.ArrayComboCreationForm"/>
       ……
 </form-beans>

 <!-- ========== Action Mapping Definitions ======== -->
 <action-mappings>
   <!-- Action To Create An Experiment -->
   <action    path="/ExperimentCreation"
              type="edu.harvard.mgh.molbio.microarray.ExperimentCreationAction"
              name="ExperimentCreationForm"
              scope="request"
              input="/experiment_creation.jsp">
             <forward name="display" path="/experiment_list.jsp" />
   </action>
   ……
</action-mappings>

</struts-config>
```

Model


Struts Flow Diagram
Browser
Request
Response
web.xml
(container config)
servlet mappings
Server
Controller
View
Strut Tag Libraries
struts-config.xml
Business Objects
Usable from
Controller, View & Model
Resource
Properties File
Model
(Keeps State)

Model

```
public class ExperimentCreationForm extends ActionForm {
      public void setexperiment_name(String experiment_name) {
         this.experiment_name = experiment_name;
      }
      public String getexperiment_name() {
         return experiment_name;
      }
      ……
      public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
         ActionErrors errors = new ActionErrors();
      ……
      if (!errors.empty()) {
            errors.add("input_error",
               new ActionError("error.input_error.exist"));
      }
         return errors;
   }
      public void reset() {
      }
}
```

Model (devam…)

```
public class ExperimentCreationAction extends Action {
   public ActionForward perform (ActionMapping mapping,
                                 ActionForm    form,
                                 HttpServletRequest request,
                                 HttpServletResponse response) {
        ActionErrors errors = new ActionErrors();
        ExperimentCreationForm theForm = (ExperimentCreationForm) form;
        String experiment_name = theForm.getexperiment_name();
        ……
        //Connect to data source and create a blank record in the database
        try {
            Context ctx = new InitialContext();
            DataSource ds =(DataSource)ctx.lookup("java:comp/env/jdbc/gmds");
            con =ds.getConnection();
            ……
            //Return a forward
            if (!errors.empty()) {
                errors.add("input_error", new ActionError("error.input_error.exist"));
                saveErrors(request, errors);
                return (new ActionForward(mapping.getInput()));
            }
            else {
                return mapping.findForward("display");
            }
        }
    }
}
```

View

```
<%@ page language="java" %>
<%@ page import="java.util.*" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/app.tld"    prefix="app" %>
……
<app:checkLogon/>

<html:html>
<body>

<html:errors property="input_error"/>

<html:form action="/ExperimentCreation" focus="experiment_name">
<table border="0" width="100%">
  <tr><td>
      <html:text property="experiment_name" size="32" maxlength="128"/>
      <html:errors property="experiment_name"/>
 </td></tr>
……
<tr><td>
      <html:submit property="submit" value="Submit"/>
</td></tr>
</table>
</html:form>

</body>
</html:html>
```

MVC Dezavantajları

- MVC framework'lerinin kurulması karmaşıktır
- Bir sayfa geliştirmek artık çok adımlı, önceden ekstra planlama ve düşünme gerektiren bir iş haline gelmiştir.

Apache Ant

Ant

- Another Nice Tool
- Apache Software Foundation tarafından Jakarta projesi kapsamında açık kaynaklı olarak geliştirilmekte ve dağıtılmaktadır.
- Unix'lerdeki Make benzeri bir araçtır.
- Genişletilebilirdir
- Tamamen Java'da, platform bağımsız olarak tasarlanmıştır
  - Herhangi bir platform'a özgü çağrılar içermez
- Ant build dosyası bir XML dosyasıdır
- Aktif Ant topluluğu tarafından Ant için sürekli yeni yetenekler eklenmektedir

Ant Kurmak

- Ant'ı şu adresten temin edebilirsiniz:
  - http://jakarta.apache.org/ant/
- Zip veya Tar dosyasını <bir_dizin> 'e açın
- PATH çevresel değişkeni içerisine <bir_dizin>/ant/bin dizinini ekleyin
- ANT_HOME çevresel değişkenini <bir_dizin>/ant olarak ayarlayın

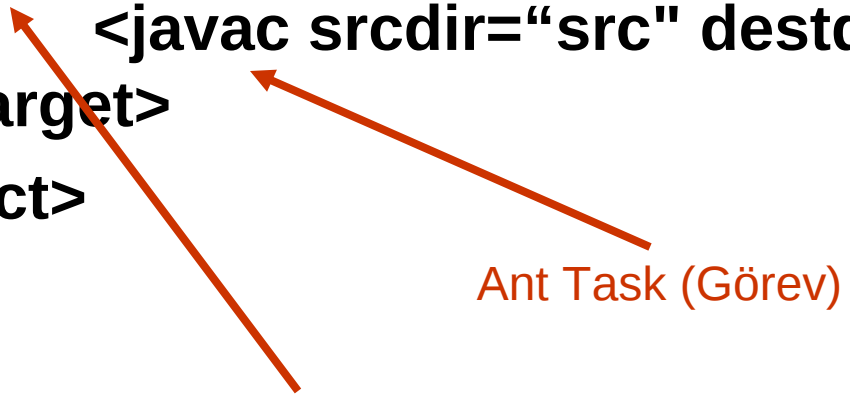
Ant'ı Çalıştırmak

- Make'e benzer şekilde:
  - ant [opsiyonlar] hedef

- Bulunulan dizindeki build.xml dosyasını kullanır.

Örnek Bir build.xml Dosyası

```
<project name=“merhaba" default="compile">
    <target name="compile">
        <javac srcdir=“src" destdir=“build" />
    </target>
</project>
```



Bağımlılık Zinciri

- Kullanıcı hedefi belirtir
- Ant sistemi o hedefin bağımlı olduğu tüm hedefleri bir kez olmak kaydıyla sırayla işler
- Dosyalar güncelse görev işletilmez
- Örneğin ‘test’ ve ‘deploy’ hedefleri ‘compile’ hedefine bağımlı olacaktır.

Yerleşik Gelen Ant Görevleri

- Ant
- AntCall
- AntStructure
- Available
- Chmod
- Copy
- Copydir
- Copyfile
- Cvs
- Delete
- Deltree
- Echo
- Exec
- ExecOn
- Fail
- Filter
- FixCRLF
- GenKey
- Get
- GUnzip
- GZip
- Jar
- Java
- Javac
- Javadoc
- Mail
- Mkdir
- Move
- Patch
- Property
- Rename
- Replace
- Rmic
- SignJar
- Sql
- Style
- Tar
- Taskdef
- Touch
- Tstamp
- Unjar

Hibernate

Hibernate

- Gavin King tarafından başlatılan bir açık kaynaklı bir Nesne-İlişkisel eşleme (ORM) Projesi
- LGPL lisansı ile dağıtılmaktadır
- Ayda 13.000 kere indirilmekte olan popüler, endüstri standardı olma yolunda ilerleyen bir teknolojidir.
- Proje sayfası hibernate.org

Modern ORM Çözümleri

- Saydam kalıcılık
  - Plain Old Java Objects (POJO)
- Kalıcı / Geçici Nesne Örnekleri
- Otomatik değişim kontrolü
- Geçişken Kalıcılık
- Çalışma zamanı SQL üretimi
- Üç farklı kalıtım eşleme stratejisi

Diğer Persistency Mekanizmalarından Üstünlükleri

- Doğal programlama modeli
- Yazılan kod miktarının daha az olması
- Uygulama sunucusu dışında da kodun geliştirilebilmesi ve test edilebilmesi
- Geliştirilen sınıfların kalıcı olmayan bağlamda da kullanılabilmesi
- Akıllı veri çekme stratejileri ile veritabanı erişimini minimize etmesi
- Model değişikliklerinin daha kolay yapılabilmesi

İlişkisel Veritabanlarının İyi Olduğu İşler

- Büyük miktardaki veri ile çalışma
  - Arama, sıralama…
- Veri kümeleri ile çalışma
  - Birleştirme, kümeleme
- Paylaştırma
  - Eş zamanlılık (Transaction)
  - Çok uygulamalılık
- Bütünlük
  - Alan kısıtları
  - Transaction Isolation

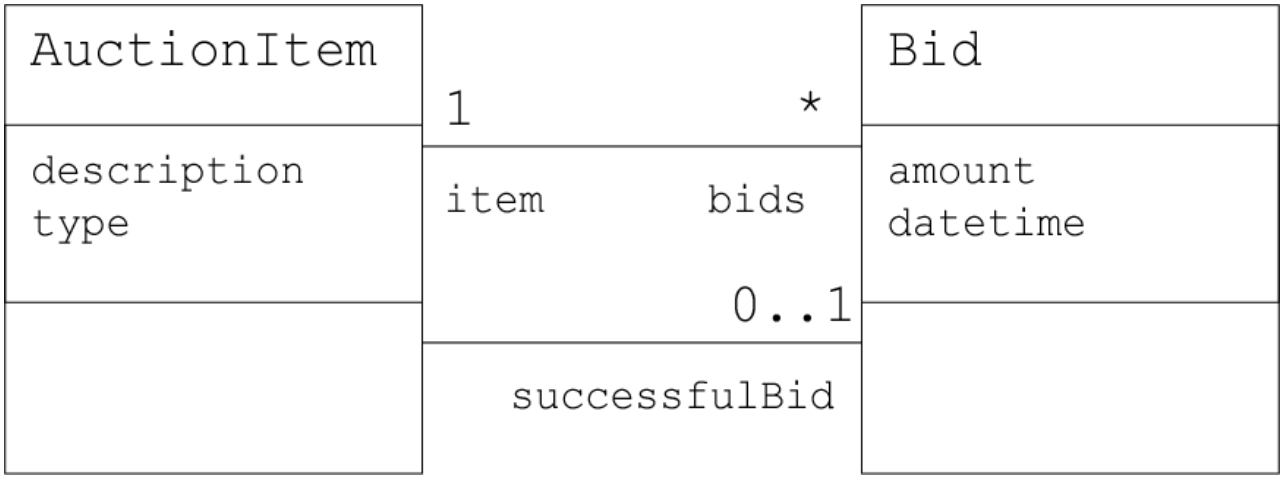
İlişkisel Veritabanlarının Kötü Olduğu Noktalar

- Modelleme
  - Polimorfizm eksikliği
  - Detaylı modeller kullanmanın güçlüğü
- İş Mantığı
  - Stored Procedure'ler uygulama mantığını dağıtmaktadır.

Hibernate

- JavaBean’ler için kalıcılık mekanizması
- Fine-Grained (İnce elenmiş) veri modellerine izin verir

Örnek Nesne Modeli



Örnek Nesne Modeli – Sınıf Kodu

- Varsayılan (parametresiz) Constructor
- Get/Set ikilileri
- Koleksiyon özellikleri sınıf değil arayüz ile tiplendirilmiş
- ID özelliği

```
public class AuctionItem {
    private Long _id;
    private Set _bids;
    private Bid _successfulBid
    private String _description;

    public Long getId() {
          return  _id;
    }
    private void setId(Long id) {
          _id = id;
    }
    public String getDescription() {
          return _description;
    }
    public void
    setDescription(String desc) {
          _description=desc;
    }
    …
}
```

Örnek Nesne Modeli – XML Eşleme Dosyası

- Okunabilir metadata
- Veritabanı sütun, tablo eşleştirmeleri
- Doğal ID numarası üretme stratejisi
- Koleksiyon metadata'sı
- Veri çekme stratejisi

```
<class name="AuctionItem"
    table="AUCTION_ITEM">
    <id name="id" column="ITEM_ID">
        <generator class="native"/>
    </id>
    <property name="description"
    column="DESCR"/>
    <many-to-one
    name="successfulBid"
    column="SUCCESSFUL_BID_ID"/>
    <set name="bids"
        cascade="all"
        lazy="true">
        <key column="ITEM_ID"/>
        <one-to-many class="Bid"/>
    </set>
</class>
```

Otomatik Değişim Kontolü

Bir AuctionItem nesnesi yükleme ve ‘description’ özelliğini değiştirme

```
Session session = sessionFactory.openSession();
Transaction tx = s.beginTransaction();

AuctionItem item =
   (AuctionItem) session.get(ActionItem.class, itemId);
item.setDescription(newDescription);

session.save(item);
tx.commit();
session.close();
```

Geçişken Kalıcılık

Bir AuctionItem nesnesi yükleme ve yeni bir Bid nesnesi yaratma

```
Bid bid = new Bid();
bid.setAmount(bidAmount);

Session session = sf.openSession();
Transaction tx = session.beginTransaction();

AuctionItem item =
   (AuctionItem) session.get(ActionItem.class, itemId);
bid.setItem(item);
item.getBids().add(bid);

Session.save(bid);
tx.commit();
session.close();
```

Nesne Ayırmak

Bir AuctionItem nesnesi yükleme ve yeni bir Bid nesnesi yaratma

```
Session session = sf.openSession();
Transaction tx = session.beginTransaction();
AuctionItem item =
   (AuctionItem) session.get(ActionItem.class, itemId);
tx.commit();
session.close();

item.setDescription(newDescription);

Session session2 = sf.openSession();
Transaction tx = session2.beginTransaction();
session2.update(item);
tx.commit();
session2.close();
```

Hibernate Sorgu Dili

- Hibernate Query Language (HQL)
- SQL'i nesneye yönelik hale getirmek için tasarlanmıştır
  - Tablo ve sütunlar yerine sınıflar ve özellikler
  - Polimorfizm
  - Sınıf ilişkileri
  - SQL'den daha yüksek seviyeli
- İlişkisel işlemler için tam destek
  - Inner/outer/full join işlemleri, kartezyen çarpımları
  - Projeksiyon
  - Kümeleme (max, avg) ve gruplama
  - Sıralama
  - Alt sorgular
  - SQL fonksiyon çağrıları

Hibernate Sorgu Dili

- En basit sorgu

  `from AuctionItem`

- ‘Bütün AuctionItem nesnelerini çek’

```
List allAuctions = session.createQuery(“from
AuctionItem”).list();
```

Hibernate Sorgu Dili

- Daha faydalı bir örnek:

```
select item from AuctionItem item join item.bids
bid where item.description like ‘hib%’ and
bid.amount > 100
```

- Diğer bir deyişle:

  Değeri 100’den büyük olan ve açıklaması ‘hib’ ile başlayan bütün AuctionItem nesnelerini çek

Hibernate Hakkında Daha Fazla Bilgi İçin

- http://hibernate.org
- Hibernate in Action (Manning, 2004)
- Araç desteği
  - http://xdoclet.sf.net
  - http://www.andromda.org

XDoclet

XDoclet Nedir?

- XDoclet bir kod üretim mekanizmasıdır
- Açık kaynaklı (flexible BSD lisansı ile) bir yazılımdır.
- JavaDoc metadata şablon mekanizması
- Attribute-Oriented Programming
- EJBDoclet projesinden türemiştir

XDoclet’in Amaçları

- Gereksiz manuel işleri kısmak
- Kod/Metadata çakışmasını engellemek
- Pragmatik Programlama Prensipleri
  - DRY (Don’t Repeat Yourself)
  - Problem alanına yakın kod yazın
  - Kod üreten kod yazın
- JSR 175 & 181

JSR 175

- Sınıflar, arayüzler, alanlar ve metodların bazı özelliklere sahip olduğu şeklinde işaretlenmesini sağlayacak bir metadata tekniği sağlanması

XDoclet Mimarisi

- XJavaDoc üzerinde inşa edilmiştir
- XJavaDoc bir Java Kod yapı analizi aracıdır.
- Modüllere ayrılmıştır
- Farklı amaçlar için şablonlar içerir
- Ant ile çalışır. Bunun için gerekli Ant görevleri tanımlanmıştır.

XDoclet Kullanımı

- Web uygulamalarında
  - web.xml konfigürasyon dosyası üretiminde
  - Tag kütüphanesi TLD dosyası üretiminde
  - Struts konfigürasyon dosyası üretiminde
- J2EE uygulamalarında
  - Deployment descriptor üretiminde
  - Enterprise Java Bean geliştirilirken
- Hibernate kullanılırken
  - Eşleme dosyaları üretilirken

Hibernate İle Kullanım

```
/**
* @hibernate.class proxy="domain.personel.Personel"
 */
public class Personel extends DomainObject {
    private String ad;
    /**
     * @hibernate.property not-null="true"
     */
    public String getAd() {
        return ad;
    }
    public void setAd(String ad) {
        this.ad = ad;
    }

    private String soyad;
    /**
     * @hibernate.property not-null="true"
     */
    public String getSoyad() {
        return soyad;
    }
    public void setSoyad(String soyad) {
        this.soyad = soyad;
    }

    private String sicilNo;
    /**
     * @hibernate.property not-null="true" unique="true"
     */
    public String getSicilNo() {
        return sicilNo;
    }
    public void setSicilNo(String sicilNo) {
        this.sicilNo = sicilNo;
    }
}
```

Ant ve XDoclet Ortak Çalıştırma

- XDoclet Hibernate modülünün tanımlanması

```
<!-- hibernate taskdef -->
<taskdef name="hibernatedoclet"
   classname="xdoclet.modules.hibernate.HibernateDocletTask">
  <classpath refid="xdoclet.classpath"/>
</taskdef>
```

Ant ve XDoclet Ortak Çalıştırma

Ant içerisinden XDoclet ile Hibernate eşleştirme dosyalarının üretilmesi

```
<!-- Execute the hibernatedoclet task -->
 <target name="hibernate.generate" depends="prepare" description="generates hibernate mappings">
  <!--    Hand crafted hibernate mappings-->
  <copy todir="${temp.dir}">
   <fileset dir="${src.hibernate.dir}"/>
  </copy>
  <hibernatedoclet
   destdir="${temp.dir}“ mergeDir="${merge.dir}">

   <fileset dir="${src.java.dir}">
    <include name="**/*.java"/>
   </fileset>

   <hibernate version="2.0"/>
  </hibernatedoclet>

  <!-- Pack the generate d *.hbm.xml mapping files into {hibernate.mappings} -->
  <jar destfile="${hibernate.mappings}" basedir="${temp.dir}">
   <include name="**/*.hbm.xml"/>
  </jar>
 </target>
```

XDoclet Hakkında Bilgi

- XDoclet proje sayfası
  - Xdoclet.sourceforge.net
- XDoclet In Action
  - http://www.manning.com/walls/

Özet

- Bu sunumda tamamen açık kaynaklı bileşenler ve araçlarla web tabanlı bir J2EE uygulamasının gerçekleştirilmesi üzerinde durulmuştur.
- Bu sunumda anlatılan teknolojiler hakkında daha fazla bilgi için ilgili teknolojinin proje sayfası, API dokümanları, ilgili forumlar, ve Java Portal'lerinden faydalanılabilir.