

Java Server Faces (JSF)



JSF Nedir?

- Web tabanlı Java uygulamaları geliştirmek için tasarlanmış bir **UI framework**'tür
- **Bileşen (component) tabanlı** UI sayfaları geliştirme imkanı sağlar
- **Event tabanlı** çalışır
- Kabiliyetleri, davranışı bir **spesifikasyon** ile tanımlanır
- **Farklı implemantasyonları** olabilir

JSF'in Özellikleri

- UI sayfalarının ihtiyaç duyduğu kodu “**managed beans**” isimli bir katman da ele almayı sağlar
- Bunlara **backing bean** adı da verilmektedir
- Bu bölüm **iş katmanı ile entegrasyonu** da sağlar
- Spring, CDI gibi çözümlerle de kolayca entegre kolabilir

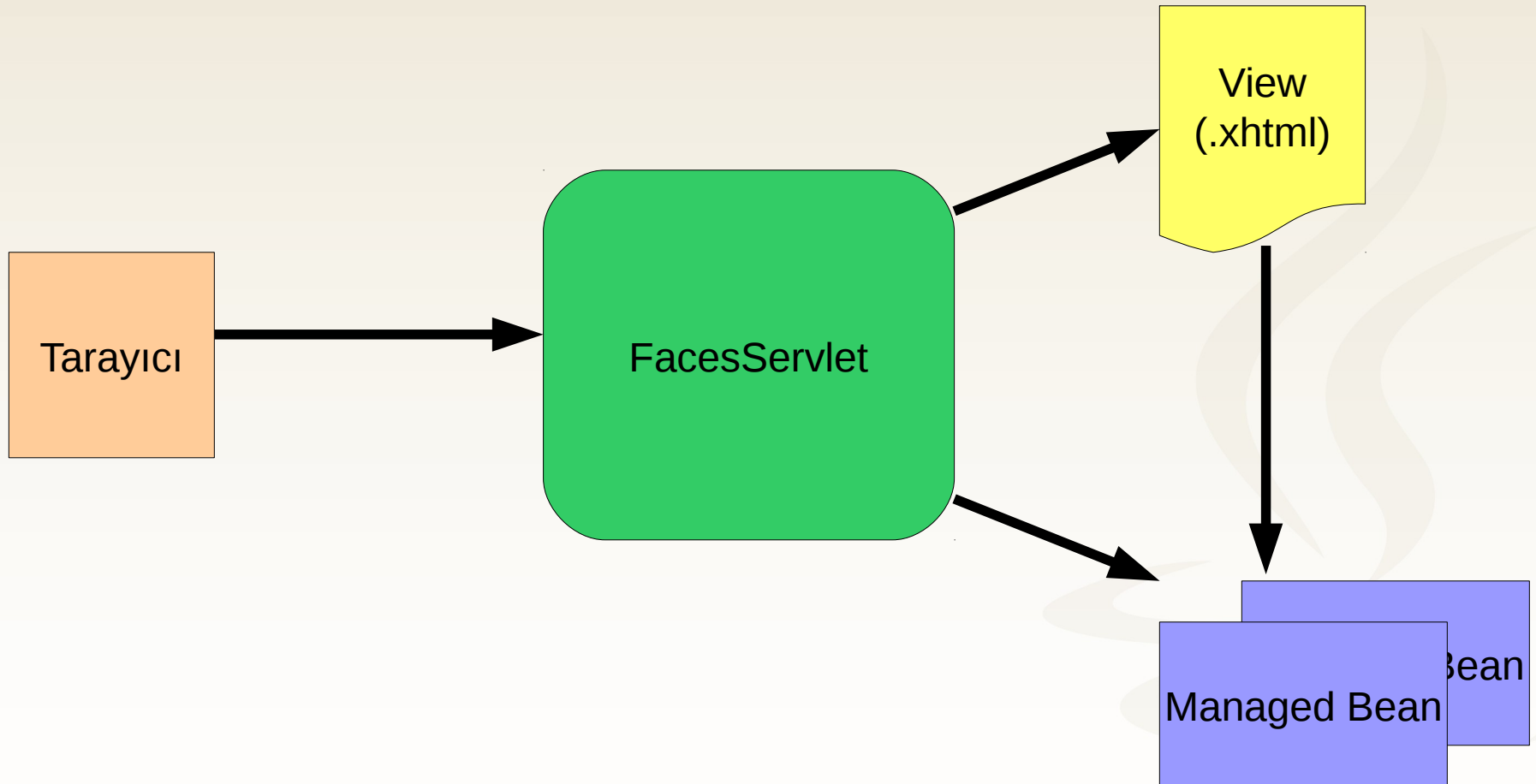
JSF'in Özellikleri

- Facelets kabiliyeti üzerinden **template tabanlı** custom component geliştirme imkanı sunar
- **Expression language** desteği vardır
- UI sayfaları içerisinde bu ifadeler kullanılarak **managed bean'ler ile etkileşim** sağlanabilir
- Built-in **ajax desteği** vardır

JSF'in Özellikleri

- Built-in HTML tabanlı **UI tag bileşen kütüphanesi** vardır
- Dışarıdan (3rd party) **bileşen kütüphaneleri** de (PrimeFaces gibi) kullanılabilir
- Event tabanlı programlama yapma imkanı sunar
- UI event'leri yakalanıp, **managed bean'ler üzerinde ele almak** mümkündür
- State yönetimi için **scope desteği** sunar

JSF'in Mimarisi



JSF Konfigürasyonu

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>
    javax.faces.webapp.FacesServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

```
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
```

JSF Konfigürasyonu

```
<context-param>
```

```
<param-name>
```

```
    javax.faces.PROJECT_STAGE
```

```
</param-name>
```

```
<param-value>Development</param-value>
```

```
</context-param>
```

Geliştirme sürecinde bug'ların tespiti için faydalı bir parametredir. Hata mesajlarının daha detaylı olmasını sağlar, client side JS hatalarını da görüntüler. Performansa negatif etkisi vardır. Gerçek ortamda "Production" değeri ile değiştirilebilir.

JSF Konfigürasyonu

- Ayrıca **faces-config.xml** isimli bir konfigürasyon dosyası da vardır
- Burada JSF'e özel davranış ayarları, managed bean tanımları, navigasyon kuralları belirtilir
- JSF 2 ile birlikte bu tanımlar **anotasyon tabanlı** olarak yapılabilmektedir
- Dolayısı ile **faces-config.xml** opsiyoneldir

JSF Konfigürasyonu

- **Birden fazla** faces-config.xml dosyası da olabilir

```
<context-param>
  <param-name>
    javax.faces.application.CONFIG_FILES
  </param-name>

  <param-value>
    /WEB-INF/faces-config1.xml,
    /WEB-INF/faces-config2.xml
  </param-value>
</context-param>
```

JSF Managed Bean Kavramı

- Kullanıcının gönderdiği input'u ele alan, işleyen, kullanıcıya gösterilecek çıktıyı oluşturan, UI event'lerini yakalayan, UI iş mantığının yer aldığı **Java nesneleridir**
- **Backing bean** adı da verilir
- Bu nesneleri yaratmak ve yönetmek **JSF container'ın görevidir**
- faces-config.xml içerisinde veya **@ManagedBean** anotasyonu ile tanımlanır

JSF Managed Bean Kavramı

- Managed bean'ler **arasında ilişki** kurulabilir
- Bu tür ilişkiler kurmak için **@ManagedProperty** anotasyonu kullanılır
- Enjekte edilecek dependency üzerinde bu anotasyon kullanılır
- Değerin enjekte edilebilmesi için **setter metoda ihtiyaç duyulur**

Managed Bean ve Scope Kavramı

- Managed bean instance'larının **ömürleri farklı farklı** olabilir
- Bazı bean'ler request boyunca, diğerleri ise session boyunca aktif olabilirler
- **Default scope request'tir**
- @RequestScoped, @ViewScoped, @SessionScoped, @ApplicationScoped anotasyonları ile değer scope'lar da kullanılabilir

JSF Scope Tanımları

- Request Scope
 - **@RequestScoped** anotasyonu ile tanımlanır
 - Bean request başında yaratılır
 - Request boyunca aynı instance kullanılır
 - Response'un sonlanması ile birlikte destroy edilir
 - **ExternalContext.getRequestMap()** metodu ile request scope'a programatik olarak da erişilebilir

JSF Scope Tanımları

- View Scope
 - **@ViewScoped** anotasyonu ile tanımlanır
 - Aynı JSF view render edildiği müddetçe aynı instance kullanılır
 - Instance'ın korunması için **Action/Listener metotlarından NULL veya void** dönülmesi gerekir
 - **UIViewRoot.getViewMap()** metodu ile view scope'a programatik biçimde erişilebilir

JSF Scope Tanımları

- Flash Scope
 - Bu scope için herhangi bir **anotasyon mevcut değildir**
 - Bean düzeyinde kullanılamaz
 - Flash scope'daki bir veri **bir sonraki request'de de korunur**, ancak müteakip request'lerde korunmaz
 - **Response.sendRedirect** kullanımları içindir
 - **ExternalContext.getFlash()** metodu ile bu scope'a programatik olarak erişilebilir

JSF Scope Tanımları

- Session Scope
 - **@SessionScoped** anotasyonu ile tanımlanır
 - HTTP Session'ın ömrü boyunca aynı bean instance kullanılır
 - **ExternalContext.getSessionMap()** metodu ile programatik biçimde erişilebilir

JSF Scope Tanımları

- Application Scope
 - **@ApplicationScoped** anotasyonu ile tanımlanır
 - Web uygulaması boyunca aynı bean instance kullanılır
 - **ExternalContext.getApplicationMap()** metodu ile programatik olarak erişilebilir

Managed Bean ve Managed Property Örneği

```
@ManagedBean
@ViewScoped
public class VetListBackingBean {

    @ManagedProperty("#{petClinicService}")
    private PetClinicService petClinicService;

    private Collection<Vet> vets;

    public void setPetClinicService(PetClinicService
petClinicService) {
        this.petClinicService = petClinicService;
    }

    @PostConstruct
    public void init() {
        vets = petClinicService.findVets();
    }

    public Collection<Vet> getVets() {
        return vets;
    }
}
```

Built-in JSF HTML Tag Kütüphanesi

- JSF spesifikasyonu içerisinde UI bileşenleri ve JSF fonksiyonları için **iki tane built-in tag kütüphanesi** vardır

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html">
```

```
<h:head>
```

```
...
```

```
</h:head>
```

```
<h:body>
```

```
...
```

```
</h:body>
```

```
</html>
```

JSF core ve html tag kütüphanelerinin namespace tanımları yapılır

JSF XHTML Sayfaları

- JSF sayfalarının **XHTML formatında** olması önerilmektedir

```
<html ...>

<h:head>
  <title>JSF 2 Test</title>
</h:head>

  <h:body>
    <h:form>
      <h:inputText
        value="#{helloBean.name}" />
      <h:commandButton
        value="click" action="welcome" />
    </h:form>
  </h:body>
</html>
```

JSF UI bileşenleri
<h:form> elemanı
içerisinde
kullanılmalıdır

JSF Tag Örnekleri

Username: `<h:inputText
value="#{userBean.username}" />
`

Password: `<h:inputSecret
value="#{userBean.password}" />
`

Age: `<h:inputText value="#{userBean.age}">
 <f:convertNumber/>
</h:inputText>
`

Weight: `<h:inputText value="#{userBean.weight}">
 <f:convertNumber minFractionDigits="2" />
</h:inputText>
`

JSF Tag Örnekleri

```
Birth Date:<h:inputText value="#{userBean.birthDate}">
    <f:convertDateTime pattern="dd/MM/yyyy" type="date"/>
</h:inputText><br/>
```

```
Gender:<h:selectOneRadio value="#{userBean.gender}">
    <f:selectItems value="#{userBean.availableGenders}"
var="g" itemValue="#{g.name}" itemLabel="#{g.label}"/>
</h:selectOneRadio><br/>
```

```
Occupations:<h:selectOneListbox
value="#{userBean.occupation}">
    <f:selectItems
value="#{userBean.availableOccupations}"/>
</h:selectOneListbox><br/>
```

```
<h:commandButton value="OK" action="show"/>
```

JSF Tag Örnekleri

```
<h:dataTable var="vet" value="${vetListBackingBean.vets}">
  <h:column>
    <f:facet name="header">First Name</f:facet>
    ${vet.firstName}
  </h:column>
  <h:column>
    <f:facet name="header">Last Name</f:facet>
    <h:outputText value="${vet.lastName}"/>
  </h:column>
  <h:column>
    <f:facet name="header">Graduation Year</f:facet>
    <h:outputText value="${vet.graduationYear}"/>
  </h:column>
  <h:column>
    <f:facet name="header">Edit</f:facet>
    <h:commandButton value="Edit"
action="#{vetListBackingBean.editVetDetail(vet.id)}" />
  </h:column>
</h:dataTable>
```


JSF Tag Örnekleri

```
<f:view locale="#{userPref.locale}"
        encoding="UTF-8" contentType="text/html">
    <h:form>
        <h:outputLabel>First Name :</h:outputLabel>
        <h:inputText value="#{vetDetailBean.vet.firstName}"/>
        <br/>
        <h:outputLabel>Last Name :</h:outputLabel>
        <h:inputText value="#{vetDetailBean.vet.lastName}"/>
        <br/>
        <h:outputLabel>Graduation Year :</h:outputLabel>
        <h:inputText
value="#{vetDetailBean.vet.graduationYear}"/>
        <br/>
        <h:commandButton action="#{vetDetailBean.save}"/>
    </h:form>
</f:view>
```

HTML Görüntüleri

Username:

Password:

Age:

Weight:

Birth Date:

Gender:

☒ Bayan ☐ Bay

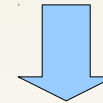
Occupations:

Avukat

Doktor

Mühendis

```
Username:#{userBean.username}<br/>
Password:#{userBean.password}<br/>
Age:#{userBean.age}<br/>
Weight:#{userBean.weight}<br/>
Birth Date:#{userBean.birthDate}<br/>
Gender:#{userBean.gender}<br/>
Occupation:#{userBean.occupation}<br/>
```



Username:q

Password:

Age:123

Weight:12.2

Birth Date:Sat Dec 01 02:00:00 EET 2001

Gender:female

Occupation:avukat

JSF ve Facelets

- JSF sayfaları için **şablon (template) kabiliyeti** sunar
- JSF bileşenlerinin gruplanarak (composition) farklı sayfalar içerisinde **yeniden kullanılabilmesini** de sağlar

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:ui="http://java.sun.com/jsf/facelets">

</html>
```

Facelets Tags

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html">
```

```
<h:head>
  <title>#{title}</title>
</h:head>
```

Şablonun içerisinde "name" ile belirtilen bölüme içeriğin eklenmesini sağlar

```
<h:body>
  <div id="top" class="top">
    <ui:insert name="top">Top Section</ui:insert>
  </div>
  <div>
    <div id="left">
      <ui:insert name="left">Left Section</ui:insert>
    </div>
    <div id="content">
      <ui:insert name="content">Main Content</ui:insert>
    </div>
  </div>
</h:body>
</html>
```

Facelets Tags

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html">
```

```
<h:body>
```

Belirtilen şablon dosyası ile bir page composition yapmayı sağlar, tag'in dışındaki kısımlar göz ardı edilir

```
<ui:composition template="./template.xhtml">
  <ui:param name="title" value="Test Facelets"/>
```

```
<ui:define name="top">
```

Şablonun içerisine parametre değeri geçmeyi sağlar

```
  Welcome to Test Page
</ui:define>
```

Şablon içerisinde name ile eşleşen bölüme bu tag'in içeriğinin insert edilmesini sağlar

```
<ui:define name="left">
```

```
  <h:outputLabel value="You are in the Left Section"/>
</ui:define>
```

```
<ui:define name="content">
```

```
  <h:outputText value="You are in the Main Content
```

```
Section"/>
```

```
</ui:define>
```

```
</ui:composition>
```

```
</h:body>
```

```
</html>
```

Facelets Tags

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html">

  <ui:include src="/templates/header.xhtml">
    <ui:param name="title" value="Test Facelets"/>
  </ui:include>

  <ui:include src="/templates/body.xhtml"/>

</html>
```

Bir xhtml sayfasının içeriğini başka bir xhtml sayfasının içerisine ekler

İletişim



www.harezmi.com.tr

www.java-egitimleri.com



info@harezmi.com.tr

info@java-egitimleri.com



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)