

Webアプリケーション開発演習A

サーブレットとJSP

情報システム専攻科2年

チェック ☒

今日の目標!

サーブレットを理解する

- JSPは「HTML + Java」なのに対してサーブレットはJavaオンリー
- サーブレットの決まりごと
 - HttpServletクラスから派生する
 - URLとサーブレットの関連付けは@WebServletで行う
 - POST通信の時はdoPost、GET通信の時はdoGetが呼ばれる



今日の目標！

サーブレットからJSPを呼び出す



目次

サーフレットからJSPを呼び出す 演習

A decorative blue header at the top of the slide, featuring various white geometric shapes like circles, squares, and diamonds, some of which are nested or overlapping.

目次

サーブレットからJSPを呼び出す 演習

サーブレットからJSPを呼び出す

前回記述したサーフレットのソース

```
@Override
protected void doGet(
    HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {

    String[] luckArray = {"超す", "勝つ", "負ける", "引当", "引き分け"};
    int index = (int)(Math.random() * luckArray.length);
    String luck = luckArray[index];

    Date date = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");
    String today = sdf.format(date);

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    out.println("<html>");
    out.println("<head>");
    out.println("<title>スッキリ占い</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<p>" + today + "の運勢は" + luck + "です</p>");
    out.println("</body>");
    out.println("</html>");
}
```

この部分はHTMLを出力しています

この部分はHTMLを出力している

サーブレットからJSPを呼び出す

**Javaのコードの中にHTMLを出力する
コードがあるのは効率が悪い**



**なんとか、サーブレットはJavaのコードだけ
HTMLの部分はJSPと言う風に
住み分けができないか??**



サーブレットからJSPを呼び出す

そこで登場するのが今回の
サーブレットからJSPを呼び出す！
です！

サーフレットからJSPを呼び出す

やってみよう!

では、実際にやってみましょう!

<https://github.com/nishino-naoyuki/2019Web>
・サーフレットJSP連携サンプル.pdf





サーブレットからJSPを呼び出す

**どういふ動きだったのかを
解説します**

サーブレットからJSPを呼び出す

リクエスト

<http://localhost:8080/servletsample2/servletjsp>



JspSample
Servlet

転送

hello.jsp

レスポンス

```
1 |
2 | <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd">
3 | <html>
4 | <head>
5 | <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 | <title>HELLO</title>
7 | </head>
8 | <body>
9 | <p>HELLO SERVLET-JSP</p>
10 | </body>
11 | </html>
12 |
```

サーブレットからJSPを呼び出す

「転送」ってどういうこと？



サーブレットからJSPを呼び出す

表示した時のURLを見てみましょう

```
http://localhost:8080/servletsample2/servletjsp
```

何か不思議に思う点はありませんか？

hello.jspを表示したのにURLに
hello.jspが**入っていない！**

サーブレットからJSPを呼び出す

リクエスト

<http://localhost:8080/servletsample2/servletjsp>



JspSample
Servlet

転送

hello.jsp

転送とは処理引き継がせること。
この場合サーブレット
では処理せず
hello.jspに処理を引
き継がせている。

レスポンス

```
1 |  
2 |<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd">  
3 |<html>  
4 |<head>  
5 |<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
6 |<title>HELLO</title>  
7 |</head>  
8 |<body>  
9 |<p>HELLO SERVLET-JSP</p>  
10|</body>  
11|</html>  
12|
```

サーブレットからJSPを呼び出す

サーブレットで転送の処理を行っているのが↓のコード

```
RequestDispatcher dispatcher =  
    request.getRequestDispatcher("WEB-INF/jsp/hello.jsp");  
dispatcher.forward(request, response);
```

JspSampleServletに記載します

重要! フレットからJSPを呼び出す

転送先を指定する。

【注意】

ここには**WebContent**フォルダからの**相対パス**を記載する

```
RequestDispatcher dispatcher =  
    request.getRequestDispatcher("WEB-INF/jsp/hello.jsp");  
dispatcher.forward(request, response);
```

転送処理を実行する

**この転送処理は今後かない出てきます。
しっかり覚えましょう**



サーブレットからJSPを呼び出す

サーブレットからJSPへ処理を転
送する~~メリット~~って何？

サーブレットからJSPを呼び出す

**サーブレット・・・複雑な処理をJavaでスッキリ書く
(JSPに複雑な処理を書くと見づらい)**

**JSP・・・HTML(画面)に関しては、JSPに書く
ほうが見やすい**

(サーブレットにHTML出力の処理を書くと見づらい)

**役割分担を明確にすることでソースコード
がスッキリする！(メンテナンス性UP!)**

サーブレットからJSPを呼び出す

ちなみに・・・

**JSPを直接サーブレットを介さず表示
しようとするとうなるでしょうか？**

サーブレットからJSPを呼び出す

hello.jspがあるフォルダは
どこにあるかと言うと・・・



WebContent

META-INF

WEB-INF

jsp

hello.jsp

lib

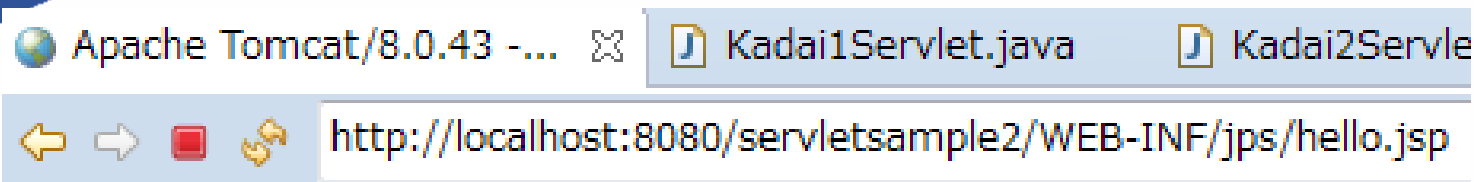
WebContentがWebアプリケーション名以下に続くURLの基準となるフォルダ
(これをルートという)

ルート(WebContent)を基準に考えるとhello.jspは
WEB-INF/jsp/hello.jsp
の場所にある！

http://localhost:8080/servletsample2/WEB-INF/jsp/hello.jsp

サーブレットからJSPを呼び出す

しかし...



HTTPステータス 404 -

type ステータスレポート

メッセージ

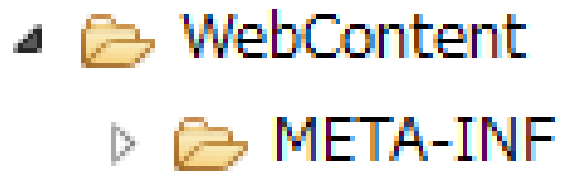
説明 The requested resource is not available.

Apache Tomcat/8.0.43

404(そんなファイル無いよエラー)

サーブレットからJSPを呼び出す

**実はWEB-INF以下のフォルダは
公開フォルダではありません。**



WebContent
META-INF



WEB-INF
jsp
hello.jsp
lib

このエリアはブラウザから直接
リクエストできない。
サーバー内部からのアクセス
(つまり転送)はOK！

重要!

フレットからJSPを呼び出す

つまり...

WebContent

META-INF

WEB-INF

jsp

hello.jsp

lib

0_500-.jpg

hello.jspは、WEB-INF
より下の階層にあるので
直接アクセス不可

0_500-.jpgは、
WebContent直下にある
ので**直接アクセス可能**

A decorative blue header at the top left of the slide, featuring various white geometric shapes like circles, squares, and diamonds, some of which are nested or overlapping.

目次

サーブレットからJSPを呼び出す 演習

やってみよう!

演習

演習でこのやり方に慣れましょう！

**<https://github.com/nishino-naoyuki/2019Web>
・サーフレット課題1.pdf**

完成したらメールで提出



まとめ

今日の目標!

サーブレットからJSPを呼び出す

- サーブレットからJSPを呼び出すのは「転送」を使う

```
RequestDispatcher dispatcher =  
    request.getRequestDispatcher("WEB-INF/jsp/hello.jsp");  
dispatcher.forward(request, response);
```

- WEB-INF以下のフォルダは公開されていない
サーバーからのみアクセスできるフォルダ