Exercises: AjaxTags I

Make a project that supports AjaxTags. The easiest approach is to edit my ajaxtags-blank project and change index.jsp to do something useful. Remember that you have to put the following somewhere in the JSP page that uses AjaxTags:

```html
<%@ taglib uri="http://ajaxtags.org/tags/ajax" prefix="ajax" %>
```

Or, you could make a new Dynamic Web project and copy the required JAR files, JavaScript files, and styles.css from my ajaxtags project into your new project. But there are quite a few files to copy. Also, remember that with most uses of AjaxTags, you never write or edit any JavaScript code. You have to make the main .jsp page that has the tags, and you have to write the server-side code (servlets or JSP), but you don’t touch the client-side JavaScript code. Pick and choose the problems that interest you below.

1. Make a pushbutton. When the user presses the button, display a random number in the current page. The random number should come from the server. Use ajax:htmlContent.

2. Make a textfield to collect a customer id. When the user presses a button, display a bulleted list showing the id, first name, last name, and balance of that customer. Display an error message for unknown customer ids. Again, my Customer and CustomerUtils classes will do most of the boring part of the work. Again, use ajax:htmlContent.

3. Make a hypertext link that says “Richest Customer”. When you click it, display a bulleted list showing the id, first name, last name, and balance of the richest customer. Remember that CustomerUtils.getRichestCustomer() already returns the Customer with the highest balance. (Copy Customer and CustomerUtils from one of your previous projects). Use ajax:anchors.

4. Make a variation of the problem 2. But add one feature: when the user starts entering a customer id, display a dropdown list of the ids matching what is typed so far.

5. Make four tabbed panels labeled “a1234”, “a1235”, “a1236”, and “a1237”. When the user clicks on the tab, show a bulleted list showing the id, first name, last name, and balance of that customer.

6. Make two select boxes (comboboxes). The first one initially has four choices: “a”, “b”, “c”, and “d”. The second one is initially empty. If the user chooses “a”, then list four words starting with a in the second combobox. Do something similar for the other three choices.