

- 1. Components
- 2. HeroesSearchComponent

- [Info](#)
- [Source](#)
- [Template](#)
- [Styles](#)
- [DOM Tree](#)

File

src/app/components/heroes-search/heroes-search.component.ts

Description

Component for heroes search

Implements

[OnInit](#)

Metadata

selector app-heroes-search
styleUrls ./heroes-search.component.scss
templateUrl ./heroes-search.component.html

Index

Properties

- Public [data](#)
- Public [hero](#)
- Public [term](#)
- Public [user](#)

Methods

- [getInfo](#)
- [ngOnInit](#)

Constructor

constructor(_heroService: [HeroService](#), _activatedRoute: [ActivatedRoute](#), _userService: [UserService](#))
Defined in [src/app/components/heroes-search/heroes-search.component.ts:33](#)

Constructor in which we inject hero service, formBuilder and router service

Parameters :

Name	Type	Optional
_heroService	HeroService	No
_activatedRoute	ActivatedRoute	No
_userService	UserService	No

Methods

getInfo

`getInfo()`

Defined in [src/app/components/heroes-search/heroes-search.component.ts:58](#)

Function to get results of the search

Returns : [void](#)

ngOnInit

`ngOnInit()`

Defined in [src/app/components/heroes-search/heroes-search.component.ts:47](#)

Start when the component inits

Returns : [void](#)

Properties

Public data

Type : [string](#)

Default value : ''

Defined in [src/app/components/heroes-search/heroes-search.component.ts:29](#)

variable to store data from search

Public hero

Type : [Hero\[\]](#)

Defined in [src/app/components/heroes-search/heroes-search.component.ts:21](#)

variable to store the hero from the search

Public term

Type : [string](#)

Default value : ''

Defined in [src/app/components/heroes-search/heroes-search.component.ts:33](#)

variable to store term from search

Public user

Type : [User\[\]](#)

Defined in [src/app/components/heroes-search/heroes-search.component.ts:25](#)

variable to store the user from the search

```
import { Component, OnInit } from '@angular/core'
import { Hero } from '../models/hero'
import { HeroService } from 'src/app/services/hero.service'
import { ActivatedRoute } from '@angular/router'
import { UserService } from 'src/app/services/user.service'
import { User } from 'src/app/models/user'

/**
 * Component for heroes search
 */
@Component({
  selector: 'app-heroes-search',
  templateUrl: './heroes-search.component.html',
  styleUrls: ['./heroes-search.component.scss']
})

export class HeroesSearchComponent implements OnInit {
  /**
   * variable to store the hero from the search
   */
  public hero: Hero[]
  /**
   * variable to store the user from the search
   */
  public user: User[]
  /**
```

```

    * variable to store data from search
    */
public data: string = ''
/**
    * variable to store term from search
    */
public term: string = ''

/**
    * Constructor in which we inject hero service, FormBuilder and router service
    */
constructor(
    private _heroService: HeroService,
    private _activatedRoute: ActivatedRoute,
    private _userService: UserService
) { }

/**
    * Start when the component inits
    */
ngOnInit() {
    this._activatedRoute.params.subscribe(params => {
        this.data = params['data']
        this.term = params['term']
        this.getInfo()
    })
}

/**
    * Function to get results of the search
    */
getInfo() {
    if (this.data === 'DC') {
        this._heroService.allDCHeroes().subscribe(
            res => {
                this.user = null
                this.hero = res
            },
            error => {
                console.log(error)
            }
        )
    } else if (this.data === 'Marvel') {
        this._heroService.allMarvelHeroes().subscribe(
            res => {
                this.user = null
                this.hero = res
            },
            error => {
                console.log(error)
            }
        )
    } else if (this.data === 'User') {
        this._userService.getUserByName(this.term).subscribe(
            res => {
                this.hero = null
                this.user = res
            },
            error => {
                console.log(error)
            }
        )
    } else {
        this._heroService.getHeroByName(this.term).subscribe(
            res => {
                this.user = null
                this.hero = res
            },
            error => {
                console.log(error)
            }
        )
    }
}
}

<div class="color1">
    <div class="searchFor">
        SEARCHING FOR: <span>{{ term }}</span>
    </div>
    <div class="containerSearch" *ngIf="hero != null && data != 'User'">
        <div class="card2" *ngFor="let hero of hero">
            <img
                *ngIf="hero.image"
                [src]="hero.image"
                [routerLink]="['/heroesDetail', hero.idHero]"
                [alt]="hero.heroName"

```

```

        onError="this.src='../.../assets/img/nogender_2.jpg'"
        class="img-fluid img2"
    />
    <div class="overImg">
        <h4>{{ hero.heroName }}</h4>
    </div>
</div>
</div>
<div *ngIf="hero == null && data != 'User'">
    <h1 class="NotFound">CHARACTER WITH GIVEN NAME NOT FOUND</h1>
</div>
<div class="containerSearch" *ngIf="user && data === 'User'">
    <div class="card2" *ngFor="let usu of user">
        <img
            *ngIf="usu.photo"
            [src]="usu.photo"
            [routerLink]="['/publicProfile', usu.idUsu]"
            [alt]="usu.alias"
            onError="this.src='../.../assets/img/nogender_2.jpg'"
            class="img-fluid img2"
        />
        <div class="overImg">
            <h4>{{ usu.alias }}</h4>
        </div>
    </div>
</div>
<div *ngIf="!user && data === 'User'">
    <h1 class="NotFound">USER WITH GIVEN ALIAS NOT FOUND</h1>
</div>
</div>

```

```
./heroes-search.component.scss
```

```

.containerSearch {
    display: flex;
    flex-direction: row;
    justify-content: space-around;
    flex-wrap: wrap;
}

```

```

.searchFor{
    text-align: center;
    font-family: "B612";
    color: #00a23eb6;
    padding-bottom: 2vw;
    font-size: 1.5rem;
}

```

```

.searchFor span {
    color: #cc4224;
}

```

```

.NotFound{
    text-align: center;
    font-family: "B612";
    color: #cc4224;
}

```

```

@media (min-width: 992px) {

    .containerSearch {
        flex-direction: row;
    }
}

```

Legend

Html element

Component

Html element with directive

result-matching ""

No results matching ""