

Java, technologie zaawansowane

JSP

JSP (JavaServer Pages)

- Dynamiczne strony www
 - „Podobieństwo” do PHP
 - Dużo rozszerzeń
 - JSF, JSTL
- Unified EL
- Język
 - Mieszanka jsp i Javy
 - tradycyjne jsp
 - xml

Fazy w JSP

- Tworzenie
 - Deploy
 - Walidacja składni
 - Translation
 - Wykonywane raz dla strony
 - Tłumaczenie jsp na Javę i kompilacja
 - Generowanie odpowiednich Servletow
 - Obsługa połączeń
 - events -> request -> reply

Przykład

pierwsza.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<html>
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8">
    <title>Ala ma Kotka</title>
  </head>
  <body>
    <b>Jakiś tekst :-)</b>
  </body>
</html>
```

Składnia jsp

tradycyjny jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
```

xml jsp

```
<?xml version="1.0" encoding="UTF-8" ?>
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.0">
<jsp:directive.page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" />
```

tradycyjny jsp

```
<%--' JSPCommentBody
<%@' DirectiveBody
<%!' DeclarationBody
<%= ' ExpressionBody
<% ' ScriptletBody
${ ' EL ExpressionBody
```

xml jsp

```
<jsp:text' XMLTemplateText
<jsp:directive.' XMLDirectiveBody
<jsp:declaration' XMLDeclarationBody
<jsp:expression' XMLExpressionBody
<jsp:scriptlet' XMLScriptletBody
#{ ' EL ExpressionBody
```

Zakresy zmiennych na serwerze

- Dane na serwerze przechowywane są w 4 zakresach:
 - Application
 - Page
 - Request
 - Session

Zmienne Scriptletów

- Zmienne dostępne w Scriplecie:
 - Application Scope
 - application
 - Page Scope
 - config, exception (Error Page), pageContext, page
 - out, response
 - Request Scope
 - request
 - Session Scope
 - session

Standardowe rozkazy JSP

- `<jsp:include>`
 - Dołącz inny plik jsp
- `<jsp:forward>`
 - Prześlij do innego jsp (zakończenie pracy tego jsp)
- `<jsp:plugin>`
 - Wstaw obiekt, flash, Java Applet,...
- `<jsp:param>`
 - Ustawienie parametrów startowych dla powyższych znaczników
- `<jsp:useBean>`
 - Fasolki Java

Parametry startowe

- Parametry startowe
 - Deklarowane w web.xml

```
<context-param>  
  <param-name> ... </param-name>  
  <param-value>....</param-value>  
</context-param>
```
- – Dostęp:
 - ServletContext
 - `getInitParameter(name)`

Parametry startowe cd.

- Zastosowanie
 - Przechwywanie danych dla całej aplikacji
 - np. Hasła dostępu
- Zalety
 - Bezpieczeństwo
 - Nie widoczne dla użytkownika
 - WEB_INF
 - Dostęp TYLKO dla serwera i aplikacji

Fasolki Jawy

- Java Beans
 - Klasy POJO
 - Metody get/set definiują własności fasoli
 - Można używać jako zmiennych sesyjnych, requesta,...
 - Można wypełnić danymi przesłanymi w parametrach

Biblioteki znaczników

- Istnieje wiele bibliotek znaczników
 - JSTL (JavaServer Pages Standard Tag Library)
- Dołączamy podobnie
 - `<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>`
- JSTL
 - Dodatkowe znaczniki:
 - `forEach`, `catch`, `choose`, `import`, `url`
 - `Functions`, `SQL`, operacje na XML'ach

Obsługa błędów

- Wskazanie pliku

- W web.xml

- ```
<error-page>
```

- ```
    <exception-type>java.lang.Exception
```

- ```
 </exception-type>
```

- ```
    <location>/blad.jsp</location>
```

- ```
</error-page>
```

- W pliku jsp

- ```
<@page ... errorPage="/blad.jsp" ... %>
```

Plik obsługi błędów

- Tradycyjny jsp z deklaracją
 - `<% page isErrorPage="true" %>`
 - Wtedy możemy użyć np.
 - `<%= exception.getMessage() %>`
- Poprzez Servlet
 - W web.xml
 - `<error-page>`
 - `<error-code>404 </error-code>`
 - `<location>/errors</location>`
 - `</error-page>`
 - Obsługa w odpowiedniej klasie Servlet z mapowaniem