Services Getting Data From the Server

Q.1) What is Service and Why it Written ?

- Service is used to separate the UI logic and business logic.
- Service is a piece of reusable code.
- We can re-use that code in different components.
- A Service is a reusable TypeScript class that can be used in multiple components across angular application.

Q.2) How to create Service in Angular ?

- Command for creating service in angular
- **ng g s serviceName.**

• **Ng**	→	stands for Angular.
• **g**	→	stands for Generate
• **s**	→	stands for Service
• **ServiceName**	→	Given any service name.

Q.3) How Many files Generated once we give this command?

- Two files are generated.

• serviceName.service.spec.ts	→	Its usable for testing purpose
• serviceName.service.ts	→	Usable for perform service operations.

Q.4) Which is the decorator we have by default in the Services ?

- @Injectable

Q.5) What is benefit of dependency injection?

- Loosely coupled.
- Easier to test.
- Reused the code with different components

Q.6) What are the Advantages of Services?

- A service provides re-usability of code/business logic.
- We can share the Business logic across the multiple components.
- Services are easier to test and debug.
- With the services, We can communicate with different components, which does not have Parent Child Relationship.

Q.7) What is used of providedIn: 'root' ?

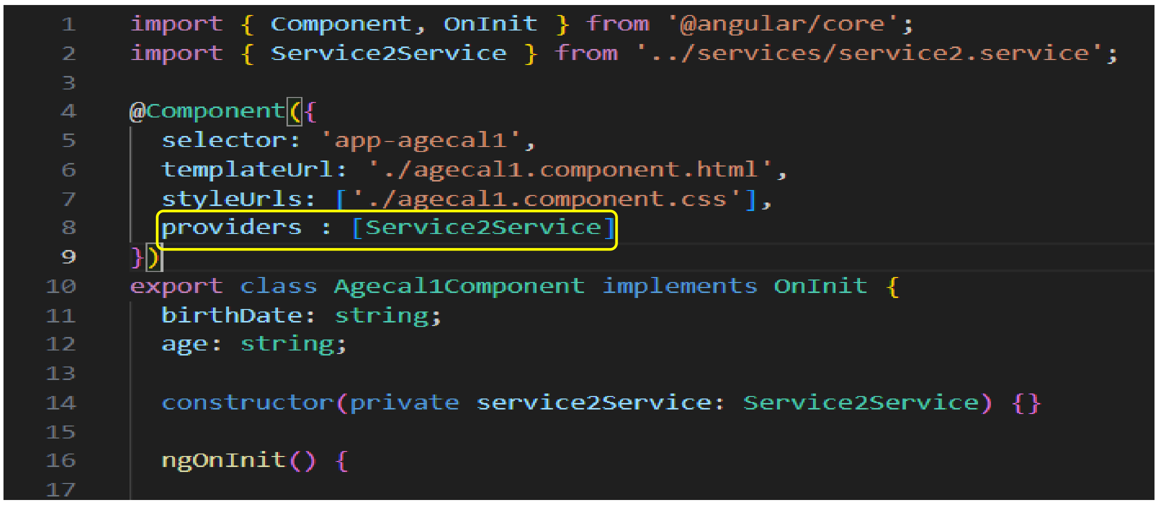
- It means Angular creates a single instance of particular service and whole application can access the same.
- providedIn option registers the service with a specific NgModule.

Services Getting Data From the Server

- And Root means Parent of all components / directives that's why whole application has access to that service.

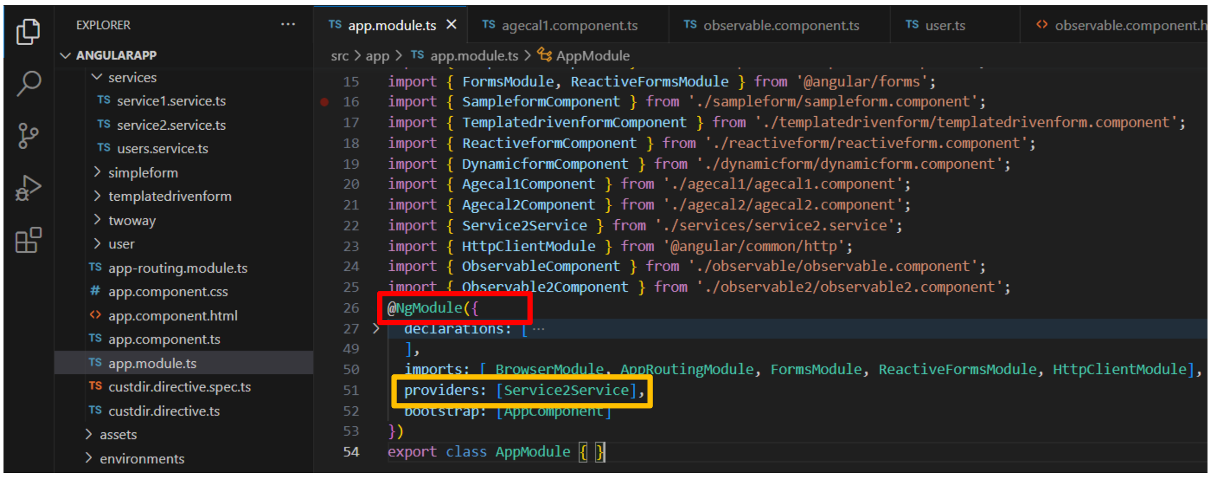
Q.8) How many ways we can register service in angular ?

- Register the Service at the Component Level.



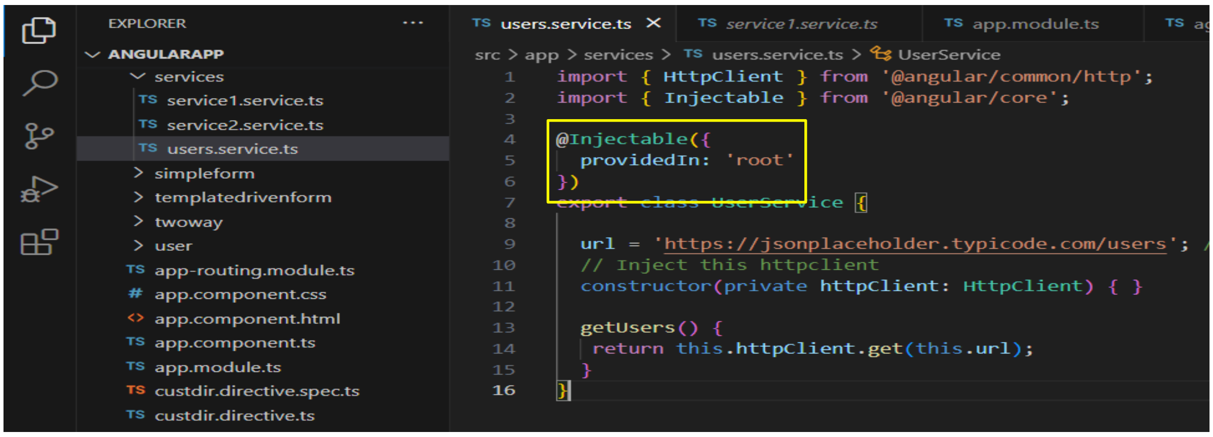
```typescript
import { Component, OnInit } from '@angular/core';
import { Service2Service } from '../services/service2.service';

@Component({
  selector: 'app-agecal1',
  templateUrl: './agecal1.component.html',
  styleUrls: ['./agecal1.component.css'],
  providers : [Service2Service]
})
export class Agecal1Component implements OnInit {
  birthDate: string;
  age: string;

  constructor(private service2Service: Service2Service) {}

  ngOnInit() {

```

- Register the Service at the Module Level.



```typescript
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { SampleformComponent } from './sampleform/sampleform.component';
import { TemplatedrivenformComponent } from './templatedrivenform/templatedrivenform.component';
import { ReactiveformComponent } from './reactiveform/reactiveform.component';
import { DynamicformComponent } from './dynamicform/dynamicform.component';
import { Agecal1Component } from './agecal1/agecal1.component';
import { Agecal2Component } from './agecal2/agecal2.component';
import { Service2Service } from './services/service2.service';
import { HttpClientModule } from '@angular/common/http';
import { ObservableComponent } from './observable/observable.component';
import { Observable2Component } from './observable2/observable2.component';
@NgModule({
  declarations: [···
  ],
  imports: [ BrowserModule, AppRoutingModule, FormsModule, ReactiveFormsModule, HttpClientModule],
  providers: [Service2Service],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

- Register the Service at the root Level.



```typescript
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class UserService {

  url = 'https://jsonplaceholder.typicode.com/users';
  // Inject this httpclient
  constructor(private httpClient: HttpClient) { }

  getUsers() {
   return this.httpClient.get(this.url);
  }
}
```

Services Getting Data From the Server

Q.8) What is Dependency Injection in Angular services?

- Dependency Injection is a design pattern.
- And it is a technique in which class receives its dependencies from external sources rather than creating itself.
- Dependency Injection is by default provided by the Angular framework and it is a best advantage of angular.

Q.9) How can we achieve dependency injection in angular ?

- We need to inject that particular service in in a constructor and for get instance
- We need to register that service in a providers array.
- Also we have another way …
- If we create a service and there is providedIn: 'root'
- Then the angular will create an single instance of that service and the
- Entire whole application will access of that particular instance of that service in any component.

Q.10) What is HttpClient ?

- HttpClient in angular is used to perform HTTP requests and handle the response received from the server.
- HttpClient is a Built-in service class available in **@angular/common/http package** in the angular framework.

Q.11) What is the Use of HttpClient?

- If we want to get the data from the server side then
- We need to send the request for to the Server so for that
- We use HttpClient → its usable for sending request to the server and get the response from the server.

Q.12) Which model we need to import while using HttpClient?

- import {HttpClientModule} from '@angular/common/http';
- Service → httpClient is a inbuilt-in service
- HttpClient → It's performing the request and response from the server.
- HttpClientModule → We need to import HttpClientModule in imports array.

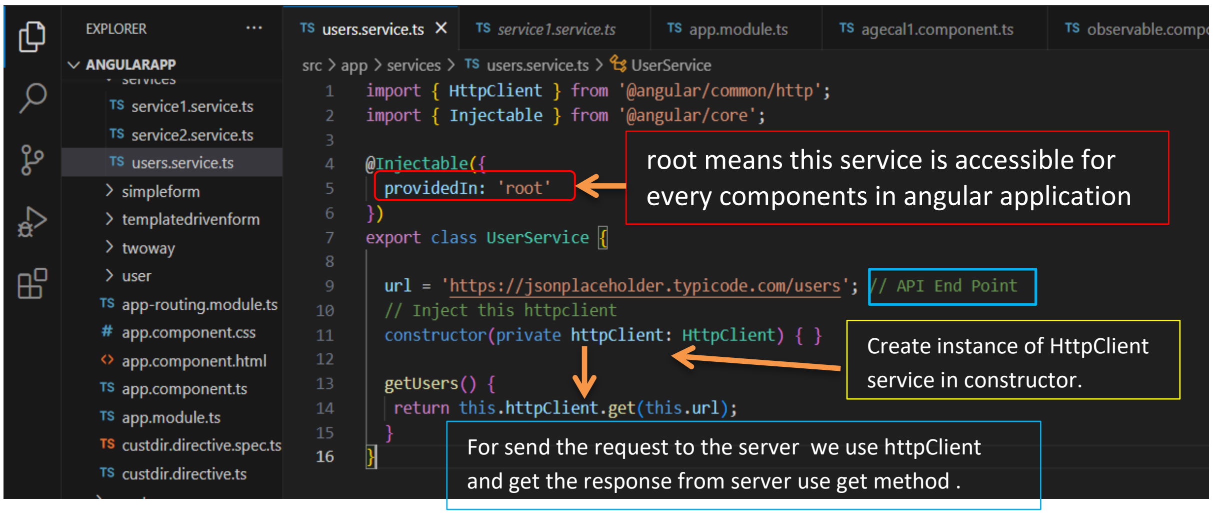
Q.13) List the HttpClient Methods in Angular?

- get() → To retrieve data from the server
- post() → To post new client data to the server
- put() → To update the data to the server
- delete() → To delete the item from the server
- patch() → To update a part of the information for the given resource

Services Getting Data From the Server

Q.14) How to get the data from the server?

- For this we need to go with some basic standards like first need to create service.
- For creating service we use ng g s users (Users is my Service name).



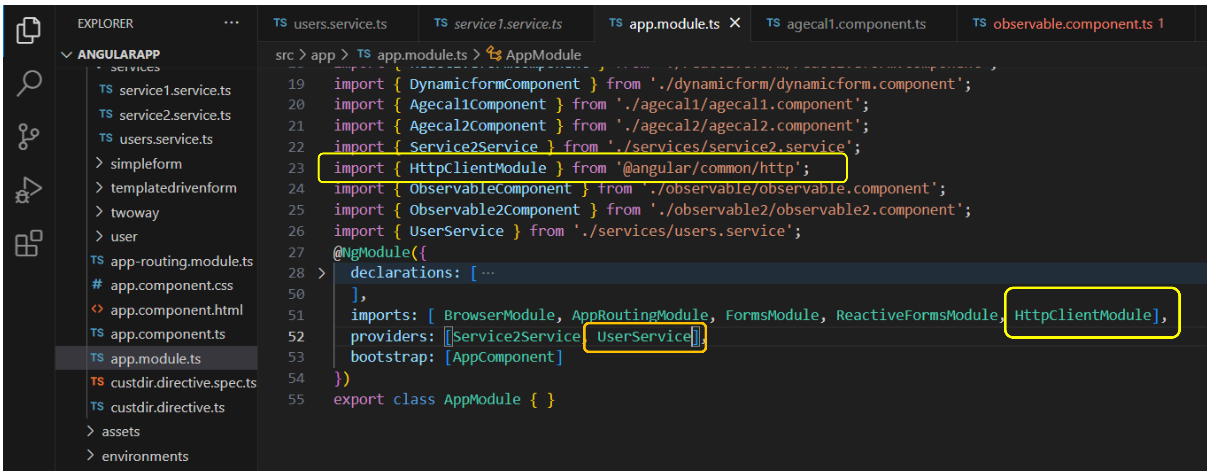
```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class UserService {

  url = 'https://jsonplaceholder.typicode.com/users'; // API End Point
  // Inject this httpclient
  constructor(private httpClient: HttpClient) { }

  getUsers() {
   return this.httpClient.get(this.url);
  }
}
```

root means this service is accessible for every components in angular application

Create instance of HttpClient service in constructor.

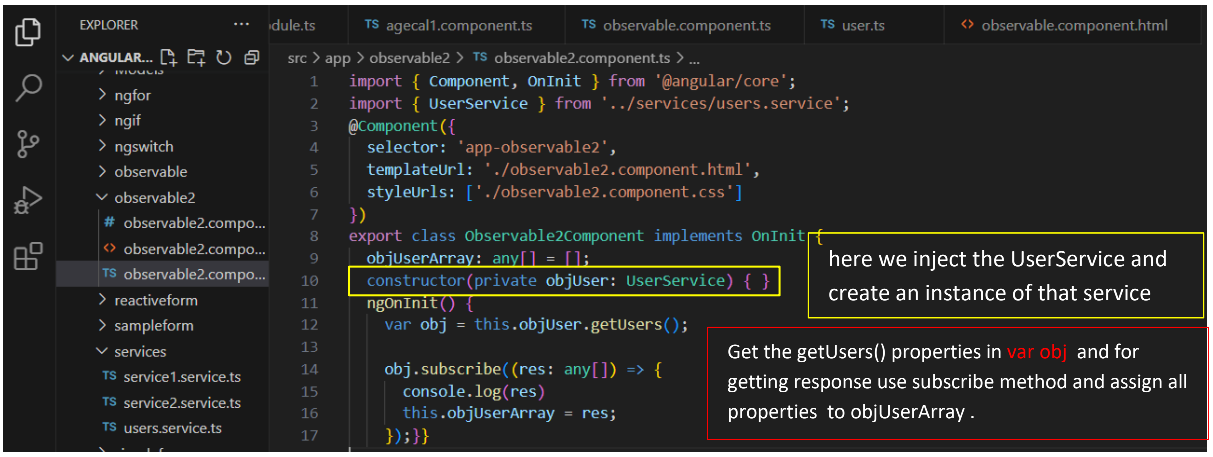
For send the request to the server we use httpClient and get the response from server use get method .

- Configure the HttpClientModule in an app.component.ts



```
import { DynamicformComponent } from './dynamicform/dynamicform.component';
import { Agecal1Component } from './agecal1/agecal1.component';
import { Agecal2Component } from './agecal2/agecal2.component';
import { Service2Service } from './services/service2.service';
import { HttpClientModule } from '@angular/common/http';
import { ObservableComponent } from './observable/observable.component';
import { Observable2Component } from './observable2/observable2.component';
import { UserService } from './services/users.service';
@NgModule({
  declarations: [ ···
  ],
  imports: [ BrowserModule, AppRoutingModule, FormsModule, ReactiveFormsModule, HttpClientModule],
  providers: [Service2Service, UserService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

- Injecting the service in Component class.



```
import { Component, OnInit } from '@angular/core';
import { UserService } from '../services/users.service';
@Component({
  selector: 'app-observable2',
  templateUrl: './observable2.component.html',
  styleUrls: ['./observable2.component.css']
})
export class Observable2Component implements OnInit {
  objUserArray: any[] = [];
  constructor(private objUser: UserService) { }
  ngOnInit() {
    var obj = this.objUser.getUsers();

    obj.subscribe((res: any[]) => {
      console.log(res)
      this.objUserArray = res;
    });}}
```

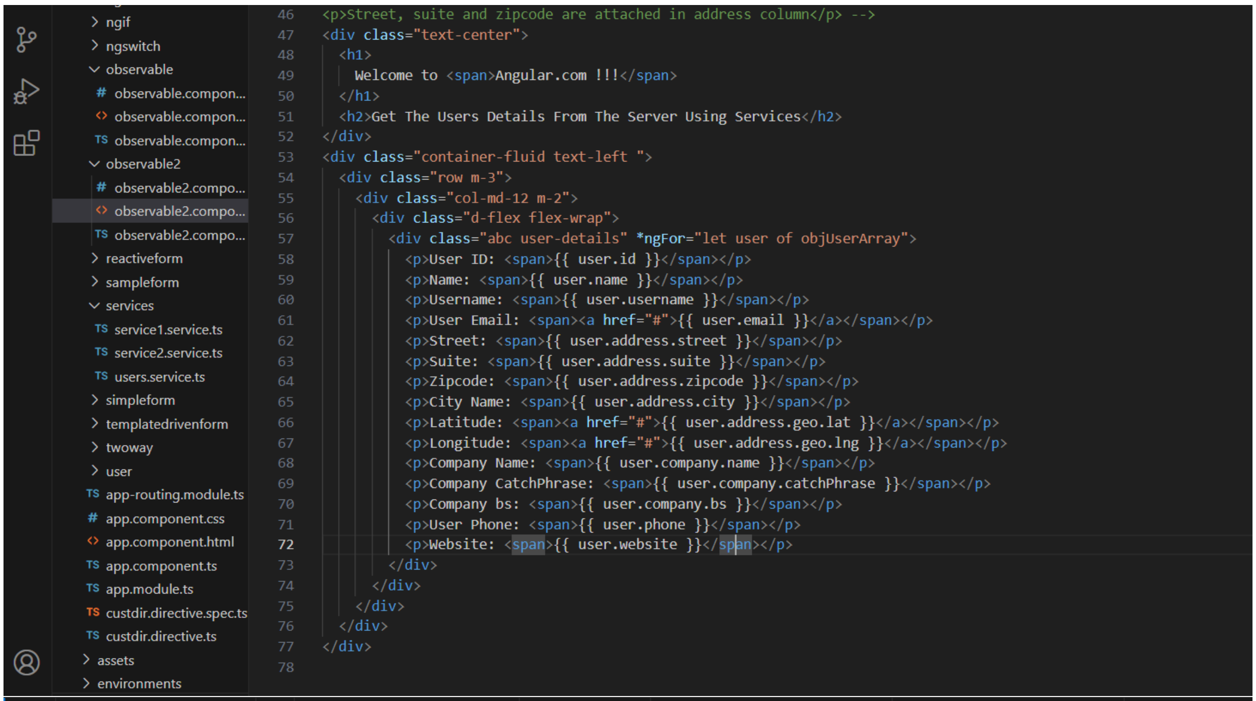
here we inject the UserService and create an instance of that service

Get the getUsers() properties in var obj and for getting response use subscribe method and assign all properties to objUserArray .

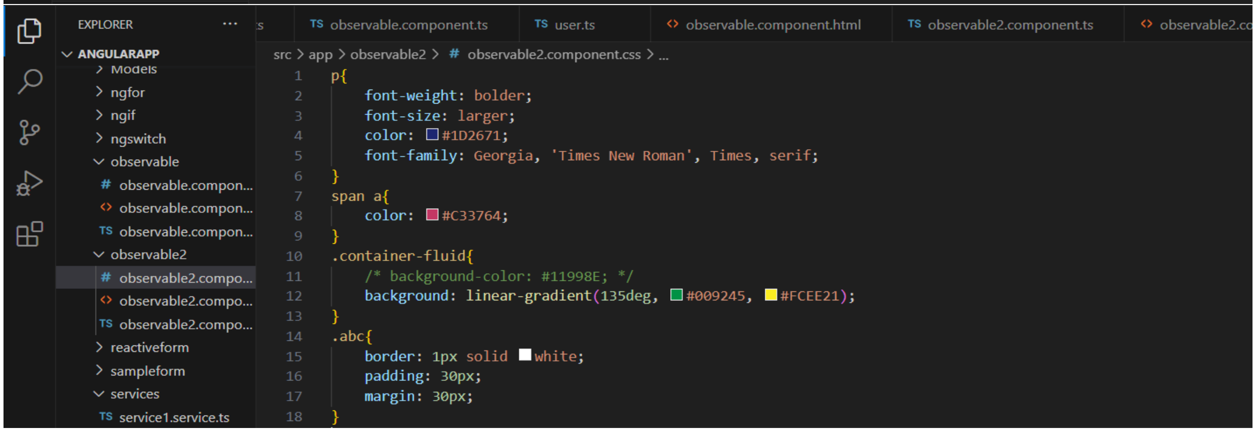
Ajay Patil

Services Getting Data From the Server

- After this parse the response by the component class we will use HTML template and angular <code> *ngFor</code> expression.



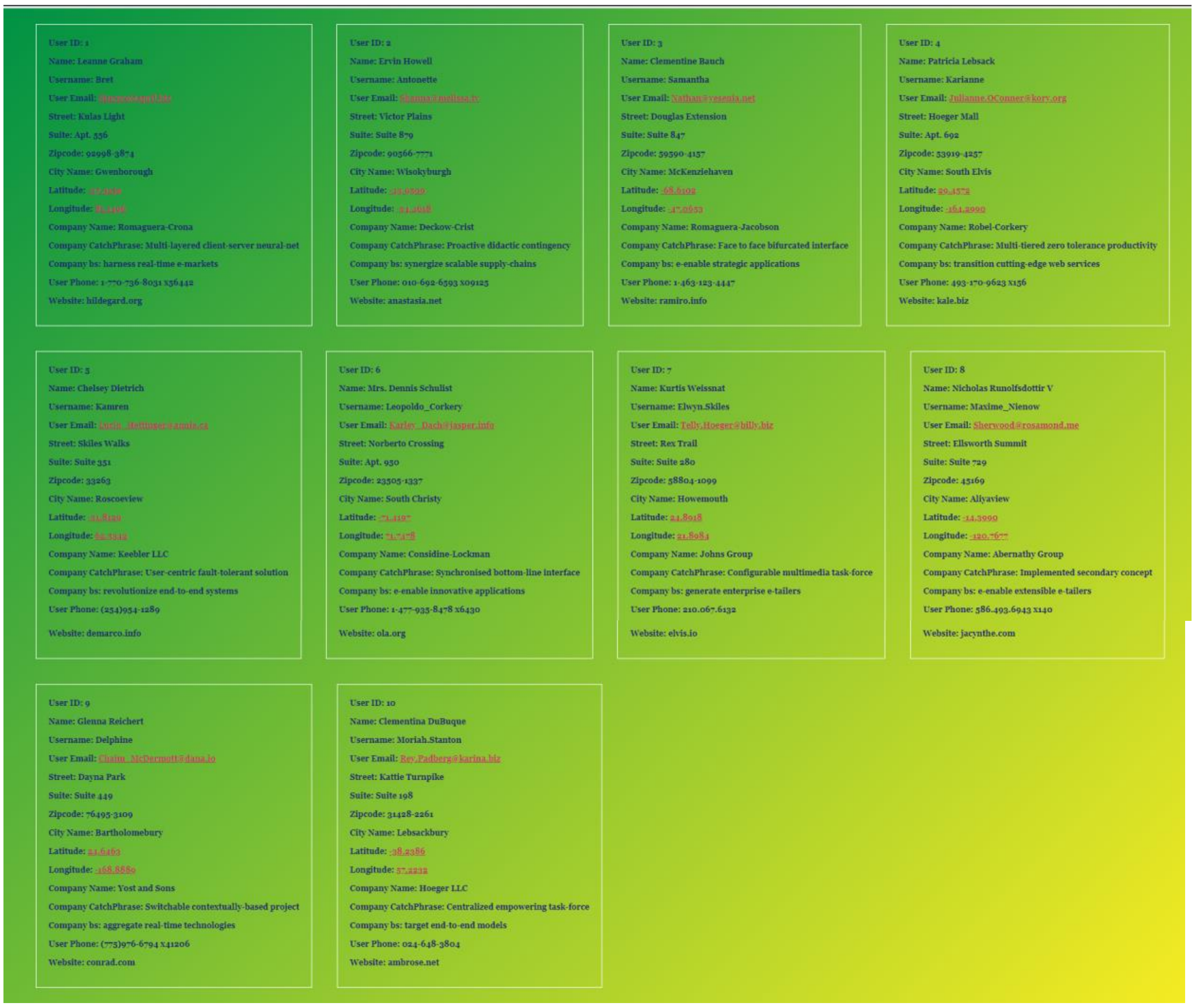
```html
<p>Street, suite and zipcode are attached in address column</p> -->
<div class="text-center">
  <h1>
    Welcome to <span>Angular.com !!!</span>
  </h1>
  <h2>Get The Users Details From The Server Using Services</h2>
</div>
<div class="container-fluid text-left ">
  <div class="row m-3">
    <div class="col-md-12 m-2">
      <div class="d-flex flex-wrap">
        <div class="abc user-details" *ngFor="let user of objUserArray">
          <p>User ID: <span>{{ user.id }}</span></p>
          <p>Name: <span>{{ user.name }}</span></p>
          <p>Username: <span>{{ user.username }}</span></p>
          <p>User Email: <span><a href="#">{{ user.email }}</a></span></p>
          <p>Street: <span>{{ user.address.street }}</span></p>
          <p>Suite: <span>{{ user.address.suite }}</span></p>
          <p>Zipcode: <span>{{ user.address.zipcode }}</span></p>
          <p>City Name: <span>{{ user.address.city }}</span></p>
          <p>Latitude: <span><a href="#">{{ user.address.geo.lat }}</a></span></p>
          <p>Longitude: <span><a href="#">{{ user.address.geo.lng }}</a></span></p>
          <p>Company Name: <span>{{ user.company.name }}</span></p>
          <p>Company CatchPhrase: <span>{{ user.company.catchPhrase }}</span></p>
          <p>Company bs: <span>{{ user.company.bs }}</span></p>
          <p>User Phone: <span>{{ user.phone }}</span></p>
          <p>Website: <span>{{ user.website }}</span></p>
        </div>
      </div>
    </div>
  </div>
</div>
```



src > app > observable2 > # observable2.component.css > ...

```css
p{
    font-weight: bolder;
    font-size: larger;
    color: #1D2671;
    font-family: Georgia, 'Times New Roman', Times, serif;
}
span a{
    color: #C33764;
}
.container-fluid{
    /* background-color: #11998E; */
    background: linear-gradient(135deg, #009245, #FCEE21);
}
.abc{
    border: 1px solid white;
    padding: 30px;
    margin: 30px;
}
```

Services Getting Data From the Server



User ID: 1
Name: Leanne Graham
Username: Bret
User Email: Sincere@april.biz
Street: Kulas Light
Suite: Apt. 556
Zipcode: 92998-3874
City Name: Gwenborough
Latitude: -37.3159
Longitude: 81.1496
Company Name: Romaguera-Crona
Company CatchPhrase: Multi-layered client-server neural-net
Company bs: harness real-time e-markets
User Phone: 1-770-736-8031 x56442
Website: hildegard.org

User ID: 2
Name: Ervin Howell
Username: Antonette
User Email: Shanna@melissa.tv
Street: Victor Plains
Suite: Suite 879
Zipcode: 90566-7771
City Name: Wisokyburgh
Latitude: -43.9509
Longitude: -34.4618
Company Name: Deckow-Crist
Company CatchPhrase: Proactive didactic contingency
Company bs: synergize scalable supply-chains
User Phone: 010-692-6593 x09125
Website: anastasia.net

User ID: 3
Name: Clementine Bauch
Username: Samantha
User Email: Nathan@yesenia.net
Street: Douglas Extension
Suite: Suite 847
Zipcode: 59590-4157
City Name: McKenziehaven
Latitude: -68.6102
Longitude: -47.0653
Company Name: Romaguera-Jacobson
Company CatchPhrase: Face to face bifurcated interface
Company bs: e-enable strategic applications
User Phone: 1-463-123-4447
Website: ramiro.info

User ID: 4
Name: Patricia Lebsack
Username: Karianne
User Email: Julianne.OConner@kory.org
Street: Hoeger Mall
Suite: Apt. 692
Zipcode: 53919-4257
City Name: South Elvis
Latitude: 29.4572
Longitude: -164.2990
Company Name: Robel-Corkery
Company CatchPhrase: Multi-tiered zero tolerance productivity
Company bs: transition cutting-edge web services
User Phone: 493-170-9623 x156
Website: kale.biz

User ID: 5
Name: Chelsey Dietrich
Username: Kamren
User Email: Lucio_Hettinger@annie.ca
Street: Skiles Walks
Suite: Suite 351
Zipcode: 33263
City Name: Roscoeview
Latitude: -31.8129
Longitude: 62.5342
Company Name: Keebler LLC
Company CatchPhrase: User-centric fault-tolerant solution
Company bs: revolutionize end-to-end systems
User Phone: (254)954-1289
Website: demarco.info

User ID: 6
Name: Mrs. Dennis Schulist
Username: Leopoldo_Corkery
User Email: Karley_Dach@jasper.info
Street: Norberto Crossing
Suite: Apt. 950
Zipcode: 23505-1337
City Name: South Christy
Latitude: -71.4197
Longitude: 71.7478
Company Name: Considine-Lockman
Company CatchPhrase: Synchronised bottom-line interface
Company bs: e-enable innovative applications
User Phone: 1-477-935-8478 x6430
Website: ola.org

User ID: 7
Name: Kurtis Weissnat
Username: Elwyn.Skiles
User Email: Telly.Hoeger@billy.biz
Street: Rex Trail
Suite: Suite 280
Zipcode: 58804-1099
City Name: Howemouth
Latitude: 24.8918
Longitude: 21.8984
Company Name: Johns Group
Company CatchPhrase: Configurable multimedia task-force
Company bs: generate enterprise e-tailers
User Phone: 210.067.6132
Website: elvis.io

User ID: 8
Name: Nicholas Runolfsdottir V
Username: Maxime_Nienow
User Email: Sherwood@rosamond.me
Street: Ellsworth Summit
Suite: Suite 729
Zipcode: 45169
City Name: Aliyaview
Latitude: -14.3990
Longitude: -120.7677
Company Name: Abernathy Group
Company CatchPhrase: Implemented secondary concept
Company bs: e-enable extensible e-tailers
User Phone: 586.493.6943 x140
Website: jacynthe.com

User ID: 9
Name: Glenna Reichert
Username: Delphine
User Email: Chaim_McDermott@dana.io
Street: Dayna Park
Suite: Suite 449
Zipcode: 76495-3109
City Name: Bartholomebury
Latitude: 24.6463
Longitude: -168.8889
Company Name: Yost and Sons
Company CatchPhrase: Switchable contextually-based project
Company bs: aggregate real-time technologies
User Phone: (775)976-6794 x41206
Website: conrad.com

User ID: 10
Name: Clementina DuBuque
Username: Moriah.Stanton
User Email: Rey.Padberg@karina.biz
Street: Kattie Turnpike
Suite: Suite 198
Zipcode: 31428-2261
City Name: Lebsackbury
Latitude: -38.2386
Longitude: 57.2232
Company Name: Hoeger LLC
Company CatchPhrase: Centralized empowering task-force
Company bs: target end-to-end models
User Phone: 024-648-3804
Website: ambrose.net

Thank U For Your Patience

Ajay Patil