# *Jdfjlasjdljalskjlkajsdlkjaslkn;aklsnd;lkjas\Web Programming III (420-C30-HR)*

# *Lab 3 – AJAX Beginning*

Date assigned: Wednesday, Sept. 7, 2016

Date due: **Wednesday, Sept 7, 2016, 6:00 p.m.**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will have:

* Understand AJAX
* Understand HTTP Request object

To do:

1. Create a folder called yourusername\_C30\_L03 in your Labs folder for this course. All files will eventually need to be stored in this folder.
2. You need to make HTTP requests throughout this lab. To do that you must work on your localhost server and use http://localhost/x/y/z where x/y/z is your path. Another (better?) option is to work on your O drive and use http://csdev.cegep-heritage.qc.ca/students/yourusername/x/y/z/file.html to access your work. Make sure you copy everything to your H drive and zip it before submitting it, however.

**Part A – Easy A-JAX**

1. Create an html file called a\_easyAJAX.html. Include a form with two fields, a first name and a last name. Add event handlers so that each time the value in either field changes an asynchronous call is made to the server program servertime.php with a GET method. The servertime.php function returns the time from the server. The return value from this method is displayed in an h2 element located to the RIGHT of the form.
2. The steps are:
   1. create an XMLHttpRequest object,
   2. set the onreadystatechange event handler to call a function,
   3. open the xmlHttpRequest object you created with a get command to call the action servertime.php program in the php folder
   4. send the object you created using its send method
   5. In the event handler for ready state, check the value of ready state is 4 and the status is 200 (both of these are attributes of the request object you sent). If both of these are true, get the value returned in the responseText value of the request object.

**Part B – Slightly Less Easy AJAX**

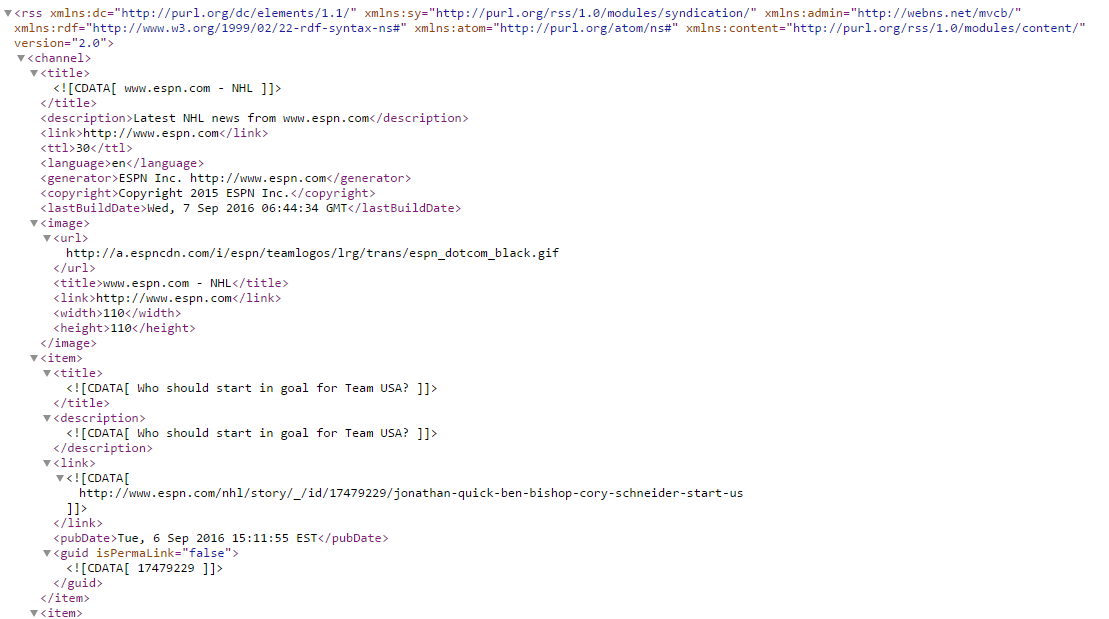
1. Create an html file called a\_moreAJAX.html. Add a form with a single button labelled Start.
2. Add an event handler so that the first time the button is pressed:
   1. The button label changes to Stop
   2. An asynchronous call is made to the server program random.php with a GET method. The random.php function randomly returns the name of someone in the class.
   3. The random.php function is called every 3 seconds until the button is pressed again (use setTimeout).
3. As each response is received from the server, keep a count of which name is returned in the responseText of the request object. Update the count of how many times each person’s name is randomly selected. The data can be displayed in a table, div or whatever you like (make it look decent). For example here is a possible output:

|  |  |
| --- | --- |
| **Student Name** | **Count** |
| Alex B | 3 |
| Alex S | 0 |
| Amir | 1 |
| Andrew | 1 |
| Brae | 2 |
| … |  |
| Troy | 3 |
| Vann | 1 |

1. Note: Only the row that changes should be updated and not the entire table. That is, if the next name is Kevin, then only the row with Kevin’s name is changed to increment the count; the rest of the table is not updated or redisplayed. (hint: use ids to get the proper row and only update it). You can get the names of all the students from the file class.txt in the php/files folder.

**Part C – AJAX with XML**

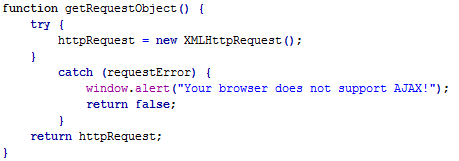
1. ESPN.com publishes separate RSS feeds for various sports, such as hockey, baseball, football, etc. These feeds return XML data. In this exercise, you will create an AJAX program that allows users to view current news for a selected sport.
   1. Open the document partC.html.
   2. It contains a form with a drop down list of five different sports. The value of the dropdown is the URL corresponding to the RSS feed for that sport. Take a couple of seconds to copy one or two to a web browser URL and see what gets returned.
   3. The return value contains a lot of data, but it a form that is tough if not impossible to read. You can see that there are multiple <item> tags corresponding to the individual stories currently sent by ESPN.



1. Create a JavaScript document called partC.js and attach it to the HTML file. Add an event handler so that, whenever the selection changes in the dropdown list, the event handler sportsUpdate is called.
   1. Create a global variable at the top of the file called httpRequest and set it to false.



* 1. Add the following function to the JavaScript file. This function creates a new XMLHttpRequest object.



* 1. Add the sportsUpdate function to the JavaScript file.
     1. This function gets the value of the selected object from the dropdown list and stores it in the variable sport.
     2. The function then aborts any current httpRequest object (in case it is being executed and has not yet returned)
     3. The function sets the onreadystatechange function to be the function getSportsNews
     4. The function opens the request to call the php function SportingNews.php in the php folder. The call must include the query string ?sport= with the sport as assigned in the step (i) above
     5. The function then sends the request.
     6. Lastly, it sets a timer (setTimeout) to recall itself (sportsUpdate) every 5 seconds.

NOTE: At this point you may want to run your program to make sure this is working. Use F12 and check the network traffic to ensure that you are getting a response from the server.

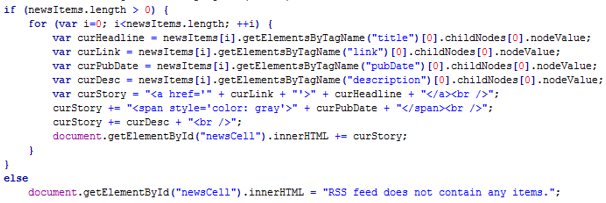
* 1. Add the getSportsNews function to the JavaScript file to handle the return.
     1. This function first checks that the request has completely returned (readystate==4) and that the request is okay (status==200). If not, the function simple ends.
     2. The function then extracts the returned information into an object called news. This requires the responseXML attribute of the response object.



* + 1. The news variable now contains an XML document. We only want the <item> elements, so we use the getElemnetsByTagName function to get the <item> elements and it’s subelements from the XML file into an array called newsItems

  
Notice that the getElementsByTagName was on the news object and not the document object.

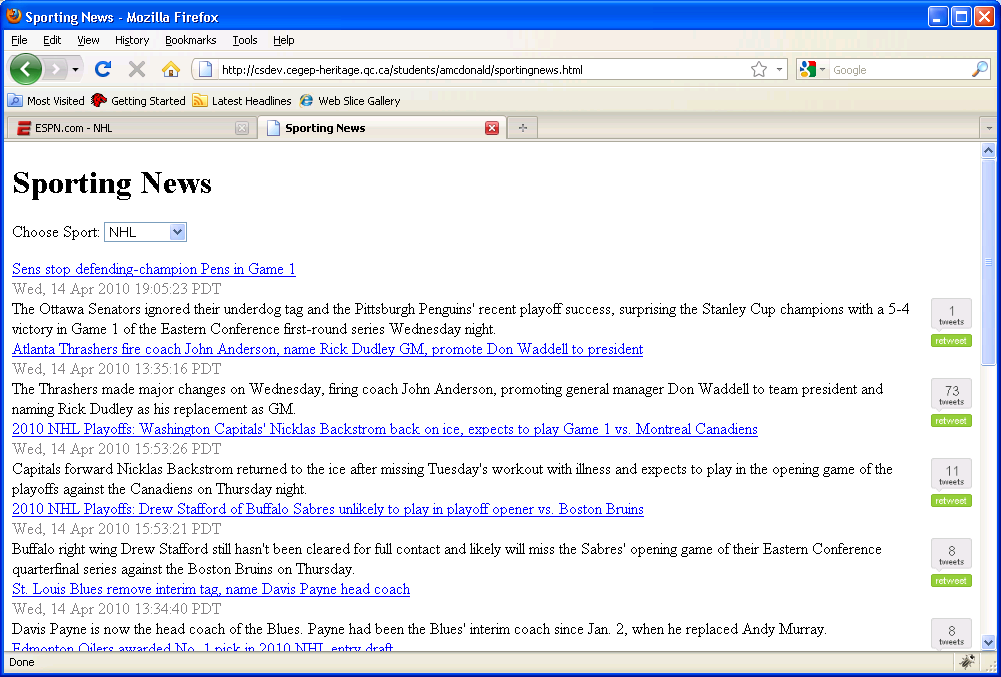
* + 1. Then we add a loop for each item through each item and extract some pieces of each item. Don’t worry too much about the fields extracted below, I just grabbed some of the more important ones. We’ll discuss the how briefly in class and you’ll do a lot more of it next year in C50.



* + 1. Lastly, we add the extracted information to the innerHTML value of the response paragraph from the html file (newsCell).

Note: It’s important that the value of the newsCell paragraph should be set to an empty string at the start of the function so you don’t just keep adding to it.

* 1. When you run your html page and choose a sport the output should be something like below (with different data).



**To submit**

Show me your work. Submit a zip file of all the files used. Remember to copy them off of the O drive.