File

`src/app/components/modals/avatar-dialog/avatar-dialog.component.ts`

Description

Component for change avatar

Implements

`OnInit`

Metadata

selector	`app-avatar-dialog`
styleUrls	`./avatar-dialog.component.scss`
templateUrl	`./avatar-dialog.component.html`

Index

Properties

- activeState
- Public data
- Public dialogRef
- Public heroImg
- Public img

Methods

- getProfilesimg
- ngOnInit
- onNoClick
- selectHero
- updateImgProfile

Constructor

`constructor(dialogRef: MatDialogRef, _UserService: UserService, _HeroService: HeroService, data: any)`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:24

Constructor in which we inject our services and different elements

Parameters :

Name	Type	Optional
dialogRef	`MatDialogRef<AvatarDialogComponent>`	No
_UserService	`UserService`	No
_HeroService	`HeroService`	No
data	`any`	No

Methods

getProfilesimg

`getProfilesimg()`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:56

function to get img profile user

Returns : `void`

ngOnInit

`ngOnInit()`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:39

Start when the component inits

Returns : `void`

onNoClick

`onNoClick()`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:84

function to close modal

Returns : `void`

selectHero

`selectHero(imgHero)`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:48

function to change img hero variable

Parameters :

Name	Optional
imgHero	No

Returns : `void`

updateImgProfile

`updateImgProfile()`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:68

function to update img

Returns : `void`

Properties

activeState

Type : `string`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:16

Public data

Type : `any`

Decorators :

`@Inject(MAT_DIALOG_DATA)`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:33

Public dialogRef

Type : `MatDialogRef<AvatarDialogComponent>`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:30

Public heroImg

Type : `any[]`

Default value : `[]`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:20

variable to store heros url img

Public img

Type : `string`

Default value : `""`

Defined in src/app/components/modals/avatar-dialog/avatar-dialog.component.ts:24

to store user img

```
import { Component, OnInit, Inject } from '@angular/core';
import { MatDialogRef, MAT_DIALOG_DATA } from "@angular/material/dialog";
import { HeroService } from 'src/app/services/hero.service';
import { UserService } from 'src/app/services/user.service';

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

export class AvatarDialogComponent implements OnInit {
  activeState: string;
  /**
   * variable to store heros url img
   */
  public heroImg: any[] = [];
  /**
   * to store user img
   */
  public img: string = "";

  /**
 * Constructor in which we inject our services and different elements
 */
  constructor(
    public dialogRef: MatDialogRef<AvatarDialogComponent>,
    private _UserService: UserService,
    private _HeroService: HeroService,
    @Inject(MAT_DIALOG_DATA) public data: any
  ) { }

  /**
 * Start when the component inits
 */
  ngOnInit() {
    this.getProfilesimg()
    this.img = this.data.userPhoto;
  }

  /**
 * function to change img hero variable
 *   @param {string} imgHero
 */
  selectHero(imgHero) {
    this.img = imgHero;
    this.activeState = imgHero;
  }

  /**
  * function to get img profile user
  */
  getProfilesimg() {
    this._HeroService.profileImgHeroes().subscribe(res => {
```

```
      this.heroImg = res;
      this.heroImg.push({image:'./../../../../assets/img/nomember.jpg'})
    }, err => {
      console.log(err)
    })
  }

  /**
  * function to update img
  */
  updateImgProfile() {

    var userImgObj = { idUsu: this.data.userId, img: this.img };

    this._UserService.newImg(userImgObj).subscribe(res => {

      localStorage.setItem('UserPhoto', this.img)
    }, err => {
      console.log(err)
    })
    this.dialogRef.close();
  }

  /**
    * function to close modal
    */
  onNoClick(): void {
    this.dialogRef.close();
  }
}

<div id="containerModal">
  <h1 mat-dialog-title>SELECT YOUR HERO</h1>
  <div mat-dialog-content>
    <div class="imgContainer">
      <div
        class="imgRedonda"
        *ngFor="let hero of heroImg"
        [ngClass]="{ red: activeState == hero.image }"
      >
        <img
          (click)="selectHero(hero.image)"
          [src]="hero.image"
          [alt]="hero.image"
          class="img-fluid"
        />
      </div>
    </div>
  </div>
  <div class="buttContainer">
    <button class="cancel" mat-button [mat-dialog-close]="">No Thanks</button>
    <button type="button" mat-raised-button (click)="updateImgProfile()">
      Send avatar
    </button>
  </div>
</div>

./avatar-dialog.component.scss

.imgRedonda {
  height: 6em;
  width: 6em;
  background-repeat: no-repeat;
  background-position: 50%;
  border-radius: 50%;
  background-size: 100% auto;
  overflow: hidden;
  border: 2.5px solid #a30023;
  margin: 0.1em;
}
.imgContainer {
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
  flex-direction: row;
}

#containerModal {
  font-family: "B612";
}

#containerModal h1 {
  color: #00a23d;
  font-size: 1.5rem;
}
```

```
#containerModal button {
  outline: none;
  background-color: #f3d403;
  border: none;
  border-radius: 10px;
  border: 2.5px solid #00a23d;
  padding: 1.5vw;
  font-size: 0.8rem;
}
.buttContainer {
  padding-top: 2vw;
  display: flex;
  justify-content: space-between;
}

#containerModal button:hover {
  background-color: #cc4224;
}

.cancel {
  background-color: #2a75b3 !important;
}

.red {
  border: 3px solid #00a23d;
}
.mat-dialog-content {
  max-height: 39vh !important;
  width: 40vh !important;
}
@media (min-width: 992px) {
  .mat-dialog-content {
    max-height: 39vh !important;
    width: 90vh !important;
  }
  #containerModal h1 {
    font-size: 2.5rem;
  }

  #containerModal button {
    font-size: 1rem;
    padding: 0.5vw;
  }
}
```

Legend
Html element
Component
Html element with directive

result-matching ""

No results matching ""