

Exercises: Ajax Custom Tags

Here are some more things to practice with custom tags, but this time they focus on Ajax. Note that the solutions are already in your Eclipse workspace. Also note that the tags will not generate the complete JavaScript code, but rather it will generate the HTML and the *calls* to the JavaScript functions that your separate file will define.

1. Make a Java-based tag that creates a simple alert button. When you press the button, the browser should contact some relative URL (specified in the tag) and pop up the result in an alert box. Note that this functionality is very, very similar to your very first Ajax exercise, but now you are using a custom tag to slightly simplify writing the HTML for the button. For example, my version looks like this: `<ajax:simpleAlert address="show-time" label="Show Time"/>` (“show-time” is the relative URL of the servlet, and “Show Time” is the label that will be on the button. When the user clicks the button, it contacts “show-time”, gets the result, and puts it in an alert box. Note that you need a separate `<% @ taglib ... %>` statement and a separate `<script ...>` in the head.)

All this tag is doing is letting you write

```
<ajax:simpleAlert address="show-time" label="Show Time"/>
```

but inserting the following into the page:

```
<input type="button" value="Show Time" onclick='simpleAjaxAlert("show-time")'/>
```

So, don't make this harder than it is!

2. Repeat the previous problem with a tag file.
3. Make a Java-based tag that creates a better alert button. In addition to specifying the URL as before, the page author should also be able to specify an input field that will be used to build data that will be sent to the URL. See the note on the next page regarding building the data string.
4. Repeat the previous problem with a tag file.
5. Make a Java-based tag that creates a simple Ajax button. When you press the button, the browser should contact some URL (specified in the tag) and place the result in some HTML region (also specified in the tag).
6. Repeat the previous problem with a tag file.

- 7.** Make a Java-based tag that creates a better Ajax button. In addition to specifying the URL and the result region, the page author should be able to specify an input field that will be used to build data that will be sent to the URL. See the note on the next page regarding building the data string.
- 8.** Repeat the previous problem with a tag file.

My solution to the previous two problems is even fancier, in that it lets the page author have an arbitrary number of input fields. If you really feel inspired (because your company doesn't give you enough other work to do), you can give that a try on your own in the evening or next week. You can either let the page author supply a list of field ids, or you can look up the form fields and their names and build the string from that. You may use either a Java-based tag or a tag file. Again, the solutions are online already.