

# Struts<sup>2</sup>

# Apache Struts

- Java EE web uygulamaları geliştirmek için kullanılan açık kaynak web uygulama framework'üdür.
- Kullanıcılara MVC mimarisini benimsetmek için Java Servlet API'sini kullanıyor.
- Model-View-Controller basit olarak nedir?
  - Model: Veritabanıyla etki halindeki uygulama mantığıdır.
  - View: İstemciye gösterilen HTML sayfalarıdır.
  - Controller: Bilgiyi model ve view arasında taşıyan aşamadır.
- Struts ile hedeflenen, model'i view ve controller'dan ayırmaktır.

# WebWork ile Birleşme

- WebWork nedir?
  - WebWork de Struts gibi bir Java web uygulama geliştirme framework'üdür.
  - OpenSymphony tarafından geliştirilmiştir.
- 2005 yılında Struts2 çalışması altında birleşmeye karar veriyorlar.



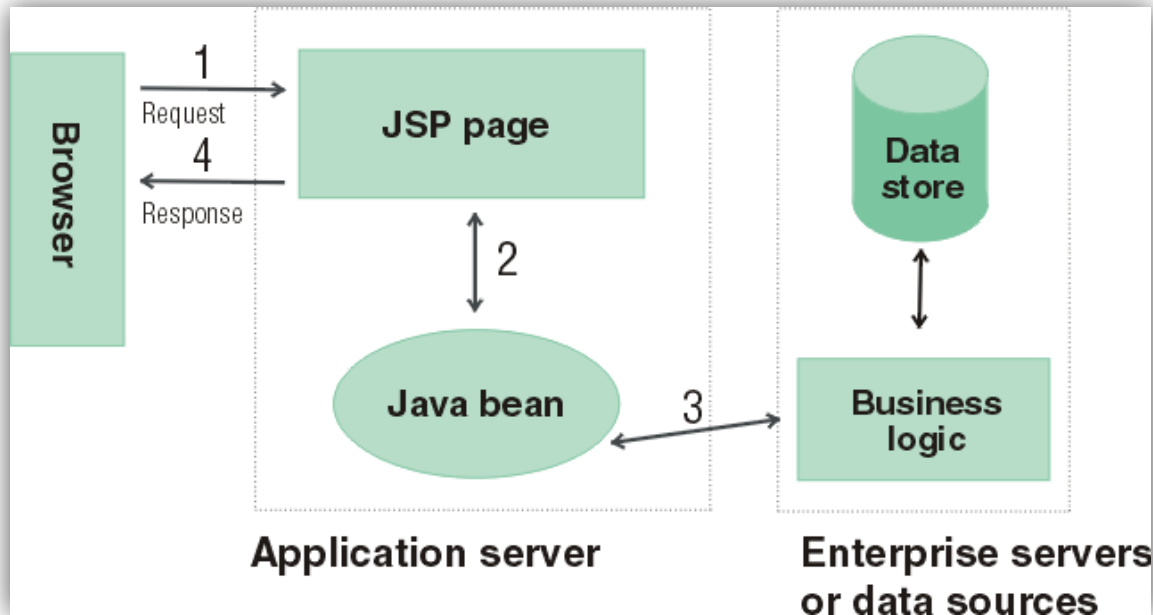
**Struts<sup>2</sup>**

# MVC Design Pattern Nedir?

- Kullanıcıya yüklü miktarda verinin sunulduğu karmaşık uygulamalarda veri ve gösterimin soyutlanması esasına dayanır. Böylece veriler (model) ve kullanıcı arayüzü (view) birbirini etkilemeden düzenlenebilir. MVC, bunu controller adı verilen ara bileşenle, veri gösterimi ve kullanıcı etkileşiminden, veri erişimi ve iş mantığını çıkarma suretiyle çözmektedir.
- **M(Model):** İş Mantığını ve veri işleme süreçlerini yürütür. C(Controller) tarafından gönderilen emirlere göre hareket eder. Bilgi işleme sürecinden sonra veriyi C (Controller)'ye, diğer modellere veya doğrudan V (View)'e gönderir.
- **V(View):** Son kullanıcıya gösterilecek olan verinin sunumu ile ilgilenir. V (View), bu bilgiyi C veya M'den alır, aynı zamanda son kullanıcıdan gelen talepleri C'ye iletir.
- **C(Controller):** Controller ise sistemin ana kısmıdır. Gelen talepleri kontrol eder ve sistemin diğer elemanlarının (M,V) bilgiyi uygun şekilde alıp, göndermelerini sağlar.

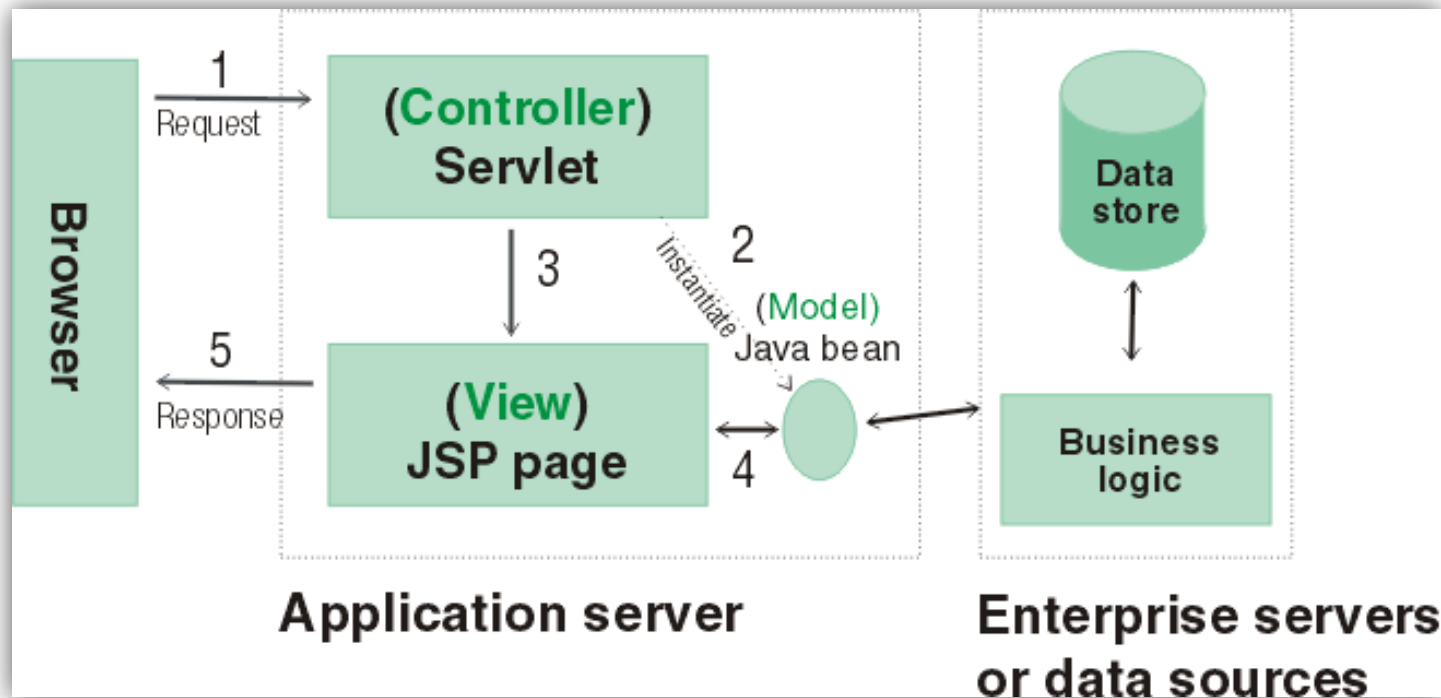
# Servlet, JSP, Model 1, Model 2

- Yalnız Servlet: Çok fazla out.println gerekiyor. Okuması güç.
- Yalnız JSP: Küçük uygulamalar için uygun görünse bile, kodun içine HTML gömmek garip. Okunması ve bakımı güç.
- JSP + Java Beans (Model 1): Küçük uygulamalar için uygun. Bean'ler model kısmını, JSP ise view kısmını oluşturuyor. Birden fazla view kullanılması gerektiğinde etkin değil.



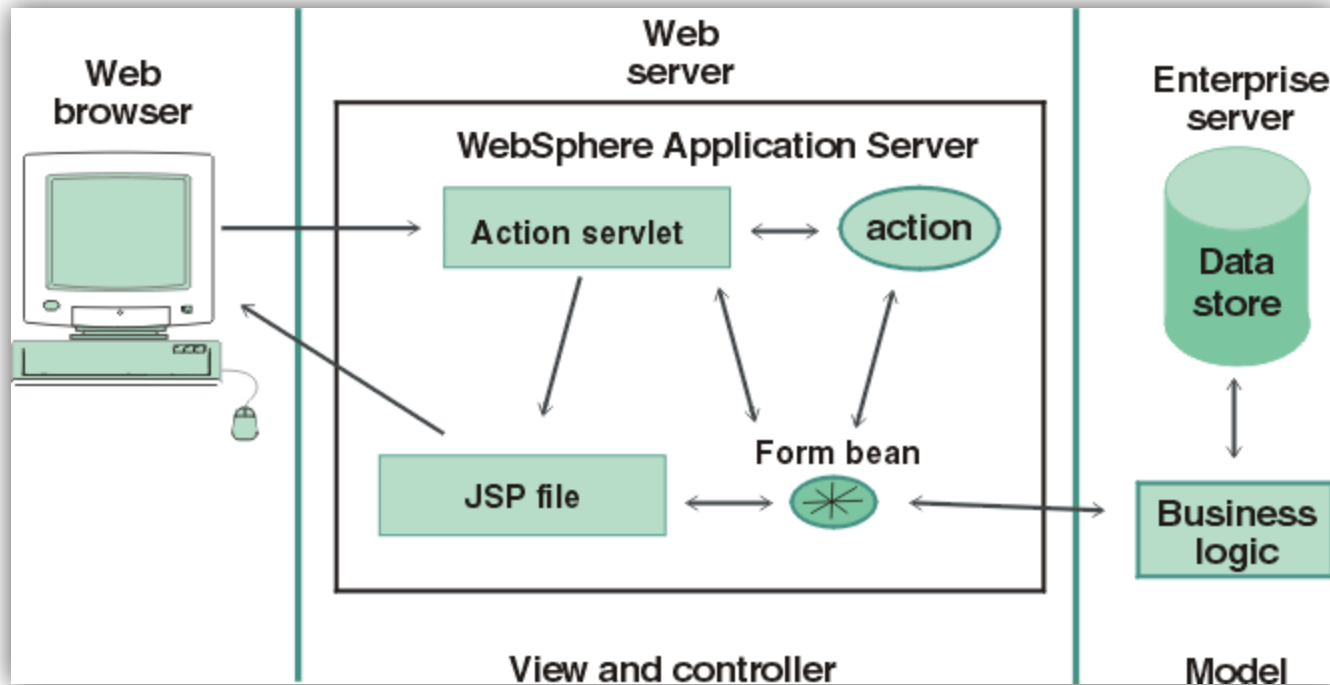
# Servlet, JSP, Model1, Model2

- JSP + Java Beans + Servlet (Model 2): İstek doğrudan servlete yapılıyor ve controller işlevi üstleniyor. Model işlevini Java Bean, view işlevini ise JSP page üstleniyor.
- Struts2'de Model2 MVC şablonu kullanılıyor.



# Struts2, Model 2'yi nasıl kullanıyor?

- Action servlet adında özelleşmiş bir servlet kullanılarak control kısmı gerçekleştiriliyor.
- Ayrıca Form bean adında özelleşmiş bir bean kullanıyor. JSP ilgili HTML'i göstereceği zaman form bean'i kullanıyor. Form bean de, forma konulacak değerleri iş mantığı kısmından almış oluyor.

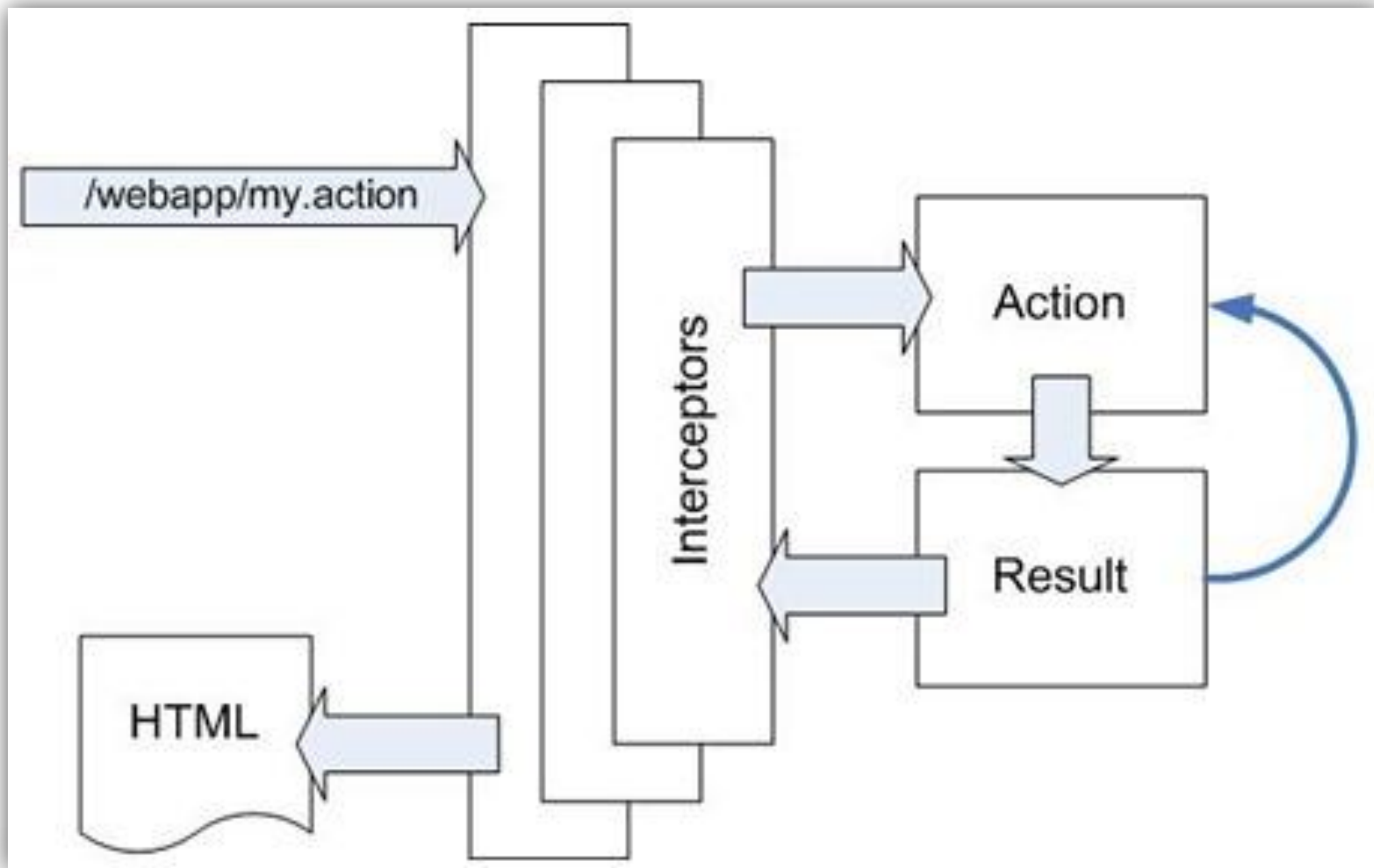


# Struts 2 Mimarisi

- Struts 2'de model, view ve control sırasıyla **Action**, **Result** ve **FilterDispatcher** ile uygulanmaktadır.
- FilterDispatcher, controller görevini kullanıcı isteğini uygun action'la eşleştirerek yapmaktadır.
- Model, veri ve iş mantığını kapsamaktadır. JDBC, EJB, Hibernate vb. ile veri erişimine olanak sağlar.
- Struts 2'de model, Action bileşeni tarafından gerçekleştirilmektedir.
- JSP, JTL, JSF, Jakarta Velocity Engine, Freemaker ya da başka sunum katmanı teknolojisi kullanarak view işlevi gerçekleştirilir.



# İsteklerin (Request) İşletilmesi



# Struts 2 Uygulamalarındaki İstek (Request) Yaşam Döngüsü

1. **Kullanıcı istek gönderir:** Kullanıcı sunucudan kaynak isteğinde bulunur.
2. **FilterDispatcher uygun action'ı seçer:** FilterDispatcher isteğe bakar ve uygun Action'ı belirler.
3. **Interceptorların kullanılması:** İş akışı, onaylama, dosya yükleme gibi fonksiyonlar Interceptor'ın yapılandırılmasıyla kullanılır.
4. **Action Çalıştırılır:** Veritabanı ile ilgili operasyonlar action metodunun çalıştırılmasıyla gerçekleştirilir.
5. **Sonuçlar render edilir:** Sonuçlar Result tarafından render edilir.
6. **İsteğin döndürülmesi:** İstekler ters sırada interceptor'lardan döner. Dönen istek sayesinde temizleme veya fazladan işlem yapabiliriz.
7. **Sonuçları kullanıcıya göster:** Son olarak kontrol servlet container'a döner ve kullanıcı browser'ına çıktıyı gönderir.

# Peki Neden Struts2?

- Açık kaynak,
- 3. parti araç desteği fazla,
- Esnek ve geliştirilebilir,
- Geniş kullanıcı ve geliştirici kitlesi,
- Zengin tag kütüphanesi (html, bean tag),
- Test etmek ve debug yapmak kolay.

# Struts2 İşimizi Nasıl Kolaylaştıracak?

Örnek bir JSP kodu:

**Jsp file:** <input name="txtAmount"> </input>

**In Bean or Jsp file:**

```
String strAmount = request.getParameter("txtAmount");  
double amount = 0.0;
```

```
try{  
    double amount = Double.parseDouble(strAmount );  
}catch(Exception e){}
```

```
bean.setAmount(amount);  
boolean flgResult =.ejbObject.processAmount(amount);  
if(flgResult){}  
else{}
```

# Struts2 İşimizi Nasıl Kolaylaştıracak?

Aynı kodun Struts2 kod örneği:

## Jsp file:

```
<s:textfield label="Amount" name="amount"  
  value="%{amount}" />
```

## Action dosyasındaki tek ihtiyacımız getter ve setter'lar.

```
double amount;  
public double getAmount(){ return amount;}  
public void setAmount(double amount){this.amount =  
  amount;}
```

## Fazladan koda ihtiyaç duymadan action method içindeki amount değişkenini doğrudan kullanabiliriz.

```
public String execute() throws Exception{  
  // use amount directly  
  return "success";  
}
```

Dinlediđiniz iin teŖekkür  
ederiz...

