

# X33EJA – Enterprise Java

**Petr Šlechta**  
Sun Microsystems  
[petr.slechta@sun.com](mailto:petr.slechta@sun.com)

# Java Server Pages JSP

# JSP

- Statický text kopírován do výstupu
- Bloky kódu (skriptlety) vykonány a výsledek vložen do stránky (dynamický obsah)
- Inspirováno PHP
- Značky `<% %>` `<%! %>` `<%= %>` `<%@ %>`
- Kompilováno do servletu (většinou při prvním přístupu na stránku)

# Ukázka

- Použití různých značek, ukázka překladu do servletu
- Faktorial pomocí JSP

# Použití JavaBeans (ne EJB!)

- Tagy pro využití JavaBeans
  - `<jsp:useBean>` ... vytvoří instanci JavaBean třídy
  - `<jsp:getProperty>`
  - `<jsp:setProperty>`
- Jiné standardní tagy
  - `<jsp:forward>`
  - `<jsp:include>`
  - `<jsp:plugin>` ... nahraje Java plugin do prohlížeče pro spuštění appletu (či jiného vloženého objektu)

# Ukázka

- Použití JavaBeans
  - Faktorial pomocí JavaBeans
  - Jednoduché oddělení logiky od prezentační vrstvy

# JSP 2.0

- Expression language
  - rozpoznán ve statickém textu a attributech tagu
  - rozšířitelný pomocí tag libraries
- Použití
  - `<%=item.getName()%>`
  - `${item.name}`
  - ```
<% Map m = (Map)pageContext.getAttribute("state");
State s = ((State)m.get("CZ"));
if( s != null ) { %>
    <%= s.getCapitol() %> <%
} %>
```
  - `${state["CZ"].capitol}`

# JSTL – Java Standard Tag Library

- Knihovny dostupné všude
  - portovatelnost aplikací, optimální implementace
- Existují i jiné implementace
  - Jakarta-Taglibs, ...
- JSTL zjednodušují tyto oblasti
  - Core (c): proměnné, podmínky, iterace, podpora práce s URL
  - I18n (ftm): locale, formátování (datum, čísla, ...)
  - DB (sql): přístup k databázím
  - XML (x): přístup k XML datům, transformace
  - Funkce (fn): kolekce, manipulace se stringy, ...



# Použití JSTL v JSP

- Deklarace použití knihovny na začátku JSP
  - `<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>`
  - `<%@ taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml" %>`
  - `<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>`
  - `<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>`
  - `<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>`



# JSTL – Core

- Jiné cykly a iterace
  - `<c:forEach var="customer" items="{customers}">`  
    `<c:if test="{customer.address.country} == 'USA'">`  
        `{customer}<br>`  
    `</c:if>`  
  `</c:forEach>`
  - `<c:forEach var="item" items="{enumeration}"`  
    `begin="2" end="10" step="2">`  
    `<c:out value="{item}"/><br>`  
  `</c:forEach>`
  - `<c:forEach var="token" items="bleu,blanc,rouge">`  
    `<c:out value="{token}"/><br>`  
  `</c:forEach>`

# JSTL – Core

- Rozdělení textu na tokeny
  - `<c:forTokens var="token" items="one,two,three" delims=", ">`  
    `<c:out value="{token}"/>`  
    `</c:forTokens>`
- Složitější výstupy
  - `<c:out value="{customer.phoneCell}" escapeXml="false">`  
    `<font color="red">no cell phone specified</font>`  
    `</c:out>`
  - `<c:out value escapeXml default="def. value"/>`
- Je-li u `c:out` parametr `java.io.Reader`, přečtou se z něj data a vloží se do výstupu

# JSTL – Core

- Větvení (switch)

- `<c:forEach var="customer" items="${customers}">`  
    `<c:choose>`  
        `<c:when test="${customer.address.country == 'USA'}">`  
            `<font color="blue">`  
        `</c:when>`  
        `<c:when test="${customer.address.country == 'Canada'}">`  
            `<font color="red">`  
        `</c:when>`  
        `<c:otherwise>`  
            `<font color="green">`  
        `</c:otherwise>`  
    `</c:choose>`  
    `${customer}</font><br>`  
`</c:forEach>`

# JSTL – Core

- Podpora práce s URL
  - `<c:import url="header.jsp">`  
    `<c:param name="pageTitle" value="myComp.com"/>`  
    `<c:param name="pageSlogan" value=" " />`  
    `</c:import>`
  - `<c:url value="base">`  
    `<c:param name="param" value="ABC"/>`  
    `</c:url>`
- Pokud nejsou povoleny cookies, tak URL bude obsahovat správně zakódovanou session
  - `base?param=ABC`
  - `base;jsessionId=E5...1F?param=ABC`

# JSTL – DB

- Pro rychlé prototypování aplikací
- Operace nad DataSource objekty
  - `<sql:dataSource var="dataSource" driver="org.gjt.mm.mysql.Driver" url="jdbc:..."/>`
  - `<sql:setDataSource var="example" driver="com.pointbase.jdbc.jdbcUniversalDriver" url="jdbc:pointbase:server://localhost:1092/jstlsample;create=true"/>`

# JSTL – DB

- Transakce
  - `<sql:transaction dataSource="${example}">`  
    `<sql:update var="newTable">`  
        create table mytable (nameid int primary key,  
                                name varchar(80))  
    `</sql:update>`  
    `<sql:update var="updateCount">`  
        INSERT INTO mytable VALUES (1,'Paul Black')  
    `</sql:update>`  
  `</sql:transaction>`



# JSTL – XML

- Pro přístup k datům z XML dokumentu (pomocí XPath výrazů)
  - `/games/athlete[sport="sailing"]...` tagy athlete s vnořeným tagem sport, který má hodnotu "sailing"
  - `country[@id="CZ"]` ... country tagy s atributem id rovno "CZ"
  - `/games/country//sport` ... všechny sport elementy z podstromu `/games/country`
- Automatické konverze v Xpath
  - node-set -> boolean, number, string
- XPath výraz může obsahovat i funkce
  - `count(node-set), ...`

# JSTL – XML

- Ukázka přístupu k datům z XML
  - `<c:set var="xmlText">`  
    `<a>`  
        `<b><c>foo</c></b>`  
        `<d>bar</d>`  
    `</a>`  
    `</c:set>`
  - `<x:parse var="a" doc="{xmlText}" />`  
    `<x:out select="$a//c"/>`  
    `<x:out select="$a/a/d"/>`
- `x:out` funguje podobně jako `c:out` (navíc XPath konverze)

# JSTL – Funkce

- Práce s řetězci
  - `<fn:length>` délka stringu
  - `<fn:toUpperCase>`, `<fn:toLowerCase>`
  - `<fn:substring>`, `<fn:substringBefore>`,  
`<fn:substringAfter>`
  - `<fn:trim>`, `<fn:replace>`
  - `<fn:indexOf>`, `<fn:startsWith>`, `<fn:endsWith>`,  
`<fn:containsIgnoreCase>`
  - `<fn:split>`, `<fn:join>` rozděl string do pole, spoj  
kolekci do stringu
  - `<fn:escapeXml>` escapování XML znaků ve stringu

# JSTL – I18n a formátování

- Z requestu lze zjistit preferované locale uživatele (ukázka nastavení ve Firefoxu)
- Nastavení locale
  - `<fmt:setLocale>` .. přepíše nastavení klienta
  - `<fmt:requestEncoding>` ... přepíše encoding klienta
- Lokalizace textů
  - `<fmt:bundle>` ... resource bundle pro stránku
  - `<fmt:message key="...">` s `<fmt:param>` podtagem `<fmt:setBundle>`

# JSTL – I18n a formátování

- Formátování čísel a datumu
  - `<fmt:formatNumber>`, `<fmt:parseNumber>`
  - `<fmt:formatDate>`, `<fmt:parseDate>`  
`<fmt:setTimeZone>`, `<fmt:timeZone >`
- Příklad použití
  - `<fmt:setLocale value="de"/>`  
`<fmt:bundle basename="cz.myapp.resources">`  
    `<fmt:message>greetingMorning</fmt:message>`  
`</fmt:bundle>`

# Ukázka

- JSP stránka využívající JSTL
  - core, XML tagy
  - podpora lokalizace
- Použití JSTL z NetBeans
  - palety
  - doplňování / nápověda

# Uživatелеm definované tagy

- Oddělit prezentaci od business logiky
- Custom tags vs. JavaBeans
  - uživatelem definované tagy mají přístup k více informacím než JavaBeans
- Uživatелеm definované tagy
  - jsou sdružené v knihovnách
  - tagu lze nastavit proměnné (jako beanu), tag může nastavovat proměnné stránky v aplikačním, session a page contextu
  - tag má přístupné všechny objekty jako JSP stránka (request, ...)
  - tagy lze zanořovat

# Uživatelsky definované tagy

- Tag handler třída
  - implementace vlastního tagu
- Tag library descriptor (TLD)
  - XML soubor popisující tagy v dané knihovně
  - knihovna tagů je součástí web aplikace
- Použití v JSP stránce
  - mapování knihovny přes prefix a uri (stejně jako u JSTL)



# Uživatелеm definované tagy

- Jednoduchý tag bez těla
  - TagSupport, doStartTag() vrací SKIP\_BODY
- Tag s tělem
  - doStartTag() vrací EVAL\_BODY\_INCLUDE
  - doEndTag() se zavolá po vyhodnocení těla
  - vrátit EVAL\_PAGE pro pokračování
  - vrátit SKIP\_PAGE pokud se zbytek stránky nemá vyhodnocovat
- Složitější tag
  - BodyTagSupport

# Ukázka

- Definice a použití uživatelem definovaných tagů
  - různé typy tagů
  - podpora v NetBeans

# Jiné reference na knihovny tagů

- web.xml
  - `<jsp-config>`
    - `<taglib>`
      - `<taglib-uri>/tlt</taglib-uri>`
      - `<taglib-location>/WEB-INF/lib.tld</taglib-location>`
    - `</taglib>`
  - `</jsp-config>`
  - `<%@ taglib prefix="tlt" uri="/tlt"%>`
- Přímá reference
  - `<%@ taglib prefix="tlt" uri="/WEB-INF/iterator.tld"%>`
- Absolutní reference
  - `<%@ taglib prefix="core"`  
`uri="http://java.sun.com/jsp/jstl/core"%>`

# Tag files

- Definice JSP fragmentu, který lze používat jako uživatelsky definovaný tag
  - překládá se na tag handler (není potřeba TLD)
  - .tld soubory do /WEB-INF/tags/
- Ukázka podpory Tag souborů v NetBeans

# Další možnosti konfigurace

- Centrální konfigurace pomocí url-patternů
  - Mapování jiných přípon než .jsp
  - Povolení / zakázání skriptletů nebo EL globálně
  - Nastavení page encoding globálně
  - Prelude / coda definice

# JSP – shrnutí

- Dostatečná funkcionality a podpora pro tvorbu web aplikací
- Mnoho web frameworků využívá JSP jako prezentační vrstvu
  - Frameworky většinou definují vlastní specifické tagy
- Web frameworky ještě více zjednodušují a automatizují tvorbu web aplikací
  - Kontrola vstupů, podpora pro zadávání dat
  - Objektový model stránky, komponenty, MVC
  - ...