

## EJS, API and Forms

For this homework you will need to call the API provided by **Typicode** to query users. In particular, we will use this endpoint:

**<https://jsonplaceholder.typicode.com/users>**

This REST endpoint responds with a JSON response that contains some **mocked up users** with various attributes of information. We are in particular interested in **three attributes**: The user's **name**, the **city** they live in, and the **company** they work for (underlined in red in the screenshot):

```
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  "address": {
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-3874",
    "geo": {
      "lat": "-37.3159",
      "lng": "81.1496"
    }
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org",
  "company": {
    "name": "Romaguera-Crona",
    "catchPhrase": "Multi-layered client-server neural-net",
    "bs": "harness real-time e-markets"
  }
},
```

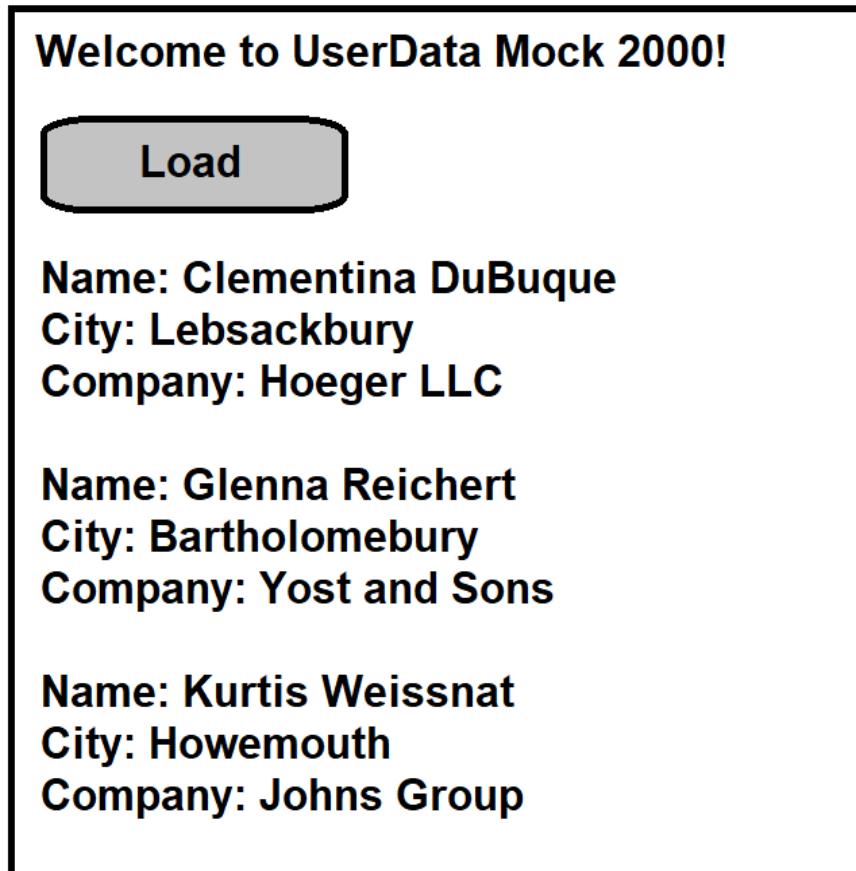
We are interested in listing **3 RANDOM** users from among the whole set that is retrieved, whenever a **button** that says “**Load**” is clicked.

Naturally we **don't** want to present the user with **JSON formatted data**, so you will need to present the JSON data to the user in **readable and formatted HTML** (e.g. in individual labels, in form of a list, as individual paragraphs, etc.) The presentation choice is yours.

The user should see a **headline** on the website stating that this is a website to list random users and their workplace. Beneath there should be the button that says “Load”. The three random results should be displayed underneath that button. The user should be able to press the button again and see three new, random users.

If you use the EJS template from class, make sure you **remove the parts of the UI that aren't needed**. For instance, there should not be an About page with cars or mascots in your project, and neither the variables for those. **Only code that's part of the assignment should be there**. If you start this project from scratch without the class template, then you don't have to worry about that.

A sample mockup screen for this assignment could look like this:



**Welcome to UserData Mock 2000!**

**Load**

**Name: Clementina DuBuque**  
**City: Lebsackbury**  
**Company: Hoeger LLC**

**Name: Glenna Reichert**  
**City: Bartholomebury**  
**Company: Yost and Sons**

**Name: Kurtis Weissnat**  
**City: Howemouth**  
**Company: Johns Group**

**Hint:** Make sure you understand the hierarchy of the JSON response: The information we need is not directly on the highest level of the JSON response.

**EXTRA CREDIT:** Add a checkbox next to the textbox. This checkbox should have the caption: “**Sort Alphabetically**” and if checked, the three results should always be shown in alphabetical order, sorted by the user's **last names**.

**Deadline is April 8<sup>th</sup> 2023, midnight (23:59).**

As always, submit often and early.

**You do not have to comment the HTML (.ejs) files, only the JavaScript code.**

## Basic Grading rubric for this homework

Item	Points
UI setup and controls	20
Code for API call on server.js	20
Logic to parse API result for the required information and display of information from API calls on UI	50
Properly committed and pushed to Github Classroom (at least 2 commits) & Link provided on BB	10
Extra Credit (showing images)	10
Total	100 (110)

Make sure your project compiles and runs.

**If the project doesn't run, you will forfeit points.**

**If your project contains unused code or variables from class, you will forfeit points.**

To set up the remote GitHub repository for your submission correctly, please follow the link below to accept the assignment via GitHub classroom.

You will then have an empty repository in GitHub.

<https://classroom.github.com/a/7JGUrN53>

### What to Turn in:

Commit your source code to your private repository in the GitHub classroom.

**Personal repos outside of GitHub Classroom will NO LONGER be accepted.**

Do not zip the files together.

Your GitHub repository should show **at least two** meaningful commits illustrating how you worked through the problem.

**Submit the link to your GitHub repository via Blackboard.**