frontend documentation

Type to search

1. Components
2. MyHeroesComponent

- Info
- Source
- Template
- Styles
- DOM Tree

File

`src/app/components/my-heroes/my-heroes.component.ts`

Description

Component that brings user followed heroes, favorites and user team

Implements

`OnInit`

Metadata

selector	`app-my-heroes`
styleUrls	`./my-heroes.component.scss`
templateUrl	`./my-heroes.component.html`

Index

Properties

- Public dialog
- Public GlobalV
- Public heroesFav
- Public heroesFol
- Public identity
- Public idUser
- Public myTeamInfo
- Public newM
- Public showName
- Public type

Methods

- addMember
- createTeam
- favorites
- following
- getTeamUsu
- getUsu
- ngOnInit

Constructor

```
constructor(dialog: MatDialog, _userService: UserService, _UserHero: UserHeroService, _Team: TeamService, GlobalV:
GlobalVariableService)
```

Defined in src/app/components/my-heroes/my-heroes.component.ts:48

Constructor in which we inject user service, modal material moduler,hero service,team service and global variables services

Parameters :

Name	Type	Optional
dialog	`MatDialog`	No
_userService	`UserService`	No
_UserHero	`UserHeroService`	No
_Team	`TeamService`	No
GlobalV	`GlobalVariableService`	No

Methods

addMember

`addMember(idHero)`

Defined in src/app/components/my-heroes/my-heroes.component.ts:100

Add member to the team

Parameters :

Name	Optional
idHero	No

Returns : `void`

createTeam

`createTeam(statusF)`

Defined in src/app/components/my-heroes/my-heroes.component.ts:142

function to open modal to create a team

Parameters :

Name	Optional
statusF	No

Returns : `void`

favorites

`favorites()`

Defined in src/app/components/my-heroes/my-heroes.component.ts:129

heros favorites

Returns : `void`

following

`following()`

Defined in src/app/components/my-heroes/my-heroes.component.ts:117

heros following

Returns : `void`

getTeamUsu

`getTeamUsu()`

Defined in src/app/components/my-heroes/my-heroes.component.ts:85

Get Team info

Returns : `void`

getUsu

`getUsu()`

Defined in src/app/components/my-heroes/my-heroes.component.ts:77

Get info user

Returns : `void`

ngOnInit

`ngOnInit()`

Defined in src/app/components/my-heroes/my-heroes.component.ts:66

Start when the component inits

Returns : `void`

Properties

Public dialog

Type : `MatDialog`

Defined in src/app/components/my-heroes/my-heroes.component.ts:54

Public GlobalV

Type : `GlobalVariableService`

Defined in src/app/components/my-heroes/my-heroes.component.ts:58

Public heroesFav

Type : `Hero[]`

Defined in src/app/components/my-heroes/my-heroes.component.ts:38

variable to store fav heroes

Public heroesFol

Type : `Hero[]`

Defined in src/app/components/my-heroes/my-heroes.component.ts:34

variable to store followed heroes

Public identity

Defined in src/app/components/my-heroes/my-heroes.component.ts:26

variable to store user identity

Public idUser

Type : `number`

Default value : `0`

Defined in src/app/components/my-heroes/my-heroes.component.ts:42

variable to save id User

Public myTeamInfo

Type : `Team`

Defined in src/app/components/my-heroes/my-heroes.component.ts:30

variable to store team info

Public newM

Type : `number`

Default value : `0`

Defined in src/app/components/my-heroes/my-heroes.component.ts:46

variable to count new members in the team

Public showName

Type : `boolean`

Default value : `true`

Defined in src/app/components/my-heroes/my-heroes.component.ts:48

Public type

Type : `boolean`

Default value : `true`

Defined in src/app/components/my-heroes/my-heroes.component.ts:47

```
import { Component, OnInit } from '@angular/core'
import { HeroService } from 'src/app/services/hero.service'
import { UserHeroService } from 'src/app/services/user-hero.service'
import { TeamService } from 'src/app/services/team.service'
import { Team } from 'src/app/models/team'
import { Hero } from 'src/app/models/hero'
import { UserService } from 'src/app/services/user.service'
import { MatDialog } from '@angular/material'
import { TeamDialogComponent } from '../modals/team-dialog/team-dialog.component'
import { Observable } from 'rxjs'
import { GlobalVariableService } from 'src/app/services/global-variable.service'

/**
 * Component that brings user followed heroes, favorites and user team
 */
@Component({
  selector: 'app-my-heroes',
  templateUrl: './my-heroes.component.html',
  styleUrls: ['./my-heroes.component.scss']
})

export class MyHeroesComponent implements OnInit {
  /**
   * variable to store user identity
   */
  public identity
  /**
   * variable to store team info
   */
  public myTeamInfo: Team
  /**
   * variable to store followed heroes
   */
  public heroesFol: Hero[]
  /**
   * variable to store fav heroes
   */
  public heroesFav: Hero[]
  /**
   * variable to save id User
   */
  public idUser: number = 0
  /**
   * variable to count new members in the team
   */
  public newM: number = 0
  public type: boolean = true;
  public showName: boolean = true;

  /**
   * Constructor in which we inject user service, modal material moduler,hero service,team service and global variables services
   */
  constructor(
    public dialog: MatDialog,
    private _userService: UserService,
    private _UserHero: UserHeroService,
    private _Team: TeamService,
    public GlobalV: GlobalVariableService
  ) {
    this.myTeamInfo = new Team(0, 0, '', '', '', '', '', '')
  }

  /**
   * Start when the component inits
   */
  ngOnInit() {
    this.getUsu()
    this.getTeamUsu()
    this.following()
    this.favorites()
    this.GlobalV.countTeamMembers = 0
  }

  /**
   * Get info user
   */
  getUsu() {
    this.identity = this._userService.getIdentity()
    this.idUser = this.identity.id
  }

  /**
   * Get Team info
   */
  getTeamUsu() {
    this._Team.getTeamInfo(this.identity.id).subscribe(
      res => {
        this.myTeamInfo = res
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Add member to the team
   * @param {number} idHero
   */
  addMember(idHero) {
    var data = { member: this.GlobalV.memberTeamNUll, codHero: idHero }
    this._Team.addMember(this.myTeamInfo.idTeam, data).subscribe(
      res => {
        this.getTeamUsu()
        this.GlobalV.countTeamMembers++
        this.newM = this.newM + 1
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * heros following
   */
  following() {
    this._UserHero.allHerosFoll(this.identity.id).subscribe(res => {
      this.heroesFol = res
    }),
      error => {
        console.log(error)
      }
  }

  /**
   * heros favorites
   */
  favorites() {
    this._UserHero.allHerosFav(this.identity.id).subscribe(res => {
      this.heroesFav = res
    }),
      error => {
        console.log(error)
      }
  }

  /**
   * function to open modal to create a team
   *  @param {string} statusF
   */
  createTeam(statusF): void {
    const dialogRef = this.dialog.open(TeamDialogComponent, {
      data: {
        idUsu: this.identity.id,
        status: statusF,
        teamInfo: this.myTeamInfo
      }
    })
    dialogRef.afterClosed().subscribe(result => {
      console.log('The dialog was closed')
      // this.idUser = this.identity.id
      if (result) {
        this.idUser = this.identity.id
        this.getTeamUsu();
      }
      // this.getTeamUsu();
    })
  }
}

<div class="color1">
  <div class="myTeam">
    <h3>MY TEAM</h3>
    <img
      *ngIf="myTeamInfo"
      class="iconBuuton"
      (click)="createTeam('delete')"
      src="../../../assets/img/trash.svg"
      alt=""
    />

    <img
      *ngIf="myTeamInfo"
      class="iconBuuton"
      (click)="createTeam('modify')"
      src="../../../assets/img/job.svg"
      alt=""
    />
  </div>
  <h5>
    Create your own team, you can add members that you have in
    <span>favorites</span> and remove them from the team whenever you want, you
    can have a maximum of 5 members in your team
  </h5>
  <br />
  <app-team
    [idUsu]="idUser"
    [newM]="newM"
    [type]="type"
    [showName]="showName"
  ></app-team>

  <button
    *ngIf="!myTeamInfo"
    type="button"
    class="UnfollowB"
    mat-raised-button
    (click)="createTeam('new')"
  >
    <span>CREATE TEAM</span>
  </button>
</div>
<div class="color2">
  <h3 class="title">My favorites</h3>
  <div class="containerSearch">
    <div class="card2" *ngFor="let hero of heroesFav">
      <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 rela">
        <h4>{{ hero.heroName }}</h4>
        <img
          matTooltip="Add member to your team"
          *ngIf="GlobalV.countTeamMembers < 5 && myTeamInfo"
          class="iconBuuton2"
          (click)="addMember(hero.idHero)"
          src="../../../assets/img/x.svg"
          alt=""
        />
      </div>
    </div>
  </div>
</div>
<div class="color1">
  <h3 class="title">FOLLOWING</h3>
  <div class="containerSearch">
    <div class="card2" *ngFor="let hero of heroesFol">
      <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>
```

./my-heroes.component.scss

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

.title {
  color: #00a23d;
  font-family: "B612";
  padding-bottom: 1em;
}

h3 {
  color: #00a23d;
  font-family: "B612";
}

.iconBuuton {
  cursor: pointer;
  height: 8vw;
}
.myTeam {
  display: flex;
  flex-direction: row;
}

.iconBuuton2 {
  cursor: pointer;
  height: 5.5vw;
  z-index: 4;
  right: 2vw;
  top: 9.5vw;
  position: absolute;
  transform: rotate(45deg);
}

h5 {
  color: rgb(223, 221, 221);
}
h5 span {
  color: #00a23d;
}

// button
$bg: #1a1a1a;
$fg: #f3d403;
$border-width: 0.2rem;
$corner-size: 3rem;
$dur: 0.3s;

.followB,
.UnfollowB {
  margin: 0 0.5vw 0 0.5vw;
  outline: none;
  font-family: "B612";
  letter-spacing: 0.02rem;
  cursor: pointer;
  background: transparent;
  border: $border-width solid currentColor;
  padding: 0.1rem 0.5rem;
  font-size: 1rem;
  color: $fg;
  position: relative;
  transition: color $dur;

  &:hover {
    color: #cc4224;
    &::before {
      width: 0;
    }
    &::after {
      height: 0;
    }
  }
  &:active {
    border-width: $border-width / 2;
  }

  span {
    position: relative;
    z-index: 2;
  }
  &::before,
  &::after {
    content: "";
    position: absolute;
    background: $bg;
    z-index: 1;
    transition: all $dur;
  }
  &::before {
    width: calc(100% - #{$corner-size});
    height: calc(101% + #{$border-width * 2});
    top: -$border-width;
    left: 50%;
    transform: translateX(-50%);
  }
  &::after {
    height: calc(100% - #{$corner-size});
    width: calc(101% + #{$border-width * 2});
    left: -$border-width;
    top: 50%;
    transform: translateY(-50%);
  }
}

@media (min-width: 992px) {
  .containerSearch {
    flex-direction: row;
  }

  .iconBuuton2 {
    height: 1.5vw;
    right: 0.5vw;
    top: 2.5vw;
  }
  .iconBuuton {
    cursor: pointer;
    height: 2vw;
  }
}
```

Legend
Html element
Component
Html element with directive

result-matching ""

No results matching ""