Name : Kavya Para

Student ID : 16326415

Mail : kpkc8@umkc.edu

GitHub Link : https://github.com/KavyaPara/Web_Development_Course/tree/main/Web-Development/ICP6

Name : Manikanta Kavitapu

Student ID : 16322502

Mail : mkmdy@umkc.edu

Github Link : https://github.com/Manikantakavitapu/Web-Development-Course/tree/main/Web-Development/ICP6

REST APIs with Angular

Intro:

A REST API (also known as a RESTful API) is a type of application programming interface (API or web API) that adheres to the REST architectural style's limitations and allows interaction with RESTful web services. REST stands for "REpresentational State Transfer."

To download or upload data and access other back-end services, most front-end applications must communicate with a server using the HTTP protocol. The HttpClient service class in @angular/common/http provides an HTTP client API for Angular applications.

Observable:

In Angular, observable is a feature that allows you to send messages between different portions of your single-page application. Because it is responsible for handling multiple values, asynchronous programming in Javascript, and event handling processes, this functionality is extensively used in Angular.

Subscribe:

Observables will be connected to observers in Angular apps, and if they observe a new value or change in data, they will execute code using Subscription, and all subscribed components will receive the updated result.

HTTP GET:

To obtain data from a server, use the HttpClient.get() method. When the response is received, the asynchronous method sends an HTTP request and returns an Observable that emits the desired data. The observe and responseType variables you supply to the call determine the return type.The get() method accepts two arguments: an endpoint URL to fetch from and an options object to customise the request.

Tasks:

1. We created a service file called SearchService.ts for interacting with Rest APIs.
2. In that service, by using HttpClient module get method we are calling an API URL.
3. In that service we have created two methods named getVenues() for getting the venues and getSearchRecipe() for getting the recipes of the food item.
4. These functions will return response as a json object.
5. When user enters the food item name and location, we passed those as parameters in the HTTP service calls. It will return a response as json object. We will store that response in an array and shown the results by using structural directive “ngFor”. It will loop through all the items in the array and show the required results by using string interpolation.

Code:

Component:

```
// For Getting the Venues
  getVenues() {
    this.recipeValue = this.recipes.nativeElement.value;
    this.placeValue = this.places.nativeElement.value;

    if (this.recipeValue !== null) {

      this.SearchService.getSearchRecipe(this.recipeValue).subscribe(item => {
        let data = item['hits'];
        this.recipeList =data;
        console.log(this.recipeList);
      })
    }

    if (this.placeValue != null && this.placeValue !== '' && this.recipeValue != null && this.recipeValue !== '') {
      this.SearchService.getVenues(this.recipeValue,this.placeValue).subscribe(item => {
        this.venueList = item['results'];
        console.log(this.venueList);
      })
    }
  }
```

Service:

```
export class SearchService {

  url:string ='https://api.foursquare.com/v3/places/search?';

  constructor(private http: HttpClient) { }
// For Getting The Venues
  getVenues(recipe, near) {
    const options = {
      method: 'GET',
      headers: {
        Accept: 'application/json',
        Authorization: 'fsq3qBsG/W/Qy1hOjDBuQ4jRMS1JfOgz6L3n57yGRFOaUpI='
      }
    };

    let finalUrl = this.url + 'near=' + near + "&query=" + recipe
    return this.http.get(finalUrl,options);
    }
  // For Getting the Recepies
    getSearchRecipe(value){
      return this.http.get('https://api.edamam.com/search?app_id=900da95e&app_key=40698503668e0bb3897581f4766d77f9&q='+value)
    }
}
```

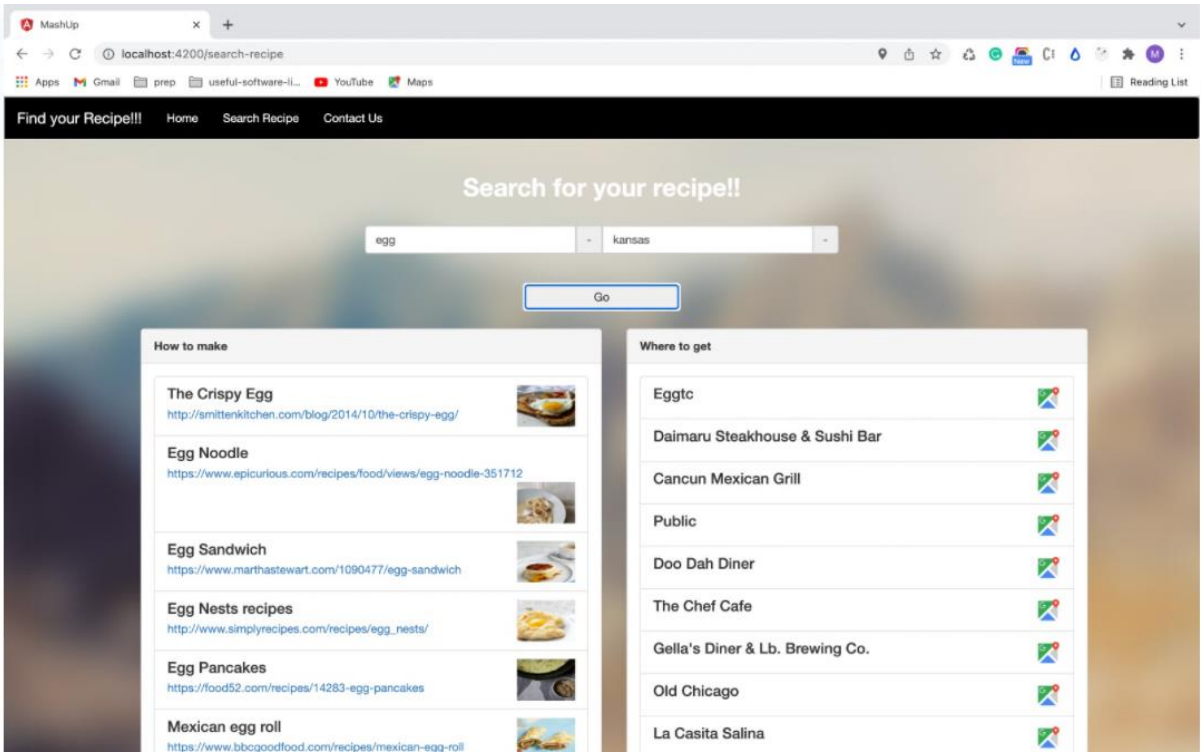
Displaying the Response:

```
<div class="parent-container">
  <div class="child-container">
    <h2><b>Search for your recipe!!
      </b></h2><br>
      <!-- Input Fields For Recipe Name and Place -->
    <form>
      <div class="form-group">
        <div class="form-group col-lg-6 col-lg-offset-3 input-group">
          <input #recipe class="form-control" placeholder="Recipe name" type="text">
          <span class="input-group-addon">-</span>
          <input #place class="form-control" placeholder="Place" type="text">
          <span class="input-group-addon">-</span>
        </div>
        <br>
        <!-- calling Get Venues -->
        <button (click)="getVenues()" class="btn btn primary col-lg-2 col-lg-offset-5 " type="button">Go</button>
      </div>
    </form>
  </div>
</div><br><br>
```

```
<!-- Recipe Wrapper Starts Here -->
<div class="panel panel-default col-6">
  <div class="panel-heading"><b>How to make</b></div>
  <div class="panel-body">
    <div *ngFor="let recipe of recipeList" class="list-group-item clearfix">
      <div class="pull-left">
        <h4 class="list-group-item-heading">{{ recipe.recipe.label }}</h4>
        <a href="{{recipe.recipe.url}}" class="list-group-item-text">{{ recipe.recipe.url }}</a>
      </div>

      <span class="pull-right">
        <img src="{{recipe.recipe.image}}" style="width: 70px; height: 50px;" align="center" />

      </span>

    </div>

  </div>
</div>
```

```
<div class="panel panel-default col-6">
  <!-- Locations of the food -->
  <div class="panel-heading"><b>Where to get</b></div>
  <div class="panel-body">

    <div *ngFor="let venue of venueList" class="list-group-item clearfix">
      <div class="pull-left">
        <h4 class="list-group-item-heading">{{ venue.name }}</h4>
      </div>

      <span class="pull-right">
        <a
          href="http://maps.google.com/maps?saddr={{currentLat}},{{currentLong}}
       &daddr={{venue.location.formattedAddress}},{{venue.location.formattedAddress}},{{venue.location.formattedAddress}}">
          <img src="../../assets/google.png" style="width: 30px; height: 30px;"></a>
      </span>

    </div>

  </div>
</div>
```

Results:



Contributions:

We have contributed equally.