SER598 Spring 2016 Lab 4
Assigned 3/31/16, due 4/10/16 at 11:59:00pm via submission to Blackboard

In this lab you are required to complete 2 tasks. The first asks you to code a simple NodeJS application. The second asks you to rewrite the same functionality in Express. You will recognize the requirements as being almost exactly the same requirements as Lab 2 (your first servlet app).

Submission:

For submission, you should submit to Blackboard a single zipfile named lab4_<asurite1>.zip (or .jar) that has within it a minimum of 2 Javascript files (one for each application) named task1.js and task2.js. You may have additional Javascript files if you choose to factor a solution out into multiple files (though I have not covered this so in no way is it required). Do not include any additional node modules outside of the ones we have discussed in class (http, url, querystring, express, jade or ejs). Basically it has to be built-in to node (except express, jade, ejs) for you to use it.

Task 1: Simple web functionality (50%)

Write a complete web application that does the following:

1. (5 pts) Provides a web form allowing a user to input a first/last name, languages the person knows how to program in, days of the week the person can meet, and at least one additional descriptive attribute you decide to add. *This is the same functionality as Lab 2, but now you must serve it from node. Map this to URL "/" and method GET.*
2. (20 pts) The web form must POST to URL */post_coder*. The POST should be handled by your node webapp, which will receive the information and store it in an array. The server should respond with a message stating the action was successful (or not) followed by a count of the number of entries in the array, and then display the same web form as step #1.
3. (25 pts) The webapp must accept a GET request to URL */coders* that lists all the entries in the array. Include the following constraints:
   a. If the browser is a Chrome browser then set a pink background color.
   b. The GET should take one or more query parameters that filter results by a substring of the attribute named in the query string. Example:
      i. "GET /coders?firstname=ob&languages=scala+lisp&days=MW&hair=blonde" on this data (this assumes the custom attribute you added was "hair"):
         1. "Bob Smith, java scala c, Monday Thursday, brunette"
         2. "Rob Roy, scala ada, Wednesday Friday, blonde"
         3. "Sue Sink, lisp C#, Monday Wednesday, black

      Would return 2 but not number 1 and 3. Note that the parameters of different attributes are AND'd together, while within a multi-valued attribute it is an OR. Note that if no parameters are given to the GET then the entire set of entries should be returned. Make certain that the results of GETs are not cached by the browser.

Task 2: Redo Task 1 in Express (50%)

Like it says, rewrite the Task 1 application using Express with Jade or EJS as the template engine. Additionally, add the following requirement:

4. Add the ability to filter based on a specific attribute using a RESTful syntax. Specifically, support a GET request on URLs:
    a. */get_coder/firstname/:name*
    b. */get_coder/lastname/:name*

    where :name is bound dynamically to the endpoint of the URL.

Constraints:

1. The web application for Task 1 should only use the http, url, and querystring modules, not express, jade, or ejs. Task 2 should use express and Jade or EJS, and avoid http module APIs.
2. Your node applications should run on port 8081.
3. Do not put Javascript or CSS in the markup for this lab. This includes AJAX. All functionality must be implemented via NodeJS.
4. No 3rd party frameworks outside of Express and Jade or EJS.