Java Server Faces
Validations
Lessons

SADRZAJ-7

- dva zadatka koja skoro svaka Web aplikacija mora da izvrši:
  - provjera da su sva polja forme popunjena i da su unesene vrijednosti u odgovarajućem formatu
  - ponovno prikazivanje forme u slučaju kada nisu unesene sve vrijednosti ili kada je unos pogrešan, zajedno sa porukama koje ukazuju na greške i sa vrijednostima koje imaju pravilan format

- ručna validacija
  - koriste se string property-iji za bean
  - sprovodi se validacija u setter metodama i/ili action controller-ima
  - vraća se null da bi se ponovo prikazala forma
  - kreiraju se odgovarjuće poruke o greškama
- implicitna automatska validacija
  - koriste se int, double, ... bean property-iji, ili se dodaje required
  - sistem ponovo prikazuje formu, ako postoji greška prilikom konverzije
  - koristi se h:message da se prikaže poruka za određeno polje
- eksplicitna validacija
  - koriste se f:convertNumber, f:convertDateTime, f:validateLength, f:validateDoubleRange, ili f:validateLongRange
  - sistem ponovo prikazuje forma u slučaju greške; opet h:message
- kreiranje sopstvenih metoda validacija

- setter metode konvertuju iz stringova
  - koriste se try/catch blokovi
  - koristi se logika specifična za aplikaciju
- Action controller provjerava vrijednosti
  - ako su vrijednosti u redu, vraća se uobičajeni izlaz
  - ako vrijednosti nedostaju ili su nelegalne, smješta se poruka o grešci u bean i vraća se null
- ulazna forma
  - prikazuje poruke o grešci
    - poruke su prazni stringovi po defaultu
    - u okviru h:outputText, treba koristiti escape="false" ako poruka sadrži HTML tagove

- alternativa kreiranju vlastitih poruka o grešci
  - kreirati FacesMessage
  - pohraniti poruku u globalnu listu poruka korišćenjem
    - facesContext.addMessage
  - vratiti null za ponovni prikaz početne forme
  - ispisati poruke o grešci pomoću
    - h:messages ili h:message
- prednosti:
  - nije potrebno čuvati poruke “na svoj način”
  - uklapa se u standardnu JSF validaciju
- nedostaci
  - manja kontrola nad formatom poruka o grešci

Implictitna automatska validacija

- definišu se bean property-iji na standardne tipove podataka
  - int, long, double, boolean, char, ...
- sistem pokušava da izvrši automatsku konverziju, kao kod jsp:setProperty
  - metode Integer.parseInt, Double.parseDouble...
- ako postoji greška, forma se ponovo prikazuje
  - i prikazuje se poruka o grešci
- može se dodati atribut required za svaki ulazni element, da bi se specificiralo da prazne vrijednosti predstavljaju grešku
  - koristi se h:message da bi se prikazale poruke o greškama
  - h:message vraća prazan string ako nema poruke
  - h:message prihvata styleClass za CSS ime stila
- dodaje se atribut immediate da bi se preskočila validacija
  - na primjer, za h:commandButton sa logout ili cancel operacijama

Eksplicitna automatska validacija

- definisati bean property-ije – prosti tipovi podataka
  - int, long, double, boolean, char, ...
- dodaje se f:validateBlah ili f:convertBlah elementima
- sistem provjerava da li polja odgovaraju pravilima
  - f:validateBlah atributi dozvoljavaju da vrši kontrola formata
- ako postoji greška prilikom validacije, forma se ponovo prikazuje
  - i smješta se poruka o grešci
- ostali pristupi su i dalje mogući
  - i dalje se može dodati atribut required za bilo koji ulazni element
  - i dalje se sa h:message prikazuju poruke
    - h:message vraća prazan string ako nema poruke
  - i dalje se dodaje atribut immediate da bi se izbjegla validacija

- i f:convertBlah i f:validateBlah provjeravaju format i prikazuju ponovo formu u slučaju greške
  - f:convertBlah mijenja i format u kome se polje prikazuje
  - f:validateBlah ima smisla samo sa h:inputText
  - f:convertBlah ima smisla koristiti i sa h:inputText i sa h:outputText
- primjer

```
<h:inputText value="#{itemBean.price}">
<f:convertNumber maxFractionDigits="2"/>
</h:inputText>
```

- prikazuje 0.75, a ne 0.749

Atributi konvertora i validatora

- f:validateLength
  - minimum
  - maximum
- f:validateLongRange
  - minimum
  - maximum
- f:validateDoubleRange
  - minimum
  - maximum

- f:convertNumber
  - currencyCode, currencySymbol
  - groupingUsed
  - integerOnly
  - locale
  - max(min)FractionDigits
  - max(min)IntegerDigits
  - pattern
  - type
    - number, currency, percentage
- f:convertDateTime
  - type
    - date, time, both
  - dateStyle, timeStyle
    - default, short, medium, long,full
  - pattern (ala SimpleDateFormat)
  - locale
  - timeZone

Korisnicki konvertori

- implementiraju Converter interfejs
  - javax.faces.convert.Converter
  - getAsObject
    - uzima String; vraća Object (konvertuje unos korisnika)
  - getAsString
    - uzima Object; vraća String
- greške pri konverziji
  - baca se ConverterException sa FacesMessage
- registracija u faces-config.xml
  - converter, converter-id, converter-class
- koristi se converter atribut

```
<h:inputText
value="#{...}" converter="someID"/>
```

Korisnicki validatori

- implementiraju Validator interfejs
- implementiranje validate metode
- greške pri konverziji
  - baca se ValidationException sa FacesMessage
- registracija u faces-config.xml
  - validator, validator-id, validator-class
- koristiti f:validator tag u JSF stranici

```
<h:inputText value="#{…}">
<f:validator validatorId="someID"/>
</h:inputText>
```