Java Server Pages

joost.vennekens@kuleuven.be

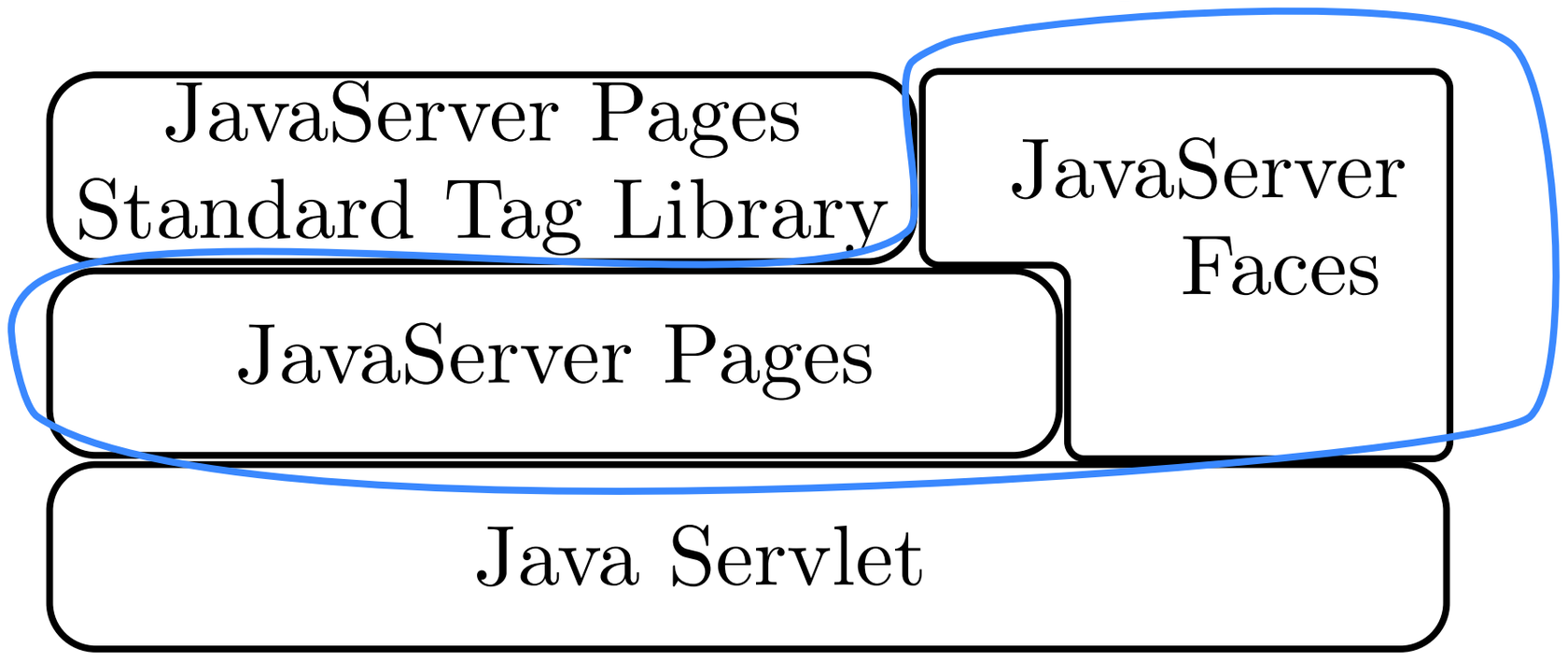
http://www.cs.kuleuven.be/~joost/DN/

Motivatie

- Maken van View
- Dynamische pagina
  - HTML
  - (Beetje) code
- Gebruiksvriendelijk formaat

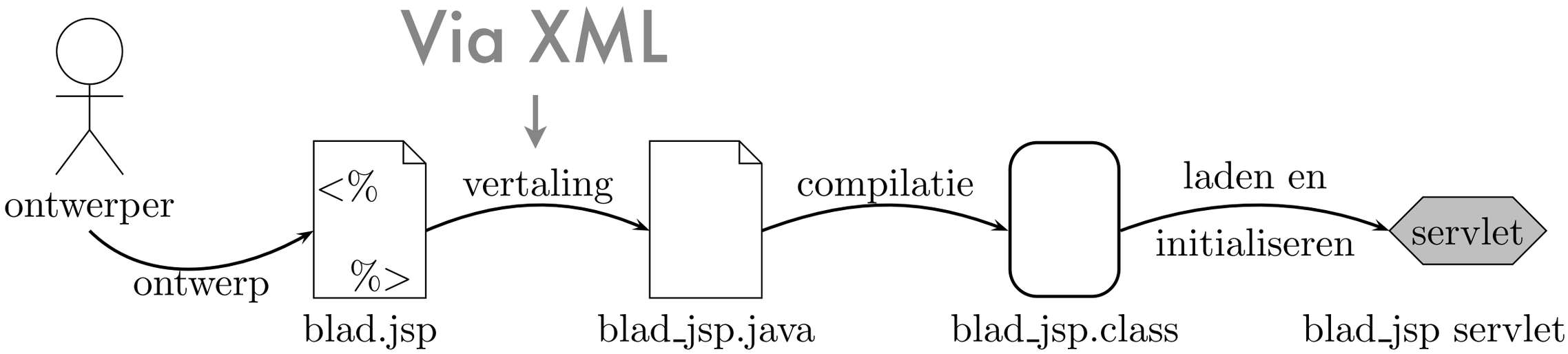
Servlet

Unified expression language



Levenscyclus

Bij eerste request voor blad.jsp:



Netbeans: right-click + view servlet

JSP syntax

```
import ...;

public class index_jsp extends ... {
  ...
  public void _jspService(...) {
    ...
    out.print(...);
    ...
  }
}
```

Directieven <%@ ... %>

Scriptlets <% ... %>

Expressies <%= ... %>

Declaraties <%! ... %>

Directieven

<%@page import=”model.*” %>

<%@page contentType=”text/html”%>

<%@page pageEncoding=”UTF–8”%>

Expressies

```
<p> Dit is het
      <%= Teller.getTeller() %>e bezoek.
</p>
```

=

```
<p> Dit is het
      <% out.print(Teller.getTeller()); %>e bezoek.
</p>
```

Voorbeeldje

teller1.jsp

Gedeelde geheugen

setAttribute(“naam”, object) & getAttribute(“naam”)

	Servlet	JSP
App	getServletContext()	application
Sessie	getSession()	session
Req	request	request
Pagina	lokale variabele	pageContext

Voorbeeld

BierExpertWeb

Java Beans en EL

Java Bean

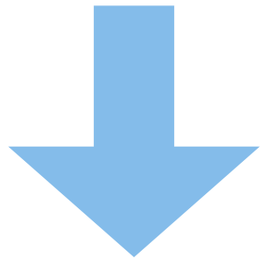
- Java klasse volgens bepaalde conventies
  - Constructor zonder argumenten
  - Voor eigenschap Klasse att naar keuze:
    - Getter: public Klasse getAtt() { ... }
    - Setter: public void setAtt(Klasse a) { ... }

Bijvoorbeeld

```
package boeken;
public class Boek {

    private String titel;
    private String auteur ;

    public Boek() { }

    public String getTitel () { return titel ; }
    public void setTitel(String titel) { this.titel = titel; }

    public String getAuteur ()    {return auteur; }
    public void setAuteur(String auteur) { this.auteur = auteur; }
}
```

Gebruik in JSP

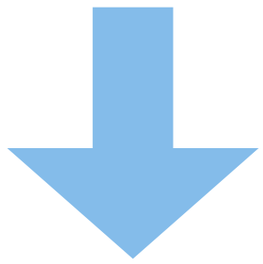
```
<jsp:useBean
    id=”eenBoek”
    class=”boeken.Boek”
    scope=”session” />
```



```
boeken.Boek eenBoek = getSession().getAttribute(“eenBoek”);
if (eenBoek == null) {
    eenBoek = new boeken.Boek();
    getSession().setAttribute(“eenBoek”, eenBoek);
}
```

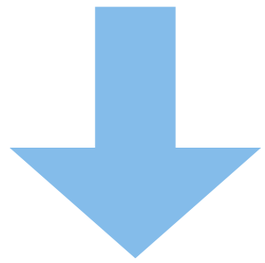
Gebruik in JSP

```
<jsp:setProperty
    name=”eenBoek”
    property=”titel”
    value=”De Avonden” />
```



```
eenBoek.setTitel(“De Avonden”);
```

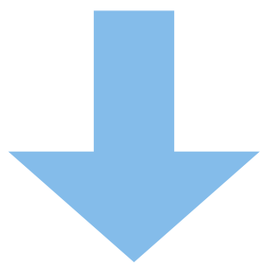
```
<jsp:getProperty
    name=”eenBoek”
    property=”titel” />
```



```
eenBoek.getTitel();
```

Initializatie van bean

```
<jsp:useBean id=”eenBoek” class=”boeken.Boek”
  scope=”session”>
   <jsp:setProperty id=”eenBoek” property=”titel”
     value=”De Avonden” />
</jsp:useBean>
```



```
boeken.Boek eenBoek = getSession().getAttribute(“eenBoek”);
if (eenBoek == null) {
    eenBoek = new boeken.Boek();
    eenBoek.setTitel(“De Avonden”);
    getSession().setAttribute(“eenBoek”, eenBoek);
}
```

teller2.java

Alternatief

- Unified expression language
  - ${...} (JSP)
  - #{...} (JSF)
- Toegang tot gegevens in Java Beans
- Oproepen van methodes
- Uitvoeren van bewerkingen

Verwijzen naar objecten

${boon}

- Proberen getAttribute(“boon”) in
  - Page scope
  - Request scope
  - Session scope
  - Application scope

Impliciete objecten

- param
- paramValues
- header
- headerValues
- cookie
- initParam
- pageContext

Map

param.naam

param[“naam”]

paramValues[“naam”][0]

request.getParameter(“naam”);

Let op!

- Impliciete objecten van JSP <> impliciete objecten van EL
- JSP: request, response, session, application, pageContext
- EL: param, paramValues, header, headerValues, cookie, initParam, pageContext
- Toegang tot JSP impliciete objecten via pageContext

Let op!

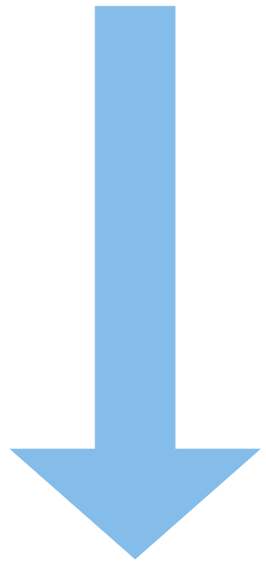
- ${pageContext.request}
- ${pageContext.response}
- ${pageContext.session}
- ${pageContext.servletContext} = application

Gedeelde geheugen

Map

- pageScope
- requestScope
- sessionScope
- applicationScope

${iets}



- Oplossen van name clashes
- requestScope[“iets.persoon”].naam ipv iets.persoon.naam

Operatoren

- rekenkundig: `+`, `-`, `*`, `/` en `div`, `%` en `mod`, en de unaire `-`;
- logisch: and, `&&`, or, `||`, not, `!`;
- relationeel: eq, `==`, ne, `!=`, lt, `<`, gt, `>`, le, `<=`, ge, `>=`;
- conditioneel: `A ? B : C` , hierbij wordt B of C geevalueerd afhankelijk van resultaat van de evaluatie van A;
- de `empty` prefix-operator bepaalt of een waarde `null` of leeg is.

Voorbeelden

EL expr	res
${1 > (4/2)}	false
${4.0 >= 3}	true
${100.0 == 100}	true
${'a' < 'b'}	true
${'hip' gt 'hit'}	false
${(10*10) ne 100}	false
${3 div 4}	0.75
${10 mod 4}	2
${1.2E4 + 1.4}	12001.4

Functies oproepen

- Static functies uit java klasse

```
package model;
public class Dobbelsteener
{
    public static int werpSteen()
    {
        return (int) ( (6 * Math.random()) + 1);
    }
}
```

Functies oproepen

```
<%@ taglib prefix="dob" uri="DobbelsteenFuncties" %>
<html> <body>
      <p>  Aantal ogen : ${dob:werpen()}  </p>
                                          </body> </html>
```

```
<taglib ... >
 <uri>DobbelsteenFuncties</uri>
   <function>
     <name>                 werpen                 </name>
     <function-class>       model.DobbelStener</function-class>
     <function-signature> int werpSteen()    </function-signature>
   </function>
</taglib>
```

Voorbeeld

index.jsp

Code herbruik

fragment

- <jsp:include page=”ander.jspf” />
- <jsp:include page=”ander.jspf”>
  <jsp:param name=”extra” value=”nog” />
  </jsp:include>
- <jsp:forward page=”ander.jsp” />

Fouten

```
<%@ page errorPage="errorPage.jsp" %>
<html><body> <p> Het volgende gaat verkeerd aflopen:
                    <% int x = 10/0;  %>                    </p>
  </body></html>
```

Als code exception gooit,
wordt er naar errorPage.jsp gegaan

Fouten

```
<%@ page isErrorPage="true" %>
<html><body>  <p> Er is een fout opgetreden.
                    Controleer uw invoerparameters. </p>
      </body> </html>
```

Toegang tot "exception" object:

```
<%@ page isErrorPage="true" %>
<html> <body>
    <p> Fout ${pageContext.exception} is opgetreden op de server.
        Controleer uw invoer parameters.
    </p>
</body> </html>
```

Alternatief

```
<error-page>
    <exception-type>java.lang.ArithmeticException </exception-type>
    <location>/rekenFout.jsp </location>
</error-page>
<error-page>
    <exception-type>java.lang.Throwable </exception-type>
    <location>/errorPage.jsp </location>
</error-page>
<error-page>
    <error-code>404</error-code>
    <location>/nietGevonden.jsp </location>        </error-page>
```

JSTL

Motivatie

- JSP: hutsepotje van HTML en Java
- Beter <jsp:getProperty name=”eenBoek” property=”titel” /> dan <%= eenBoek.getTitel() %>
- Meer mogelijk maken zonder Java code

Taglib

- Verzameling tags
- Die bepaalde functionaliteit aanbieden
- Beschikbaar via taglib declaratie

<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>

Overzicht

area	subfunctie	prefix	uri
core	variabele	c	`http://java.sun.com/jsp/jstl/core`
	flow control		
	URL beheer		
xml	basis	x	`http://java.sun.com/jsp/jstl/xml`
	flow control		
	transformatie		
I18N	locale	fmt	`http://java.sun.com/jsp/jstl/fmt`
	formattering		
databank	SQL	sql	`http://java.sun.com/jsp/jstl/sql`
functie	collectie lengte	fn	`http://java.sun.com/jsp/jstl/functions`
	string bewerking		

core library

- Variabelen manipuleren
- Controlflow
- URL management
- Varia

Variabelen

Default: page

- <c:set var=”boeknr” scope=”session”
  value=”${param.nummer}” />
- <c:remove var=”boeknr” scope=”session” />
- <c:set target=”${persoon}” property=”naam”>
  ${iets .naam}
  </c : set>

Control flow

- <c:if test="${!empty param.nummer}">
  <c:set var="boeknr" scope="session"
  value="$param.nummer" />
  Boek met nummer ${boeknr} is toegevoegd.
  </c:if>
- Maar geen “else”

Control flow

- <c:choose>
  <c:when test=”${param.cat == ’trial’}”>
  <h2>Proberen </h2></c:when>
  <c:when test=”${param.cat == ’lid’}”>
  <h2>Leden </h2></c:when>
  <c:otherwise>
  <h2>Gasten</h2></c:otherwise>
</c:choose>

Control flow

Map, Collection, array,..

- <c:forEach
  var=”movie” items=”${filmLijst}”>
  <tr> <td> ${movie} </td> </tr>
</c:forEach>

URL management

Doet zonodig herschrijving

- <c:url var=”flip” value=”/verwerk.do” >
  <c:param name=”voorNaam”
  value=”${voor}” />
  <c:param name=”naNaam”
  value=”${na}” /></c:url>
- <a href=”${flip}” Klik hier

Varia

- <c:catch>
- <c:out value=”${naam}”
  default=”niemand”
  escapeXML=”true” />

VB: viewtags.jsp

SQL

- Normaal: EJBs
- Voor rapid prototyping: sql tags
- <sql:query var=”boeken”>
  select * from PUBLIC.boeken where id = ?
  <sql :param value=”${bid}” />
  </sql : query>

Resultaat

- public interface Result {
  public String [] getColumnNames();
  public int getRowCount();
  public Map[] getRows();
  public Object[][] getRowsByIndex() ;
  public booleanisLimitedByMaxRows () ;
}

Bijvoorbeeld

- <sql:query var=”boeken”>
  select * from PUBLIC.boeken where id = ?
  <sql:param value=”${bid}” />
  </sql:query>

-
  <c:forEach var=”bk”
  items=”${boeken.rows}” >
   ${bk.titel}
  </c:forEach>

Netbeans

- Services -> Databases
- New -> Glassfish -> JDBC resource
  - Resultaat in glassfish-resources.xml en (na restart) in Services
- Toevoegen aan web.xml (tab resources)

VB: viewtagssql.jsp

SQL - andere

- <sql:update var=”boeken”
  sql=”update PUBLIC.boeken set inv = inv – ?
  where id = ?”>
  <sql :param value=”${hoeveelheid}” />
  <sql :param value=”${bid}” />
  </sql : update>

- <transaction> voor transacties

Lokalizering

- <jsp:useBean id=”nu” class=”java.util.Date” />
- <p>Standaard :
  <fmt:formatDate value=”${nu}” type=”date”
  dateStyle=”full” />
- <p>Spaans :
  <fmt:setLocale value=”es” />
  <fmt:formatDate value=”${nu}” type=”date”
  dateStyle=”full” />

Function

- <c:if test=”${fn:length(param.gebruiker) > 0}” >

Functies om bewerkingen met strings te doen:

- toUpperCase, toLowerCase
- substring, substringAfter, substringBefore
- trim, resplace
- indexOf, startsWith, endsWith
- contains, containsIgnoreCase
- split , join