

高度情報処理演習B

Java Servlet/Java Server Pages

Servlet/JSP(再掲)

- Servlet?
 - Javaでウェブアプリケーションをつくるための仕組み
 - Javaのクラスで作成する
 - 最新 - Servlet 3.0 (2011/8)
- JSP?
 - HTMLの中にJavaでロジックを組み込む
 - 初回アクセス時にServletにコンパイルされる
 - 最新 - JSP 2.2 (2011/8)

Servlet/JSPの使い分け

- Servlet
 - HTML文書もすべてJavaで記述
 - 手の込んだ画面(View)の作成が困難
- JSP
 - HTMLページの中にJavaのコードを埋め込む
 - 処理が必要な部分だけプログラムを記述

ビジネスロジック → Servlet
画面(ビュー) → JSP

JSPの例

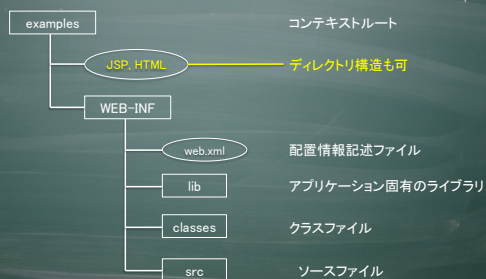
```
<%@ page contentType="text/html;charset=utf-8" %>

<html>
<head>
<title>Hello World!</title>
</head>
<body>
<h1>Hello World!</h1>

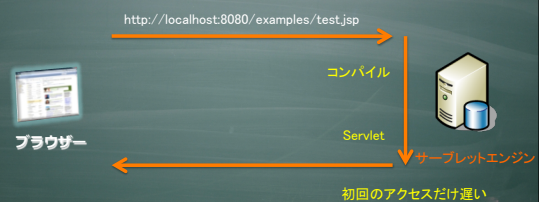
<%
  out.println(new java.util.Date());
%>

</body>
</html>
```

JSPの配置



JSP実行のプロセス



JSPの基本

- ディレクティブ
 - page, include, taglib
- スクリプティング
 - スクリプトレット, 式の評価, 変数の宣言
 - メソッドの宣言, コメントの記述
- アクションタグ
 - ここでは省略
- 暗黙オブジェクト

ディレクティブ

- JSPに関する制御文を記述する
 - **pageディレクティブ**
 - includeディレクティブ
 - 他のJSPをインクルードする場合
 - taglibディレクティブ
 - タグライブラリを利用する場合

pageディレクティブ

設定したい項目だけpageディレクティブに設定する

```
<%@ page
[ language="java" ]
[ extends="package.class" ]
[ import="{package.class | package.*}, ..." ]
[ session="true|false" ]
[ buffer="none|8kb|sizekb" ]
[ autoFlush="true|false" ]
[ isThreadSafe="true|false" ]
[ info="text" ]
[ errorPage="relativeURL" ]
[ contentType="mimeType [ ; charset=characterSet ]" |
"text/html ; charset=ISO-8859-1" ]
[ isErrorPage="true|false" ]
[ pageEncoding="characterSet | ISO-8859-1" ]
[ isELIgnored="true|false" ]
%>
```

pageディレクティブ詳細1/4

- language
 - JSPで使用する言語を指定する。現状ではJavaしか指定できないため設定する必要はない
- extends
 - JSPを他のサブクラスにする場合、スーパークラスを指定する
- import
 - Javaのクラスをインポートするために利用する。以下のパッケージはimportしなくても利用可能
 - java.lang.*
 - javax.serv.et.*
 - javax.servlet.jsp.*
 - javax.servlet.http.*

pageディレクティブ詳細2/4

- session
 - セッションの有効/無効を切り変える。デフォルトはtrue
- buffer
 - 出力データをバッファしてから返すかどうか指定する
- autoFlush
 - バッファがいっぱいになった場合クライアントに送信するか否かを指定する。デフォルトはtrue
 - java.lang.*
 - javax.serv.et.*
 - javax.servlet.jsp.*
 - javax.servlet.http.*

pageディレクティブ詳細3/4

- isThreadSafe
 - スレッドセーフで実行するかどうか指定。デフォルトはtrue
- info
 - JSPページに関する任意の情報を記述する
- errorPage
 - エラーが発生した場合に表示するページへのパスを指定する
- isErrorPage
 - JSPページがエラー表示用のページかどうか指定する。デフォルトはfalse.

pageディレクティブ詳細4/4

- **contentType**
 - JSPページのデータ型と文字コードを指定する
- **pageEncoding**
 - JSPページの文字コードを指定する。デフォルト値はcontentTypeで指定した文字コードとなるため、あえて指定は不要
- **isElIgnored**
 - EL式を適用するか否かを指定する。デフォルトはfalse

補足) pageディレクティブ-import

- 次のpageディレクティブはいずれも有効

```
<%page contentType="text/html;charset=utf-8" import="java.util.Date" %>
```

同一のディレクティブに異なる種類を指定

```
<%page import="java.util.*, java.io.*" %>
```

importする複数のパッケージをカンマ区切り

```
<%page import="java.util.*" %>
```

```
<%page import="java.io.*" %>
```

importする複数のパッケージ複数行に分けて指定

スクリプティング

- **スクリプトレット**
- **式の評価を出力**
- **変数の宣言**
- **メソッドの宣言**
- **コメントの記述**

スクリプトレット

- Javaのコードを記述する部分

- **<% コード %>**

```
<%@ page contentType="text/html;charset=Shift_JIS" %>
<html>
<%
int sum = 0;
for (int i = 0 ; i <= 10 ; i++){
    sum = sum + i;
}
%>
1から10まで加えた合計の値は

<%
out.println(sum);
%>
です。
</body>
</html>
```

式の評価を出力

- 式の評価(変数の値など)を出力するための書式

- **<%= 式 %>**

```
<%@ page contentType="text/html;charset=Shift_JIS" %>
<html>
<%
int sum = 0;
for (int i = 0 ; i <= 10 ; i++){
    sum = sum + i;
}
%>
1から10まで加えた合計の値は

<%= sum %>

です。
</body>
</html>
```

変数/メソッドの宣言

- 変数やメソッドをJSP内に定義し呼び出すことができる

- **<%! 変数やメソッド %>**

- 変数はJSPページの静的変数になる(2度目に呼び出されても値を保持)

```
<%!
int count = 0;
int getTotal(int num){
    int sum = 0;
    count++;
    for (int i = 1 ; i <= num ; i++){
        sum += i;
    }

    return sum;
}
%>
```

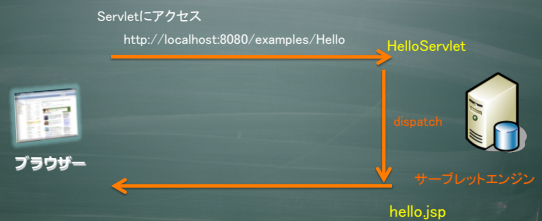

Servlet/JSPの使い分け(再掲)

- Servlet
 - HTML文書もすべてJavaで記述
 - 手の込んだ画面(View)の作成が困難
- JSP
 - HTMLページの中にJavaのコードを埋め込む
 - 処理が必要な部分だけプログラムを記述

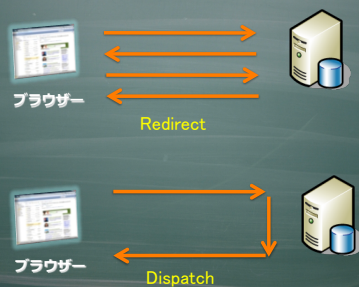
↓

ビジネスロジック → Servlet
画面(ビュー) → JSP

ServletとJSPの連携



補足) RedirectとDispatch



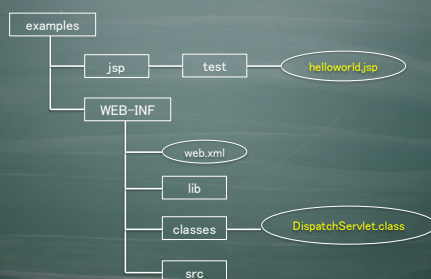
最も簡単なServlet/JSP連携

```

public class DispatchServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        RequestDispatcher dispatcher
            = request.getRequestDispatcher("/jsp/test/helloworld.jsp");
        dispatcher.forward(request, response);
    }
}
  
```

Dispatch先の指定
Dispatchの実行

Servlet/JSPの配置



JSPに値を渡す - Servlet

- `HttpServletRequest.setAttribute(String key, Object o)`

```

public class HowAreYouServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        RequestDispatcher dispatcher
            = request.getRequestDispatcher("/jsp/test/howareyou.jsp");
        request.setAttribute("name", "Michael Jackson");
        dispatcher.forward(request, response);
    }
}
  
```

JSPに値を渡す - JSP

- `HttpServletRequest.getAttribute(String key)`

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    String name = (String)request.getAttribute("name");
%>

<html>
<head>
<title>How are you</title>
</head>
<body>
    こんにちは <%= name %> さん
</body>
</html>
```

JSPに任意のオブジェクトを渡す

- 任意のオブジェクト - `OriginalObject`

```
public class OriginalObject {
    private String name;
    private int age;

    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}
```

JSPに任意のオブジェクトを渡す

- `DispatchObjectServlet`

```
public class DispatchObjectServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {

        RequestDispatcher dispatcher
            = request.getRequestDispatcher("/jsp/test/myobj.jsp");

        OriginalObject myObj = new OriginalObject();
        myObj.setName("Taro");
        myObj.setAge(20);

        request.setAttribute("myobj", myObj);

        dispatcher.forward(request, response);
    }
}
```

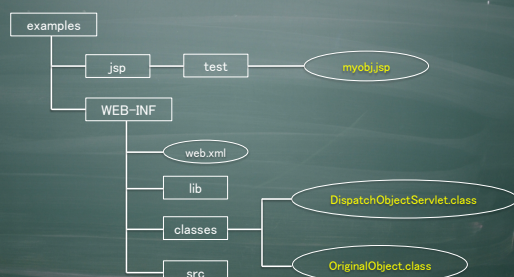
JSPに任意のオブジェクトを渡す

- `myobj.jsp`

```
<%@ page contentType="text/html; charset=utf-8" %>
<%@ page import="jsperv.bean.*" %>
<%
    OriginalObject myobj = (OriginalObject)request.getAttribute("myobj");
%>

<html>
<head>
<title>How are you</title>
</head>
<body>
    こんにちは <%= myobj.getName() %>さん<br/>
    <%= myobj.getAge() %>さいですね。
</body>
</html>
```

Servlet/JSPの配置



演習

- ServletとJSPを連携して会員登録アプリケーションを作成する
- 条件
 - POSTメソッドを利用すること
 - すべてServlet-JSP連携で実装すること
 - URLは任意でよい

