2/2/2020

Homework for Ajax Lesson

1. Answer the review questions on this lesson and go over them with a friend or one of the TAs.
    1.1. You do not need to turn these in, but you should review the answers that I post and ask me if you have any questions.

2. Do the "section 8" exercises:
    2.1. http://mumstudents.org/cs472/2016-03-AS-KL/Sections/section08ajax-prototype.php
    2.2. Try to do these exercises without looking at the solutions for at least 10 minutes. Then check out the hints and answers if necessary. Do at least the following 2 exercises:
        2.2.1. Ajax Pets
        2.2.2. Bootloader
    2.3. The Ajax exercises, Ajax Pets and BootLoader are using a PHP server that is running on mumstudents.org. Because of the same domain origin policy for JavaScript Ajax calls your calls must go back to your own web application.
    2.4. In order to get an image to display in an HTML page all you have to do is insert an <img> element onto the page with the desired url for the image. The browser will then retrieve the image from the target url. You can use an absolute url for images (no cross domain restrictions on images) or you can copy the images to your website folders. You can find the absolute URL by looking in the Chrome console at the HTTP request messages.
    2.5. Note that Ajax Pets is using a library called Prototype instead of jQuery. The $ identifier in its solution is referring to the Prototype library instead of jQuery. You can treat that as pseudocode. A jQuery solution will be slightly different.

3. Use the attached IntelliJ project , guestListAjax.zip, to create a web application that uses Ajax to dynamically add names to a guest list. Inspect it or try running it to see what remains to be done. This was a recent final exam programming question, so is good preparation for the final. Also, it is an excellent template for writing the server-side code that you will need for the project. It will show you where to put your JavaScript files and how to work with JSON in Java.

4. Do the Parsing JSON exercise below

Parsing JSON
Make use of the JSON service http://jsonplaceholder.typicode.com. Write a page that processes this simulated JSON blog data. Try to get at least part 2.1 completed. 2.2 is more complex. Do that if you have time and inspiration. Included below is a screen shot of one possible solution (also includes 2.2). You do not have to make it look exactly like this. Do this as a basic HTML application—HTML, CSS, JavaScript. You do not need servlets.

1. Create a Single Page Application (SPA) with an input form to take a userId from the browser
2. Display user name and email and address and all posts belonging to this userId
    2.1. retrieve all posts from selected user /posts?userId=1. For efficiency remember to attach entire lists to DOM rather than each list item

2.2. For every post, include a button (show comments), upon clicking the button show all comments for the specific post. (optional)

    2.2.1. retrieve all comments from selected post /comments?postId=1

    2.2.2. Hint: hide postId in data- attribute in each post.

    2.2.3. Consider using event delegation for the list of posts

    2.2.4. You may need to use $(DOM).on() to attach event handler with functions to newly created elements

WHAT TO TURN IN:

1. Turn in your status report to Sakai, and
2. zip and attach your Ajax solutions for AjaxPets and Bootloader
3. zip and attach your solution to the GuestList
4. upload your Parsing JSON exercise solution to your github pages website. Include a link to that in your status report.