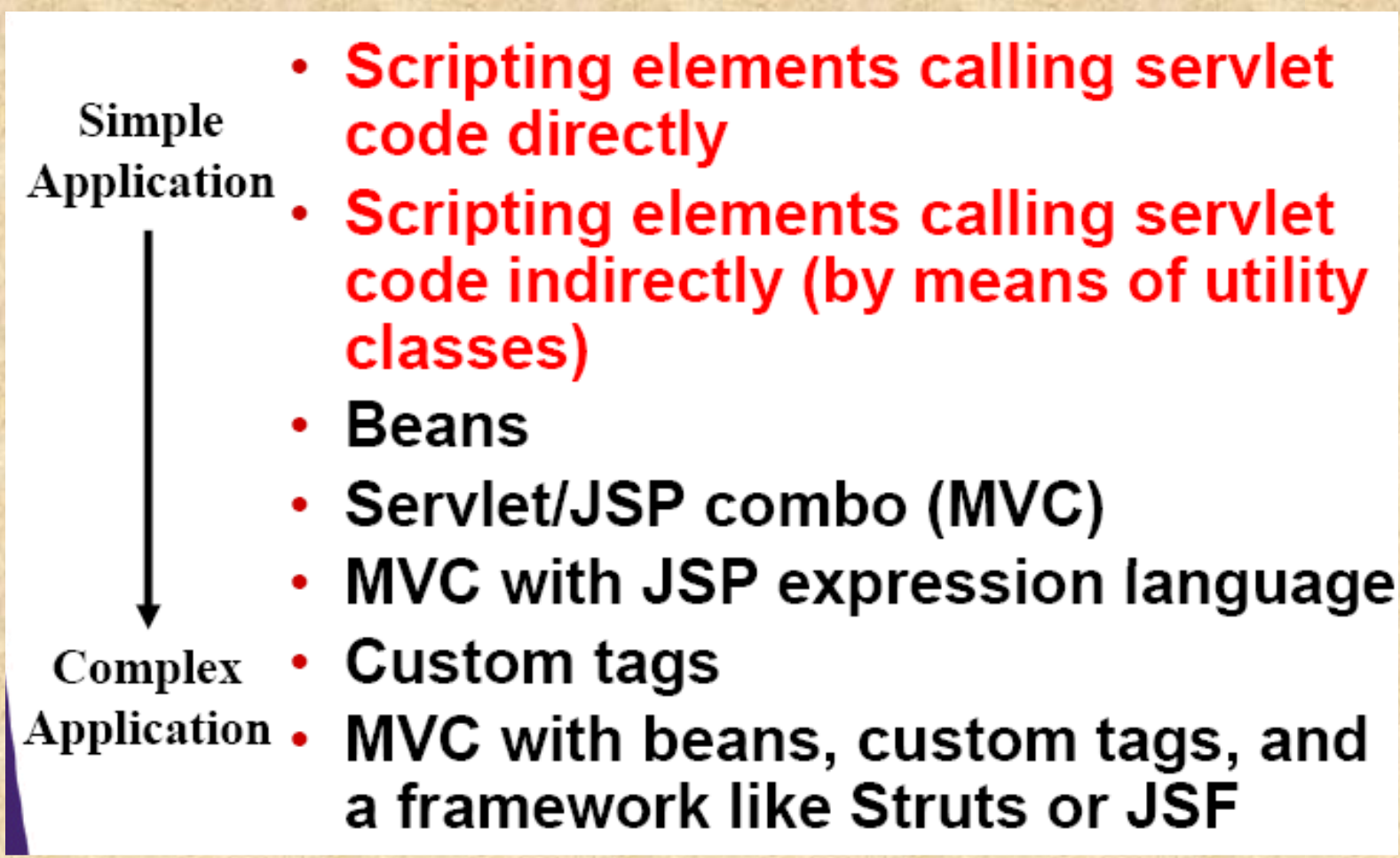
Invoking Java Code from JSP



Invoking Java Code from JSP

Types of JSP Scripting Elements

JSP Scripting elements let you insert Java code into the servlet that will be generated from the JSP page. There are three forms.

1) Expressions <%= Java Expressions %>
2) Scriptlets <% Java code %>
3) Declarations <%! Field/Method Declaration %>

Limiting the amount of Java Code in JSP

Using different kinds of packages we can limit the number lines of program in JSP. Because of followings:

1) **Development**
2) **Compilation**
3) **Debugging**
4) **Division of Labor**
5) **Testing**
6) **Reuse**

Limiting the amount of Java Code in JSP

The Importance of Using Packages

Whenever we write Java classes, the classes files are deployed in WEB-INF/classes/packages.

However, with code called from JSP, you should always use packages. And, since when you write a utility for use from a servlet.

Using JSP Expressions

A JSP expressions are used to insert values directly into the output. <%= Java Expression %>

The expression is evaluated, converted to String, and inserted in to the page. This evaluation is performed at runtime. Ex:

Current Time: <%= new java.util.Date() %>

Predefined Variables

One can use a number of predefined variables. The system simply tells you what name it will use for local variables in _jspService. The most important ones are:

request

response

session

out

application

JSP/Servlet Correspondence

The JSP expression basically become print statement in the servlet that results from the JSP page.

The simplified definition of the out variable: out in a JSP page is a JspWriter, So you have to modify the slightly simpler PrintWriter that directly returns from a call to getWriter.

Tomcat autogenerated source code

localhost/webapp/name

Using Scriptlets to make part of the JSP page conditional

An another use of scriptlets is to conditionally output HTML or other content that is not within any JSP tag. The key approaches are

1) The code inside the scriptlet are get inserted into servlet's _jspService method
2) Any static HTML text before or after a scriptlet gets converted to print statement.

The import attribute

Basically, this attribute of the page directive imports the java packages and it's classes more and more. You can import more than one java packages and classes by separating with comma (,). You can set the name of the class with the package name directly like **packagename.classname** or import all classes of the package by using **packagename.***.

Content type & pageEncoding Attributes

The contentType Attribute sets the Content-Type response header, indicating the MIME type of the document being sent to the client. The content type attribute can set as follows:

<%@ page contentType="text/plain" %>

<%@ page contentType="application/vnd.ms-excel" %>

The pageEncoding attribute is used to change the character set.

<%@ page pageEncoding ="Shift_JIS" %>

Generating Excel Spreadsheet

The page that uses the contentType attribute and tab separated data to generate excel output: Note that the page directive are at the bottom so the carriage returns at the end of the lines.

First	Last	e-mail
SVM	Udg	xyz@yahoo.com
MUM	Udg	mno@gmail.com

```
<%@ page contentType="application/vnd.ms-excel" %>
```

The Session Attribute

The Session Attribute controls whether the page participates in HTTP sessions. Use of this attribute takes one of the following two forms:

<%@ page session="true" %>

<%@ page session="false" %>

The value true (default) signifies that the predefined variable session (HttpSession) should be bound to the existing session if one exists.

The session="false" may save significant amount of server memory on high traffic sites. It does not disable the session tracking it merely prevents from creating new sessions for users.

The isELIgnored Attribute

This controls whether the JSP Expression Language (EL) is ignored (True) or evaluated normally (False).

Use of the attribute takes one of the following forms:

<%@ page isELIgnored="false" %>

<%@ page isELIgnored="true" %>

JSP 2.0 introduced a concise expression language for accessing request parameters, cookies, HTTP headers, bean properties etc.

Alternatively, the isELIgnored page directive attribute, or the <el-ignored> configuration element can be used to deactivate EL for entire translation units.

The errorPage & isErrorPage Attribute

The errorPage attribute specifies a JSP page that should process any exceptions thrown but not caught in the current page. It is used as follows:

<%@ page errorPage="URL" %>

The error page will automatically be available to the designated error page by means of the *exception* variable.

The isErrorPage indicates whether or not the current page can act as the error page for another JSP page. It is used as follows:

<%@ page isErrorPage="true" %>

<%@ page isErrorPage="false" %> default