Jose Fonseca

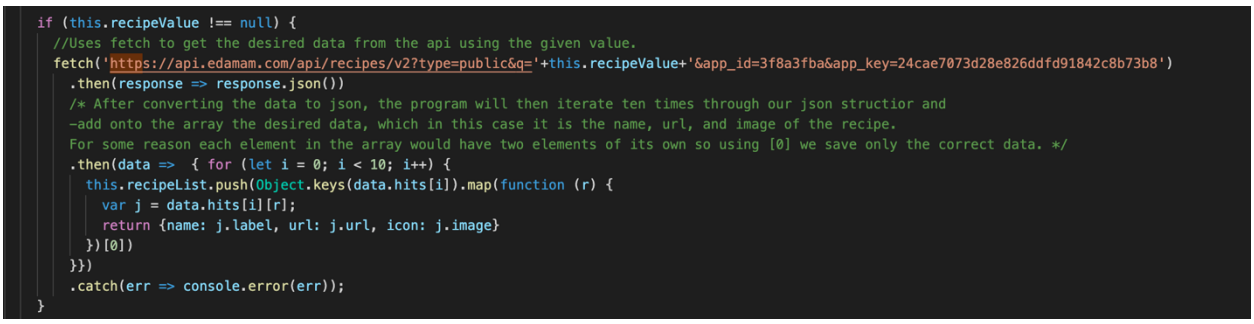
jrfvd6@umsystem.edu

https://github.com/JoseFonseca23/WebMobile-2022Spring/tree/main/2022Spring-Web/Web_ICP6

CS 490 - Web and Mobile Programming

ICP6 Documentation

In this ICP we mainly focused on using RESTful APIs in Angular. We were tasked to create a webpage that would take in users input for a recipe and their location in order to retrieve information from several APIs. The two APIs that were used were Foursquare API for retrieving information on venues according to user input, as well as EDAMAM API for the retrieval of food recipes. We were given starting code but tasked to change the UI as much as possible. I decided to get rid of unnecessary pages, like the about page. Once that was done, I began to work on the main aspect of the web page and that was retrieval of API information. I used Angular's HttpClient in order to do all the fetch requests from the APIs. After getting my custom API link using the app key and app id given to me by the API's owners, I simple stored the information into json for later use. Once the json was stored, I used promises to then iterate through the json and save the information that will be displayed.



```
if (this.recipeValue !== null) {
  //Uses fetch to get the desired data from the api using the given value.
  fetch('https://api.edamam.com/api/recipes/v2?type=public&q='+this.recipeValue+'&app_id=3f8a3fba&app_key=24cae7073d28e826ddfd91842c8b73b8')
    .then(response => response.json())
    /* After converting the data to json, the program will then iterate ten times through our json structior and
    -add onto the array the desired data, which in this case it is the name, url, and image of the recipe.
    For some reason each element in the array would have two elements of its own so using [0] we save only the correct data. */
    .then(data =>  { for (let i = 0; i < 10; i++) {
      this.recipeList.push(Object.keys(data.hits[i]).map(function (r) {
        var j = data.hits[i][r];
        return {name: j.label, url: j.url, icon: j.image}
      })[0])
    }})
    .catch(err => console.error(err));
}
```

Everything after this is simple. All that was to be done is to make sure that the html file properly called in the information and displayed it. Some quality of life improvements I made was to add keyup.Enter event listener to the input fields. That was done so that the user can run the search by simple pressing enter on their keyboard instead of pressing the button. Another small feature I added was to change the URL that would be outputted when the user choses a venue to look for. The reason I did so was to make sure that when and if a user decides not to allow the webpage to take their location, then I google maps URL will still work for them. In conclusion this ICP was really helped further understand Angular and the importance of using API's for web development.