

**Anusha Nelluri** – [anhx2@umsystem.edu](mailto:anhx2@umsystem.edu)

GitHub Link - [https://github.com/NelluriAnusha/Demo\\_Remote/tree/main/Webpart/ICP6](https://github.com/NelluriAnusha/Demo_Remote/tree/main/Webpart/ICP6)

**Achyuth Kumar Valeti** – [avgh3@umsystem.edu](mailto:avgh3@umsystem.edu)

GitHub Link - [https://github.com/AchyuthValeti/Demo\\_Remote/tree/main/Webpart/ICP6](https://github.com/AchyuthValeti/Demo_Remote/tree/main/Webpart/ICP6)

## ICP6 (REST APIs with Angular)

### Introduction:

In Angular, the HTTP requests are handled by the HttpClient API. HTTP protocol provides a high level of flexibility when communicating with a server. We use the HttpClient API to handle the data on the remote server by performing GET, POST, PUT, and Delete operations. In order to consume REST APIs, we must import and configure the HttpClient service in our Angular project.

### Tasks:

- In this ICP, we are going to develop the following two applications using routers, services, HTTP and RESTful APIs in Angular.
  1. Create an application in Angular which displays nearby restaurants
  2. Create an application in Angular which displays recipes
- Firstly, we have created applications on Four Square API & EDAMAM API to get the Recipe & location data.

### Steps to create an account in above mentioned APIs:

1. Created an account using: <https://foursquare.com/>
- After creating an account, logged into the account and created a new project.
- Once the project creation done, generated API Key for place search.
- When making API request call, we used this generated API Key value in HTTPHeaders field.

The screenshot shows the Foursquare developer console interface. At the top, there's a navigation bar with 'FSQ/developer', 'Support', 'Docs', 'Resources', and 'My Account'. Below this is a breadcrumb trail: 'Home / Anusha Nelluri / Projects / Restaurant / Project Settings'. A yellow banner message states: 'Developer Organizations are now available in the console! You can now manage multiple projects and collaborate with other developers. Learn more'. The main content area is titled 'Places API Keys' and features a sidebar with a dropdown menu set to 'Restaurant'. The sidebar has links for 'General', 'Project Settings' (which is active), 'Places API', and 'Usage'. The main panel displays a table of API keys:

API Key Name	Date Generated	Task
Place_search	02/25/2022	<a href="#">Revoke Access</a>

Below the table is a link: [Generate Additional API Key](#).

← → ↻ developer.foursquare.com/reference/place-search

FOURSQUARE My Dashboard → Support

Documentation < API Explorer

JUMP TO CTRL-/

GENERAL

- Overview
- Authentication
- Rate Limits
- Errors
- Pagination
- Session Tokens
- Localization
- Response Fields
- Places Photos Guide

PLACES APIS

- Places
  - Find Nearby Places
  - Recent Nearby Places

## Place Search

GET <https://api.foursquare.com/v3/places/search>

Search for places in the FSQ Places database using a location and querying by name, category name, taste label, or chain name. For example, search for "coffee" to get back a list of recommended coffee shops.

You must pass a location with your request by using one of the following options:

- ll & radius (circular boundary)
- near (geocodable locality)
- ne & sw (rectangular boundary)

QUERY PARAMS

query string

A string to be matched against all content for this place, including but not limited to venue name, category, taste, and

LANGUAGE

- Shell
- Node
- Ruby
- PHP
- JavaScript

AUTHENTICATION

Header Authorization

FETCH

REQUEST

```
1 const options = {method: 'GET', headers: {Accept: 'applicati
2
3 fetch('https://api.foursquare.com/v3/places/search?query=piz
4 .then(response => response.json())
5 .then(response => console.log(response))
6 .catch(err => console.error(err));
```

## 2. Created an account using <https://developer.edamam.com/>

- After creating an account, logged into the account and created a new application to get the recipe details.
- Once the application creation is done, APP ID & APP Keys will be generated.
- The APP ID & APP keys are used when making API request call to get the recipe data.

EDAMAM API Developer Portal Nutrition Wizard DASHBOARD SIGN OUT

DASHBOARD > Applications Services Messages Stats Account

Create a new application

Recipe Search API View Edit

Name	ID	State
API signup	4de073a2	live
Anusha	bcd3289c	live

Recipe Search API View Edit

EDAMAM API Developer Portal Nutrition Wizard DASHBOARD SIGN OUT

DASHBOARD > Applications Services Messages Stats Account

< Anusha

Application ID

bcd3289c

This is the application ID, you should send with each API request.

Application Keys

Create new key

173e0a544ed6bdcca394e88fe452e25f —

These are application keys used to authenticate requests.

Properties

Edit

State: live

### Implementation:

- I have implemented this application with the help of provided source code.
- Four components were created in this application as 'home', 'header' & 'search-recipe' and 'contact'.
- In 'home' component, just created a user interface using html file and user can enter an email id for subscription. The content in this home page is static.
- In 'header' component, used the 'nav-bar' which mainly contains header name or application name.

```
<a routerLink="/home" class="navbar-brand bg-primary" style="color: white">
  Find your Recipe!!!
</a>
```

- Used routing functionality and created 3 menu options and the page gets routed based on the user selection.
- The 'nav-bar' class has been used for 3 links, 1<sup>st</sup> is for 'home', 2<sup>nd</sup> is for 'search-recipe' and 3<sup>rd</sup> is for 'contact-us'.

```
<div class="collapse navbar-collapse tx">
  <ul class="nav navbar-nav">
    <li><a routerLink="/home">Home</a></li>
    <li><a routerLink="/search-recipe">Search Recipe</a></li>
    <li><a routerLink="/contact">Contact Us</a></li>
  </ul>
</div>
```

- In 'app.module.ts' file, all the created components information was included along with paths.
- Import the 'HttpClientModule' also in 'app.module.ts' file to use the HttpClient.
- In 'contact' component, just created a user interface using html & css file and user can enter their contact details.
- Coming to the **search-recipe** component, this html file has two forms.
- The form contains two input fields which allows us to give 'Recipe name' and 'Place'.
- Once we enter these values, there is a submit button named it as 'Go'.
- When we click on this button, **getVenues()** method will be called.

```
<div class="form-group col-lg-12 input-group">
  <input #recipe class="form-control" placeholder="Recipe name" type="text">
  <input #place class="form-control" placeholder="Place" type="text">
</div>
<button (click)="getVenues()" class="form-button " type="button">Go</button>
```

- The above mentioned two API calls (foursquare & EDAMAM) will be happening by calling the **getVenues()** method.
- As we have already registered for both the accounts, we need to mention the unique ID information in URLs.
- The httpClient, which is mentioned in the constructor is already initiated in load of the page.
- In 'ngOnInit()' method, used the geolocation function of getLocation to retrieve the current position's longitude and latitude coordinates.
- Using **nativeElement.value**, we can take the values of user entered Recipe name & Place name.
- We have two if conditions to check whether if the user entered recipe value or place value should not be null.
- If the values are not null, we are making the API calls.
- **The first API call is EDAMAM**, we are directly entering APP ID & APP Key in URL and passing 'recipe value'.
- After then, we call the http **get** method followed by subscribe callback which receives the response data in 'result' variable.
- Then, I have assigned the 'hits' in the result object to the 'output\_recipeData' variable. And, pushing all these values to the 'recipeList', which is used to display the values in html.

```

if (this.recipeValue !== null) {
  // To get the recipe details by using get http
  this.recipeList = [];
  this._http.get(this.recipe_url + '&q=' + this.recipeValue).subscribe( options: result => {
    const output_recipeData = result['hits'];
    console.log('Recipe Data', output_recipeData);
    output_recipeData.map(each => {
      this.recipeList.push({
        name: each.recipe.label,
        url: each.recipe.url,
        icon: each.recipe.image
      });
    });
    console.log(this.recipeList);
  });
}

```

- The second API call is FourSquare API, we are passing 'Recipe name' & 'Place value' and unique API Key with headers.
- After then, we call the http **get** method followed by subscribe callback which receives the response data in 'result' variable.
- Then, I have assigned the 'results' in the result object to the 'output\_recipeData' variable. And, pushing all these values to the 'venueList', which is used to display the values in html.

```
// Code to get the Restaurant details by using get http
this.venueList = [];
const HTTP_headers = new HttpHeaders({
  Accept: "application/json",
  Authorization: "fsq3yq0CskKTh2VGw61HN6IiS03JHd1RJrIuHBtEyFAULLs=",
});
this._http.get(this.location_url + this.recipeValue + '&near=' + this.placeValue + "&v=20220222",
  {headers: HTTP_headers}).subscribe( options: {result: any} => {
  const output_locationData = result['results'];
  console.log('Location:', output_locationData);
  output_locationData.map(each => {
    this.venueList.push({
      name : each.name,
      location : {
        formattedAddress : [
          each.location.formatted_address,
          each.location.locality]
        }
    });
  });
});
```

- Coming to the **search-recipe.component.html**, we were using ng directive **'\*ngFor'** which is used to check the value in 'recipeList' or 'venueList' .
- If the 'recipeList' is not null, we print name of the recipe, URL of the recipe which has all the information (like preparation & ingredients list) related to the recipe.
- We are also displaying the image of the recipe.
- If the 'venueList' is not null, we print the name of the venue and formatted address.
- The anchor tag will help us to make a call to maps of google by passing formatted address as parameter. So that when we click on the link or image, user will be redirected to map of the venue Location.
- We were also displaying google map icon using <div> tag.

### CSS Changes:

- **navbar-dark:** This class changes the text's color to light. When choosing a dark background color, this is applied.

```
<nav class="navbar navbar-dark bg-primary">
```

- **.bg-primary:** This will set the color to the primary color.
- Background image of the app has been changed using below code in **app.component.css**

```
.image{
  height: 100vh;
  background-size:cover;
  width:100%;
  background-image:url('https://www.synopsys.com/blogs/software-security/wp-content/uploads/2019/07/open-source-ma-h');
  background-position:50% 50%;
}
```

- Various form-groups in CSS are modified using the below code:

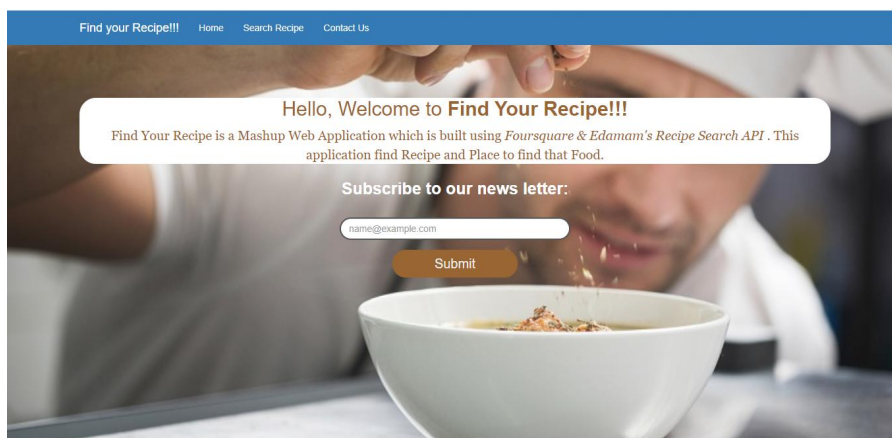
```

14  .form-control {
15      border-radius: 5px;
16      align-items: center;
17      padding: 5px;
18      margin: 5px;
19  }
20
21  .form-group{
22      border-radius: 50px;
23      background-color: #2b669a;
24  }
25
26  .form-button {
27      width: 18%;
28      padding: 10px 20px;
29      display: flex;
30      justify-content: center;
31      align-items: center;
32      border-radius: 20px;
33      background-color: #4caf50;
34      color: black;
35      font-size: 15px;
36      margin: 5px 2px;
37      cursor: pointer;
38      margin-left: 40%;
39  }
40
41  .form-group {
42      padding-top: 20px;
43      position: center;
44      text-align: center;
45  }
46
47  .form-group label {
48      color: white;
49      font-size: 25px;
50  }
51
52  .form-control {
53      border: 2px solid ;
54      border-radius: 20px;
55  }
56
57  .btn {
58      border-radius: 20px;
59      background-color: #996633;
60      color: #eeeeee;
61      font-size: 20px;
62  }

```

## Output Pages:

- Here is the home-screen, user can enter an email id for subscription.
- The content in this home page is static and the home page looks like below.



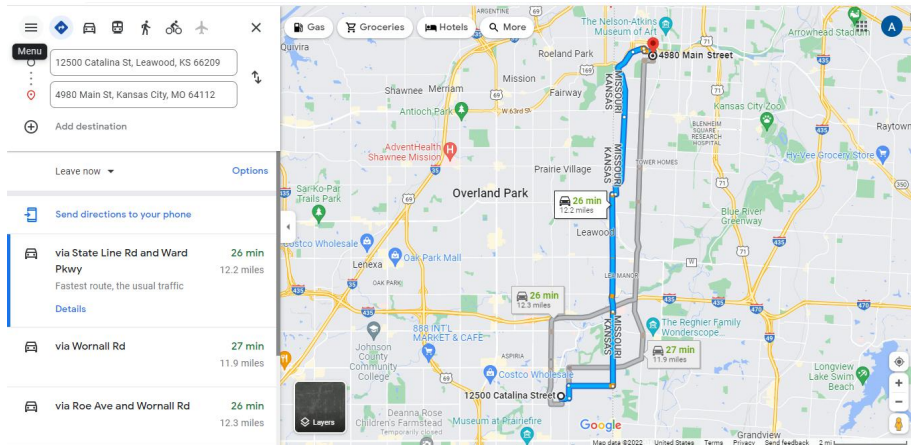


- Here is the search-recipe screen. When we click on 'Search Recipe' button from the home screen, user gets navigated to the below page.

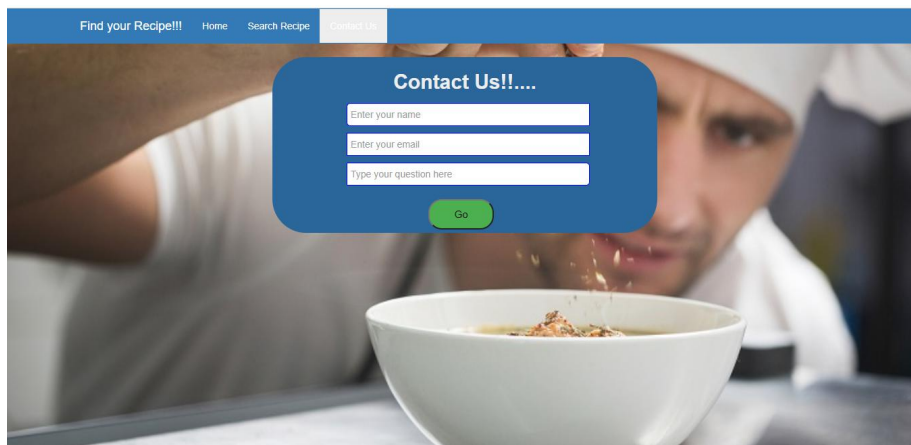
- When we enter recipe name & place, click on the 'Go' button, recipe lists and venue lists get displayed as below.

- When we click on the link that is displayed below the recipe name, user will be redirected to that particular recipe site which has all the ingredients information & preparation steps as well.

- When we click on google map icon, the page will be redirected to google maps page and shows us the destination map from the current location.



- Here is the contact-us screen, the content in this page is static.



## Contribution:

We have contributed equally.

## Conclusion:

- In this ICP, we have learned routers, services, HTTP and RESTful APIs in Angular and developed a web application using the same.

## Challenges:

We faced some challenges while retrieving the information from FourSquare API. We could not be able to get the Venues values when we make the API request call using HTTP get method. Earlier it was working fine but it is not working at present. Since Venues is not working, we used places APIs and keys to get the values for location.

Apart from this, we have not faced any major challenges.



