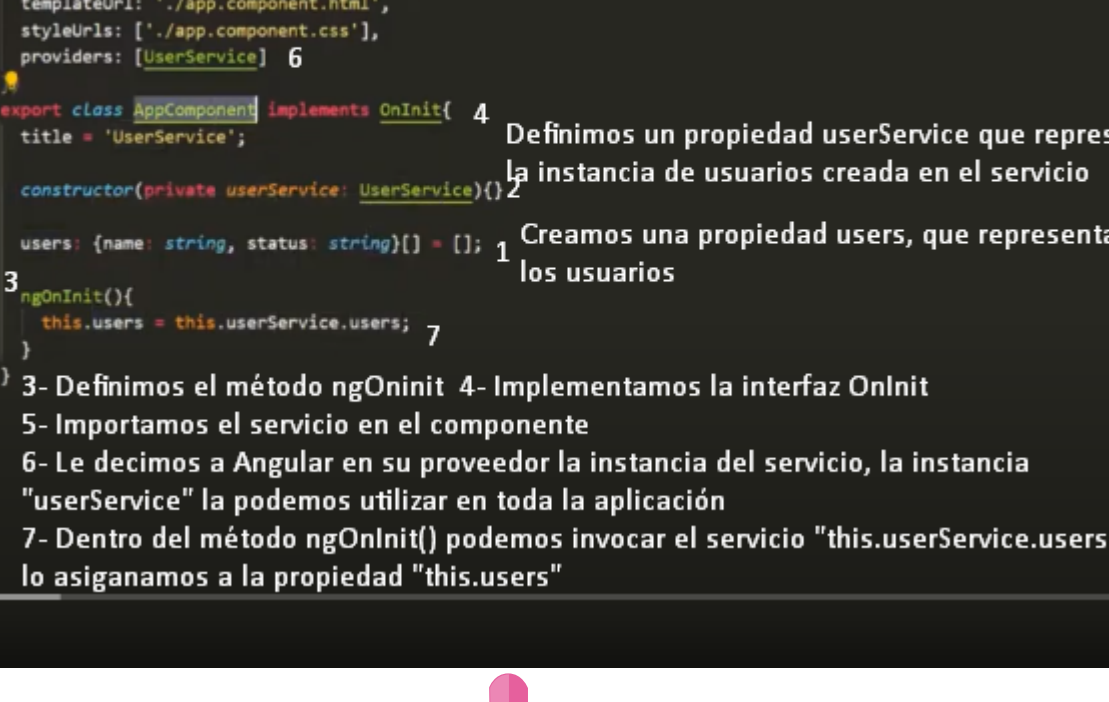




Services

- user.service.ts
- app.component.css
- app.component.html



vamos a agregar algunos usuarios desplegaremos un cuadro de texto en la página web podemos ingresar al usuario, luego mostramos una lista desplegable donde podemos poner si el usuario está activo o inactivo y agregaremos un botón para agregar usuarios



```

1 app.component.html | user service.ts | app.component.ts | adduser.component.html
2
3 <input type="text" value="add user" /> <button> Add User </button>
4
5 <div class="form">
6   1
7   2 <input type="text" placeholder="Username" id="username">
8   3
9   4
10  5
11  <input type="text" id="status">
12  <input value="Active" checked="" type="radio"/> Active
13  <input value="Inactive" type="radio"/> Inactive
14  </div>
15  </div>
16  <div>
17  <button> Create User / Add User </button>
18  </div>
19  </div>
20

```

```
6 <select name="status" id="status">
7 <option value="active">Active</option>
8 <option value="inactive">Inactive</option>
9 </select>
```

```

app.component.html / div.container / app-adduser
class="container">
app-adduser</app-adduser>

```

Probamos


John

active
inactive
active

Mark

Steve

que se quiere es crear un usuario y que se muestra en la lista



```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { User } from '../models/user';

@Injectable({
  providedIn: 'root'
})
export class UserService {
  private apiUrl = 'http://localhost:3000/users';

  constructor(private http: HttpClient) {}

  getUsers(): Observable<User[]> {
    return this.http.get<User[]>(this.apiUrl);
  }

  createUser(user: User): Observable<User> {
    return this.http.post<User>(this.apiUrl, user);
  }
}
```

vamos a crear un método que va a recibir dos parámetros en nombre y status
 arreglo users utilizo el método push para agregar un objeto que contiene
 nombre y el status.

[illegible]

Run Terminal Help • app.modules.ts - UserService - Visual Studio Code

```
src > app > TS app.module.ts > AppModule
1 import { NgModule } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
```

```
19  })
20  export class AppModule { }
21
```



```
export class AddUserComponent implements OnInit {
  username: string = '';
  status: string = 'active';
}
```

help • address.component.html - UserService - Visual Studio Code

metodo "AddUser" de su componente adduser.component.ts

```

11 styleUrls: [ './assets/component.css' ],
12 providers: [ UserService ]
13 })
14
15 export class AdduserComponent implements OnInit {
16     username: string = '';

```

↓

[illegible]

```
{ Component, OnInit } from '@angular/core';
{ UserService } from '../Services/user.service';
```

1- Estamos creando una instancia del s

↓

[illegible]

```
console.log(this.USER_DEFAULT.users);
```

```
export class AppComponent implements OnInit {
  title = 'UserService';
}
```

component.html TS user.service.ts TS app.component.ts **adduser.component.html** TS adduser.component.ts X

```

    @adduser > TS adduser.component.ts > AdduserComponent > constructor
    import { Component, OnInit } from '@angular/core';
  
```

```
AddUser(){
  this.userService.AddNewUser(this.username, this.status);
  console.log(this.userService.users);
}
```

Aquí que cuando estamos agregando nuevos usuarios

```
service.AddNewUser(this.username, this.status);  
(this.userService.users);
```

Active	
Create User	
hn	active
ark	inactive
eve	active

as por eso que no estamos viendo los usuarios que se estan agregando en la página

ib

↓



browser address bar: `app.component.ts` `app.component.ts` `adduser.component.html` `adduser`