

WEB222 Assignment 6: AJAX

Due date

Section F: Thursday, August 9, 2018 – 11:50 PM

Grade value: 5% of your final course grade

Objective

- To work on JavaScript coding with DOM
- to practice an AJAX call to a Web service

Specification:

In this assignment you will implement the JavaScript code to request JSON data from an online web service using AJAX, to render a table of users consisting of 4 "pages". The first page of data will load when the user first opens up the html page (ie, window.onload). If the user then clicks any of the "paging" buttons at the bottom of the table, the data for the current page will be removed from the table and new data will be loaded. For example:

Show Page 1

When the user first opens assignment6.html, and/or when the user presses the "1" button, or the "first" button

Show Page 2

When the user presses the "2" button

Assignment 7: AJAX

User ID	First Name	Last Name	Avatar
1	George	Bluth	
2	Janel	Wheeler	
3	Emma	Wong	

First 1 2 3 4 Last

Show Page 3

When the user presses the "3" button

Assignment 7: AJAX

User ID	First Name	Last Name	Avatar
4	Eve	Holt	
5	Charles	Morris	
6	Tracey	Ramos	

First 1 2 3 4 Last

Show Page 4




When the user presses the "4" button, or the "last" button:

Assignment 7: AJAX

User ID	First Name	Last Name	Avatar
7	Michael	Lawson	
8	Lindsay	Ferguson	
9	Tobias	Funk	

First 1 2 3 4 Last

Assignment 7: AJAX

User ID	First Name	Last Name	Avatar
10	Byron	Fields	
11	George	Edwards	
12	Rachel	Howell	

First 1 2 3 4 Last

To get started:

1. Download the template files **Assignment6.zip** from the Blackboard.
2. Extract the files and open the assignment6.js file from inside the **js** folder.
3. Note the areas indicated with a "TODO:" comment – these are the areas where you will be writing your code (there is no need to touch the HTML or CSS files). Pay attention to the comments, as these are there to help guide your solution.

4. NOTE: you can test your solution by opening the assignment6.html file on you PC in any browser because this web service supports cross-origin calls. However you must have an internet connection active to see any results from your AJAX requests.

AJAX URL(s) & JSON Data

In order to obtain our data, we will be issuing AJAX requests to a server hosted by <https://regres.in/> - a free REST-API loaded with test data that we can use to make requests for JSON formatted data. The specific url's for each "page" follow the pattern:

- Page 1: <https://regres.in/api/users?page=1>
- Page 2: <https://regres.in/api/users?page=2>
- Page 3: <https://regres.in/api/users?page=3>
- Page 4: <https://regres.in/api/users?page=4>

Click each of the links to see the JSON formatted response in the browser. You will find that the data follows the same consistent format (ie, for page 1):

```
{
  "page": "1",
  "per_page": 3,
  "total": 12,
  "total_pages": 4,
  "data": [
    {
      "id": 1,
      "first_name": "george",
      "last_name": "bluth",
      "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/calebogden/128.jpg"
    },
    {
      "id": 2,
      "first_name": "lucille",
      "last_name": "bluth",
      "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/josephstein/128.jpg"
    },
    {
      "id": 3,
      "first_name": "oscar",
```

```

        "last_name":"bluth",
        "avatar":"https://s3.amazonaws.com/uifaces/faces/twitter/olegpogodaev/128.jpg"
    }
]
}

```

Once the data is parsed, if we wish to obtain the information for the 2nd user on the current page (1) for example, we can access it with our jsData object (see assignment6.js) using the syntax:

```

jsData.data[1].id // 2
jsData.data[1].first_name // "lucille"
jsData.data[1].last_name // "bluth"
jsData.data[1].avatar // "https://s3.amazonaws.com/uifaces/faces/twitter/josephstein/128.jpg"

```

Other Requirements

- You are not permitted to use `element.innerHTML` except to clear out the contents of the table, ie: **using `element.innerHTML = ""`**
- If your code fails to fetch any data or update the DOM with the results, your assignment will be given a grade of zero (0).
- All JavaScript code must not contain any **run-time errors/exceptions**.

Final Submission Guidelines:

- Zip all of your files as **Assignment6.zip**.
- Upload the zip file to Blackboard/My.Seneca under Assignments -> Assignment 6 (same submission procedure as all previous assignments).
- **NO LATE SUBMISSIONS** for assignments. Late Assignment submissions will not be accepted and will receive a grade of zero (0).