

# ICP – 6 REST API in Angular

## Team details:

Sarath Chandra Kolisetty – [sk83g@umsystem.edu](mailto:sk83g@umsystem.edu)

Github link: <https://github.com/sarath98-lab/spring-2022/tree/main/Web/ICP6>

Eeshwara Sai Tota – [ettkv@umsystem.edu](mailto:ettkv@umsystem.edu)

Github link: <https://github.com/SaiKicks/WebMobile-Spring2022/tree/main/web/ICP6>

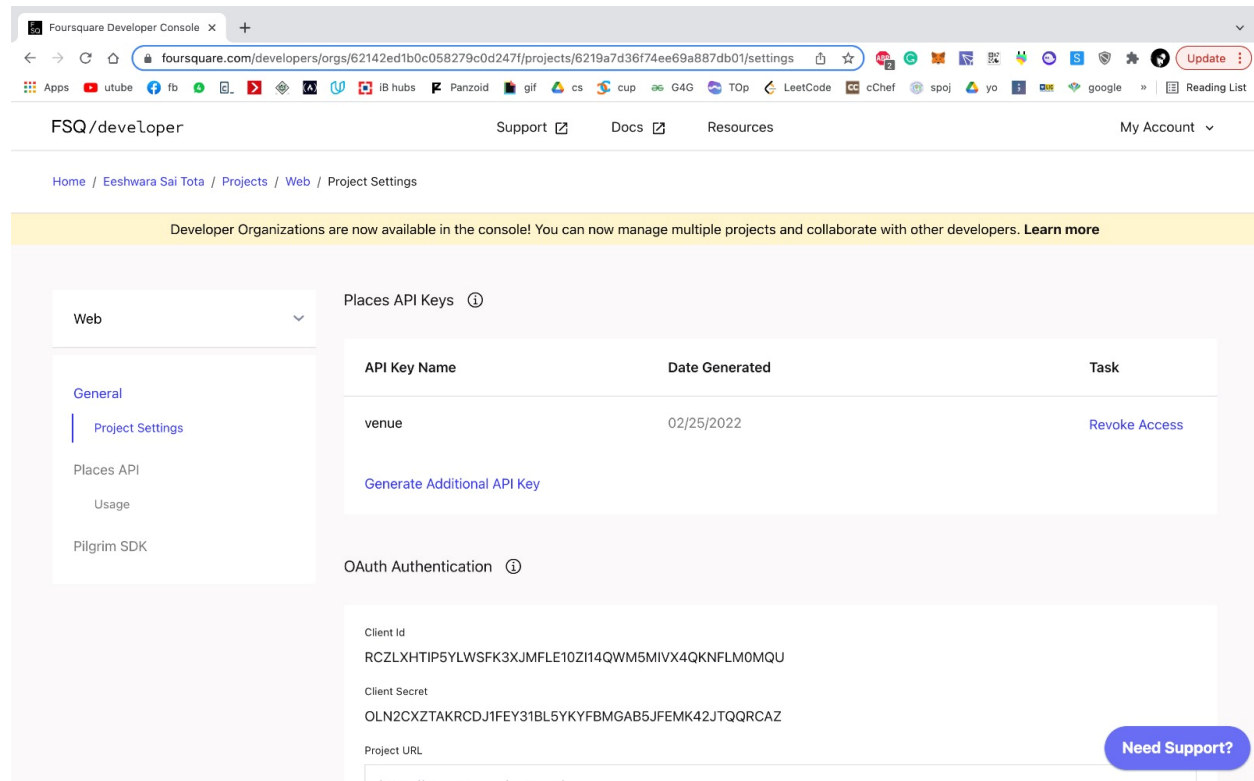
## Description:

Current ICP helps in understanding REST API and make API calls using Angular framework. REST API is a web API that complies with REST architecture. In the current ICP, an interactive web application is created to display nearby restaurants and recipes using Foursquare and EDAMAM API.

## Foursquare API:

In order to use foursquare API first user needs to create an account and create a project.

Client id and client secret for that specific project will be generated. In order to access v3/places/search API, an API key must be generated which is used in header while making places search call. This API returns list of nearby restaurants and their location.



FSQ/developer Support Docs Resources My Account

Home / Eeshwara Sai Tota / Projects / Web / Project Settings

Developer Organizations are now available in the console! You can now manage multiple projects and collaborate with other developers. [Learn more](#)

Web

General

Project Settings

Places API

Usage

Pilgrim SDK

Places API Keys

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

[Generate Additional API Key](#)

OAuth Authentication

Client Id  
RCZLXH5YTLW5FK3XJMFLE10ZI14QWM5MIVX4QKNFLM0MQU

Client Secret  
OLN2CXZTAKRCDJ1FEY31BL5YKYFBMGAB5JFEMK42JTQQRCAZ

Project URL  
<https://www.uvinniant.com/>

[Need Support?](#)

Using the above generated information, recipe name and place the API returns restaurant locations. As show below

RESPONSE 200 Try It

```
1 {
2   "results": [
3     {
4       "fsq_id": "4b019e70f964a520ff4322e3",
5       "categories": [
6         {
7           "id": 13064,
8           "name": "Pizzeria",
9           "icon": {
10            "prefix": "https://ss3.4sqi.net/img/categ
11            "suffix": ".png"
12          }
13        },
14        {
15          "id": 13236,
16          "name": "Italian Restaurant",
17          "icon": {
18            "prefix": "https://ss3.4sqi.net/img/categ
19            "suffix": ".png"
20          }
21        }
22      ],
23      "chains": [],
24      "distance": 1040,
25      "geocodes": {
26        "main": {
27          "latitude": 39.108536040464436,
28          "longitude": -94.58266156479525
29        }
30      },
31      "location": {
32        "address": "427 Main St",
33        "country": "US",
34        "cross_street": "at E 5th St",
35        "dma": "Kansas City",
36        "formatted_address": "427 Main St (at E 5th S
37        "locality": "Kansas City",
38        "neighborhood": [
39          "Downtown"
40      ],
```

## Edamam API:

The Edamam API displays the list of recipes. Similar to the foursquare, we generated credentials in the Edamam API also.

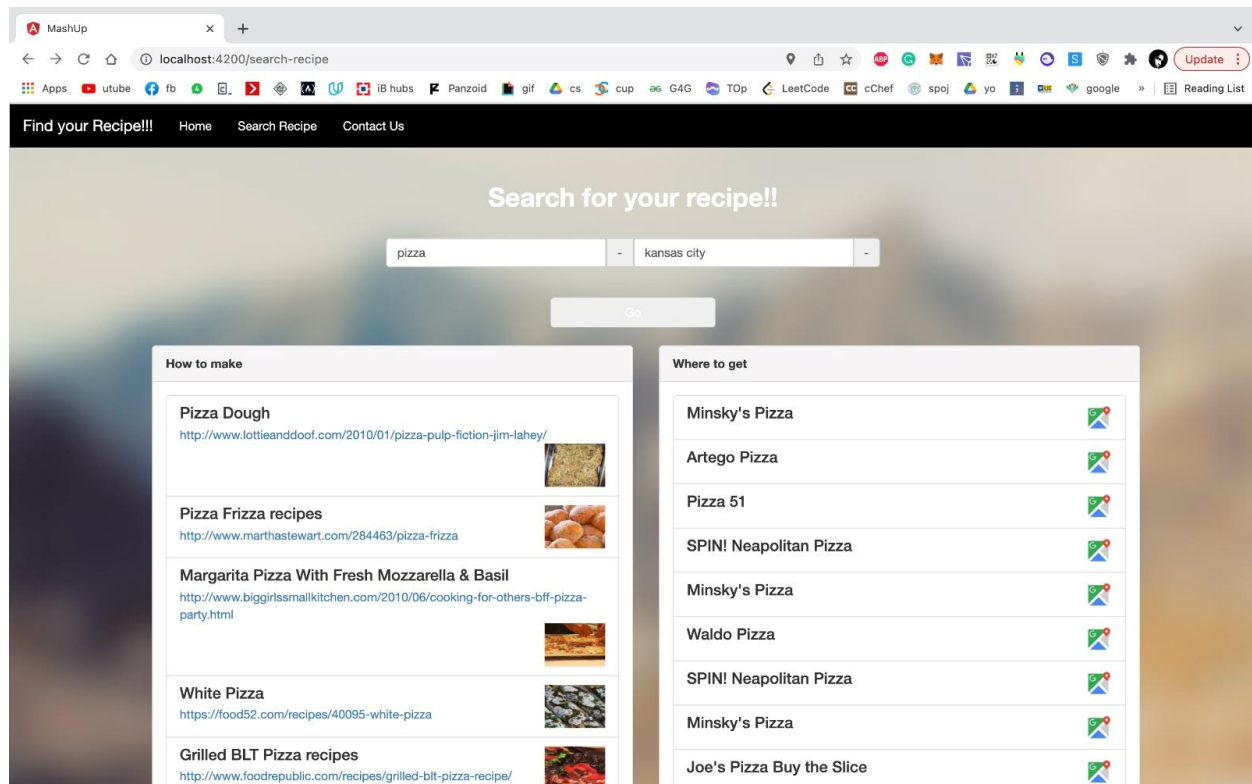
< API signup

Application ID	46e25959	
<small>This is the application ID, you should send with each API request.</small>		
Application Keys	689efccfeaeace910779467d1512aed	Create new key
<small>These are application keys used to authenticate requests.</small>		
Properties	State: live	Edit
Plan		Change

Support

Using Application ID, keys and recipe name we get list of venues through Edamam API.

Using the above two APIs we built have built the website.



## Type script file to fetch places and recipes

```
if (this.placeValue != null && this.placeValue != '' && this.recipeValue != null && this.recipeValue != '') {  
  /==  
  + Write code to get place  
  +/  
  const options = {  
    method: 'GET',  
    headers: {}  
  };  
  Accept: 'application/json',  
  Authorization: 'fsq32gSY49GCP40der033NB+b070Vezbn6jAqNG3wQ='  
};  
let venueUrl = "https://api.foursquare.com/v3/places/search?" +  
"query=" + this.recipeValue + "&nearby=" + this.placeValue;  
const placeUrl = venueUrl + this.recipeValue + "&nearby=" + this.placeValue;  
fetch(placeUrl, options).  
  .then(response => response.json())  
  .then(response => {  
    console.log(response)  
    const venues = response['results'];  
    console.log(venues)  
    venues.map(ele => {  
      const venobj = {  
        name: ele.name,  
        location: {  
          formattedAddress: [ele.location.address, ele.location.city, ele.location.country]  
        }  
      };  
      this.venueList.push(venobj);  
    });  
  });  
  .catch(err => console.error(err));  
}
```

From the response received from the API we only use required information which involves:

Recipe's label, URL and Image

```

if (this.recipeValue !== null) {
  /**
   * Write code to get recipe
   */
  let recipeUrl = "https://api.edamam.com/search?" +
    "app_id=" +
    "46e25959" +
    "&app_key=" +
    "689efcfeaeace910779467d1512aed" +
    "&q="
  const foodUrl = recipeUrl + this.recipeValue;
  this._http.get(foodUrl).subscribe(data => {
    const recipes = data['hits'];
    recipes.map(ele => {
      let recipe = ele['recipe'];
      const recobj = {
        name : ele .recipe.label,
        url: ele.recipe.url,
        icon: ele.recipe.image
      };
      this.recipeList.push(recobj);
    });
  });
}

```

From the response received from Foursquare API we use venue name, address city and country.

### **Learnings from the lesson:**

- HTTPS and REST API in Angular.

### **Issue with the lesson:**

Constructors, ngInIt etc features in Angular must have been discussed.

Discussed v2 version of venue API has issues. This is solved using places API which is not discussed.

### **Contribution:**

Equal contribution