

Exercises: Including Files and Applets

Problem #1 is quite easy, and is the most important one of the group: it is how you most commonly use `jsp:include`. Problems 2 and 3 look at more advanced but much less commonly used features of `jsp:include`. Problems 4 and 5 don't even use `jsp:include` at all, and are thus the least important of all. Remember that with `jsp:include` in general, the main page (the one that you access on the address line) goes in `WebContent`, but the included pages (the ones that main page refers to, but that users don't directly know about) normally go in `WEB-INF`.

1. Make an HTML “signature” block with your name and email address. Include it in two JSP pages.
2. The value of the `page` attribute of `jsp:include` is allowed to be a JSP expression. That is, you are permitted to do this:

```
<jsp:include page="<%= someVariable %>" />
```

Use this idea to make a JSP page that includes a “good news” message or a “bad news” page at random. You don't need `jsp:include` to just output a random message; the idea here is that you are computing (at random) which *page* to include in the main page.
3. Suppose that you have two different JSP pages that do two different things. However, for both pages you want to let the user supply a `bgColor` attribute to set the background color of the page. Implement this, but use an include mechanism to avoid repeating code. For example:
White background: `http://host/path/page1.jsp`
White background: `http://host/path/page2.jsp`
Red background: `http://host/path/page1.jsp?bgColor=RED`
Yellow background: `http://host/path/page2.jsp?bgColor=YELLOW`
For testing, I do not care if you write an HTML form to collect the `bgColor` parameter or if you simply attach it onto the end of the URL “by hand.”
4. Make two separate JSP pages that have bulleted lists containing random ints in a certain range. Avoid repeating code by including a page that defines a `randomInt` method.
5. If you are familiar with applets, make a trivial one that does nothing but set the background color to blue and print a string derived from the `MESSAGE` parameter embedded in the HTML by means of a `PARAM` element. Convert it to a version that uses the Java Plug-In. Note that, if this runs at all, it proves that you are correctly accessing the Plug-In. You don't need to use `Swing` or `Java2D` to verify that you are using the Plug-In, since the tag generated by `jsp:plugin` is incompatible with the standard virtual machine used by Firefox and IE. Try both Firefox and Internet Explorer to see which (if any) of them has the Plug-In installed. Reminder: applets run on the client, not on the server. So your applet's `.class` files can't go in the server's `WEB-INF/classes` directory. These `.class` files work the same way as for regular applets: they go in the same directory as the JSP/HTML file that uses the applet tag. This is nothing specific to JSP, but is just the normal way applets work.