

JAVA SERVER FACES TABLES AND MESSAGES LESSONS

SADRZAJ – 8

JSF Korisnicke poruke

JSF Table

JSF Korisnicke poruke

- ▶ **kreira se i učitava globalna .properties datoteka**

```
<application>  
<message-bundle>...</message-bundle>  
</application>
```

- ▶ **koriste se eksplicitna imena property-ija u .properties datoteci**

```
javax.faces.component.UIInput.CONVERSION  
javax.faces.component.UIInput.REQUIRED
```

- ▶ **primjer**

```
javax.faces.component.UIInput.CONVERSION=  
Pogresan format!  
javax.faces.component.UIInput.REQUIRED=  
Unesite vrijednost!!!
```

▶ upotreba

- kada se na serverskoj strani kreira rezultat sa nepoznatim brojem elemenata

▶ alternative

- bez petlji

```
<h:outputText
  value="#{bankCustomer.depositTable}"/>
<mytags:showMytTable custID="..." month="..."
  styleClasses="..."/>
```

- sa petljama

```
<% for(...) { ... %>
```

HTML kod

```
<% } %>
```

JSTL

▶ h:dataTable

- definicija za jednu vrstu – JSF je ponavlja onoliko puta koliko je potrebno
- value: kolekcija podataka (obično lista bean-ova) – dozvoljeni tipovi kolekcija:
 - Array
 - List (e.g., ArrayList, LinkedList)
 - ResultSet (moraju biti scroll-insensitive)
 - Result (wrapped ResultSet from JSTL)
 - DataModel (in javax.faces.model)
- var: vezan za svaki ulaz kolekcije
 - ovaj ulaz treba da bude nešto što JSF EL može da prikaže
 - Bean, array, List, Map
- drugi atributi
 - standardni TABLE atributi
 - border, bgcolor, width, cellpadding, cellspacing, frame, ...
 - style sheet
 - rowClasses, headerClass, footerClass

▶ h:column

- obično okružuju h:outputText elemente

```
<h:column>
  <h:outputText value="#{rowVar.colData}"/>
</h:column>
```

- mogu okruživati druge h:Xxxx elemente

- h:inputText, ili bilo koje druge elemente

▶ regularni HTML sadržaj mora biti u okviru f:verbatim

```
<h:column>
<f:verbatim>First Name: </f:verbatim>
<h:outputText value="#{rowVar.firstName}"/>
</h:column>
```

▶ header i footer tabele se specificira pomoću f:facet

▶ header i footer tabele

▶ header

- koristi se f:facet sa name="header"
- vrijednost može da bude f:verbatim ili h:outputText
- i dalje je potreban h:outputText za ne-heading vrijednosti

▶ footer

- koristi se f:facet sa name="footer"
- vrijednost može da bude f:verbatim ili h:outputText

▶ primjer

```
<h:column>
<f:facet name="header">
<f:verbatim>...</f:verbatim>
</f:facet>
<h:outputText value="#{rowVar.colVal}"/>
</h:column>
```


- ▶ CSS
- ▶ eksplicitno formatiranje u JSP
 - zahtjeva f:verbatim prije i poslije svakog entry-ja
- ▶ formatiranje ugrađeno u bean
 - nije dostupno Web dizajneru
 - nekonzistentnost sa style sheet-ovima
 - bean-ovi su dio modela, ne view-a
- ▶ koristiti rowClasses, headerClass, footerClass
 - rowClasses: zarezima razdvojena lista CSS stilova – primjenjuje se na svaku vrstu do kraja liste, a onda se ponavlja
 - headerClass: CSS stil za header tabele
 - footerClass: CSS stil za footer

- ▶ prikaz sadržaja iz baze podataka
- ▶ uzeti podatke iz result set-a i smjestiti ih u niz ili List-u
 - sporo
 - zahtjeva bean koji će reprezentovati podatke koji čine vrstu tabele
- ▶ moguće je koristiti ResultSet kao h:dataTable value
 - podržano je, prema specifikaciji
- ▶ moguće je koristiti i JSTL ResultSupport klasu za pretvaranje ResultSet-a u Result
 - samo jedna dodatna linija koda
 - robusno i pouzdano
 - konekciju je moguće zatvoriti ili vratiti u pool konekcija, prije nego što se počne koristiti Result