

1. Components
2. MatDialogComponent

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

File

src/app/components/modals/fight-dialog/fight-dialog.component.ts

Implements

[OnInit](#)

Metadata

selectorapp-fight-dialog

styleUrls./fight-dialog.component.scss

templateUrl./fight-dialog.component.html

Index

Properties

- Public [data](#)
- Public [dialogRef](#)
- Public [heroes](#)
- Public [HeroSelectId](#)
- Public [sessonHero](#)

Methods

- [getHeroByName](#)
- [ngOnInit](#)
- [onNoClick](#)
- Async [saveHeroStorage](#)
- [selectHero](#)

Constructor

constructor(dialogRef: [MatDialogRef](#), \_HeroService: [HeroService](#), data: [any](#))

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:17](#)

Parameters :

Name	Type	Optional
dialogRef	<a href="#">MatDialogRef&lt;FightDialogComponent&gt;</a>	No
_HeroService	<a href="#">HeroService</a>	No
data	<a href="#">any</a>	No

Methods

getHeroByName

getHeroByName (name)

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:49](#)

Parameters :

Name Optional

name No

Returns : [void](#)

ngOnInit

ngOnInit ()

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:27](#)

Returns : [void](#)

onNoClick

onNoClick ()

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:115](#)

## Async saveHeroStorage

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:79](#)

Name Optional

**Returns :** any

```
selectHero(idHero)
```

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:64](#)

Name Optional

idHero No

**Returns :** void

## Public data

Type : any

### Decorators :

```
@Inject (MAT_DIALOG_DATA)
```

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:22](#)

Public dialogRef

Type : [MatDialogRef<FightDialogComponent>](#)

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:19](#)

## Public heroes

Type : Hero[]

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:15](#)

**Public HeroSelectId**

*Type :* number

*Default value : 0*

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:17](#)

## Public sessionHero

Type : Hero

Defined in [src/app/components/modals/fight-dialog/fight-dialog.component.ts:14](#)

```

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 { Hero } from 'src/app/models/hero';
import { FormBuilder, Validators, FormGroup } from '@angular/forms';

@Component({
  selector: 'app-fight-dialog',
  templateUrl: './fight-dialog.component.html',
  styleUrls: ['./fight-dialog.component.scss']
})
export class FightDialogComponent implements OnInit {
  public sessionHero: Hero;
  public heroes: Hero[];
  // public searchForm: FormGroup;
  public HeroSelectId: number=0;
  // public sessionHero2: Hero;
  constructor(public dialogRef: MatDialogRef<FightDialogComponent>,
    // private formBuilder: FormBuilder,
    private _HeroService: HeroService,
    @Inject(MAT_DIALOG_DATA) public data: any) {
    this.sessionHero = new Hero(0, '', '', 0, 0, 0, 0, 0, 0, 0, 0, '', '', '', '', '', '', '', '', '', '', null, 0)
    // this.sessionHero2 = new Hero(0, '', '', 0, 0, 0, 0, 0, 0, 0, 0, '', '', '', '', '', '', '', '', '', null, 0)
  }

  ngOnInit() {
    // this.searchForm = this.formBuilder.group({
    //   heroName: [
    //     '',
    //     [Validators.required, Validators.minLength(1), Validators.maxLength(100)]
    //   ]
    // });
  }

  // getErrorMessage(dato) {
  //   var result: string;
  //   if (this.searchForm.controls[dato].hasError('required')) {
  //     return (result = 'This information is required');
  //   } else if (this.searchForm.controls[dato].hasError('minlength')) {
  //     return (result = 'You must enter at least 1 characters');
  //   } else if (this.searchForm.controls[dato].hasError('maxlength')) {
  //     return (result = 'The maximum of characters is 100');
  //   } else {
  //     return (result = '');
  //   }
  // }

  getHeroByName(name) {

```

```

console.log("llego")
this._HeroService.getHeroByName(name).subscribe(res => {

    // console.log(res.idHero)
    this.heroes = res;
    console.log(JSON.stringify(this.heroes, null, 2));
    // console.log(this.hero);
},
    error => {
        console.log(error)
    }
)

selectHero(idHero){
    this.HeroSelectId=idHero
}
// getHero(id) {
//     this._HeroService.getHeroById(id).subscribe(res => {
//         this.sessonHero = res;
//         // console.log(JSON.stringify(this.hero, null, 2));
//     },
//     error => {
//         console.log(error)
//     }
// )
// }

async saveHeroStorage(id) {
    // await this.getHero(id)
    this._HeroService.getHeroById(id).subscribe(res => {
        this.sessonHero = res;
        if (this.data.position === 1) {
            console.log("hero1")
            console.log(this.sessonHero)
            sessionStorage.setItem('Hero', JSON.stringify(this.sessonHero));

        } else {
            console.log("hero2")
            sessionStorage.setItem('Hero2', JSON.stringify(this.sessonHero));

        }
        // console.log(JSON.stringify(this.hero, null, 2));
    },
        error => {
            console.log(error)
        }
    )
    // if (this.data.position === 1) {
    //     console.log("hero1")
    //     console.log(this.sessonHero)
    //     sessionStorage.setItem('Hero', JSON.stringify(this.sessonHero));

    // } else {
    //     console.log("hero2")
    //     sessionStorage.setItem('Hero2', JSON.stringify(this.sessonHero));

    // }

    // }

    onNoClick(): void {
        this.dialogRef.close();
    }
}

<h1>Search Hero by name</h1>
<form>
    <mat-form-field appearance="outline">
        <mat-label>heroName</mat-label>
        <input
            matInput
            type="search"
            placeholder="Search"
            aria-label="Search"
            #searchQuery
            (keyup)="getHeroByName(searchQuery.value)"
        />
    </mat-form-field>

    <button type="submit" (click)="getHeroByName(searchQuery.value)">
        Search Hero
    </button>
</form>

<div *ngFor="let hero of heroes">
    <!-- <h1>{{ hero.idHero }}</h1> -->
    <h3>{{ hero.heroName }}</h3>
    <img class="imgc" *ngIf="data.action==='edit'" (click)="selectHero(hero.idHero)" [src]="hero.image" [alt]="hero.heroName" />

    <img *ngIf="data.action==='fight'" (click)="saveHeroStorage(hero.idHero)" [src]="hero.image" [alt]="hero.heroName" class="img-fluid" />
</div>

<button mat-button (click)="onNoClick()">cancel</button>

<button *ngIf="data.action==='edit'" mat-button [mat-dialog-close]="HeroSelectId" cdkFocusInitial>Select Hero</button>

```

```
./fight-dialog.component.scss
```

```
.imgc{  
  height: 2em;  
}
```

- Legend**  
Html element  
Component  
Html element with directive

**result-matching ""**

**No results matching ""**