Java Server Faces (JSF)

Veerle Ongenae

Industrieel Ingenieur Informatica,
Hogeschool Gent

JSF

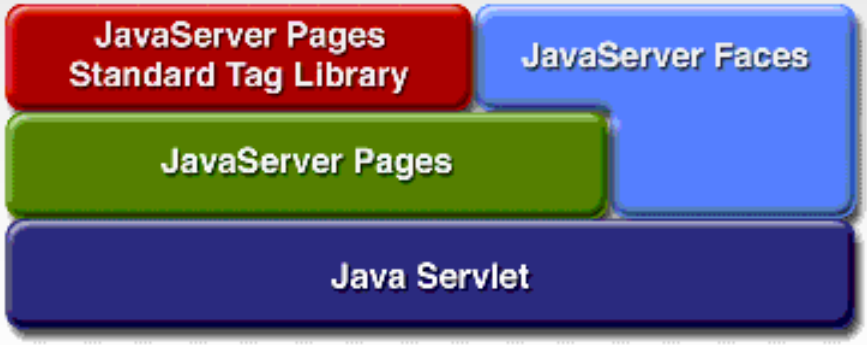
- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

JSF

- Technologie
  - Webapplicaties in Java
  - Componenten voor gebruikersinterface
  - Server side
- API
  - UI-componenten
- Tag-bibliotheken

JSF

- Verder uitwerken scheiding presentatie ↔ functionaliteit (logica)
- Meestal ingebed in JSP's, hoeft niet
  - Gebouwd op servlet-technologie

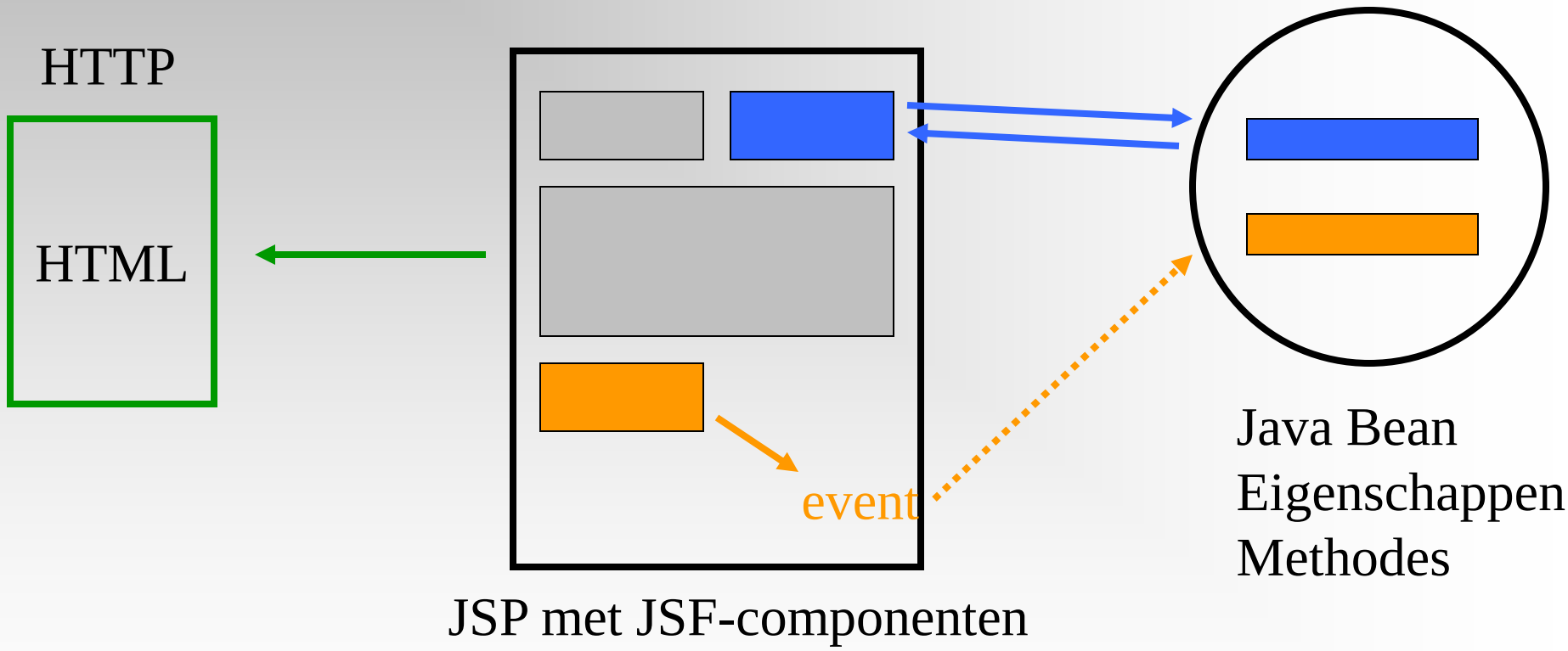


(java.sun.com)

Functionaliteit

- Opmaken UI met herbruikbare en uitbreidbare componenten
- Koppeling UI-componenten aan data (server)
- Bewaren van de status van UI-componenten tussen verschillende HTTP-aanvragen
- Koppeling events (client) aan functionaliteit (server)

Principe


HTTP
HTML
event
Java Bean
Eigenschappen
Methodes
JSP met JSF-componenten

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

Voorbeeld

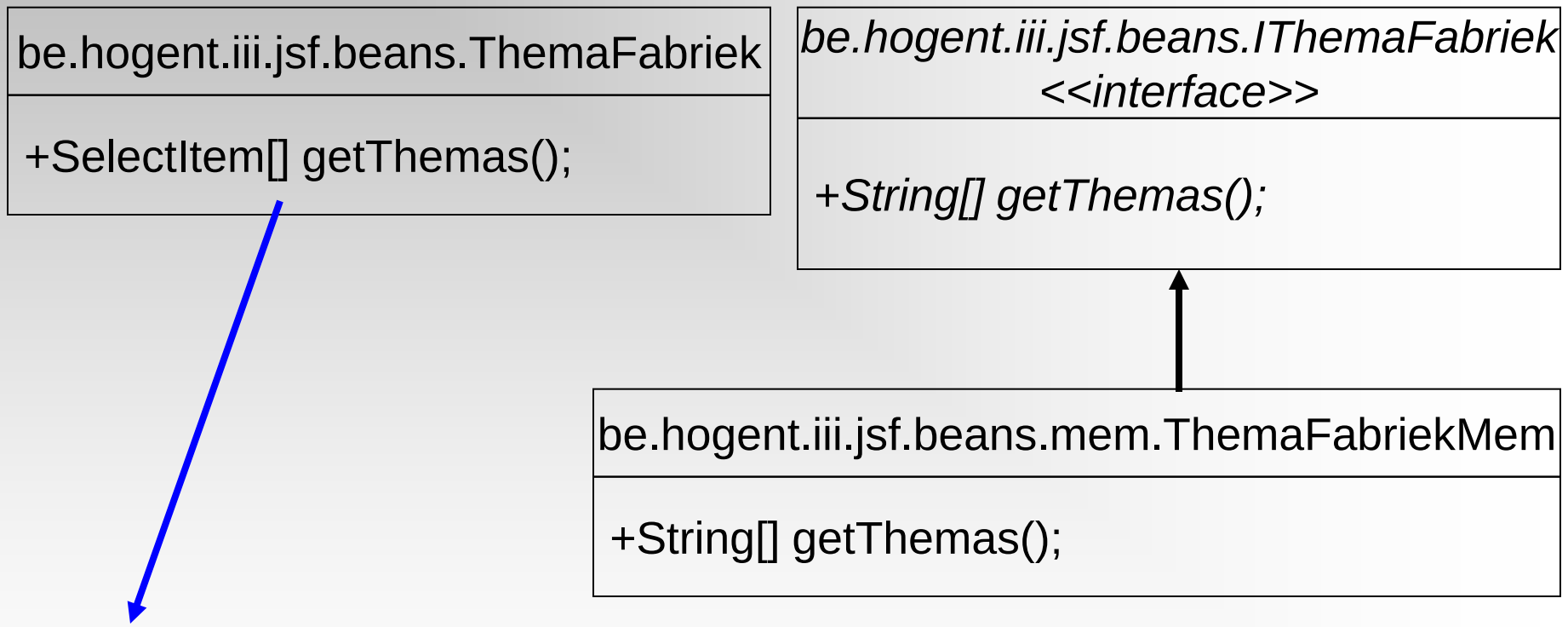

Keuze - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers
Maak hier uw keuze
Naam:
Thema: sport
Kies
Klaar
Keuze - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers
Maak hier uw keuze
Naam: ann
Thema: actualiteit
Kies
sport
actualiteit
muziek
Klaar
Keuze - Mozilla Fir
Bestand Bewerken Beeld
Maak hier uw keuze
Naam: els
Thema: muziek
Kies
Hallo els, u koos muziek.
Klaar

Voorbeeld

- http://localhost:8084/jsf/index.jsp
  - Doorsturen naar keuzeJSF.jsp

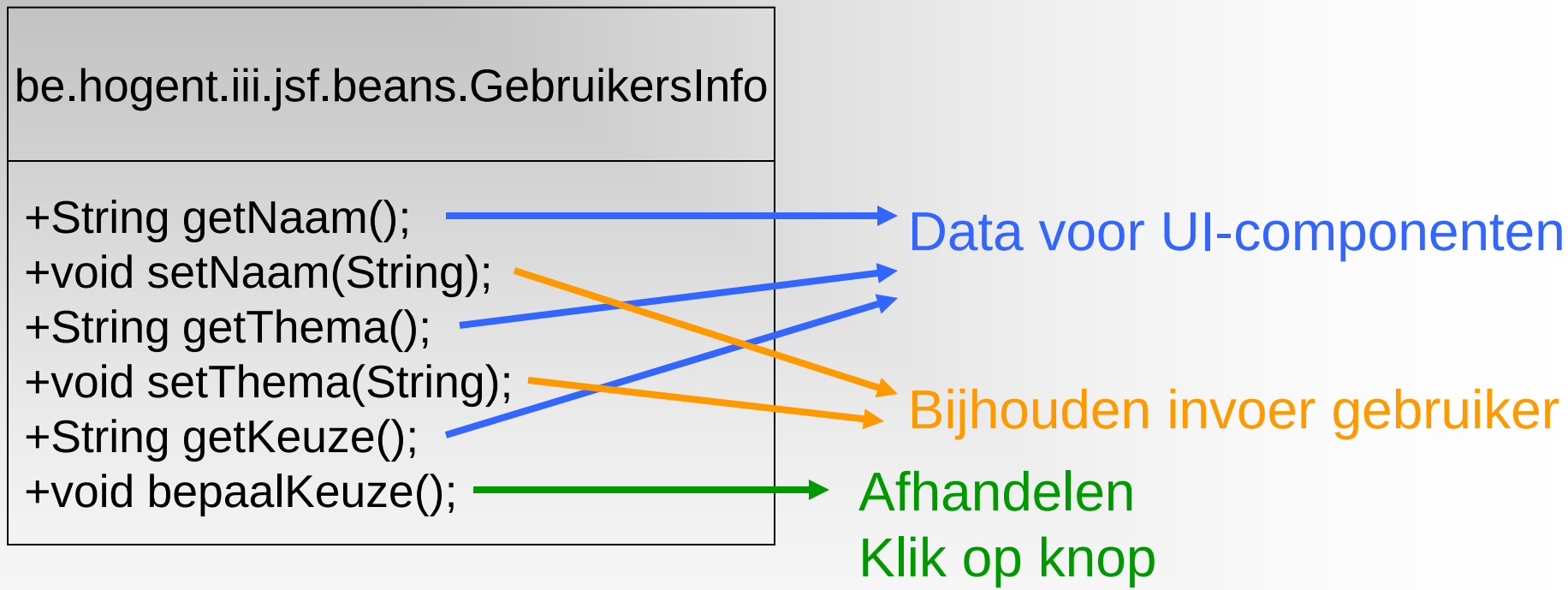
  <jsp:forward page="faces/keuzeJSF.jsp"/>
- http://localhost:8084/jsf/faces/keuzeJSF.jsp maakt gebruik van 2 beans
  - GebruikersInfo
  - ThemaFabriek

Gebruikte beans: ThemaFabriek



Thema's voor keuzelijst (drop down list)

Gebruikte beans: GebruikersInfo



keuzeJSF.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
     charset=UTF-8">
        <title>Keuze</title>
    </head>
    <body> … </body>
</html>
```

```
<body><f:view>
   <h1><h:outputText value="Maak hier uw keuze" /></h1>
     <h:form><table>
       <tr><td>Naam:</td>
            <td><h:inputText id="invoerNaam"
                      value="#{GebruikersInfo.naam}" /></td></tr>
       <tr><td>Thema:</td>
           <td><h:selectOneMenu id="invoerThema"
                      value="#{GebruikersInfo.thema}">
                    <f:selectItems value="#{ThemaFabriek.themas}"/>
                 </h:selectOneMenu>
           </td></tr></table>
       <h:commandButton value="Kies" action="keuzeJSF.jsp"
           actionListener="#{GebruikersInfo.bepaalKeuze}" />
       <br>
       <h:outputText value="#{GebruikersInfo.keuze}"/>
   </h:form></f:view></body>
```

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

JSP met JSF

- f:view
  - Omvat alle andere componenten
  - boomstructuur
- h:form
  - Data sturen naar server
  - Genereert attributen form-tag
  - Bevat de andere componenten
- h:outputText
  - Label

Gebruikte componenten

- h:inputText
  - Invoer tekst
- h:selectOneMenu
  - Eén selecteren uit lijst
  - f:selectItems
    - Items waaruit de lijst bestaat
    - value-atttibuut
      - Unified EL
        - Wijst naar eigenschap
        - List SelectItem
        - Array SelectItem

Gebruikte componenten

- h:commandButton
  - action
    - volgende pagina
    - String
      - Letterlijk
      - Resultaat methode (Unified EL)
  - actionListener
    - Methode die event afhandelt
    - Unified EL

value-attribuut

```
<h:outputText value="Maak hier uw keuze" />
<h:inputText id="invoerNaam"
    value="#{GebruikersInfo.naam}" />
<h:selectOneMenu id="invoerThema"
    value="#{GebruikersInfo.thema}"> …
```

- Tekst
  - Data voor component
- unified EL – uitdrukking
  - #{…}
  - Data voor component
  - Bijhouden invoer gebruiker
  - Eigenschap bean

Unified EL

- #{…}
- Gelijkaardig EL
  - value-binding
    - Eigenschappen beans

```
<h:inputText id="invoerNaam"  value="#{GebruikersInfo.naam}" />
```

  - method-binding
    - Methodes beans

```
<h:commandButton value="Kies" action="volgendeActie"
        actionListener="#{GebruikersInfo.bepaalKeuze}" />
```

  - Andere EL-elementen

Componenten

- Tags ↔ UI-componenten (server)
  - Genereren uitvoer
    - Eventueel ophalen data uit bean
  - Bij indienen pagina
    - Data opslaan in bean
    - Eventueel informatie opslaan in verborgen veld formulier

UI-componten

- Registreren
  - Event listeners
  - Validators
  - Convertors

Overzicht UI-componenten

- UIColumn: kolom in UIData
- UICommand: actie uitvoeren
- UIData: tonen groep data
- UIForm: groep componenten, ~HTML-form
- UIGraphic: figuur
- UIInput: invoer, afgeleid van UIOutput
- UIMessage: bericht tonen
- UIMessages: berichten tonen
- UIOutput: uitvoer op pagina
- UIPanel: zorgt voor layout kindcomponenten
- UIParameter: parameter
- UISelectBoolean: invoer logische waarde, afgeleid van UIInput
- UISelectItem: een item
- UISelectItems: meerdere items
- UISelectMany: invoer meerdere items uit keuze, afgeleid UIInput
- UISelectOne: invoer item uit keuze, afgeleid UIInput
- UIViewRoot: basis componentboom

Renderen UI-componenten

- UI-component bevat NIET hoe hij gerenderd moet worden
- Aparte renderer
- UICommand
  - 2 renderers
    - Link
    - Knop
  - 2 corresponderende tags
    - commandButton
    - commandLink

Overzicht tags

- column: kolom in tabel
- commandButton: submit-knop
- commandLink: link
- dataTable: tabel
- form: formulier
- graphicImage: figuur
- inputHidden: verborgen veld
- inputSecret: wachtwoordveld
- inputText: tekstveld
- inputTextarea: invoer meerdere lijnen
- message: bericht
- messages: berichten
- outputLabel: label
- outputLink: link (zonder actie)
- outputFormat: bericht
- outputText: één lijn tekst
- panelGrid: groepeert componenten in tabelvorm
- selectBoolean: checkbox
- selectItem: één item in lijst
- selectItems: meerdere items in lijst
- selectManyCheckbox: lijst checkboxen
- selectManyListbox: lijst, meerdere items selecteren mogelijk
- selectManyMenu: keuzelijst, meerdere items selecteren mogelijk
- selectOneListbox:lijst, één item selecteren
- selectOneMenu: keuzelijst, één item selecteren
- selectOneRadio: lijst radiobuttons

Managed Beans

- Java Beans beschikbaar voor JSF
- Eigenschappen
  - Data UI-componenten
  - Invoer Gebruiker
- Methodes
  - Afhandelen events
  - …
- Hoe gekend in JSP-pagina?
  - faces-config.xml
  - In WEB_INF/

Configuratie Managed Beans

```
<managed-bean>
      <description>
         Bevat de gegevens die de gebruiker ingaf.
      </description>
      <managed-bean-name>GebruikersInfo</managed-bean-name>
      <managed-bean-class>
          be.hogent.iii.jsf.beans.GebruikersInfo
       </managed-bean-class>
      <managed-bean-scope>session</managed-bean-scope>
   </managed-bean>
```

Naam, gebruikt in JSP

klassenaam

scope

Eigenschappen bean initialiseren

```
<managed-bean>
    <managed-bean-name>customer</managed-bean-name>
    <managed-bean-class>…</managed-bean-class>
    <managed-bean-scope> request </managed-bean-scope>
    <managed-property>
        <property-name>mailingAddress</property-name>
        <value>#{addressBean}</value>
    </managed-property>
</managed-bean>
<managed-bean>
    <managed-bean-name>addressBean</managed-bean-name>
    <managed-bean-class> … </managed-bean-class>
    <managed-bean-scope> none </managed-bean-scope>
</managed-bean>
```

Eigenschappen bean initialiseren

- In configuratie
  - <managed-property>
    - Naam
    - Waarde
      - Constante
      - Unified EL
        - Object in zelfde of ruimere scope
        - Nieuw aan te maken (scope = none)

Scope bean	Scope verwijzing
none	none
application	none, application
session	none, application, session
request	none, application, session, request

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

web.xml

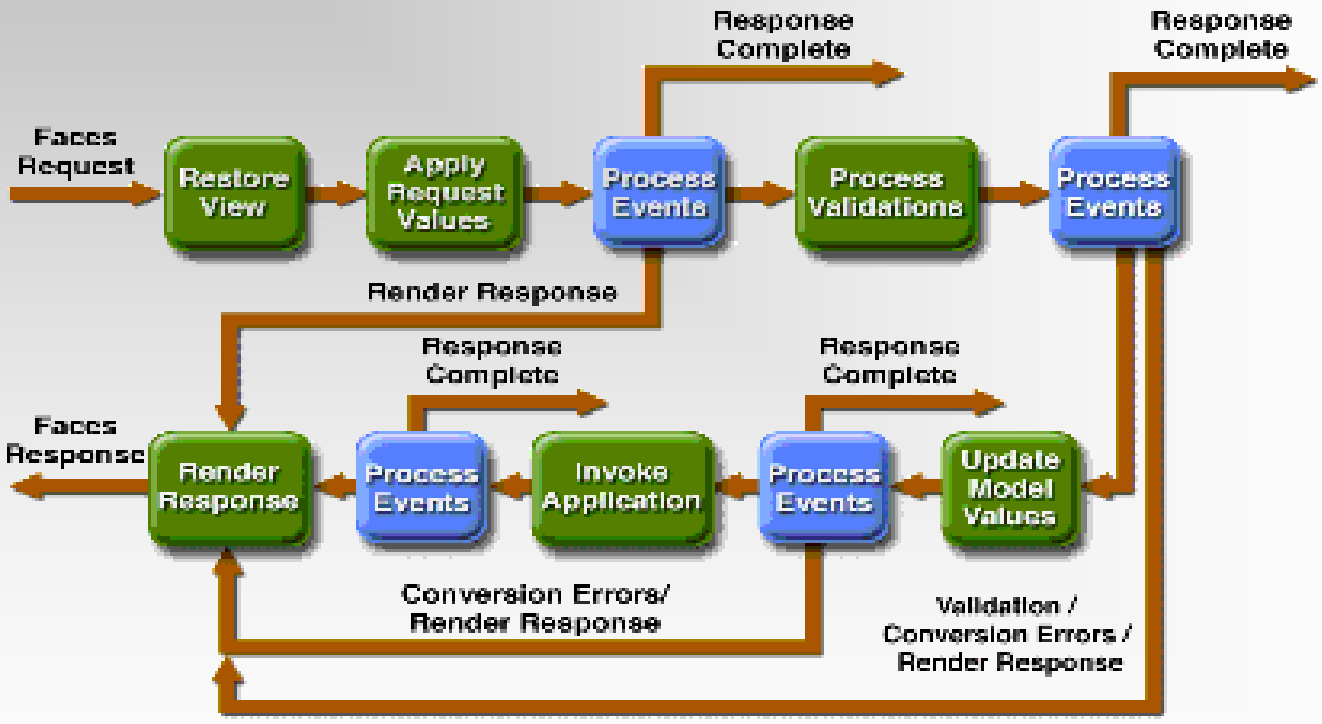
- Contextparameters
  - JSF-configuratie
  - Andere configuratie voor applicatie
    - Vb. implementatieklasse ThemaFabriek
- Alle aanvragen doorsturen naar één servlet (FacesServlet)

ThemaFabriek

- Constructor
  - Ophalen contextparameter
    - Via FacesContext
      - Informatie voor één aanvraag
    - Via ExternalContext
      - Toegang tot applicatieomgeving: sessie, ServletContext, aanvraag, antwoord, …

```
FacesContext fctx = FacesContext.getCurrentInstance();
ExternalContext ectx = fctx.getExternalContext();
String contextParam = ectx.getInitParameter("klasseThemaFabriek");
```

Levensloop JSF-pagina



(java.sun.com)

Postback ↔ eerste aanvraag

- Eerste aanvraag
  - Restore View
  - Render Response
- Postback
  - Formulier indienen naar zelfde url

Restore View

- Opbouwen view (indien postback)
  - Boomstructuur van componenten
    - Informatie ophalen (verborgen veld, server, …)
  - Binden van luisteraars, validators, … aan componenten
- Aanmaken nieuwe lege view
- Bewaren view in FacesContext
  - Informatie ivm één aanvraag

Apply Request Values

- HTTP-parameters → nieuwe waarde voor sommige componenten
  - Eventueel foutboodschap op FacesContext (bv. String → Integer mislukt)
- Events genereren
  - Doorsturen naar luisteraars
- Indien (methodes FacesContext)
  - renderResponse → Render Response fase
  - responseComplete → externe redirect

Process validations

- Validatie componenten
  - Indien gespecifieerd
- Validatie mislukt
  - Foutboodschap op FacesContext
  - Naar Render Response fase
- Events genereren
  - Doorsturen naar luisteraars
- Indien
  - renderResponse → Render Response fase
  - responseComplete → externe redirect

Update Model Values

- Data component (~value-attribuut) → data Java Beans
  - Mislukt? → Render Response faze
- Events genereren
  - Doorsturen naar luisteraars
- Indien
  - renderResponse → Render Response fase
  - responseComplete → externe redirect

Invoke Application

- Application level events
  - Bv.
    - Indienen formulier
    - Link naar andere pagina
  - Luisteraar voor bijhorende actie oproepen
  - Te genereren pagina bepalen
- Events gekoppeld aan componenten afhandelen

Render Response

- Opbouwen view (indien eerste aanvraag)
  - Boomstructuur van componenten
- JSP-container
  - Overloopt componenten
    - Genereren uitvoer
- Fouten in vorige fazen
  - Originele pagina + foutberichten
- Bewaren statusinformatie

FacesServlet

- Registratie FacesServlet in web.xml
  - Handelt alle aanvragen af (zie levensloop)

```
<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>
```

Bibliotheken

- jsf-api.jar
- jsf-impl.jar
- jstl.jar
- standard.jar
- commons-beanutils.jar
- commons-collections.jar
- commons-digester.jar
- commans-loggins.jar

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

Events

- Event-object
  - Gegenereerd door component
  - Informatie gebeurtenis
- Luisteraar registreren
  - Verwittigd bij optreden event
- Drie types
  - Value-change events: veranderen waarde component
  - Action events: klikken op knoppen, links, …
  - Data-model events: selecteren rij, …

Luisteraar

- Twee mogelijke methodes
  - Methode managed bean
  - Luisterklasse schrijven

Luisteraar

- Methode managed bean
  - Attribuut actionlistener van component
    - Met unified EL verwijzen naar methode

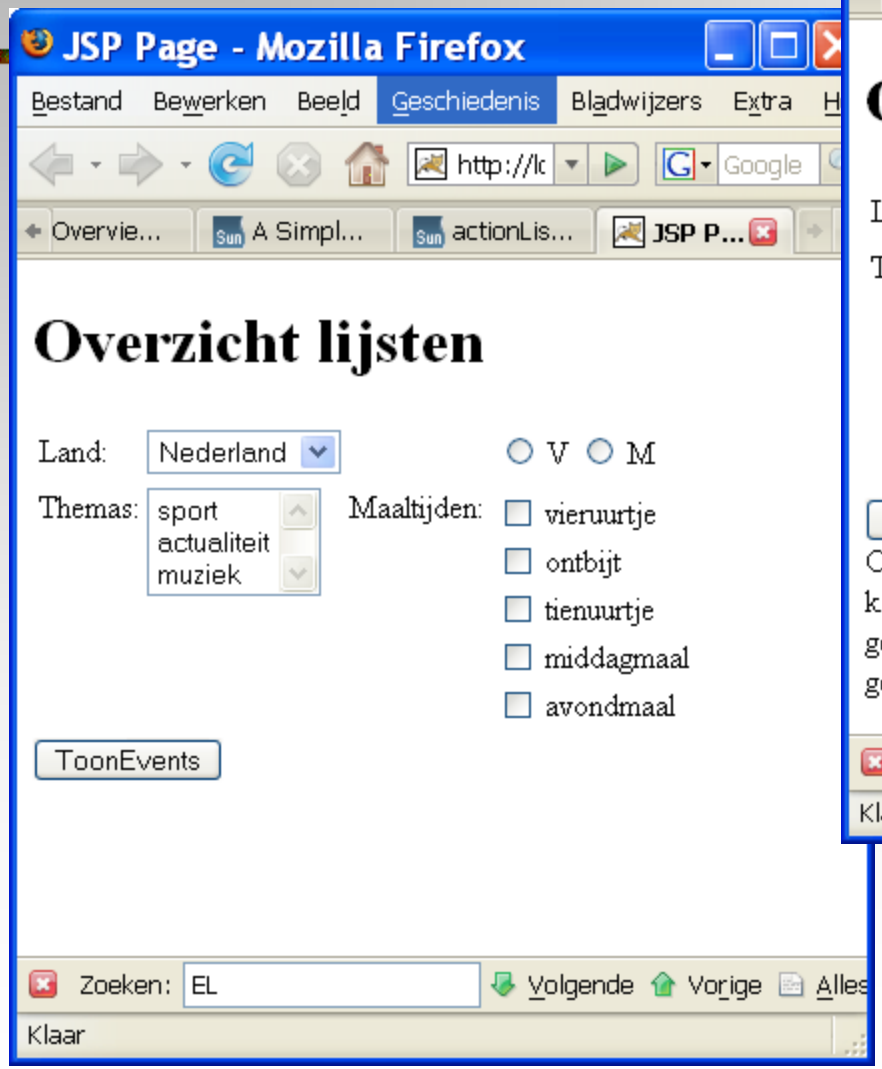
```
<h:commandButton  actionListener="#{GebruikersInfo.bepaalKeuze}" />
```

  - Signatuur methode in bean

```
public void bepaalKeuze(ActionEvent e) {
      keuze = "Hallo " + naam + ", u koos " + thema + ".";
}
```

  - javax.faces.event.ActionEvent

Voorbeeld


JSP Page - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra H
Overzicht lijsten
Land: Nederland
V M
Themas: sport actualiteit muziek
Maaltijden: vieruurtje ontbijt tienuurtje middagmaal avondmaal
ToonEvents
Zoeken: EL
Volgende
Vorige
Klaar


JSP Page - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
Overzicht lijsten
Land: België
V M
Themas: sport actualiteit muziek
Maaltijden: vieruurtje ontbijt tienuurtje middagmaal avondmaal
ToonEvents
Component keuzeLand geselecteerde waarde: BE Component keuzeGeslacht geselecteerde waarde: V Component keuzeThema geselecteerde waarden: sport muziek Component keuzeMaaltijden geselecteerde waarden: 16U 10U
Zoeken: EL
Volgende
Vorige
Klaar

Voorbeeld

- http://localhost:8084/jsf/faces/lijsten.jsp

```
<h:selectOneRadio layout="lineDirection" id="keuzeGeslacht"
  valueChangeListener="#{InfoLijsten.itemGeselecteerd}">
     <f:selectItems value="#{ThemaFabriek.geslachten}" />
</h:selectOneRadio>
```

- be.hogent.iii.jsf.beans.InfoLijsten
  - Reageren op acties + bericht tonen

```
public void itemGeselecteerd(ValueChangeEvent e)
    throws AbortProcessingException  {  … }
public void itemsGeselecteerd(ValueChangeEvent e)
    throws AbortProcessingException  { … }
```

- be.hogent.iii.jsf.beans.ThemaFabriek
  - Opvullen lijsten

Luisteraar

- Luisterklasse schrijven
  - Implementatie van de interface javax.faces.event.ValueChangeListener of javax.faces.event.ActionListener

public void processValueChange(ValueChangeEvent event) throws AbortProcessingException

public void processAction(ActionEvent event) throws AbortProcessingException

  - Registreren
    - Tag valueChangeListener of actionListener
      - Attribuut type: klasse luisteraar

```
<h:inputText  …>
   <f:valueChangeListener type="listeners.NameChanged" />
</h:inputText>
```

Luisteraar

- Methode managed bean
  - Attribuut actionlistener van component
    - Met unified EL verwijzen naar methode
- Luisterklasse schrijven
  - Implementatie van de interface ValueChangeListener of ActionListener
  - Registreren
    - Tag valueChangeListener of actionListener
      - Attribuut type: klasse luisteraar

JSF

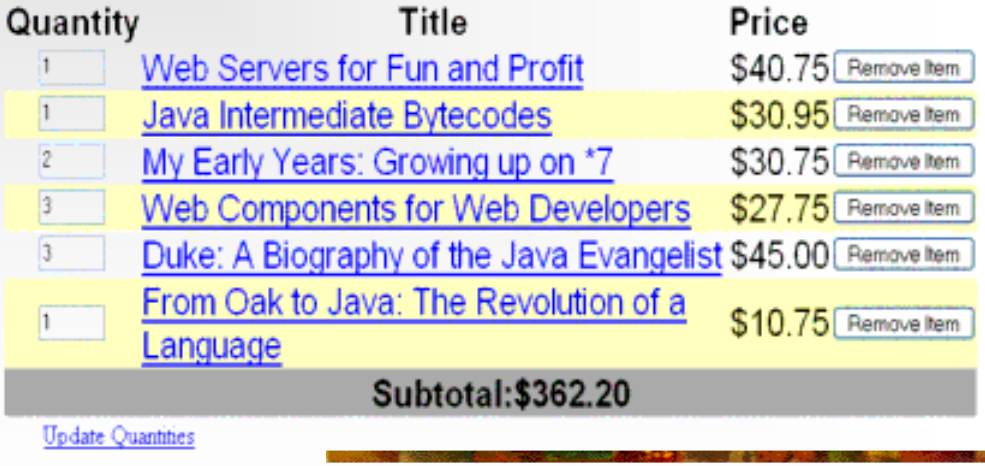
- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

UIData component

- Voorbeeld
  - toonboeken.jsp
- Corresponderende tag
  - h:dataTable
- Toont verzameling (=value-attribuut) van data-objecten in tabelvorm

UIData component

- h:column
  - Beschrijft layout kolom
  - Kan ook knop, link, … zijn
- f:facet
  - Headers beschrijven
    - In <h:column>
  - Footers beschrijven
    - Na <h:column>-tags



JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

Taal-afhankelijke informatie in bundels

- Labels per taal in .properties bestanden
  - Properties-bestand
    - Tekstbestand
    - Per lijn een sleutel-waarde-paar

      sleutel=waarde
    - Commentaar na '#'
- Referentie naar bundel

```
<f:loadBundle basename="be.hogent.iii.jsf.beans.labels" var="labels"/>
```

- Label ophalen uit bundel

```
<f:facet name="header">
  <h:outputText value="#{labels.titel}"/>
</f:facet>
```

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

Navigatie

- Opeenvolging pagina's
- Regels
  - Wat na klikken op knop of link?
  - In configuratie (faces-config.xml)

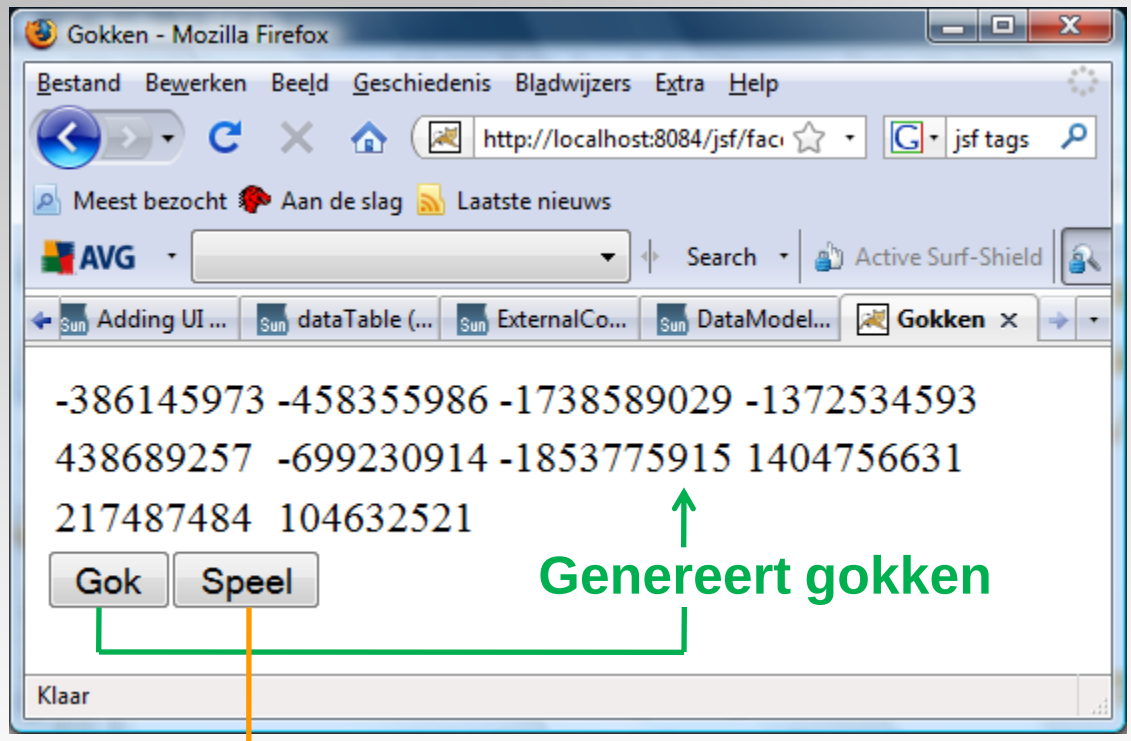
```
<navigation-rule>
  <from-view-id>/greeting.jsp</from-view-id>
  <navigation-case>
    <from-outcome>success</from-outcome>
    <to-view-id>/response.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

Navigatie

- Welke regel?
  - action-attribuut
    - String
    - Resultaat methode

```
<h:commandButton id="submit" action="success" value="Submit" />
<h:commandButton value="Submit“ action="#{cashier.submit}" />
```

Voorbeeld



Genereert gokken

Methode speel van een Casino-object

Navigatie in faces-config.xml

```
<navigation-rule>
    <from-view-id>/gokken.jsp</from-view-id>
    <navigation-case>
        <from-action>#{Casino.speel}</from-action>
        <from-outcome>pech</from-outcome>
        <to-view-id>/speelOpnieuw.jsp</to-view-id>
    </navigation-case>
    <navigation-case>
        <from-action>#{Casino.speel}</from-action>
        <from-outcome>gewonnen</from-outcome>
        <to-view-id>/gewonnen.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
```

Casino

- faces-config.xml

```
<managed-bean>
    <managed-bean-name>Casino</managed-bean-name>
    <managed-bean-class>be.hogent.iii.jsf.beans.Casino</managed-bean-class>
    <managed-bean-scope>application</managed-bean-scope>
</managed-bean>
```

- Haalt gokken van de sessie
- Gewonnen?

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

Validatie

- Controle gebruikersinvoer
- keuzeJSF.jsp

Geen invoer

Enkel spaties

Validatie

- Attribuut required = true
  - Automatisch foutboodschap indien niet opgegeven
- Attribuut validator
  - Methode

```
public void controleerNaam(FacesContext context,
    UIComponent toValidate, Object value) { … }
```

    - Validatie mislukt instellen
    - Foutboodschap

Ingebouwde validatie

- validateDoubleRange-tag
  - Interval (reëel)
- validateLength-tag
  - Lengte (string)
- validateLongRange-tag
  - Interval (geheel)
- Attributen
  - Minimum
  - Maximum

```
<h:inputText id="quantity" size="4" value=
    "#{item.quantity} " >
    <f:validateLongRange minimum="1"/>
</h:inputText>
<h:message for="quantity"/>
```

Andere methode eigen validatie

- Implementeren Validator-interface

```
public void validate(FacesContext context, UIComponent component,
    Object toValidate)
```

- Koppelen aan component

```
<h:inputText ... >
  <f:validator validatorId="customValidator" />  ...
</h:inputText>
```

- Registeren bij applicatie (faces-config.xml)

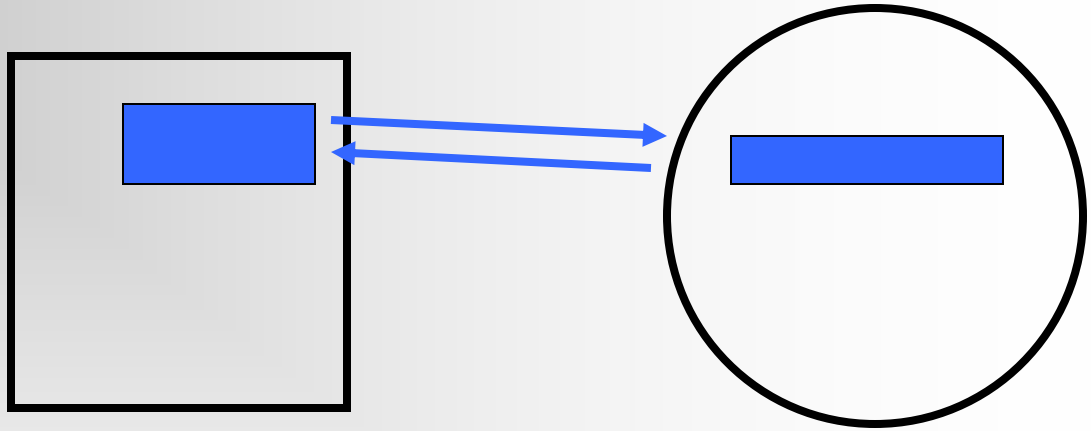
```
<validator>
  ...
  <validator-id>customValidator</validator-id>
  <validator-class>
    validators.CustomValidator
  </validator-class>
</validator>
```

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

Convertor

- Data component
  - Model view
    - Bv. int, long
  - Presentatie view
    - Bv. datum in bepaald string-formaat
- Automatische conversie



JSP met JSF-componenten

Java Bean
Eigenschappen

Automatische conversie

- Bean

```
Integer age = 0;
public Integer getAge(){ return age;}
public void setAge(Integer age) {this.age = age;}
```

- Component

```
<h:inputText value="#{MijnBean.age}" />
```

- Achter de scherm wordt de IntegerConverter gebruikt

Expliciete conversie

- Gebruik converter-attribuut

```
<h:inputText converter="javax.faces.convert.IntegerConverter“ … />
```

- Gebruik converter-tag

```
<h:inputText value="#{LoginBean.age}" />
  <f:converter converterId="Integer" />
</h:inputText>
```

Conversie datums

- convertDateTime-tag

```
<h:outputText value="#{cashier.shipDate}">
  <f:convertDateTime pattern="EEEEEEEE, MMM dd, yyyy" />
</h:outputText>
```

Conversie getallen

- convertNumber-tag

```
<h:outputText value="#{cart.total}" >
  <f:convertNumber type="currency“/>

</h:outputText>
```

- Bv. aantal cijfers voor of na komma, …

Eigen converter

- Koppelen aan component

```
<h:inputText converter="CreditCardConverter“ …>
  ...
</h:inputText>
```

- Registreren in applicatie (faces-config.xml)

```
<converter>
  <description> … </description>
  <converter-id>CreditCardConverter</converter-id>
  <converter-class>
    converters.CreditCardConverter
  </converter-class>
</converter>
```

Eigen converter - klasse

- Implementatie Converter-interface
- Twee methodes
- Presentatie → Model

Object getAsObject(FacesContext context, UIComponent component,
String value)

- Model → Presentatie

String getAsString(FacesContext context, UIComponent component,
Object value)

JSF

- Inleiding
- Voorbeeld
- Componenten en beans
- Levensloop
- Events
- UIData
- Taal-afhankelijkheid
- Navigatie
- Validatie
- Conversie
- Componenten

rendered-attribuut

- Logische uitdrukking
  - Bepaalt of de component uitvoer genereert

```
<h:commandLink id="check"
    ...
    rendered="#{cart.numberOfItems > 0}">
  <h:outputText
   value="#{bundle.CartCheck}"/>
</h:commandLink>
```

immediate attribuut

- In welke faze events, validaties, … uitvoeren
- Indien true
  - Apply request values faze
  - Vervroegd uitvoeren
- Gebruikt bij commandLink
  - Voorkomt bv. dat formuliergegevens verwerkt worden (Logout, Annuleren, …)

Componenten binden

- Normaal binden waarde component aan eigenschap Java Bean
- Soms component zelf binden aan eigenschap Java Bean
  - type = type component
  - Component kan gemanipuleerd worden vanuit Java Bean

```
<inputText binding="#{UserNumberBean.userNoComponent}" />
```

```
UIInput userNoComponent = null;
public void setUserNoComponent(UIInput userNoComponent) {
    this.userNoComponent = userNoComponent; }
public UIInput getUserNoComponent() {
    return userNoComponent; }
```

Informatie

- The Java EE 5 Tutorial (http://java.sun.com/javaee/5/docs/tutorial/doc/bnaph.html)
  - Hoofdstuk 10 – 15
- JSF Tag Library (http://java.sun.com/javaee/javaserverfaces/1.2_MR1/docs/tlddocs/index.html)