JavaServer Faces (JSF)

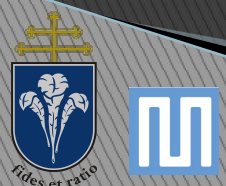
Tartalom

- Hibakezelés
- Validátorok
- Összetett komponensek
- Template használat
- Komponens könyvtárak

Hibakezelés – szerver oldal

- FacesMessage (JSF hibaüzenet)
  - FacesContext-ben tároljuk a hibaüzeneteket egy message queue-ban (alkalmazás által vagy kézzel generált)
  - Severity: INFO, WARN, ERROR, FATAL…
  - Summary
  - Detail

```
FacesContext facesContext =
FacesContext.getCurrentInstance();
    facesContext.addMessage("message1",
        new FacesMessage(FacesMessage.SEVERITY_INFO,
"Info message.", "Example of an info message."));
```



Hibaüzenet megjelenítése – kliens oldal

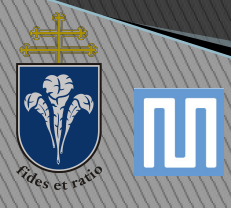
```
<h:form>
 <h:messages/>
    <h:message errorStyle="color:red; display:block" for="myInput" />
    <h:inputText id="myInput" required="true" label="My Input" />
    <h:commandButton value="Click" />
</h:form>
```

Validátorok

1. Beépített validátorokon keresztül

```
<h:inputText id="name" value="#{bean.value}" >
    <f:validateRequired />
    <f:validateLength minimum="5" maximum="10" />
</h:inputText>
```

- DoubleRangeValidator
- LongRangeValidator
- LengthValidator
- + required



Validátorok

2. Alkalmazásszintű validáció

- Az eseménykezelő metódusban, programozottan validálunk
- Komplexebb, üzleti-logika jellegű validációt érdemes így kezelni

```
public void submit(){
        …
        FacesContext.
                getCurrentInstance().
                addMessage("myinput",
                new FacesMessage(FacesMessage.SEVERITY_ERROR,
                "Invalid value", "Value is invalid"));
}
```

Validátorok

3. Inline validáló metódus

```
<h:inputText id="name" value="#{bean.value}"
    validator="#{bean.validate} ">
</h:inputText>
<h:message for=„name" />
```

```
public class MyBean
{
public void validate(FacesContext context, UIComponent
component, Object value) {
 if (value==null) {
    UIInput input = (UIInput) component;
    input.setValid(false);
    context.addMessage(
    component.getClientId(context),
    new FacesMessage(FacesMessage.SEVERITY_ERROR,
    "The value must not be null.",
    "The value must not be null."));
    }
}
```

Validátorok

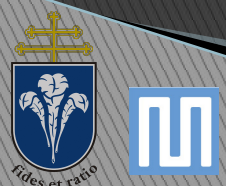
4. Saját validátor

```
@FacesValidator("emptyInputValidator")
public class EmptyInputValidator implements Validator
{
@Override
public void validate(FacesContext context, UIComponent
component, Object value) {
    if (value==null) {
     UIInput input = (UIInput) component;
     input.setValid(false);
     context.addMessage(
     component.getClientId(context),
     new FacesMessage(FacesMessage.SEVERITY_ERROR,
     "The value must not be null.",
     "The value must not be null."));
     }
}
```

```
<h:inputText id=„username” value="#{userBean.userName}">
      <f:validator validatorId="emptyInputValidator"/>
</h:inputText>
```

Összetett komponensek építése

- JSF keretrendszer segítségével építhetünk saját komponenseket is **(composite components):**
- **Webapp/resources** mappában létrehozni egy mappát – a mappa nevével fogjuk tudni elérni a komponenskönyvtárunkat pl. **resources/mycomp**
- A fenti mappában létre kell hoznunk a komponensünket– pl. **resources/mycomp/myInputComp.xhtml**



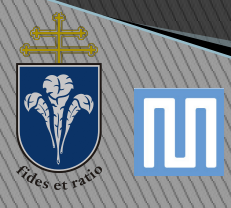
Összetett komponens építése

- Komponens deklarálása

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:composite="http://xmlns.jcp.org/jsf/composite"
xmlns:h="http://xmlns.jcp.org/jsf/html">

<composite:interface>
    <composite:attribute name= "labelValue" required= "true"/>
     <composite:attribute name= "dataValue" required="true"/>
</composite:interface>

<composite:implementation>
    <h:outputLabel value="#{cc.attrs.labelValue}„/>
    <h:inputText value="#{cc.attrs.dataValue}"></h:inputText>
</composite:implementation>

</html>
```

Összetett komponens használata

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:mycomponents="http://xmlns.jcp.org/jsf/composite/mycomponents">

<mycomp:myInputComp labelValue="Name: "
dataValue="#{customerBean.name}" >

</html>
```

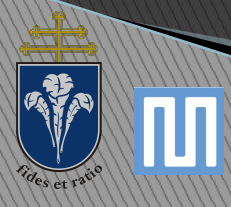


Facelet Template

- A template file valójában egy xhtml file, amely definiálja az oldal struktúráját, kinézetét. Egységes kinézetet ad az oldalaknak.
- Template-ek használata:
  - Template file definiálása (ui:insert)
  - Közös oldalak definiálás (ui:composition)

Használható tagek

- **ui:insert** – template file-ban használandó, olyan tartalmat határoz meg, amit a template-et alkalmazó oldal fog kitölteni. A tartalmat az "ui:define" tag-el lehet specifikálni. Megadható default megvalósítás, amit a template-et használó oldalak felüldefiniálhatnak.
- **ui:define** – tartalmat definiál, amelyet a template megfelelő nevű "ui:insert" tag-jébe szúrunk be – a name attribútum kapcsolja össze a kettőt
- **ui:include** – másik xhtml, facelet oldal szúrható be
- **ui:composition** – "template" attribútummal használva betölti a templatet. Egyébként elemek olyan csoportját definiálja, amelyek beszúrhatóak más oldalakba.



Facelet template minta

- Template.xhtml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      >
    <h:head>
   <h:outputStylesheet name="common-style.css" library="css" />
    </h:head>
    <h:body>
    <div id="page">
   <ui:insert name="header" > Default header </ui:insert>
     <ui:insert name= "menu" ></ui:insert>
   <ui:insert name="content" >
       <ui:include src="/commonContent.xhtml" />
   </ui:insert>
         <ui:insert name="footer" ></ui:insert>
</h:body>
</html>
```

Template-et használó oldal

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      >
    <ui:composition template="template/template.xhtml">
        <ui:define name="header">
                <h2>This is page1 header</h2>
            </ui:define>
        <ui:define name="menu">
                <h2>This is page1 menu</h2>
            </ui:define>
        <ui:define name="content">
                <h2>This is page1 content</h2>
            </ui:define>
        <ui:define name="footer">
                <h2>This is page1 Footer</h2>
            </ui:define>
        </ui:composition>
</html>
```

Komponens könyvtárak

- Primefaces – http://www.primefaces.org/showcase
- Richfaces – http://showcase.richfaces.org/ (JBOSS)
- IceFaces – http://icefaces-showcase.icesoft.org