JAVA SERVER FACES
EVENT HANDLING
LESSONS

CONTENT-6

JSF Event Handling
JSF Action Listener
JSF Value Change Listener
JSF H Library
JSF H Form
JSF H Input Text
JSF H Input Secret
JSF H Command Button
JSF H Command Link
JSF Value Change Listener Atrribute
JSF Combo Boxes and List Boxes
JSF H Output Text and H Output Format

- dvije vrste događaja koje generiše korisnik:
  - događaji koji startuju back-end procesiranje
  - događaji koji utiču samo na format korisničkog interfejsa
- JSF kod koji upravlja ovim akcijama dijeli na:
  - action kontroler-e
  - event listener-e
- action kontroleri obrađuju submit forme
  - okidaju se poslije popunjavanja bean-ova
  - okidaju se poslije validacione logike
  - vraćaju stringove koji utiču na dalju navigaciju
- event listener-i obrađuju UI događaje:
  - obično se okidaju prije popunjavanja bean-ova
  - obično izbjegavaju validacionu logiku
  - nikad direktno ne utiču na navigaciju

- event handler-i su jedna prepoznatljiva karakteristika JSF-a
- Ajax je često mnogo efikasniji
  - u mnogim situacijama samo mali broj elemenata se mijenja – u tom slučaju ajax će obezbijediti veći stepen interakcije

- tipovi event listener-a
- ActionListener
  - poziva se pomoću submit dugmeta, slika, i linkova sa odgovarajućim JavaScript kodom

```
<h:commandButton value="..." .../>
<h:commandButton image="..." .../>
<h:commandLink .../>
```

  - automatski submit-uju formu
- ValueChangeListener
  - poziva se pomoću combo box-ova, checkbox-ova, radio dugmadi, tekst polja, ...

```
<h:selectOneMenu .../>
<h:selectBooleanCheckbox.../>
<h:selectOneRadio .../>
<h:inputText .../>
```

  - ne submit-uju formu automatski

- korišćenje ActionListener-a
- neka dugmad samo submit-uju formu i pokreću backend procese
  - koristi se <h:commandButton action="..." ...>
- druga dugmad utiče samo na UI
  - koristi se <h:commandButton actionListener="..." .../>
  - obično se želi da se ovaj proces izvršava prije nego što se popune bean-ovi i posebno prije validacije
  - zato se koristi atribut "immediate" da bi se naznačilo izvršavanje prije navedenih operacija
  - <h:commandButton actionListener="..." immediate="true" .../>

- listeneri su obično u form bean klasi
  - mogu biti i u odvojenoj klasi ako se koristi FacesContext da bi se dobio request ili session objekat i potražio form bean eksplicitno
- ActionEvent kao argument metode
  - ove metode ne vraćaju rezultat (nije String kao kod action kontrolera)
  - ActionEvent nalazi se u javax.faces.event paketu

```
public void someMethod(ActionEvent event) {
  doSomeSideEffects();
}
```

JSF Value Change Listener

- ValueChangeListener
  - nije vezan za dugme
  - veže se za combobox, listbox, radio button, checkbox, textfield, itd.
  - forme se ne submit-uju automatski
  - potrebno je dodati JavaScript za automatski submit onclick="submit()" ili onchange="submit()“

- obično su u form bean klasi
  - mogu biti i u odvojenoj klasi ako se koristi FacesContext da bi se dobio request ili session objekat i potražio form bean eksplicitno
- uzimaju ValueChangeEvent kao argument
  - korisne ValueChangeEvent metode
    - getComponent
    - getOldValue – prethodna vrijednost GUI elementa
    - getNewValue – trenutna vrijednost GUI elementa

```
public void someMethod(ValueChangeEvent event) {
  boolean flag =
((Boolean)event.getNewValue()).booleanValue();
  takeActionBasedOn(flag);
}
```

- h:form
  - ne specificira se ACTION atribut
  - mora se koristiti POST metoda
- h:inputText
  - NAME se generiše automatski
  - VALUE je u formi #{beanName.propertyName}
- h:inputSecret
  - NAME se generiše automatski
  - VALUE se primjenjuje samo na izlaz, ne na ulaz
- h:commandButton
  - ACTION odgovara atributu ACTION taga forme
- h:outputText
  - prikazuju se property-iji bean-a

- odgovarajući HTML element
  - <FORM …>
- mogućnosti
  - ne specificira se akcija
- atribut
  - enctype – tip kodiranja
    - application/x-www-form-urlencoded (default)
    - multipart/form-data
    - text/plain
  - target – frame dio u kojem se prikazuje rezultat
  - onsubmit. JavaScript kod koji se izvršava prije slanja forme
    - može da se izvrši validacija polja na klijentskoj strani

h:inputText

- odgovarajući HTML element
  - <INPUT TYPE="TEXT" …>
- atributi
  - value – odgovarajuća vrijednost bean-a
    - value="#{beanName.propertyName}"
    - koristi se i za inicijalnu vrijednost i pri slanju serverskoj komponenti
  - valueChangeListener – definiše koji se metod izvršava kada se šalje forma i kada je prethodna vrijednost promijenjena
  - onchange – JavaScript koji se izvršava pri promjeni vrijednosti
  - immediate – koristi se da označi da listener zaobilazi validaciju
  - required – definiše da li korisnik mora da unese vrijednost
    - ako je ne unese, forma se ponovo prikazuje i prikazuje se poruka o grešci
  - validator – metod koji izvršava validaciju

h:inputSecret

- odgovarajući HTML element
  - <INPUT TYPE="PASSWORD" …>
- atributi
  - isti kao za h:inputText, s tim što se value atribut koristi samo za submit

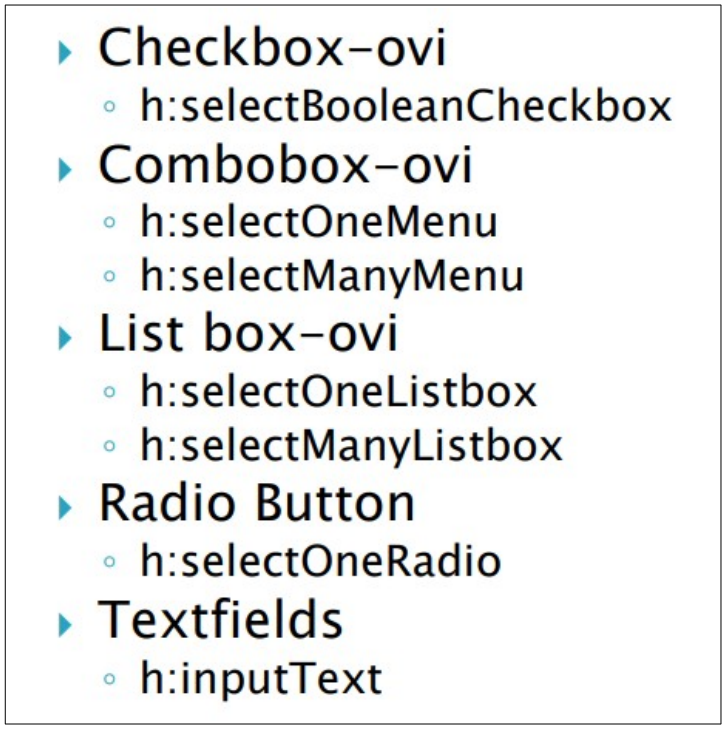
h:commandButton

- odgovarajući HTML element
  - <INPUT TYPE= "SUBMIT" …>
- atributi
  - value – natpis na dugmetu – često je statički string, ali se može i dinamički mijenjati
  - action – metod action controller-a koji se izvršava kada se šalje forma – izvršava se kada svi listener-i završe svoj posao
  - actionListener – metod listener-a koji se izvršava kada se šalje forma – izvršava se prije action controller-a

h:commandLink

- odgovarajući HTML element
  - <A HREF="..."> sa povezanim JavaScript kodom koji šalje formu kada se link aktivira
  - JavaScript se ubacuje automatski pomoću JSF
- atributi
  - value – tekst linka – često je statički string, ali se može i dinamički mijenjati
  - action – metoda action controller-a koja se izvršava kada se šalje forma – izvršava se kada svi listener-i završe svoj posao
  - actionListener – metod listener-a koji se izvršava kada se šalje forma – izvršava se prije action controller-a

JSF Elementi sa Value Change Listener – ом / атрибутом



- navođenjem opcije po opcije

```
<h:selectOneMenu ...>
<f:selectItem itemValue="..." itemLabel="..."/>
<f:selectItem itemValue="..." itemLabel="..."/>
<f:selectItem itemValue="..." itemLabel="..."/>
</h:selectOneMenu>
```

- navođenjem svih opcija u jednom redu

```
<h:selectOneMenu ...>
  <f:selectItems value="..."/>
</h:selectOneMenu>
```

- value atribut h:selectOneMenu određuje selected vrijednost

- vrijednost za f:selectItems mora biti List ili niz elemenata tipa java.faces.model.SelectItem
- SelectItem ima dva osnovna konstruktora
  - jedan samo specificira String
    - vrijednost koja se prikazuje i vrijednost koja se šalje setter metodi bean-a su isti
  - drugi specificira Java objekat i String
    - string se prikazuje, a Java objekat se šalje kao vrijednost setter metodi bean-a (mapiranje se izvršava automatski pomoću JSF-a)

```
SelectItem[] listboxItems =
{
  new SelectItem(realValue1, "Choice 1"),
  new SelectItem(realValue2, "Choice 2"),
...
```

- h:outputText
  - prikazuje se bean property ili element kolekcije
    - vrijednosti iz properties fajla se upisuju u kolekciju pomoću f:loadBundle
  - Atributi
    - styleClass ili style: rezultat je SPAN element
    - escape: vrijednost false označava da se ne koristi filtera za < i >
- h:outputFormat
  - prikazuje poruke o grešci sa parametrima
  - parametri se prosljeđuju sa f:param