

Spring Struts Entegrasyonu



Struts Nedir?

- Struts **request ve action** tabanlı bir web framework'tür
- 2001'de çıkmıştır ve döneminde oldukça popüler hale gelmiştir
- UI teknolojisi olarak **JSP ve taglib**'ler kullanılmaktadır.
- **Controller katmanında** web request'leri **action nesneleri** tarafından ele alınmaktadır

Struts 1.x vs 2.x

- Daha sonraki dönemlerde Struts üzerindeki **iyileştirme ve yenileme çalışmaları** söz konusu olmuştur
- Struts 2.x ortaya çıkmıştır
- Struts 2.x günümüz yeni nesil **event oriented UI framework**'lerine daha çok benzemektedir
- Orijinal Struts Web Framework Struts 1.x (**Struts Classic**) olarak anılır olmuştur

Struts 1.x Mimarisi

- Bütün request ve action tabanlı web framework'leri gibi J2EE **front controller** örüntüsünü temel almaktadır

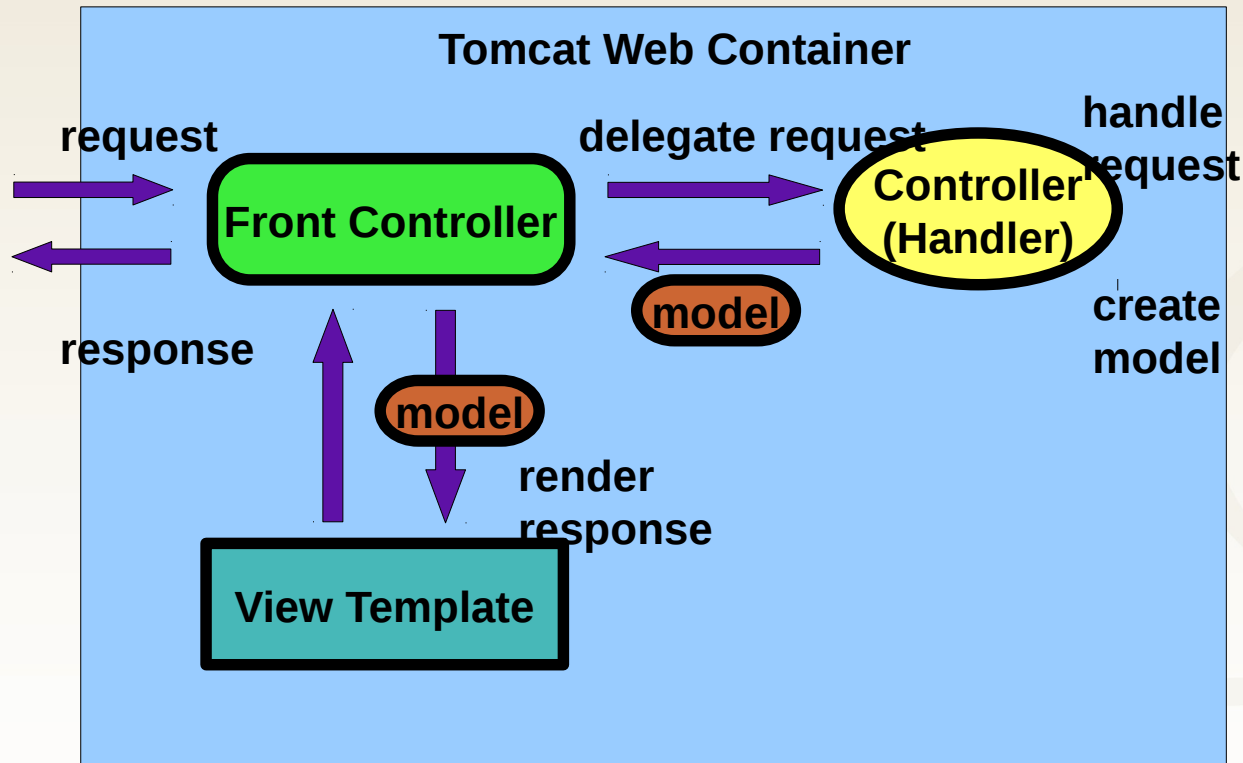
Front Controller



Web uygulamasına gelen bütün requestler Front Controller tarafından ele alınır

Bu sayede uygulamanın ihtiyaç duyduğu güvenlik, internationalization, view render gibi işlemler tek ve merkezi bir noktadan halledilmiş olur

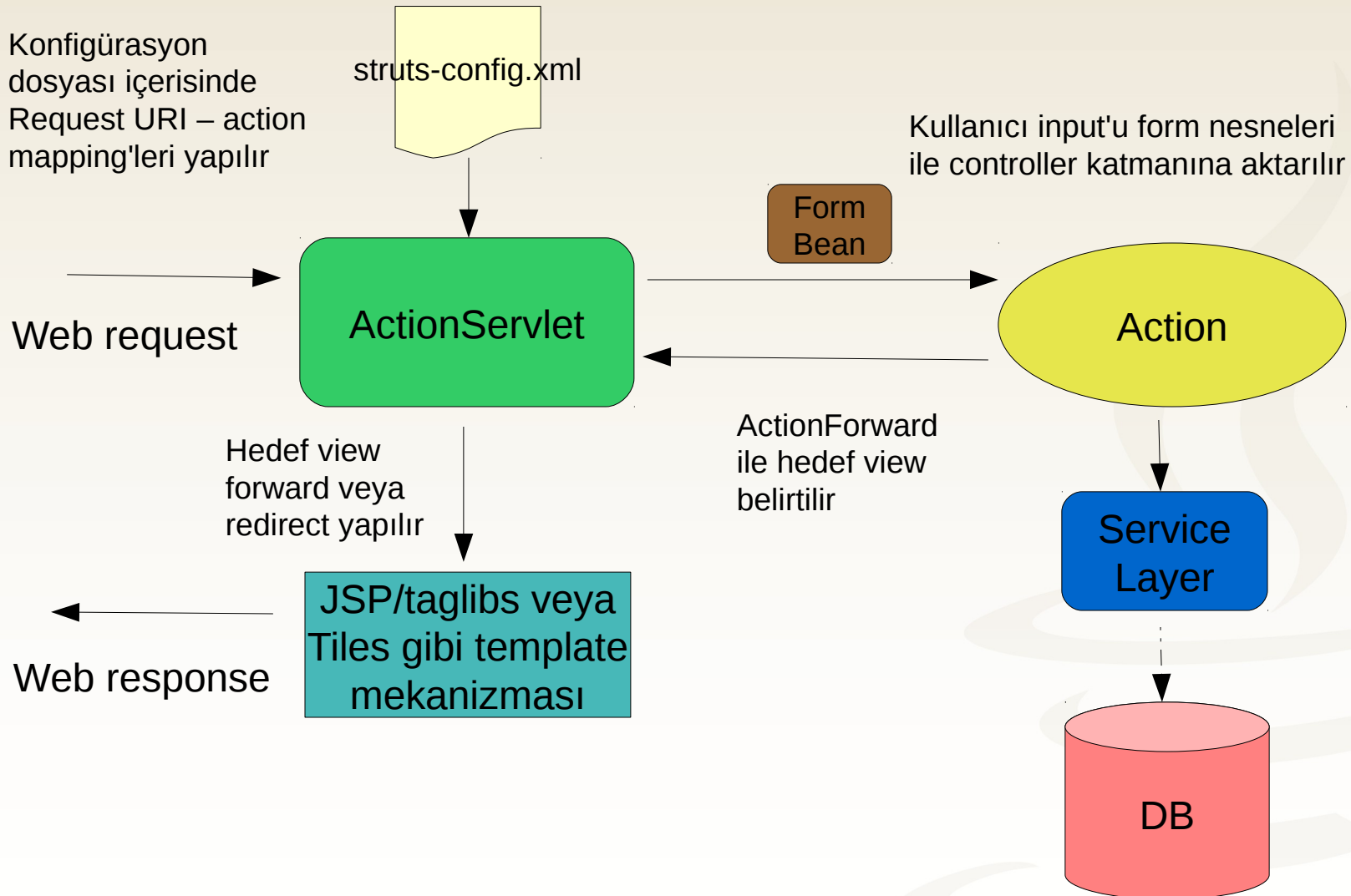
Front Controller



Front Controller genellikle bir Servlet olur, uygulamaya gelen bütün request'leri handle eder

Request'in hangi komuta karşılık geldiğini çözümlemesi için request'in path'inden, parametrelerinden vs yararlanır

Struts 1.x Mimarisi



Spring ve Struts 1.x

- Spring – Struts 1.x entegrasyonunda iki farklı yol izlenebilir
 - **ContextLoaderPlugin** kullanarak
 - **ActionSupport** kullanarak

ContextLoaderPlugin

- Struts 1.1+ plugin'idir
- Spring bean konfigürasyon dosyasını **ActionServlet** için yükler
- Oluşturulan ApplicationContext, **parent container** olarak ContextLoaderListener ile yüklenen **WebApplicationContext**'i görür
- İstenirse bütün konfigürasyon dosyaları bu plugin tarafından da yüklenebilir

ContextLoaderPlugin Konfigürasyonu

```
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">  
  <set-property property="contextConfigLocation"  
    value="/WEB-INF/action-servlet.xml,/WEB-INF/applicationContext.xml"/>  
</plug-in>
```

struts-config.xml

içindeki plugin
tanımı

Default konfigürasyon ActionServlet'i ismi-servlet.xml
isimli bir dosyayı WEB-INF altında arar

```
<controller>
```

```
  <set-property property="processorClass"  
    value="org.springframework.web.struts.DelegatingRequestProcessor"/>
```

```
</controller>
```

struts-config.xml

içindeki
controller
tanımı

Bu sayede action nesneleri doğrudan Spring
container içerisinde resolve edilecektir

ContextLoaderPlugin Kullanımı

```
<action path="/users" .../>
```

struts-config.xml içindeki action tanımı

Action path ile bean name eşleşmelidir

```
<bean name="/users" .../>
```

Spring container içerisindeki bean tanımı

Bean scope'unun prototype olması önemlidir

Eğer DelegatingRequestProcessor kullanılmıyor ise, bu durumda action tanımı struts-config.xml içerisinde aşağıdaki gibi yapılabilir

```
<action path="/user" type="org.springframework.web.struts.DelegatingActionProxy"
  name="userForm" scope="request" validate="false" parameter="method">

  <forward name="list" path="/userList.jsp"/>
  <forward name="edit" path="/userForm.jsp"/>

</action>
```

ActionSupport Kullanımı

- İkinci alternatif Action sınıflarınızın Spring'in **ActionSupport** sınıflarından uygun olan birisini extend etmesidir
- Bu durumda action nesnesi içerisinde `WebApplicationContext`'e **`getWebApplicationContext()`** metodu vasıtası ile erişmek mümkün olacaktır
- Daha sonra da **bean lookup** gerçekleştirilebilir

Spring ve Struts 2.x

- Struts 2.x built-in Spring plugin'i ile gelmektedir
- Bu sayede **Action**, **Interceptor** ve **Result nesneleri** doğrudan **Spring bean'ları** olarak tanımlanıp yönetilebilmektedir
- Etkinleştirmek için struts2-spring-plugin-x-x-x.jar'ın classpath'de mevcut olması yeterlidir

İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

