

Aplicații Web cu Java Server Faces JSF

JavaServer Faces este un framework de componente server-side care permite dezvoltarea de aplicații Web bazate pe tehnologii Java. JavaServer Faces se compune din următoarele:

- API pentru reprezentarea componentelor și controlul stării acestora; pentru tratarea evenimentelor, pentru validări server-side și conversii de date, pentru definirea modulului de navigare; internaționalizare și accesibilitate.
- Biblioteci de tag-uri pentru adăugarea de componente la paginile Web și pentru conectarea componentelor la obiecte server-side.

Prin urmare, JavaServer Faces oferă un model de programare bine-definit și mai multe biblioteci de tag-uri, ușurând crearea și întreținerea aplicațiilor Web.

O aplicație JSF obișnuită cuprinde următoarele:

- Un set de pagini Web în care se așează componentele.
- Un set de tag-uri pentru a adăuga componentele pe pagina.
- Un set de "backing beans", componente JavaBeans care definesc proprietățile și funcțiile componentelor unei pagini.
- Un fișier web.xml (deployment descriptor).
- Fișierele opționale de configurare (faces-config.xml).
- Diverse componente opționale (componente custom, validatori, convertori, interceptori).
- Un set de tag-uri dedicate (custom) pentru reprezentarea obiectelor custom în pagină.

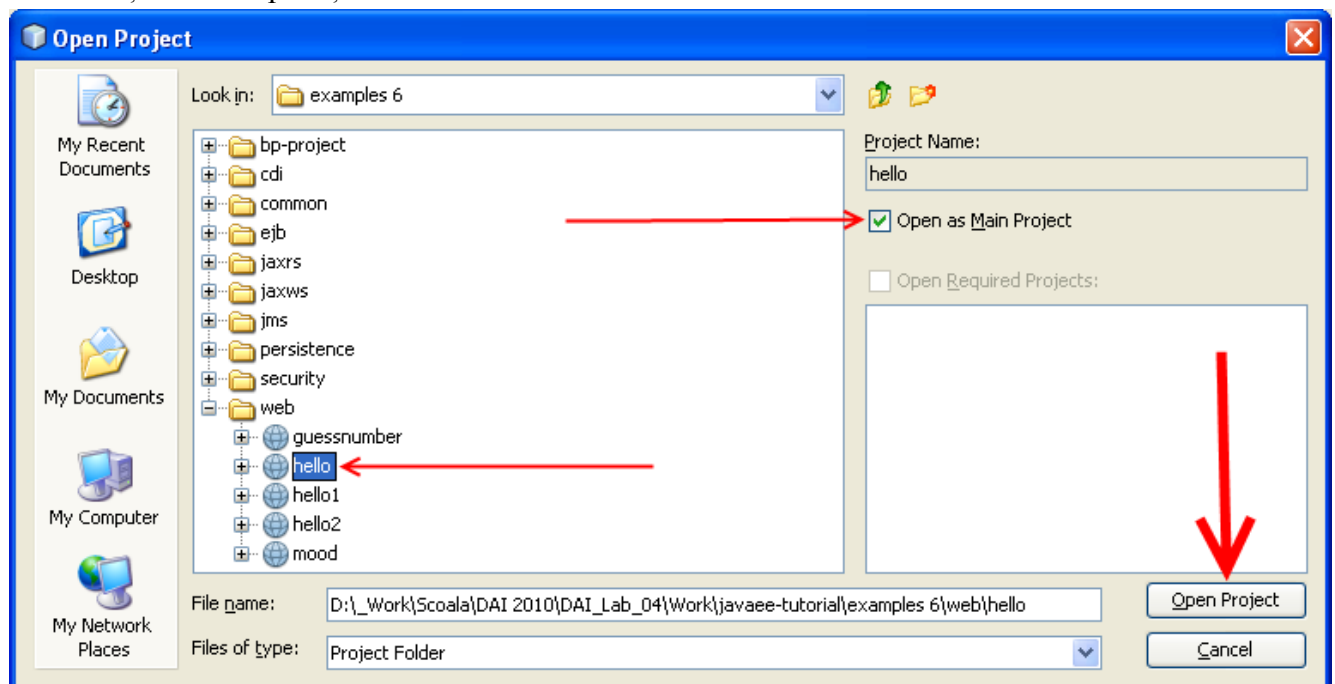
1. Aplicație simplă JSF (Hello)

O aplicație simplă JSF presupune realizarea următorilor pași:

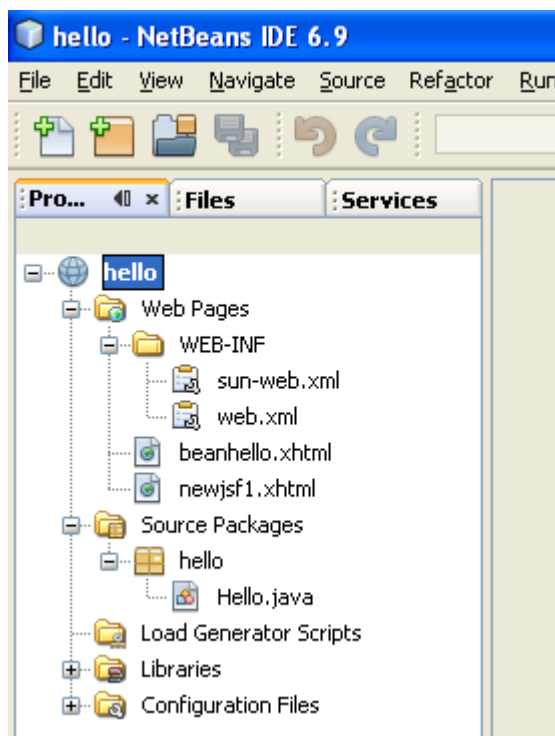
- Creare backing beans.
- Creare pagini Web.
- Mapări.

Aplicația Hello din directorul "DAI_Lab_04\javaee-tutorial-examples.zip\examples 6\web\hello\" este o astfel de aplicație simplă, care conține un backing bean și o pagină Web.

Deschideți această aplicație folosind NetBeans.



Observați organizarea proiectului.



Analizați fișierul "Hello.java", backing bean-ul proiectului.

```

10 package hello;
11
12 import javax.faces.bean.ManagedBean;
13
14 @ManagedBean
15 public class Hello {
16     final String world = "Hello DAI World!!!";
17
18     public String getworld() {
19         return world;
20     }
21 }
22

```

Adnotarea "@ManagedBean" înregistrează acest backing bean ca resursă pentru JavaServer Faces. Analizați fișierul "beanhello.xhtml", o pagină XHTML.

```

1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml"
5       xmlns:h="http://java.sun.com/jsf/html">
6 <h:head>
7     <title>Facelets Hello World</title>
8 </h:head>
9 <h:body>
10     #{hello.world}
11 </h:body>
12 </html>

```

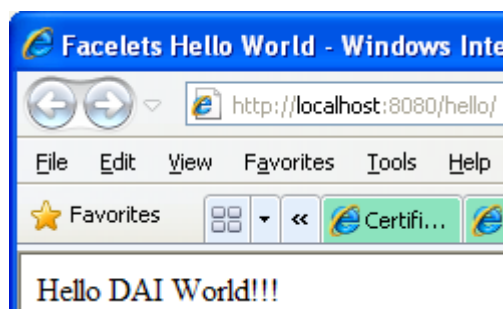
Observați tag-urile folosite.

Preluarea valorii din backing bean se face prin conectarea paginii la bean prin intermediul Expression Language (EL), care folosește expresia `#{hello.world}` pentru a prelua valoarea proprietății `world`.

Observați că se folosește "hello" pentru a referi bean-ul Hello. Dacă nu se precizează nimic la adnotarea @ManagedBean, backing bean este accesat cu prima literă minusculă.

Analizați fișierul web.xml.

Rulați aplicația Hello.



2. Aplicație simplă JSF (Hello1)

Aplicația Hello1 din directorul "DAI_Lab_04\javaee-tutorial-examples.zip\examples_6\web\hello1\" este o aplicație simplă, care conține un backing bean, două pagini Web și o resursă imagine.

Identificați aceste componente.

Deschideți fișierul de plecare (index.xhtml).

```

1 |<?xml version='1.0' encoding='UTF-8' ?>
2 |<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 |  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 |<html xmlns="http://www.w3.org/1999/xhtml"
5 |      xmlns:h="http://java.sun.com/jsf/html">
6 |  <h:head>
7 |    <title>Facelets Hello Greeting</title>
8 |  </h:head>
9 |  <h:body>
10 |    <h:form>
11 |      <h:graphicImage url="duke.waving.gif"/>
12 |      <h2>Hello, my name is Duke. What's yours?</h2>
13 |      <h:inputText id="username"
14 |                  value="#{hello.name}"
15 |                  required="true"
16 |                  requiredMessage="A name is required."
17 |                  maxLength="25">
18 |    </h:inputText>
19 |    <p></p>
20 |    <h:commandButton id="submit" value="Submit" action="response">
21 |    </h:commandButton>
22 |    <h:commandButton id="reset" value="Reset" type="reset">
23 |    </h:commandButton>
24 |  </h:form>
25 | </h:body>
26 |</html>

```

Observați că introducerea datelor se face în câmpul "username", care este obligatoriu și cu o lungime limitată superior.

Rețineți valoarea atributului "action" din butonul de submit.

Deschideți fișierul "response.xhtml".

```

1 |<?xml version='1.0' encoding='UTF-8' ?>
2 |<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 |  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 |<html xmlns="http://www.w3.org/1999/xhtml"
5 |      xmlns:h="http://java.sun.com/jsf/html">
6 |  <h:head>
7 |    <title>Facelets Hello Response</title>
8 |  </h:head>
9 |  <h:body>
10 |    <h:form>
11 |      <h:graphicImage url="duke.waving.gif"/>
12 |      <h2>Hello, #{hello.name}!</h2>
13 |      <p></p>
14 |      <h:commandButton id="back" value="Back" action="index" />
15 |    </h:form>
16 |  </h:body>
17 |</html>
18 |

```

Rețineți valoarea atributului "action" din butonul de revenire.

Deschideți fișierul hello1.java.

```

10 package hello1;
11
12 import javax.faces.bean.ManagedBean;
13 import javax.faces.bean.RequestScoped;
14
15 @ManagedBean
16 @RequestScoped
17 public class Hello {
18     private String name;
19
20     public Hello() {
21     }
22
23     public String getName() {
24         return name;
25     }
26
27     public void setName(String user_name) {
28         this.name = user_name;
29     }
30 }

```

Deschideți fișierul web.xml.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.0"
3     xmlns="http://java.sun.com/xml/ns/javaee"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
6     http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
7     <context-param>
8         <param-name>javax.faces.PROJECT_STAGE</param-name>
9         <param-value>Development</param-value>
10    </context-param>
11    <servlet>
12        <servlet-name>Faces Servlet</servlet-name>
13        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
14        <load-on-startup>1</load-on-startup>
15    </servlet>
16    <servlet-mapping>
17        <servlet-name>Faces Servlet</servlet-name>
18        <url-pattern>/faces/*</url-pattern>
19    </servlet-mapping>
20    <session-config>
21        <session-timeout>
22            30
23        </session-timeout>
24    </session-config>
25    <welcome-file-list>
26        <welcome-file>faces/index.xhtml</welcome-file>
27    </welcome-file-list>
28 </web-app>

```

Rulați aplicația.

Modificați valoarea atributului "action" și observați efectul.

3. Aplicație simplă JSF (GuessNumber)

Aplicația GuessNumber din directorul "DAI_Lab_04\javaee-tutorial-examples.zip\examples 6\web\guessnumber\" este o aplicație simplă, care conține un backing bean, două pagini Web și mai multe resurse.

Identificați aceste componente.

Deschideți fișierul de plecare (greeting.xhtml) și identificați bibliotecile de tag-uri utilizate.

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

Identificați unde este marcat faptul că fișierul de plecare nu mai este cel implicit (index.xhtml).

Rulați aplicația.

Modificați aplicația astfel încât numărul să fie resetat automat după ce a fost ghicit.

Modificați aplicația astfel încât numărul să fie resetat la apăsarea unui nou buton.

Modificați mesajul care apare la validare.

Modificați aplicația astfel încât la o încercare nereușită să se ofere un hint ("Dați un număr mai mare").

Modificați aplicația astfel încât să se afișeze numărul total de încercări reușite și nereușite.

Resurse utile:

- <http://download.oracle.com/javaee/6/javaserverfaces/2.0/docs/pdldocs/facelets/h/tld-summary.html>
- <http://www.coreservlets.com/JSF-Tutorial/jsf1/>
- <http://www.coreservlets.com/JSF-Tutorial/jsf2/>
- <https://javaetutorial.dev.java.net/servlets/ProjectDocumentList>
- <http://download.oracle.com/javaee/6/javaserverfaces/2.0/docs/pdldocs/facelets/>
- <http://netbeans.org/kb/docs/web/jsf20-intro.html>