

# JSF Validation



# Data Validation

- Ekrandan girilen veya değiştirilen **verinin** arka tarafta iş katmanının kullanımına sunulmasından evvel **doğru, geçerli ve anlamlı olup olmadığının denetlenmesi** gerekir
- Bu işleme **validasyon** adı verilir
- JSF'de validasyon UI component tree oluşturulduktan veya restore edildikten ve **form datası ilgili UI bileşenlerine aktarıldıktan sonra** gerçekleşir

# Data Validation

- Validasyon sırasında herhangi bir **hata ortaya çıkarsa** JSF'in yaşam döngüsü yarıda kesilip **hata mesajları** görüntülenecektir
- JSF içerisinde validasyon **dekleratif veya programatik** biçimde gerçekleştirilebilir

# Data Validation

```

<h:form>
  <h:outputLabel>First Name :</h:outputLabel>
  <h:inputText id="firstName"
    value="#{vetDetailBean.vet.firstName}"
    required="true"
    requiredMessage="First Name must be entered"/>
  <h:message for="firstName"/><br />

  <h:outputLabel>Last Name :</h:outputLabel>
  <h:inputText id="lastName"
    value="#{vetDetailBean.vet.lastName}"
    <f:validator
      validator="#{vetDetailBean.validateLastName}"/>
  </h:inputText>
  <h:message for="lastName"/><br />

  ...

```

Standard required validation

Managed bean içerisinde  
Validator metod

# Data Validation

```
@ManagedBean
@ViewScoped
public class VetDetailBean {

    public void validateLastName(
        FacesContext context,
        UIComponent component,
        Object value) throws ValidatorException {

        String lastName = (String) value;
        boolean valid = false;
        //...validate last name value
        if(!valid) {
            throw new ValidatorException(
                new FacesMessage(FacesMessage.SEVERITY_ERROR,
                    "Last name invalid", "Invalid last name value entered"));
        }
    }
    ...
}
```

# Data Validation

Custom validator

```
<h:inputText id="email"
    value="#{vetDetailBean.vet.email}">
    <f:validator validatorId="emailValidator"/>
</h:inputText>
<h:message for="email"/><br/>
```

```
<h:outputLabel>Graduation Year :</h:outputLabel>
<h:inputText id="gy"
    value="#{vetDetailBean.vet.graduationYear}">
    <f:validateLongRange minimum="1999" maximum="2016"/>
</h:inputText>
<h:message for="gy"></h:message><br />
```

```
<h:commandButton value="Save"
    action="#{vetDetailBean.save}" />
```

```
<br/>
<h:messages/>
```

Built-in range validator

```
</h:form>
```

# Custom Validator Implementasyonu


```
@FacesValidator("emailValidator")
public class EmailValidator implements Validator {

    @Override
    public void validate(
        FacesContext context,
        UIComponent component,
        Object value) throws ValidatorException {
        String email = (String) value;
        boolean valid = false;
        //...validate e-mail value
        if(!valid) {
            throw new ValidatorException(
                new FacesMessage(FacesMessage.SEVERITY_ERROR,
                    "E-mail invalid", "Invalid e-mail value
entered"));
        }
    }
}
```

# JSR-303 Bean Validation API Desteği

```
<h:inputText id="email"
              value="#{vetDetailBean.vet.email}">
</h:inputText>
<h:message for="email"/><br/>
```

JSR-303 bean validation



```
public class Vet {

    private String firstName;

    private String lastName;

    @NotNull(message="E-mail girilmeli")
    @Email(message="Geçerli bir e-mail adresi girilmeli")
    private String email;

    private Integer graduationYear;

    ...
}
```



# İletişim



[www.harezmi.com.tr](http://www.harezmi.com.tr)

[www.java-egitimleri.com](http://www.java-egitimleri.com)



[info@harezmi.com.tr](mailto:info@harezmi.com.tr)

[info@java-egitimleri.com](mailto:info@java-egitimleri.com)



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

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