420-320 Web Development Assignment 2

Prepared by Nasreddine Hallam Modified by Myles Burgess

Purpose

- To access and fetch data from remote web server API and then Dynamically display it on the web page, as well as store historical requests in local storage.

Objectives

- HTML and CSS
- Coding using DOM and Event handling
- Working with fetch to request data using API Endpoint

Keywords: html, CSS, JavaScript, DOM, event programming, fetch API.

Description of the API

We will be using the Exchange rates API (exchangerate.host site), which is a simple and lightweight free service for current and historical foreign exchange rates and has many endpoints.

In this assignment, you will be working with **Convert Endpoint**. The Convert Endpoint - "Currency conversion endpoint, can be used to convert any amount from one currency to another. When you access the site to discover the API requirements you will find the Convert Endpoint in the Convert Currency tab. You will need to download and discover the structure of the JSON yourself (I.E. download and display the entire download as a string to examine how the data is structured).

In order to convert currencies, you will need to append the **from, to,** and **amount** parameters to the URL so that the web API knows what currency to convert from, what currency to convert to, and the amount you wish to get the result for.

For example,

- The main endpoint API is: https://api.exchangerate.host/convert
- The parameters for this end point are stated in the Query String: from, to and amount. So, converting 35 Dollars from U.S. dollars to Canadian dollars would require the following parameters:
    - From = USD
    - To = CAD
    - Amount = 35
- Therefore, the query string that needs to be sent to the web API will be calculated by combining those four parameters into one query string:
    - https://api.exchangerate.host/convert?from=USD&to=CAD&amount=35
- Everything except the URL will be supplied by the user through the form interface of your web site and you will have to add the ? and & symbols.

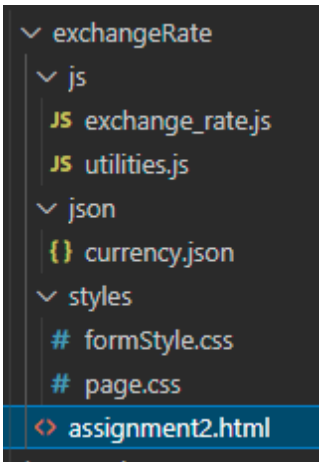
Due Date: November 22, 2022

420-320 Web Development Assignment 2

Prepared by Nasreddine Hallam Modified by Myles Burgess

Files system structure

Figure 1 - Sample file structure

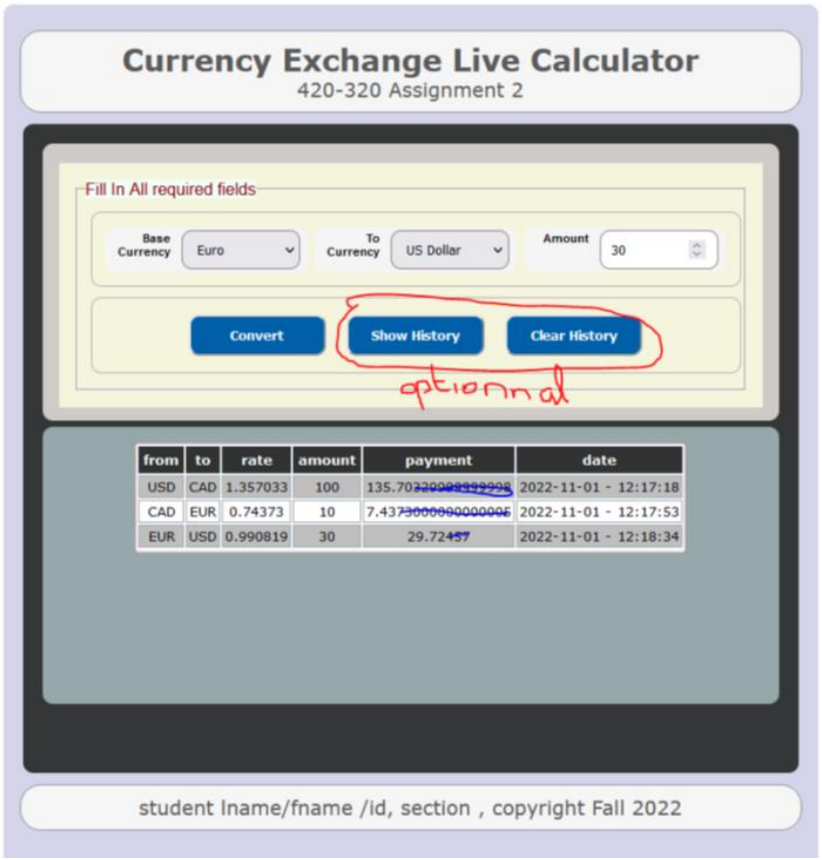


- Your index.html must be at the top of the file structure for this assignment
- Your CSS should be in its own folder inside the file structure for this assignment.
- Your JavaScript should be inside its own folder in the file structure for this assignment
- The supplied JSON file should be in its own folder in the file structure for this assignment.

HTML and CSS

- **No** inline /embedded CSS is allowed.
- **No** inline /embedded JS code is allowed.
- You **must** use **flex** layout.
- Your layout should resemble the GUI shown below (figure 2).
- The colors do not have to match the ones in the image but should meet the contrast requirements of 4.5 for background and text colors, as well go well together
- The selection boxes for **base** and **to** currencies will have the options added by JavaScript (I.E., code the HTML for the select fields but do not code the HTML for the options, they will be added by the JavaScript)
- PS: The image indicates that two of the displayed buttons are optional. Ignore the scribble, as those buttons are not optional and will have their functionality described in the JavaScript requirements section below.

Figure 2 - Sample GUI



Due Date: November 22, 2022

420-320 Web Development Assignment 2

Prepared by Nasreddine Hallam Modified by Myles Burgess

JavaScript requirements

1. Using the supplied JSON file (currency.json) you will dynamically build the options for both the **base currency** and **to currency** selection items in the form. Initially set the CAD option as the default option for each select field.

2. Create an object called urlObject (it can have a global scope). Its (keys /values) are as follows:
   mainURL : `https://api.exchangerate.host/convert`
   fromQry : the value of the **base** currency selection field
   toQry: the value of the **to** currency selection field
   amountQry : the value of the **amount** field from the form
   absoluteURL: a method that will calculate and return the query string required to retrieve information from the web API

   The values for keys fromQry, toQry and amountQry are set dynamically and taken from their corresponding DOM HTML elements when those HTML elements are changed. When you first create this object, you should call whichever function you will be using for the dynamic retrieval of this data, to get the initial values from the form.

3. Whenever the **Convert** button is clicked, you should make a fetch request (do not use XMLHttpRequest) to the web API and retrieve the requested data. You should use the **urlObject** method to build the correct query string for the fetch statement. Since we will not be using the history feature of the web API, you should record the current date and time of the request as it will be needed to help you build your history in local storage.

4. When you retrieve data from the web API, you convert it into usable data and then append the data in a table row that you build dynamically in the lower section of your web page. To build the row you should pass the necessary data to a function specifically designed to create the required HTML elements and append the row to the table. See the GUI figure above to determine which data must be displayed.

   The data should also be appended to whatever history information you have in local storage. (Local storage should not have single keys for each data retrieval but should instead have 1 key that has an array as its value).

   You should use the date and time that you recorded when the request was made as the data for the last field in the table row.

5. When the **Show History** button is clicked, you should replace the currently displayed table with the data in the local storage. You should again pass the required data to the function described in #4 above which builds the row and appends it to that table.

6. When the **Clear history** button is clicked, you should remove all the current table rows from the web site and delete the data stored in local storage.

Due Date: November 22, 2022

420-320 Web Development Assignment 2

Prepared by Nasreddine Hallam Modified by Myles Burgess

Submission

To submit your assignment, place the folder that contains the file structure of your web site into a Windows compressed folder. Submit the Windows compressed folder to LEA JavaScript Assignment 2.

NOTE Do not use winrar or winzip as sometimes the files get corrupted by OMNIVOX, use the Windows **Compressed (zipped) folder** option. If you are not sure where to find this, follow the following steps:

a) In Windows explorer (I.E. the file system) Select the file(s)/folder(s) that you wish to add to the compressed folder
b) Right click on the blue selection area
c) Select **Send to** from the drop-down menu
d) Select Compressed (zipped) folder from the side menu
e) Give the new Compressed folder a name that follows the following convention Yourname_webdev23_A2 (obviously you will replace Yourname with your actual name, or you will not get any marks for the assignment).

Due Date: November 22, 2022