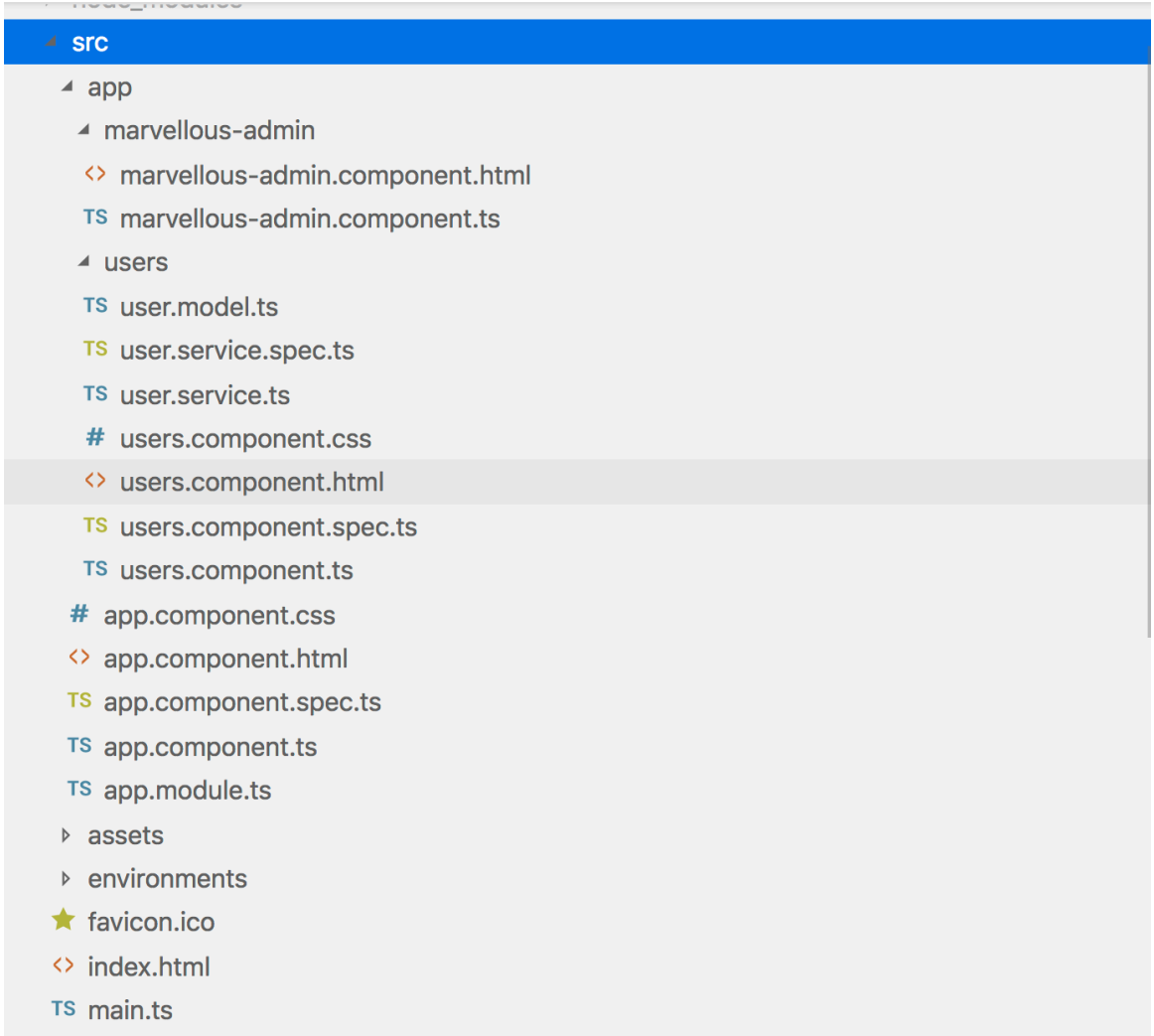
Augury

- Augury is a tool designed specifically for Angular apps.
- It's an open-source debugging and profiling tool for Angular 7 applications.
- Augury is just a Chrome extension that's quite simple to use.
- Augury visualizes our app's structure in a form of a tree, showing how components and their dependencies relate to each other.
- It also allows us to inspect properties of our objects and change them on the fly.
- On top of that, we can easily view the source code of a specific component, insert breakpoints as needed, work with events, and more.
- Lastly, We can browse the application's routing system, as well as view the full list of all utilized modules.
- We can install Augury from https://augury.rangle.io/
- After that, we may open the developer tools by pressing Ctrl + Shift + I (Windows/Linux) or Cmd + Opt + I (macOS).
- On browser new tab called Augury has appeared.
- After switching to this tab, we will either see the application's structure or the phrase "This application is not an Angular application".

Consider our Angular application MarvellousAugury to understand working of Augury.

About **MarvellousAugury** Application :

Consider below project layout of our application.



- This application contains two custom components as MarvellousAdmin & User.
- There is one model named as User model. Which contains one class which maintains information about user.

```
export class User
{
  id: number = 0;
  first: string = '';
  last: string = '';
}
```

- There is one user defined service named as UserService. This service return a list of hardcoded users.

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

import { User } from './user.model';
```

```
@Injectable()
export class UserService
{

  constructor() { }

  getUsers(): User[] {
    return [
      {
        id: 1,
        first: 'Piyush',
        last: 'Khairnar'
      },
      {
        id: 2,
        first: 'Pooja',
        last: 'Khairnar'
      },
      {
        id: 3,
        first: 'Madhavi',
        last: 'Khairnar'
      }
    ]
  }
}
```

- we allowing the users to be selected by clicking on them. The currently selected user is stored in a separate selectedUser attribute.

```
<div *ngFor="let user of users" (click)="onSelect(user)"
[class.selected]="user === selectedUser">
        <p>
                {{user.last}}, {{user.first}} (ID: {{user.id}})
        </p>
</div>

<div *ngIf="selectedUser">

  <h3>Edit Information of Marvellous Users</h3>

  <label for="first"> First Name </label>

  <input [(ngModel)]="selectedUser.first" placeholder="First name" id="first">

  <label for="last"> Last Name </label>

  <input [(ngModel)]="selectedUser.last" placeholder="Last name" id="last">

</div>
```

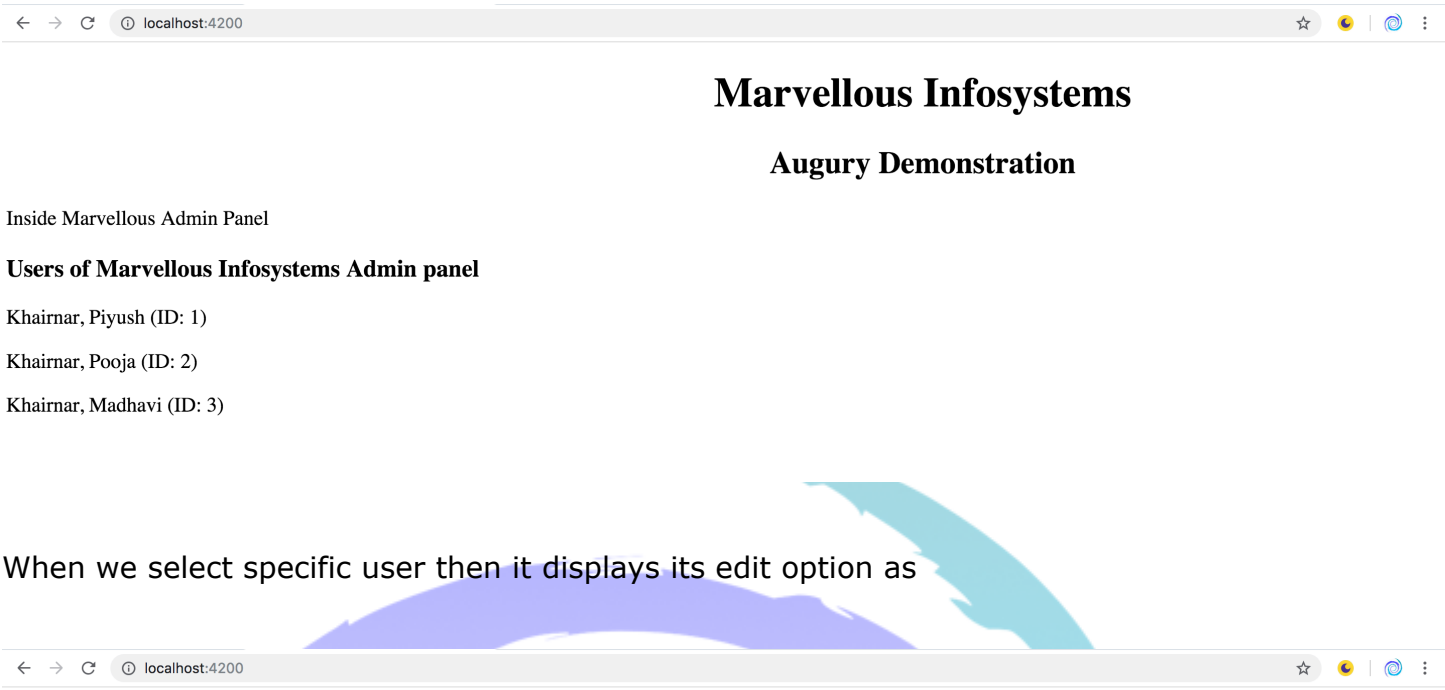
• We access userService from User component using the dependency injection mechanism.

```
import { Component, OnInit } from '@angular/core';

import { User } from './user.model';

import { UserService } from './user.service';

@Component({
  selector: 'app-users',
  templateUrl: './users.component.html',
  styleUrls: ['./users.component.css']
})
export class UsersComponent implements OnInit
{
  users: User[];

  selectedUser: User;

  onSelect(user: User): void
  {
    this.selectedUser = user;
  }

  constructor(private userService: UserService) { }

  ngOnInit()
  {
    this.getUsers();
  }

  getUsers(): void
  {
    this.users = this.userService.getUsers();
  }
}
```

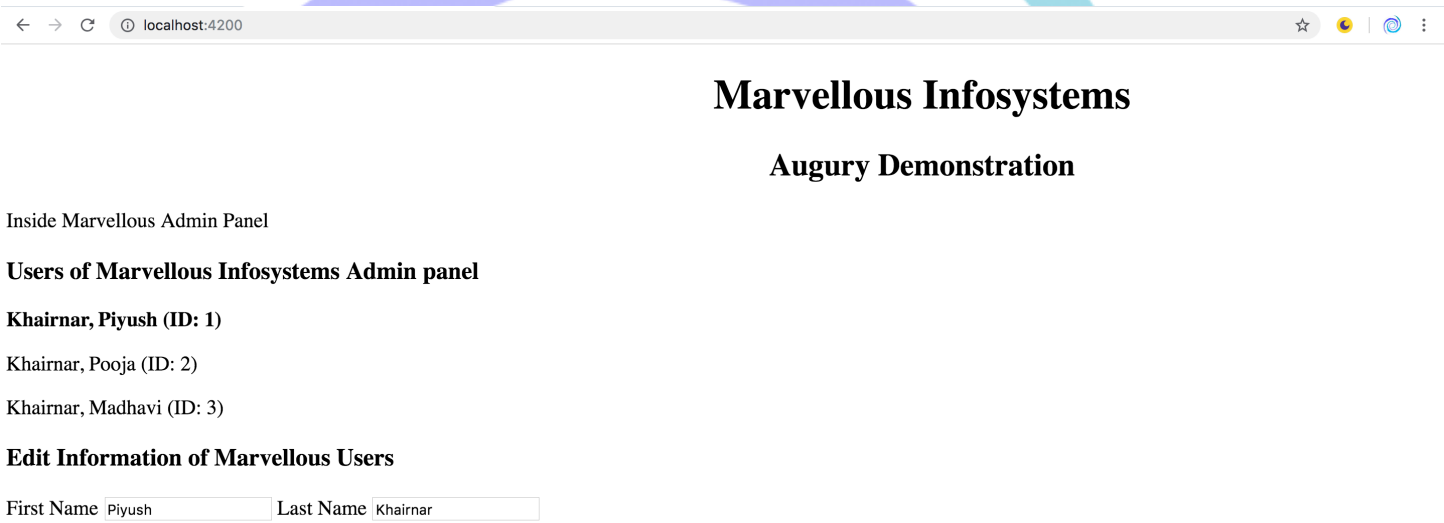
• Both the components User and MarvellousAdmin are rendered from app component.

```
<div style="text-align:center">

  <h1>
    Marvellous Infosystems
  </h1>

  <h2>
    Augury Demonstration
  </h2>

</div>

<app-marvellous-admin></app-marvellous-admin>

<h3>Users of Marvellous Infosystems Admin panel</h3>

<app-users></app-users>
```

Our application looks like

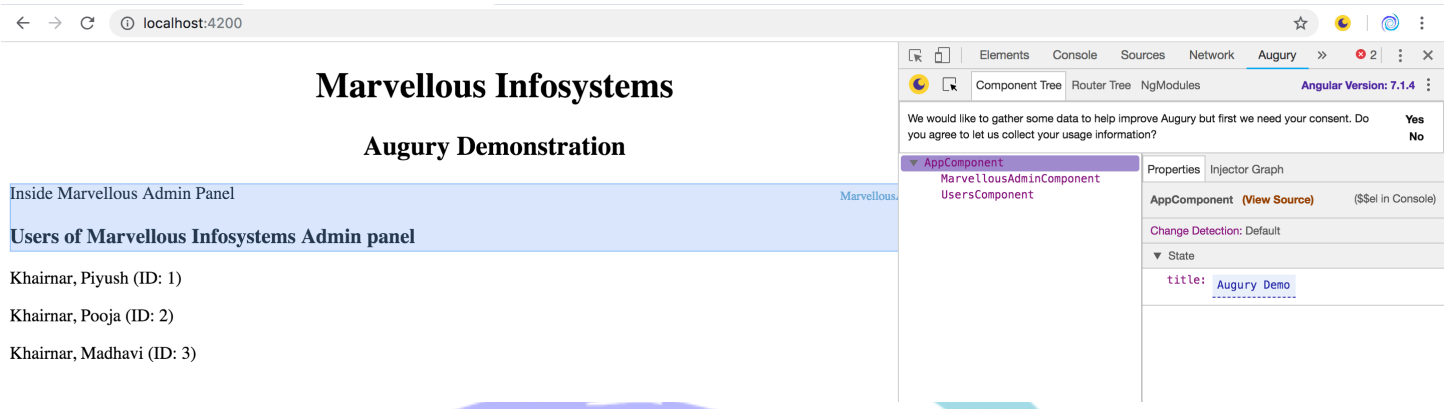


When we select specific user then it displays its edit option as

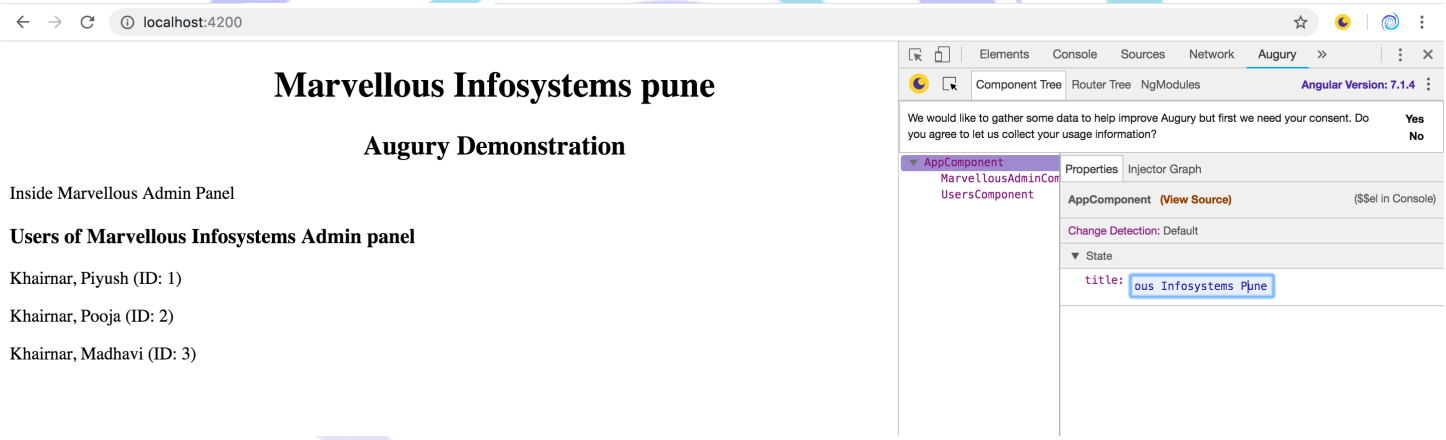


Now we use above MarvellousAugury Application to demonstrate features of Augury

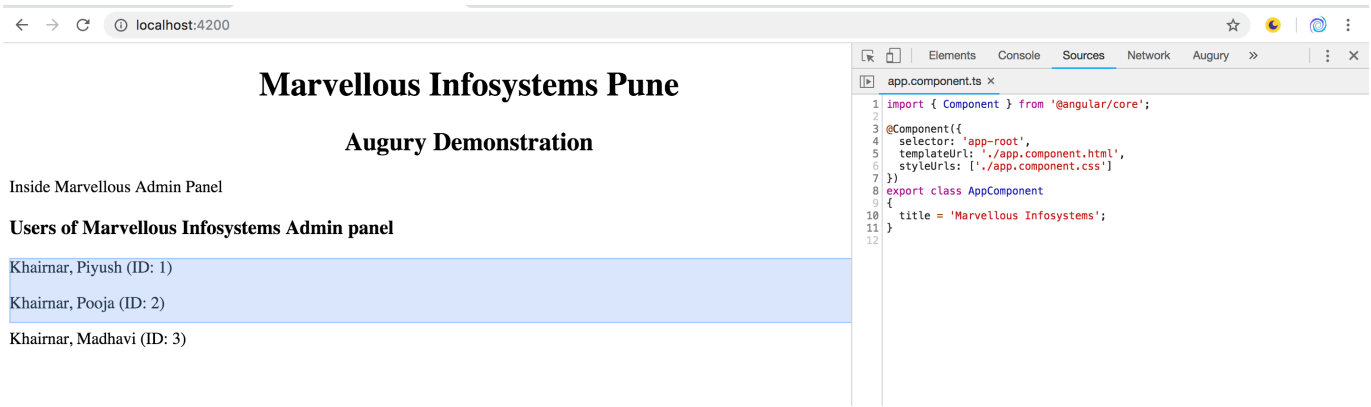
For component view Press Ctrl + Shift + I or Cmd + Opt + I, switch to the Augury tab, and press AppComponent in the left pane, under the Component Tree



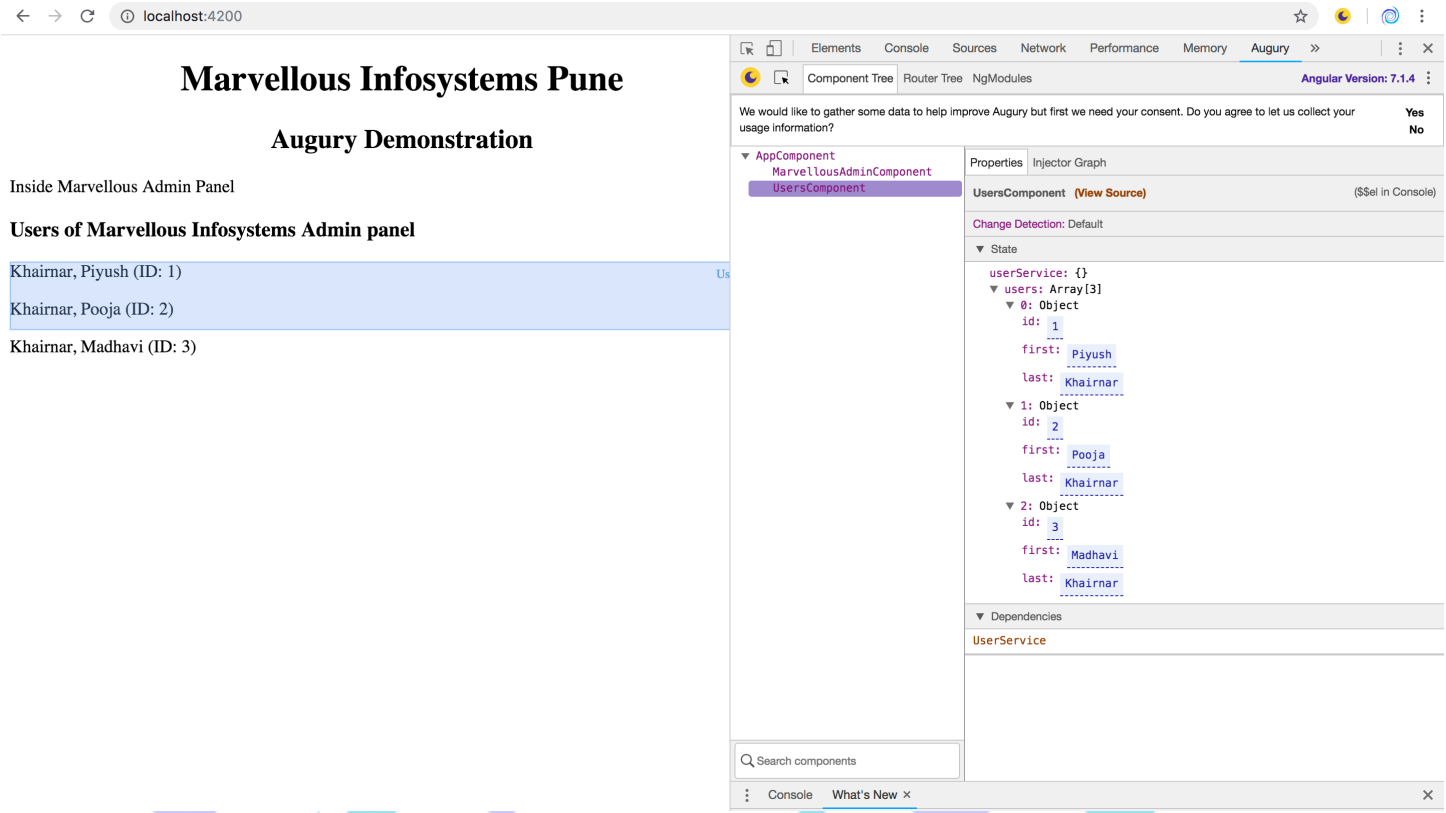
We can also change the contents from Augury panel so that changes gets reflected in the targeted output.



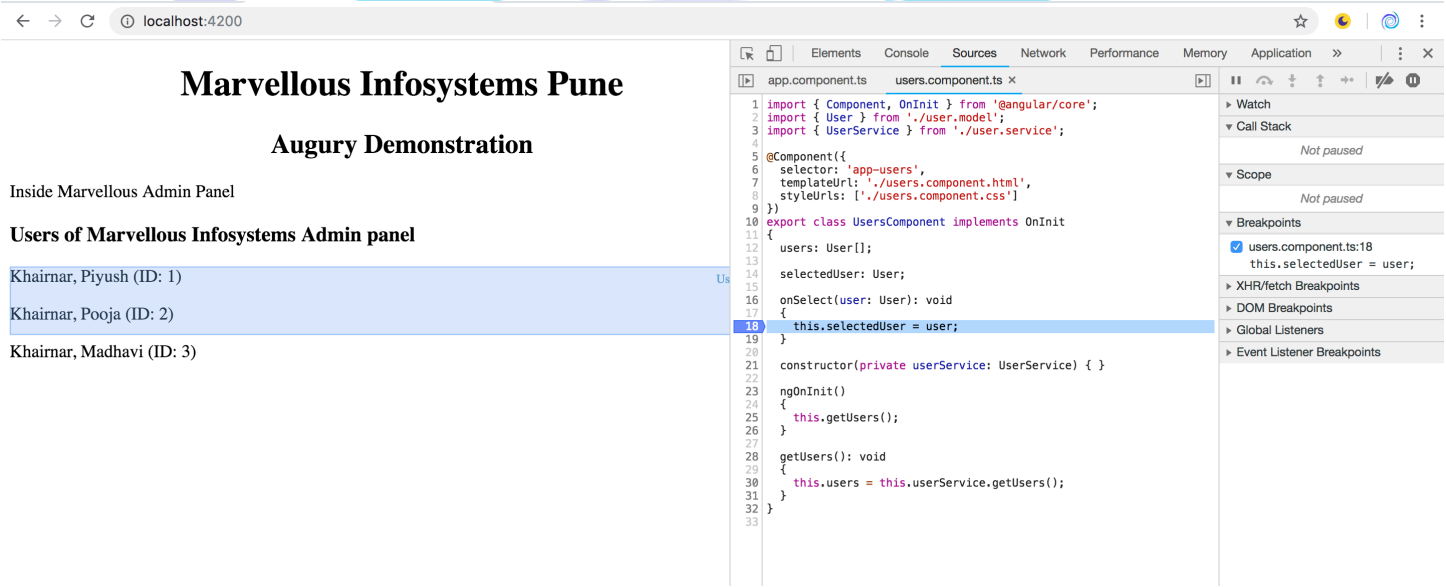
When we click one view source link it displays source code of our component.



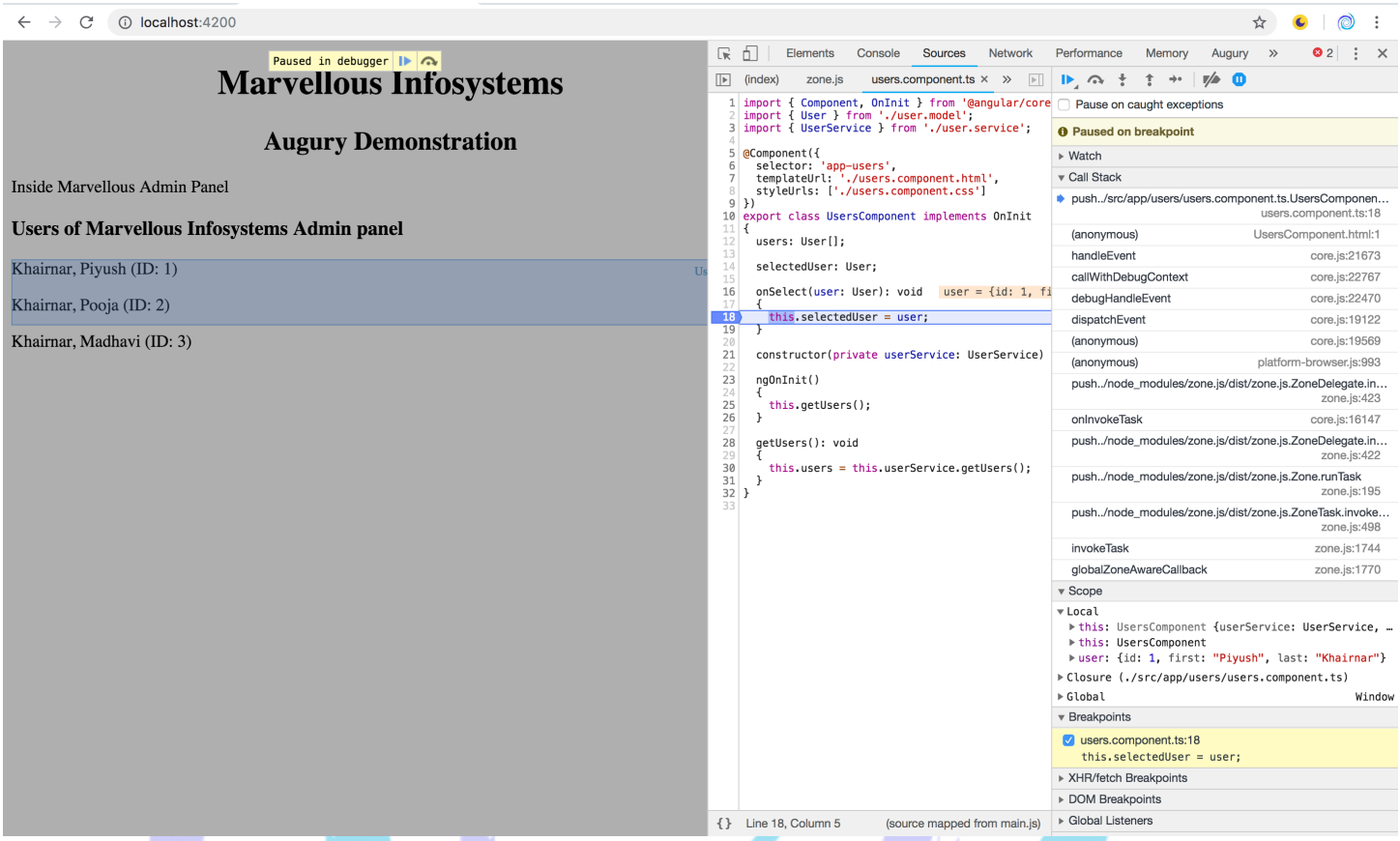
When we select User component then we get all dependencies and values from the array which is returned by the service.



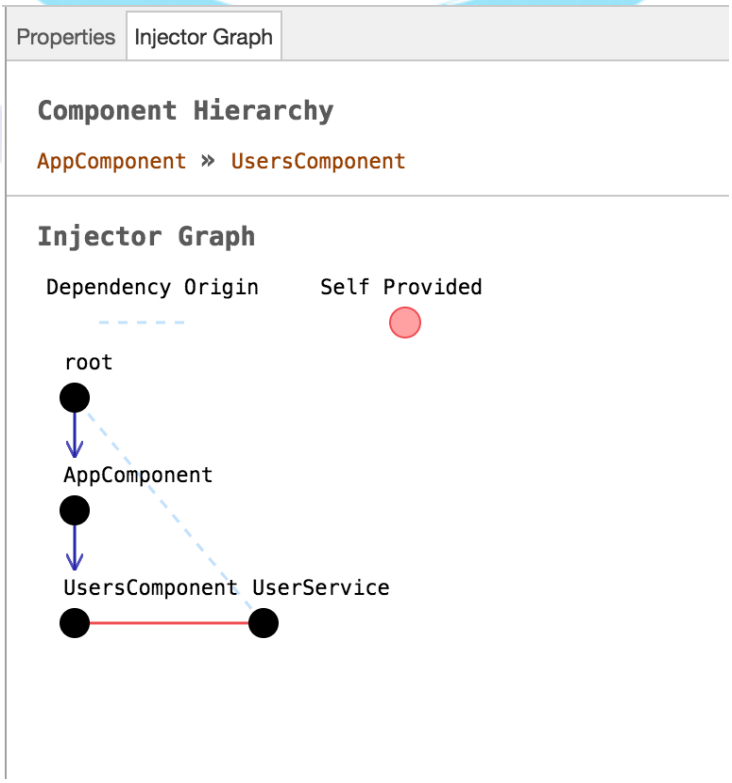
From the source code of user component we can apply breakpoint by clicking on left side of the line.



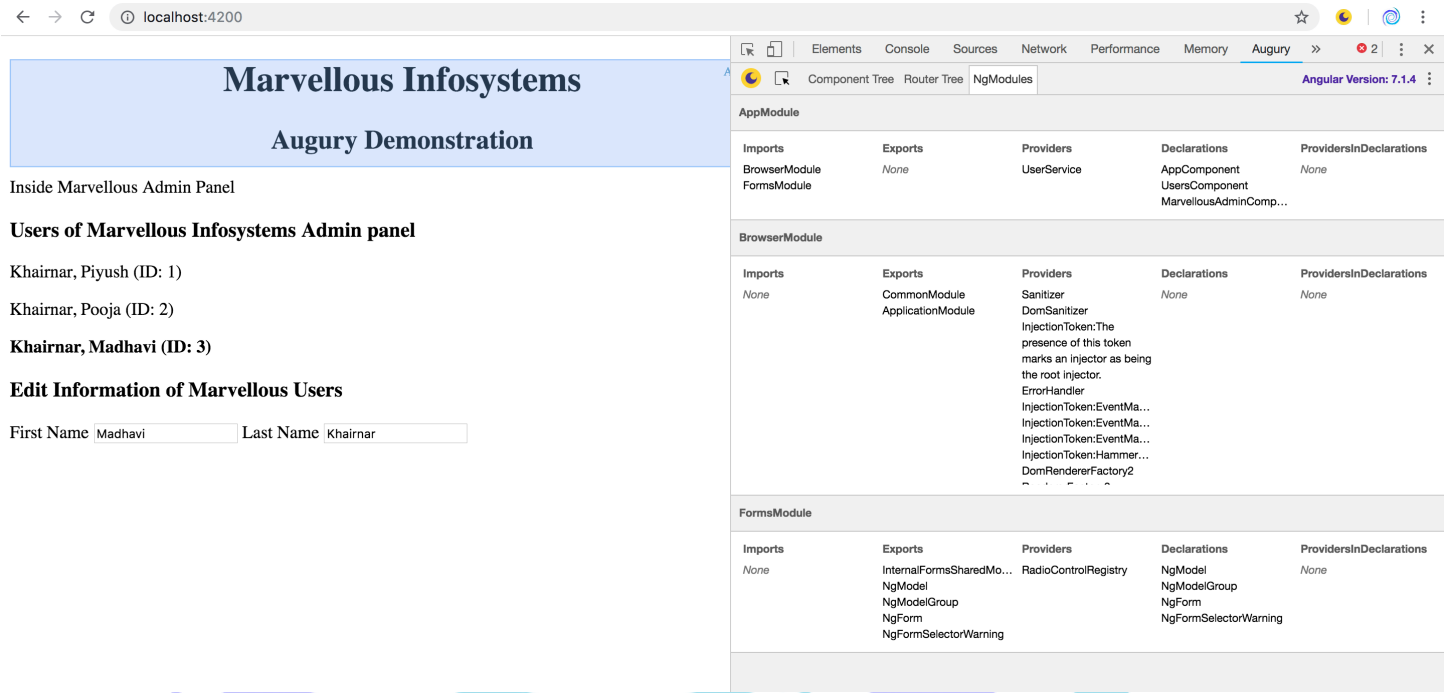
After applying the breakpoint when we click specific user then our code execution automatically stops at the breakpoint.



As we inject User service into user component we can get that information by selecting injector graph of Augury.



We can also check list of all used modules of our application by selecting ngModule tab.



If our application contains routes then using augury we can get the information about all the routes.

As there is no route in our application we use another Angular application as MarvellousWildcards which contains 4 routes.