



# Tag Libraries - Verwendung

## Java EE Grundlagen

## **Inhalte dieses Kapitels**

Allgemeines

JSP Standard Tag Library (JSTL)

Zusammenfassung

# Allgemeines [1|3]

## Motivation und Konzept

- Komplexe Logik in JSPs (bei Generierung der Antwort)
  - Möglichkeiten von XML begrenzt
- Doppelter Code in JSPs
  - Zentralisierung
  - Wiederverwendbarkeit
- Tag Libraries
  - Sammlungen von JSP-Tags zur Realisierung bestimmter Funktionen
  - Auslieferung als JAR
    - Einbindung über Classpath

## Allgemeines [2|3]

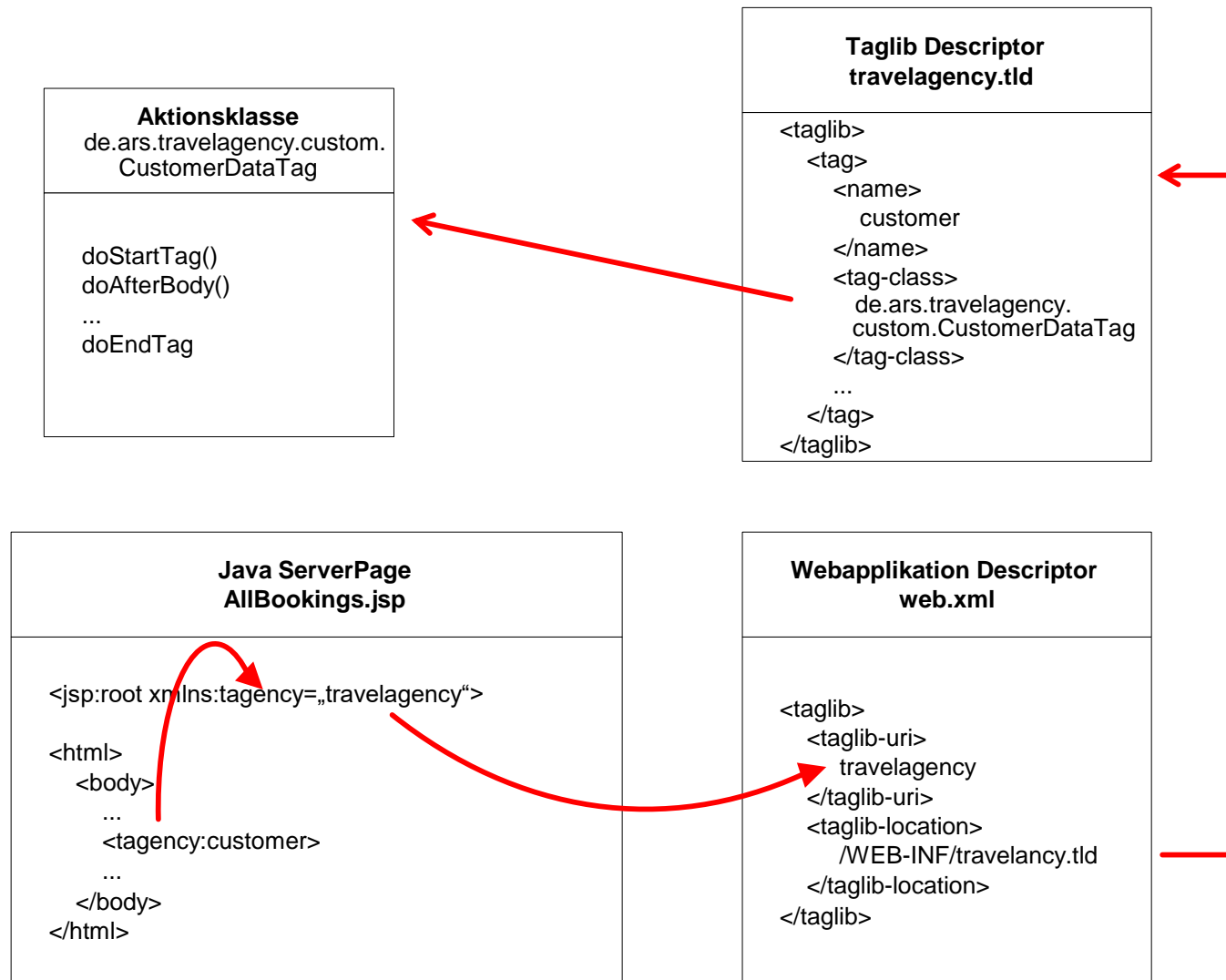
### Einbinden von Tag Libraries

- Einbinden als XML-Namespace
  - Best Practise: Einbinden am Wurzelelement
- Tag Library (JAR) mit URI muss im Classpath liegen
- Custom Tag kann
  - Logik ausführen und/oder
  - Ausgabe generieren

```
<jsp:root xmlns:ui="http://www.ars.de/web/ui">  
  <ui:text label="Anzahl Teilnehmer" name="count" />  
</jsp:root>
```

# Allgemeines [3|3]

## Ermitteln der Custom-Tag-Implementierung über Classloader



# JSP Standard Tag Library [1|11]

## Bedeutung

- Tag Libraries für häufige Aufgaben in JSPs
  - Schleifen, Fallunterscheidungen
  - Operationen auf Zeichenreihen
  - Internationalisierung
- Standardisierung im Rahmen des Java Community Process
- Reduzierung der Notwendigkeit zur Entwicklung eigener Tag Libraries
- Aufteilung nach Aufgabenbereich
- Bereitstellung durch Webcontainer

# JSP Standard Tag Library [2|11]

## Bestandteile

- Core (Kernaufgaben)
  - Iterationen
  - Fallunterscheidungen
- Internationalisierung und Formatierung
  - Ausgeben und Einlesen von landesspezifischen Informationen
  - Zahlen- und Datumsformatierung
- Funktionen
  - Operationen aus Zeichenreihen
- XML processing (nicht empfohlen)
  - Zugriff auf XML-Dokumente, XSLT
- Relational database access (nicht empfohlen)
  - Zugriff auf relationale Datenbanken

# JSP Standard Tag Library [3|11]

Area	Subfunction	Prefix	TLD
Core	Variable Support	c	<a href="http://java.sun.com/jsp/jstl/core">http://java.sun.com/jsp/jstl/core</a>
	Flow Control		
	URL management		
	Miscellaneous		
XML	Core	x	<a href="http://java.sun.com/jsp/jstl/xml">http://java.sun.com/jsp/jstl/xml</a>
	Flow control		
	Transformation		
l18n	Locale	fmt	<a href="http://java.sun.com/jsp/jstl/fmt">http://java.sun.com/jsp/jstl/fmt</a>
	Message formatting		
	Number and date formatting		
Database	SQL	sql	<a href="http://java.sun.com/jsp/jstl/sql">http://java.sun.com/jsp/jstl/sql</a>
Functions	Collection length	fn	<a href="http://java.sun.com/jsp/jstl/functions">http://java.sun.com/jsp/jstl/functions</a>
	String manipulation		



# JSP Standard Tag Library [4|11]

## Core Tags

Name	Beschreibung
catch	Exception abfangen
out	Texte und Ausdrücke auf JSP-Seite ausgeben
forEach	Collections durchlaufen
if	if-Anweisung
choose when otherwise	Mehrfachauswahl (analog zu switch/case)
import	Ressourcen einbinden (müssen nicht in der gleichen Webapplikation sein)
redirect	Redirect an angegebene URL
url	URL Encoding durchführen
set remove	Variable in Sichtbarkeitsbereich ablegen bzw. entfernen

# JSP Standard Tag Library [5|11]

## Core Tags – Beispiel

```
<!-- Setzen einer Variablen -->
```

```
<c:set var="firstName" value="Lorenz"/>
```

```
<!-- Wert der Variable über Body -->
```

```
<c:set var="firstName">
```

Lorenz

```
</c:set>
```

```
<!-- Setzen einer Variable per EL -->
```

```
<c:set var="firstName" value="${param.firstName}"/>
```

```
<!-- Löschen einer Variablen-->
```

```
<c:remove var="firstName"/>
```

```
<!-- Angabe eines Gültigkeitsbereichs -->
```

```
<c:set var="firstName" value="Lorenz" scope="session"/>
```

# JSP Standard Tag Library [6|11]

## Core Tags – Beispiel

```
<c:forEach items="${sessionScope.startFlights}" var="flight">
  <c:if test="${flight.seats ge sessionScope.order.persons}">
    <tr>
      <td>
        <input type="radio" name="flightsHome" value="${flight.flightID}">
      </td>
      <td>
        <c:out value="${flight.flightNumber}"/>
      </td>
      [...]
    </tr>
  </c:if>
</c:forEach>
```

# JSP Standard Tag Library [7|11]

## Core Tags – Beispiel

<!-- Es wird eine neue URL erzeugt.

Sie verweist auf die Resource, wie in "value" angegeben.

Im Body des Tags können z.B. Request-Parameter mitgegeben werden.

Die URL wird in der Variable namens "url" gespeichert.

-->

```
<c:url var="url" value="/insertBook" >
  <c:param name="book" value="${bookId}" />
</c:url>
<p><a href="${url}">Buch einfügen</a></p>
```

# JSP Standard Tag Library [8|11]

## l18n

Name	Beschreibung
bundle	Resource Bundle für den Bereich des Tags festlegen
setBundle	Resource Bundle für einen wählbaren Bereich festlegen
setLocale	Vorgabe des Clients für eine Locale überschreiben
requestEncoding	Codierung des Requests überschreiben
message	String für eine Locale auf JSP-Seite ausgeben
formatNumber parseNumber	Zahlen internationalisiert ausgeben bzw. einlesen
formatDate parseDate	Datum internationalisiert ausgeben bzw. einlesen
timezone	Zeitzone für den Bereich des Tags festlegen
setTimezone	Zeitzone für einen wählbaren Bereich festlegen

# JSP Standard Tag Library [9|11]

## l18n – Beispiele

```
<fmt:bundle basename="nls.Messages">  
  <td><fmt:message key="selection" /></td>  
  <td><fmt:message key="flightNumber" /></td>  
</fmt:bundle>
```

```
<td>  
  <fmt:formatNumber value="${flight.price}"  
    maxFractionDigits="2" minFractionDigits="2" />  
</td>
```

# JSP Standard Tag Library [10|11]

## EL-Funktionen

Name	Beschreibung
contains containsIgnoreCase	Prüfen, ob String eine Teilzeichenreihe enthält
endsWith startsWith	Prüfen, ob String mit angegebener Zeichenreihe endet bzw. beginnt
escapeXML	Alle relevanten Zeichen durch XML Entities ersetzen
indexOf	Ermittelt, an welcher Stelle des Strings eine Teilzeichenreihe enthalten ist
length	Länge eines Strings ermitteln
replace	Ersetzt Zeichenreihe innerhalb des Strings
substring	Teilzeichenreihe erzeugen
substringBefore	Teilzeichenreihe erzeugen, die sich vor der angegebenen Zeichenreihe befindet
toLowerCase/uppercase	String in Klein- bzw. Großbuchstaben umwandeln
trim	Leerzeichen am Anfang und Ende des Strings entfernen

# JSP Standard Tag Library [11|11]

## Expression-Language-Funktionen

```
<!-- Umwandeln von Sonderzeichen in XML-Entitäten -->
```

```
<c:set var="output" value="${fn:escapeXml(book.title)}"/>
```

```
<!-- Ermitteln der Anzahl an Elementen in Collections -->
```

```
<c:set var="size" value="${fn:length(books)}"/>
```

```
<!-- Operationen auf Zeichenketten -->
```

```
<c:set var="firstName" value="${fn:substringBefore(name, ' ')}"/>
```

```
<!-- Verschachteln von Funktionen -->
```

```
<c:set var="output" value="${fn:trim(fn:substring(title, 4))}"/>
```



## Kontrollfragen

- Was sind Tag Libraries und wofür werden sie verwendet?
- In welchem Format werden Tag Libraries ausgeliefert, wie werden sie eingebunden?
- Was ist die JSTL?

