Lab 19: Introduction to jQuery Ajax

In this lab, you will use jQuery to make Ajax requests.

Objectives:

- To write jQuery Ajax programs

Part 1: Getting Started with jQuery Ajax

Steps:

_1. Use *File - Close All* to close all open editing windows.

_2. In RAD, create a new Dynamic Web Project named **Lab19Web** associated with **Lab19EAR**.

_3. Copy the jQuery JavaScript file from a previous lab into the new project's WebContent folder.

_4. In the WebContent folder, create a JavaScript Source File **studentJSON.js** that will simulate a JSON server. Initialize the file with JSON text that represents a student - name, studentID and GPA. Make sure that you have the syntax correct, then ignore errors in the editor - it's trying to parse the file as JavaScript.

_5. Create a new HTML File, **ajax-json.html**. In the *body*, create empty *div* elements for each of the student properties (each div should have a unique ID).

The write a script in the *head* with a jQuery *ready()* method. In the ready() method, issue a JSON Ajax request, then update the *div* elements with the student information. Note that you will not need a loop since the input JSON file defines a single object.

Run and test.

_6. Create a new XML File named **student.xml**, and enter XML content for a student (root element) with child elements for studentID, name and gpa.

_7. Create a new HTML File, **ajax-xml.html** that has the same *body* content as ajax-json.html. Write a script with a ready() method that issues an Ajax request for the XML, then updates the div elements as before. Again, you will not need a loop.

Hint: You can get the text from an XML element like:

```
var gpa = $(xml).find("gpa").text();
```

Part 2: More Practice

Steps:

_1. Create a new HTML File, **guitars.html**, that implements two hierarchical *select* elements. The first should list brands of guitars: Gibson, Fender and Rickenbacker. The second should list models for the selected brand (use Google to find some sample model names). When the user selects a brand, your page should use Ajax to retrieve the models.

You should write two servlets that override the *doGet* method - one that provides the brand list, and the other the model list. The "model-list" servlet should accept a "GET" parameter for the currently selected brand. See either the complete JSON or XML samples in the chapter for inspiration.

You can create a servlet in RAD by right-clicking on your Web project and choosing *New* - *Servlet* and completing the wizard. We recommend that you put your servlets in a package named **servlet** and assign meaningful names to their classes (the class name will act as the relative URL).

_2. Once you have the above working, add a progress "div" and an Ajax error-handling function.