JSF aplikace

- Součástí webové aplikace – samostatné nebo jako modul Enterprise app.
- Projekt musí obsahovat JSF knihovny
- Nastavení se provádí v konfiguračních souborech
  - '*web.xml*', '*sun-web.xml*' (závislé na aplikačním serveru)
  - '*faces-config.xml*'
  - uložené ve Web/WEB-INF
  - v NetBeans přístupné také v Configuration Files

Konfigurace '*web.xml*'

- Nastavení webové aplikace (obdoba '*ejb-jar.xml*')

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
    <context-param>
        <param-name>com.sun.faces.verifyObjects</param-name>
        <param-value>false</param-value>
    </context-param>
    <context-param>
        <param-name>com.sun.faces.validateXml</param-name>
        <param-value>true</param-value>
    </context-param>
    <context-param>
        <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
        <param-value>client</param-value>
    </context-param>
    <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>/faces/*</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>faces/welcomeJSF.jsp</welcome-file>
        </welcome-file-list>
    </web-app>
```

při deploy na server jsou vytvářeny objekty (komponenty, validátory, renderery apod.)

při deploy na server jsou xml soubory validovány

ukládání stavu client | server

kladné číslo – pořadí v jakém se má servlet loadovat

JSF servlet bude zpracovávat stránky s URL '/faces/' - tzv. prefixové mapování, nebo s nějakou příponou ***'*.jsf'*** nebo ***'.faces'*** apod.

Doba platnosti session v minutách (0 nebo záporné hodnoty – platnost nekončí)

Úvodní stránka

Konfigurace '*faces-config.xml*'

```
<faces-config version="1.2"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd">


 <navigation-rule>
        <from-view-id>/students.jsp</from-view-id>
        <navigation-case>
            <from-outcome>edit-student</from-outcome>
            <to-view-id>/student-edit.jsp</to-view-id>
            <redirect />
        </navigation-case>
        <navigation-case>
            <from-outcome>edit-teacher</from-outcome>
            <to-view-id>/teacher-edit.jsp</to-view-id>
            <redirect />
        </navigation-case>
    </navigation-rule>
</navigation-rule>

</faces-config>
```

Navigační string (návratová hodnota z metody definované v action)

Při přesměrování změnit URL

Edit-teacher není v aplikaci využit, jedná se o příklad řetězení navigation-case

Tvorba aplikace - přehled

1) Struktura aplikace, model, business metody využívající model
2) Webový projekt, backing beans – třídy spolupracující s uživatelským rozhraním (zpracovávají vstupy a výstupy)
3) Registrace tříd do konfiguračního souboru
4) Zobrazení – JSP stránky, používající JSF komponety
5) Vytvoření navigace
6) Validace
7) Internacionalizace

Základ aplikace

- Založit nový enterprise projekt
  - File – New Project – kategorie Java EE – Enterprise Application 'x*33eja-jsf*' pouze s modulem WEB
- EJB modul
  - otevřít EJB projekt z předchozího cvičení
  - k enterprise projektu přidat daný EJB modul:
    - kontextové menu (KM) – na Java EE Modules enterprise projektu – Add Java EE Module
  - zkontrolovat připojení do databáze
    - Persistence Unit resp. *'sun-resources.xml'* (ve složce Server Resources)
    - Username: jpa, password: test, database: jpa_example

Webový modul – konfigurace

- Přidat JavaServer Faces framework (ve vlastnostech webového projektu v kategorii Frameworks)
  - ponechat výchozí Servlet URL Pattern např. *'/faces/*'*
  - ponechat zaškrtnuté Validate XML
  - záložka Libraries – nastavit JSF 1.2
- Projít konfigurační soubory (Slide 2, 3), poté nastavit

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

- Přejmenovat *'welcomeJSF.jsp'* na *'students.jsp'*
- Do *'index.jsp'* přidat přesměrování <jsp:forward page="students**.jsf**" />

Backing bean #1

- Vytvořit klasickou třídu *'StudentsBack.java'*
- Registrace ve *'faces-config.xml'* pod jménem *'students'*:
  - zobrazit XML, *KM* – Java Server Faces – Add Managed Bean

```
<managed-bean>
   <managed-bean-name>students</managed-bean-name>
   <managed-bean-class>cz.cvut.x33eja.back.StudentsBack</managed-bean-class>
   <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
```

- Totéž lze File – New File – Categories – JSF – JSF Managed Bean
- Přidat EJB projekt jako knihovnu webového modulu

Backing bean #2

- *KM* – Insert Code – Call Enterprise Bean – vybrat z EJB projektu StudentSessionBean

```
@EJB
StudentSessionLocal ssl;

public List<Student> getAllStudents() {
   return ssl.getAllStudents();
}
```

- Ke každé datové položce třídy vygenerovat getter a setter (má-li být možnost nastavovat hodnoty – pomocí Insert Code nebo Refactor – Encapsulate Fields) př. Student student = null; …

- Přístup k jiné backing bean (v příkladu nepoužito)

```
StudentsBack sb = (StudentsBack)FacesContext.getCurrentInstance()
                  .getApplication().GetELResolver()
                  .getValue(context.getELContext(), null, "students");
```

Zobrazení dat #1

- *Zobrazení dat v 'students.jsp'*
  - z palety JSF (není-li zobrazena – Window - Palette) komponentu JSF Data Table
    - Zvolit 'Table Generated from Entity Class' – Browse, zadat *'Student'* a zvolit odpovídající entitu z balíčku *'x33eja.model.Student'*, použít registrovanou backing bean *'students'* – automaticky se vygeneruje tabulka
  - <h:dataTable>
    - atribut *var* s hodnotou *'item'* se odkazuje na jednotlivé property dané entity
    - Do atribut *value* vložit *'#{**students**.allStudents}'* (pokud je backing bean správně zaregistrovaná objeví se v Code completion po napsání #{ })
    - U sloupce Supervisor zobrazovat jeho jméno a příjmení (nikoli výstup toString)

Zobrazení dat #2

- Přidat nad tabulku tlačítků *'New Student'* a do tabulky sloupec s tlačítkem *'Edit'*
- V backing bean vytvořit metody 'new*Student'* a '*editStudent'* s návratovým typem String

```
public String newStudent() {
   student = new Student();
   return "edit-student";
}
```

- Vytvořit stránku *'student.jsp'* pro úpravu údajů studenta,
  - komponenty <h:form>, <h:panelGrid>, <h:outputText>, <h:inputText required=”true”>, <h:commandButton>
- **<h:messages />** - doporučuji pro ladění umístit na každé stránce, kde se upravují data pro zobrazení možných chyb
- Umožnit navigaci zpět na seznam studentů (h:commandLink)

Navigace

- *'faces-config.xml'*:
  - Tvorba pravidel – vizuálně nebo v XML
  - KM - JSF – Add Navigation Rule: from view ID: *'/*'*
  - KM - JSF – Add Navigation Case: *'edit-student'*
  - Po vytvoření pravidel provést deploy – navigace bude fungovat, nicméně zvolený detail se nebude zobrazovat – chybí provázání na konkrétní záznam

```
<navigation-rule>
   <from-view-id>/*</from-view-id>
   <navigation-case>
      <from-outcome>edit-student</from-outcome>
      <to-view-id>/student.jsp</to-view-id>
      <redirect />
   </navigation-case>
   <navigation-case>
      <from-outcome>list-students</from-outcome>
      <to-view-id>/students.jsp</to-view-id>
      <redirect />
   </navigation-case>
</navigation-rule>
```

Navigace je platná pro všechny stránky

Návratová hodnota z metody definované v action

Při přesměrování změnit URL

DataModel

- Vytvořit DataModel a nastavit jej jako value do students.jsp (místo listu)

```
public DataModel getAllStudentsModel() {
   allStudentsModel = new ListDataModel(getAllStudents());
   return allStudentsModel;
}
```

- Tlačítko edit (h:commandButton) volá akci:

```
public String editStudent() {
    student = (Student) allStudentsModel.getRowData();
    return "edit-student";
 }
```

- Pokud bychom nechtěli používat DataModel, je možné místo h:commandButton použít h:commandLink s vloženým parametrem pomocí: <f:param name=”id” value=”111” />. Toto řešení pro semestrální práci nedoporučujeme.

Komponenta selectOneListbox

- Přidání supervizora

Backing Bean:

```
public List<SelectItem> getAllTeachersSelectList() {
   List<SelectItem> items = new ArrayList<SelectItem>();
   for (Teacher t : ssl.getAllTeachers()) {
      items.add(new SelectItem(t, t.getSurname() + " " + t.getFirstName()));
   }
   return items;
}
```

JSP:

```
<h:selectOneListbox value="#{students.student.hasSupervisor}" size="1">
   <f:selectItems value="#{students.allTeachersSelectList}" />
</h:selectOneListbox>
```

JSF - konvertor

- ***Conversion Error setting value 'x33eja.model.Teacher@475b7d' for 'null Converter'***
  – nepovedla se automatická konverze (String, Integer apod.)

```
public class TeacherConverter implements Converter {
   public Object getAsObject(FacesContext context, UIComponent component, String value){
      return lookupStudentSessionBean().getTeacherWithId(value);
   }

   public String getAsString(FacesContext context, UIComponent component, Object value){
      return String.valueOf(((Teacher)value).getBirthNumber());
   }
}
```

- Registrace konvertoru ve faces-config.xml

```
<converter>
   <converter-id>TeacherConverter</converter-id>
   <converter-class>cz.cvut.x33eja.util.TeacherConverter</converter-class>
</converter>
```

- JSF

```
<h:selectOneListbox value="#{students.student.hasSupervisor}" size="1">
   <f:selectItems value="#{students.allTeachersSelectList}" />
   <f:converter converterId="TeacherConverter" />
</h:selectOneListbox>
```

JSF - validace

- Atribut required – nenulová hodnota vstupu
- f:validateLength
- Definice vlastního validátoru:

Java:

```
public class BirthdateValidator implements Validator {
   public void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException
{
   throw new ValidatorException(new FacesMessage(FacesMessage.SEVERITY_ERROR, "Too long number", null));
      }
   }
}
```

faces-config.xml:

```
  <validator>
     <validator-id>BirthdateValidator</validator-id>
     <validator-class>cz.cvut.x33eja.util.BirthdateValidator</validator-class>
  </validator>
```

JSF:

```
  <h:inputText id="birth" value="#{students.student.birthNumber}" required="true"
               disabled="#{!(students.student.birthNumber==0)}">
     <f:validator validatorId="BirthdateValidator"/>
  </h:inputText>
  <h:message for="birth" />
```

JSF - lokalizace

- File – New – category Other – Properties File – Translation
- KM – Add Locale cs_CZ

students=Students
student_add=Add Student
student_edit=Edit

- faces-config.xml

```
<application>
   <locale-config>
      <default-locale>en</default-locale>
      <supported-locale>cs</supported-locale>
   </locale-config>
   <resource-bundle>
      <base-name>translation</base-name>
      <var>msgs</var>
   </resource-bundle>
 </application>
```

- JSP

Použití: <h:outputText value="#{**msgs**.students}"/>

Nebo: <f:loadBundle basename="translation" var="msgs" /> - nefunguje doplňování kódu

Přehled možných výjimek #1

- *'org.apache.jasper.JasperException: java.lang.RuntimeException: Cannot find FacesContext'*
  - ve web.xml – servlet mapping – Faces Servlet zkontrolovat url-pattern:
    - Pro: '**/faces/***' nutné jej také zadat do URL např. *'aplikace-war/welcomeJSF.jsp'* nahradit: *'aplikace-war/faces/welcomeJSF.jsp'*
    - Pro ***'*.jsf'*** místo *'aplikace-war/welcomeJSF.jsp'* nastavit *'aplikace-war/welcomeJSF.jsf'*
- *'j_id_id47: Validation Error: Value is not valid'*
  - h:selectOneListbox (ID konkrétní komponenty lze zjistit např. pomocí pluginu FireBug kliknutím na daný element stránky)
  - Chyba se zobrazí, je-li na stránce komponenta h:messages
  - Pro danou entitu implementovat equals viz entita Person
    (KM v dané entitě - Insert Code – Equals … a zvolit např. primární klíč)

Přehled možných výjimek #2

- *'java.io.NotSerializableException: x33eja.model. ...'*
  - implementovat pro entity rozhraní Serializable