CS338 Developing User Interfaces

Lab 5: jQuery for JSON

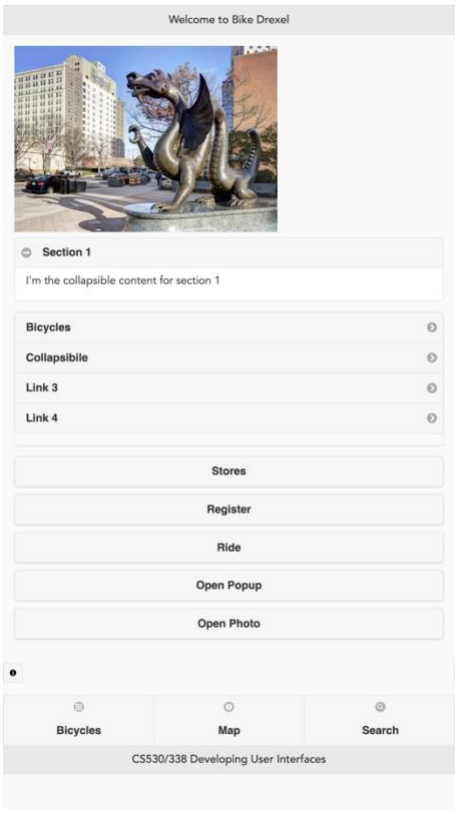
As we learned in the class, one important way to transfer data between a web application and a server is through JSON format. We can also store simple data structures in text files that are in JSON format. JSON files are lightweight, text-based, human-readable, and can be edited using a text editor. One way to get access to JSON files and retrieve data is through the jQuary command, **getJSON**, which "*Load JSON-encoded data from the server using a GET HTTP request."* In this lab, we get more familiar with Jason format and the getJSON command to build a semi dynamic page. Also, we use a table for designing a page.

Note: This is our final lab for the class, and it is more like an assignment. We don't have the steps like before.

Before you start, log in to tux and create a new directory called "Lab5" in your public_html/cs338 directory. Copy all the files from Lab4 into the new Lab5 directory. Grab all the files from Lab5 on BBLearn and save them in the Lab5 project directory you just created.

Your downloaded package contains a file, **data.json**, which has the data structure that we will be working on in this lab. The data structure is in JSON format. Please open it with an editor and take a look inside. It should have information for four bikes.

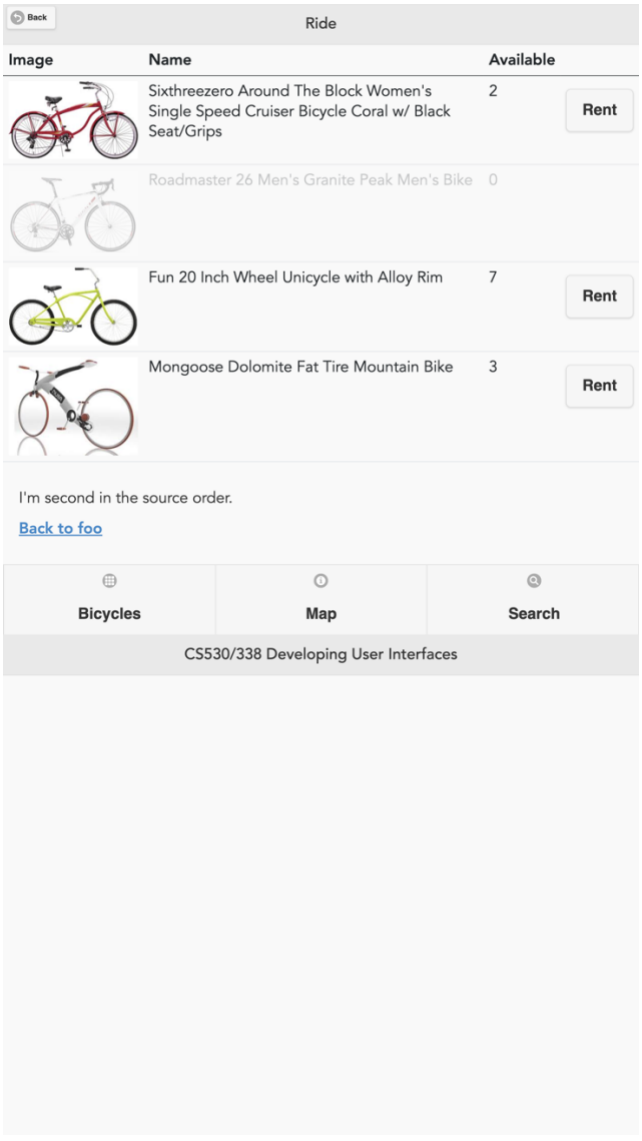
So far, your home page from lab four should look like something like this:



“Ride” page

In this lab, we will be working with the “Ride” page that you created before, but there was no content so far. Please locate the code for the “Ride” page in your code.

“Ride” page should generate dynamic content differently, namely using the getJSON command and JavaScript to populate the page table and handle events on the page. The end result should look as follows:



To create this page, you have to read from the data.json file and store that data in a table. To build the table, write code that iterates over each bike and, using jQuery and JavaScript, adds table rows with the content shown above: bike image, bike name, number available, and a “Rent” button. For this part, you can refer to the WebDev3 lecture, pages 17 and 18, to refresh your memory on how to

work with jQuery. For getting access to the data in the JSON file, your code should look something like this:

```
$(function () {
  $.getJSON('data.json', function (data) {
    $.each(data.bikes, function (i, bike) {

            <!-- Add your code here -->
            <!-- Add your code here -->
            <!-- Add your code here -->
            <!-- Add your code here -->
            <!-- Add your code here -->

        });
  }).error(function () {
    console.log('error');
  });

});
```

If the availability is 0 for any bike, you should add a class **unavailable** to the entire table row and then give this row an opacity of 0.25. I would recommend having a css file to take care of styles in these situations. You should also skip adding the "Rent" button since this function would be disabled.

When the "Rent" button is clicked, the code should decrement that bike's availability by 1. You can use jQuery to access this element and then change it to the updated number. Once the availability goes down to 0, you should add the **unavailable** row class and remove the "Rent" button as described above. (For a real web site, the system would need to communicate these changes to a back-end database, but we are ignoring such issues in this practice.)

When you are done, make sure your lab files are on tux and submit your URL for Lab 5 in BBLearn.