Exercises:
jQuery UI Autocompleters

First, make a new Dynamic Web Project and set things up as described on the earlier jQuery UI Setup handout.

1. Put an accordion panel in the page, with two sections. The first autocompleter (problems 2 and 3) will go in the first accordion section and the second autocompleter (problem 4) will go in the second accordion section.

2. Make a JavaScript array that contains a few city names beginning with A and B. (See http://usatoday30.usatoday.com/weather/climate/usa/wusaclim.htm for a short list.) Make a local autocompleter that lets you enter the name of a city.

3. Add a styled button that, when pressed, will send the city name to Google.

4. Repeat the process, but use an Ajax autocompleter where the list of cities is on the server. Again, send the city name to Google when you press a button. This is one of the few times where you have to write significant server-side code, so skip this problem if you are not a Java programmer or don't want to attempt this. But if you want to try, here are the steps for actually writing the Java code:

- Go to the jquery-ui-autocompleters project and copy both the coreservlets and org.json packages to your new project.
- In the coreservlets package, delete the three Contact-related classes, and rename LanguageCompleter to CityCompleter (R-click on the class name, then Refactor, Rename).
- Put this line at the top of the class: @WebServlet("/city-completer")
- Modify the array and the findLanguages method to search for cities instead of languages (optionally, rename the variable and method names to make it more clear)
- Use source: "city-completer" for your autocompleter.

An alternative is to steal my server-side code, but to still write the client-side code yourself. To do this, go into my jquery-ui-autocompleters-exercises project and make a new HTML file and a corresponding new JavaScript file to be used for the Ajax-based autocompleter. The relative address of the server-side code will be "city-completer".