



## InternetBeans Express

- Komponentenbasierter, datengesteuerter Ansatz mit bekannten Mechanismen aus dbSwing
- Datensensitivität wie bei dbSwing durch dataSet- und columnName-Eigenschaft
- DataExpress-Funktionalität anwendbar



3

## Inhalt

- **Architektur der InternetBeans Express Technologie**
- InternetBeans Express mit Servlets
- InternetBeans Express mit JSPs

4

### Klassen und Interfaces lassen sich in drei Kategorien unterteilen

- InternetBeans Komponenten
- JSP-Tag Behandlungsroutinen
- Unterstützung von Datenmodellen

5

- zuständig für die dynamische Generierung von HTML und der HTTP-Semantik
- dienen als Ersatz für die „out.println“-Anweisungen in Servlets

6

## JSP-Tag Behandlungsroutinen

- rufen die InternetBeans und deren unterstützende Infrastruktur intern auf
- werden von den JSP-Tag Erweiterungen in der Tag-Library von InternetBeans verwendet

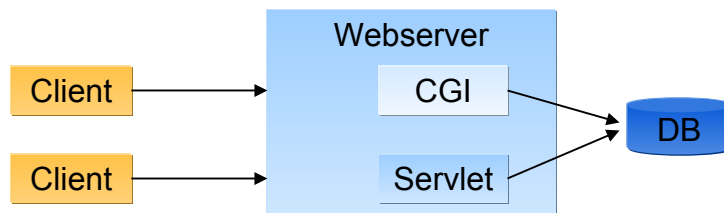
## Unterstützung von Datenmodellen

- DataExpress
- DataModules

- Architektur der InternetBeans Express Technologie
- **InternetBeans Express mit Servlets**
- InternetBeans Express mit JSPs

## Java Servlets

- Ein Mittel zur Erweiterung der Funktionalität von WWW-Servern
- Das Gegenstück zu einem Applet das auf einem Server aktiviert wird.
- Sie werden ähnlich wie das Common Gateway Interface (CGI), zur Erstellung von dynamischen Webseiten verwendet.

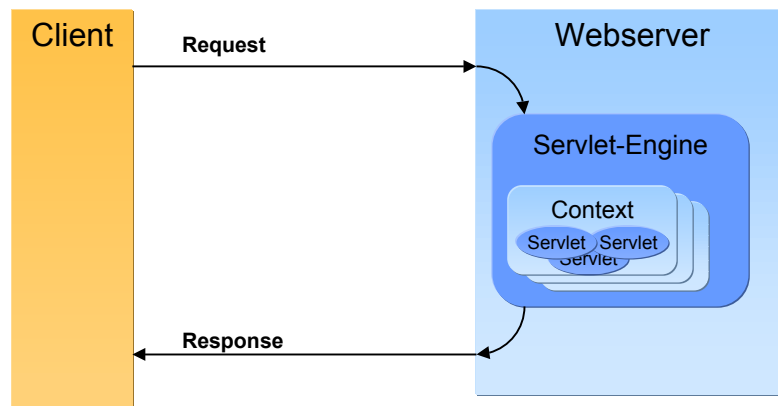


## Vorteile von Servlets

- Sicherheit durch die Sprache Java
- Plattformunabhängigkeit
- Standard API 2.1 → neue API 2.3
- Problemloser Datenbankzugriff (über JDBC)
- Objektorientiert
- Sehr ressourcensparend, da mehrere gleichzeitige Anfragen durch Threads sehr effizient verarbeitet werden können

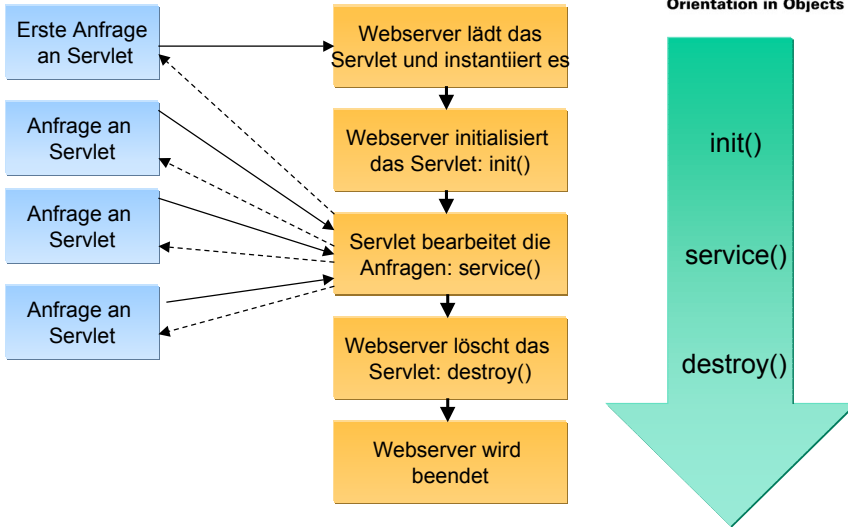
11

## Komponenten

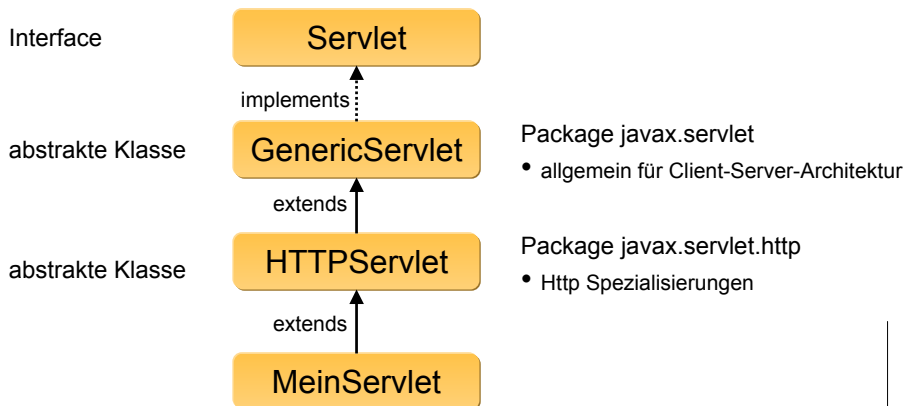


12

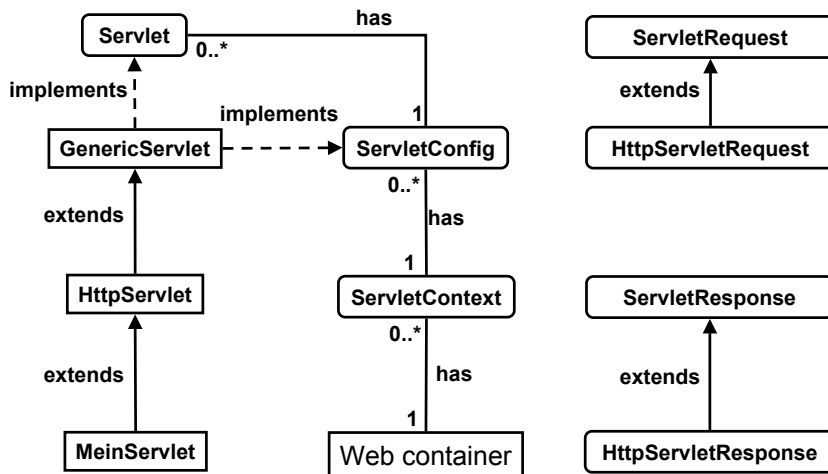
# Servlet Lebenszyklus



# Implementierungshierarchie



## Servlet API in UML



15

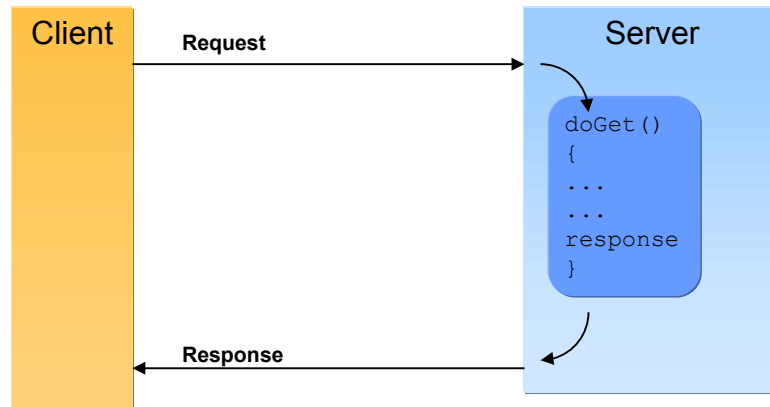
## Die Klasse HttpServlet

Als Unterklasse zu Servlet können hier folgende Methoden überschrieben werden:

- doGet
- doPost
- doPut
- delete

Hier erfolgt jeweils die Übergabe des Request- und des Response-Objektes.

16



17

- Zum Schreiben der Antwort-Seite wird ein `PrintWriter` verwendet
- `PrintWriter` ist Teil von `java.io`
- Der `Printwriter` wird vom `response`-Objekt zurückgeliefert

### Anwendung

```
PrintWriter out = new  
PrintWriter(response.getOutputStream());  
...  
  
out.println("<html><body>");  
  
...  
  
out.close();
```

18

## Schwächen von Servlets

- Daten und Präsentation sind nicht entkoppelt
- Änderung im Aussehen bedeutet Änderung im Code
- keine XML-Integration

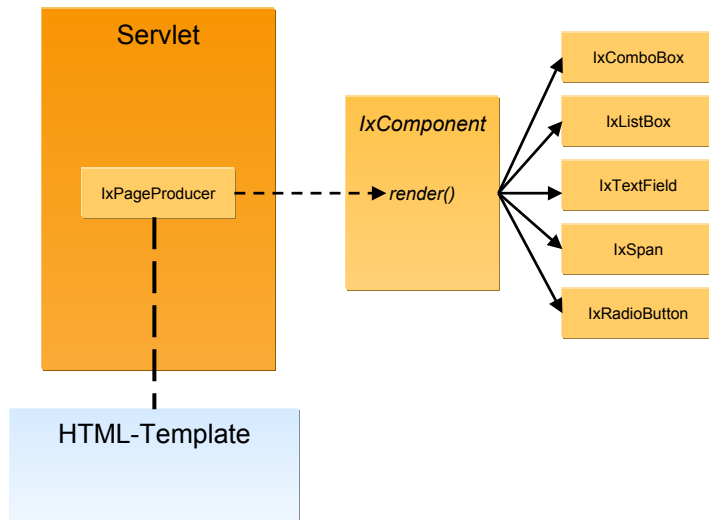
## Ix-Applikation mit Servlets erstellen

1. Eine HTML-Seite erstellen
2. HTML-Elemente mit id-Attributen versehen (Platzhalter)
3. Servlet erstellen
4. Im Servlet einen IxPageProducer generieren
5. Eine Datenquelle oder Model angeben
6. InternetBeans Express Komponenten (Tabellen, Listboxen, Comboboxen) initialisieren und deren properties füllen

- InternetBeans Komponenten werden in die HTML-Seite eingebunden und sind nun datensensitiv, ohne das sich das Aussehen der Seite geändert hat
- für <table>, <span> und andere HTML Tags müssen dann ID Attribute gesetzt werden
- jeder Button ist durch seinen Namen an die InternetBeans Komponente gebunden

- Der ixPageProducer erzeugt die dynamische HTML-Seite
- Parst das template File und ersetzt die identifizierten spans mit dynamischen Inhalt aus den ix-Komponenten
- Die doGet()-Methode reduziert sich oft auf einen Zweizeiler

## ixPageProducer Schema



23

## doGet()

```
public void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException
{
    response.setContentType (CONTENT_TYPE) ;
    ixPageProducer1.servletGet (this, request, response) ;
}
```

24

## doPost()

```
public void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException
{
    response.setContentType (CONTENT_TYPE);
    ixPageProducer1.servletPost (this, request, response);
    doGet (request, response);
}
```

25

## ixControl

- Emuliert jedes HTML-Steuelement
- kann anstelle aller InternetBeans Express Komponenten genutzt werden
- controlName und Name müssen gesetzt werden

26

## Schema der ix-Technologie

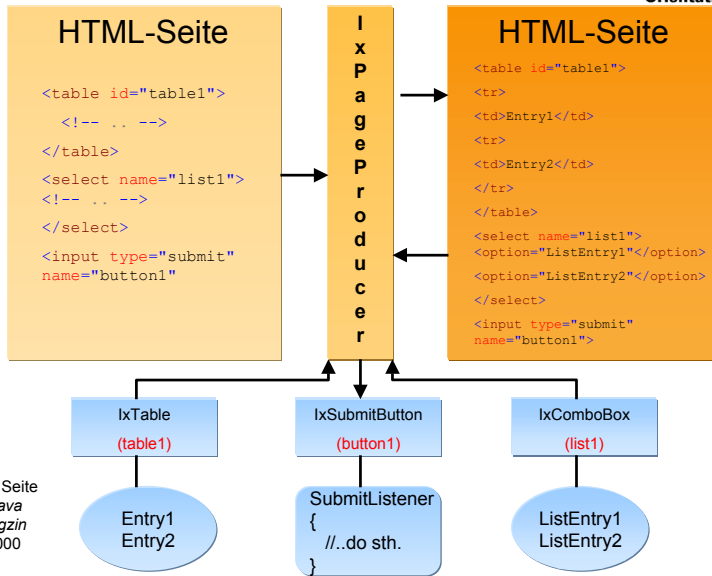


Abb. 3, Seite  
76, Java  
Magazin  
12.2000

27

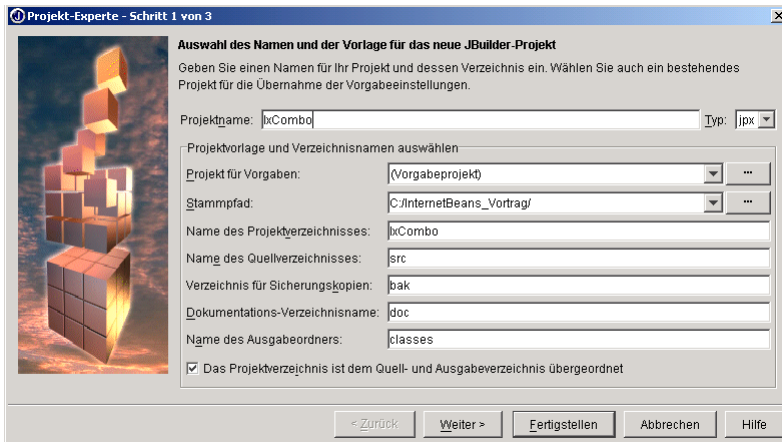
## Vorteile

### Vorteile:

- HTML-Seite dient als Template mit Platzhaltern, die ersetzt werden
- Layout kann ohne Codeänderungen geändert werden
- MVC-Paradigma wird eingehalten
- Anlehnung an dbSwing und damit verbundene Vorteile

28

## Servlet-Beispiel IxCombo - Projekt erstellen



**Projekt-Experte - Schritt 1 von 3**

**Auswahl des Namen und der Vorlage für das neue JBuilder-Projekt**

Geben Sie einen Namen für Ihr Projekt und dessen Verzeichnis ein. Wählen Sie auch ein bestehendes Projekt für die Übernahme der Vorgabeeinstellungen.

Projektname:  Typ:

Projektvorlage und Verzeichnisnamen auswählen

Projekt für Vorgaben:

Stammpfad:

Name des Projektverzeichnisses:

Name des Quellverzeichnisses:

Verzeichnis für Sicherungskopien:

Dokumentations-Verzeichnisname:

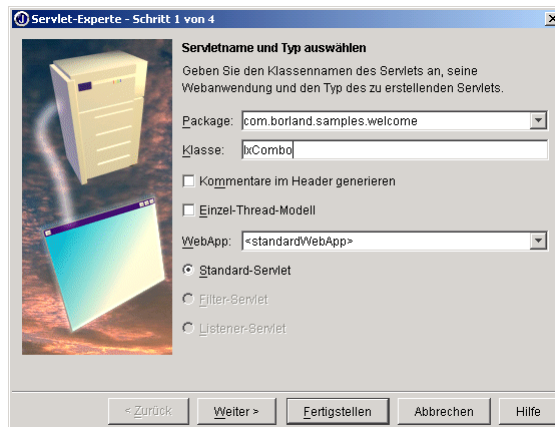
Name des Ausgabeordners:

Das Projektverzeichnis ist dem Quell- und Ausgabeverzeichnis übergeordnet

< Zurück Weiter > Fertigstellen Abbrechen Hilfe

29

## Servlet-Beispiel IxCombo - Servlet-Experte



**Servlet-Experte - Schritt 1 von 4**

**Servletname und Typ auswählen**

Geben Sie den Klassennamen des Servlets an, seine Webanwendung und den Typ des zu erstellenden Servlets.

Package:

Klasse:

Kommentare im Header generieren

Einzel-Thread-Modell

WebApp:

Standard-Servlet

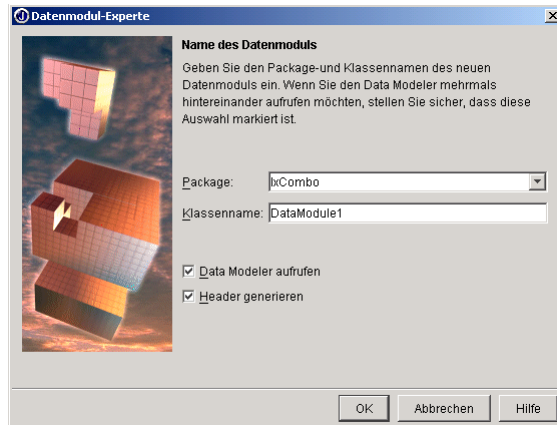
Filter-Servlet

Listener-Servlet

< Zurück Weiter > Fertigstellen Abbrechen Hilfe

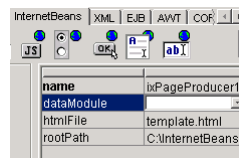
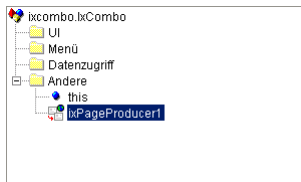
30

# Servlet-Beispiel IxCombo - Datenmodul



31

# Servlet-Beispiel IxCombo - IxPageProducer



32

- Architektur der InternetBeans Express Technologie
- InternetBeans Express mit Servlets
- **InternetBeans Express mit JSPs**

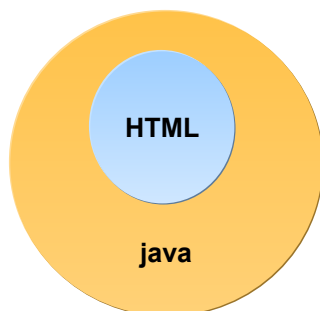
- JSP sind eine Erweiterung der Servlet-API
- Ziel ist die einfache Entwicklung portabler, dynamischer Webanwendungen
- Trennung zwischen Funktionalität und Layout
- Wiederverwendbare Komponenten: ‚Custom Tag Libraries‘
- dadurch Trennung des Programmiererprofils:  
**Java-Programmierer ↔ Webdesigner**

## JSP - technisch gesehen

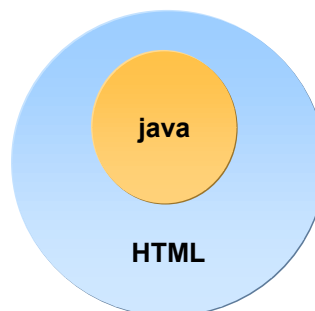
- JSP ist eine Skriptsprache, die in ‚Template‘-Text eingebettet wird
- JSP-Seiten werden vor der ersten Ausführung in Servlets übersetzt (eine JSP-Seite → ein Servlet)

35

## Servlets & JSP



Servlets



JSP

Mit JSPs lassen sich Servlets ‚inside out‘ programmieren

36

## Servlets & JSP

Legende:

HTML

Java

```
writer.print („<html><body>  
Bei diesem Klick war es  
„+new  
Date()+“.</body></html>“);
```

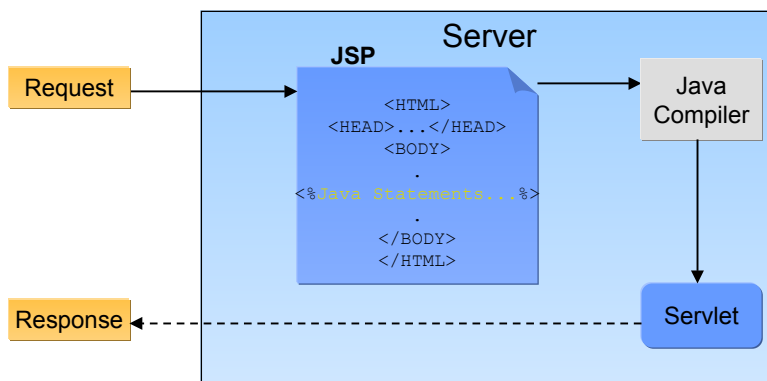
Servlets

```
<html><body>Bei diesem  
Klick war es <%=new Date()  
%>.</body></html>
```

JSP

37

## JavaServer Pages



38

## JSP Elemente

<b>Standard-HTML</b>	→	<code>&lt;html&gt;</code>
<b>JSP Direktive</b>	→	<code>&lt;%@page import = "com.wombat.JSP.*" %&gt;</code>
<b>JSP Action-Tags, z.B. Beans zu laden</b>	→	<code>&lt;body&gt;</code> <code>&lt;h1&gt;Willkommen&lt;/h1&gt;</code> <code>&lt;p&gt;Heute ist der &lt;/p&gt;</code> <code>&lt;jsp:useBean id = "clock" class = "calendar.jspCalendar" /&gt;</code> <code>&lt;ul&gt;</code> <code>&lt;li&gt;Tag: &lt;%= clock.getDayOfMonth() %&gt;&lt;/li&gt;</code> <code>&lt;li&gt;Jahr: &lt;%= clock.getYear() %&gt;&lt;/li&gt;</code> <code>&lt;/ul&gt;</code>
<b>Ausdrücke</b>	→	<code>&lt;% if (Calendar.getInstance().get(Calendar.AM_PM) ==</code> <code>Calendar.AM) {%&gt;</code> <code>    Guten Morgen</code> <code>&lt;% } %&gt;</code>
<b>Scriptlets</b>	→	<code>&lt;/body&gt;</code>
<b>Kommentar</b>	→	<code>&lt;/html&gt;</code> <code>&lt;!-- Kommentar %&gt;</code>

39

## Taglibs

- Erstellen eigener Tags
- Auslagern von Logik in Taglibs
- Vereinfachen von komplexen JSP-Seiten
- Ein Tag - eine TagKlasse
- Syntax:
  - `<Namespace:TagName AttributName= Wert/>` oder
  - `<Namespace:TagName AttributName= Wert>`  
    ...Rumpf...  
    `</Namespace:TagName AttributName= Wert>`
- Semantik wird durch Tag-Klasse (JavaBean) angegeben

40

- JSP-Fragment, welches Taglib verwendet

```
<%@ taglib uri="hellolib.tld" prefix="test"%>
<html>
  <body>
    <test:hello user="Gernot" />
  </body>
</html>
```

- Logik erzeugt direkt die Ausgabe
- Keine Abstraktion von der Programmiersprache

Basisidee:

JSP als HTML-Template mit speziellen Tags ohne Java Code zu erstellen, die dann dynamischen Inhalt liefern können.

Dadurch entsteht eine klare Trennung von Web-Repräsentation und Businessobjekten.

1. JSP-Seite erstellen
2. database und query Tags anwenden und/oder DataModules
3. Mit dem control Tag HTML-Elemente und/oder andere InternetBeansExpress Komponenten umschliessen
4. Skriptlets für Eventhandler schreiben

## database-Tag

```
<ix:database id="database1"
driver="com.borland.datastore.jdbc.DataStoreDriver"
url="jdbc:borland:dslocal:C:\\BeispielDB.jds"
username="test" >
...
</ix:database>
```

45

## query-Tag und table-Tag

```
<ix:query id="aktien" statement="select * from aktien">
  <ix:table dataSet="aktien" >
    <table id="depottable" align="Left" cellspacing="0"
cellpadding="7">
      <tr>
        <th>Name:</th> <th>WKN:</th> <th>Kurs:</th>
      </tr>
    </table>
  </ix:table>
</ix:query>
```

46

## internetbeans.tld

- XML-Datei, welche die InternetBeans Express custom tag library definiert
- besteht aus einer Reihe von Tags um in einer JSP InternetBeans zu benutzen
- dienen als Templates für dynamische Daten

## internetbeans.tld Datei I

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
    PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag
    Library 1.1//EN"
    "http://java.sun.com/j2ee/dtds/web-
    jsptaglibrary_1_1.dtd">

<taglib>
  <tlibversion>1.0</tlibversion>
  <jspversion>1.1</jspversion>
  <shortname>ix</shortname>

<uri>http://www.borland.com/jbuilder/internetbeans.tld</u
ri>
  <info>
    JSP tag extensions for InternetBeans Express
  </info>
```

```
<tag>
  <name>
    database
  </name>
  <tagclass>
    com.borland.internetbeans.taglib.DatabaseTag
  </tagclass>
  <info>Database connection object</info>
  <attribute>
    <name>id</name>
    <required>true</required>
    <rtexprvalue>>false</rtexprvalue>
  </attribute>
</tag>
```

## Tag-Bibliothek im Detail

- enthält sechs Tags, jedoch gibt es 17 InternetBeans Komponenten
- kein Nachteil, denn das control-Tag wird einer IxControl-Komponente zugeordnet, die Informationen an alle anderen steuerungsspezifischen InternetBeans weitergibt

## Tags im Detail I

Name des Tags	Beschreibung	Attribute
<b>database</b>	Definiert eine Database-Komponente in DataExpress	id driver url username password
<b>query</b>	Definiert eine QueryDataSet - Komponente in DataExpress	id database statement
<b>control</b>	Definiert eine IxControl-Komponente in InternetBeans Express	id tupleModel dataSet columnName

51

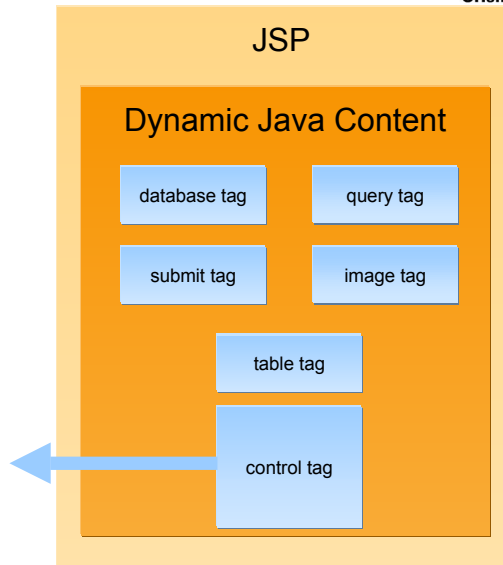
## Tags im Detail II

Name des Tags	Beschreibung	Attribute
<b>image</b>	Definiert eine IxImage-Komponente in InternetBeans Express	id tupleModel dataSet columnName
<b>submit</b>	Definiert eine IxSubmitButton - Komponente in InternetBeans Express	id methodName
<b>table</b>	Definiert eine IxTable-Komponente in InternetBeans Express	id dataSet tableModel

52

## Ix-Schema für JSPs

Kann alle anderen spezifischen Ix-Komponenten ersetzen, z.B. ixCheckbox



53

## JSP-Beispiel IxAktien - Projekt erstellen

Projekt-Experte - Schritt 1 von 3

**Auswahl des Namen und der Vorlage für das neue JBuilder-Projekt**

Geben Sie einen Namen für Ihr Projekt und dessen Verzeichnis ein. Wählen Sie auch ein bestehendes Projekt für die Übernahme der Vorgabeeinstellungen.

Projektname:  Typ:

Projektvorlage und Verzeichnisnamen auswählen

Projekt für Vorgaben:  ...

Stammpfad:  ...

Name des Projektverzeichnisses:

Name des Quellverzeichnisses:

Verzeichnis für Sicherungskopien:

Dokumentations-Verzeichnisname:

Name des Ausgabeordners:

Das Projektverzeichnis ist dem Quell- und Ausgabeverzeichnis übergeordnet

< Zurück Weiter > Fertigstellen Abbrechen Hilfe

54

## JSP-Beispiel IxAktien - JSP-Experte

**JSP-Experte - Schritt 1 von 3**

**JSP-Dateidetails bearbeiten**

Füllen Sie die folgenden Felder aus, um eine JSP-Datei zu erstellen, und wählen Sie die optional zu erzeugenden Komponenten.

WebApp:

Name:

Generieren eines Submit-Formulars

Deklarieren der InternetBean-Tag-Bibliothek, Präfix:

Hintergrund:

Optionale Komponenten:

Generieren der Fehler-Seite

Generieren eines Beispiel-Bean

< Zurück   Weiter >   Fertigstellen   Abbrechen   Hilfe

55

## JSP-Beispiel IxAktien - HTML Teil

```
<%@ page
import="com.borland.internetbeans.*,com.borland.dx.dataset.*,com.borland.d
x.sql.dataset.*" %>

<%@ taglib uri="/internetbeans.tld" prefix="ix" %>

<html>
<head>
<title>
AktienJSP
</title>
</head>
<body>
<h1>
Aktien aus einer DB auslesen
</h1>
```

56

## JSP-Beispiel IxAktien - Ix-Tags

```
<ix:database id="database1"
driver="com.borland.datastore.jdbc.DataStoreDriver"
url="jdbc:borland:dslocal:C:\\InternetBeans_Vortrag\\IxAktien\\Aktien.jds"
username="test" >

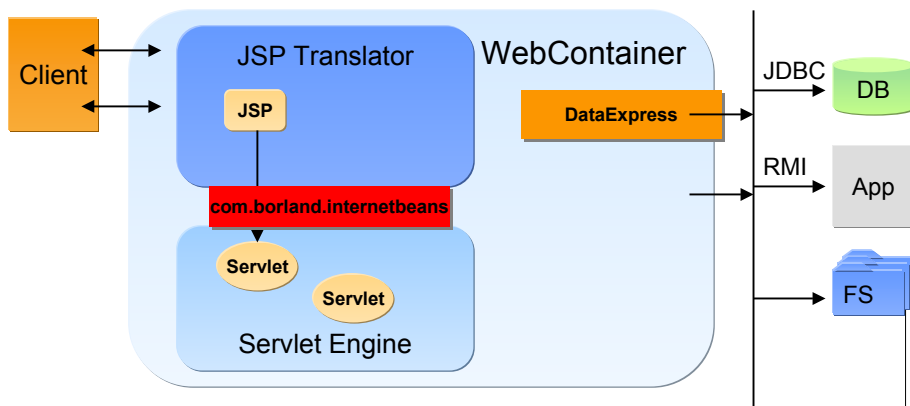
<ix:query id="aktien" statement="select * from aktien">

.....

</ix:query>
</ix:database>
```

57

## Architektur



58

## Fazit

- einfaches, vielversprechendes Konzept
- durch DataExpress ist Kommunikation mit anderen Modellen möglich (z.B. EJB)
- leider gibt es zuwenig praxistaugliche Beispiele
- Dokumentation ist nicht ausreichend

## Literatur und Quellen

- Notes on using InternetBeans Express  
<http://community.borland.com/article/0,1410,26373,00.html>
- InternetBeans Express  
<http://www.drbob42.com/Jbuilder/jbexpr01.htm>
- InternetBeans Express  
<http://www.drbob42.com/Jbuilder/jbexpr02.htm>
- *Java Magazin* 12/2000  
Swing im Web: Hammock und InternetBeans Express
- *Java Magazin* 01/2001  
JBuilder 4.0: neuer Komfort bei Servlets und JSP



**Vielen Dank für die  
Aufmerksamkeit!**