

1. Components
2. selectHeroComponent

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

File

src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts

Description

Component to select a Hero

Implements

[OnInit](#)

Metadata

selector        select-hero-dialog  
styleUrls      ./select-hero-dialog.component.scss  
templateUrl   ./select-hero-dialog.component.html

Index

Properties

- [activeState](#)
- Public [data](#)
- Public [dialogRef](#)
- Public [heroes](#)
- Public [HeroSelectId](#)
- [selectedIndex](#)
- Public [sessionHero](#)

Methods

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

Constructor

constructor(dialogRef: [MatDialogRef](#), \_HeroService: [HeroService](#), data: [any](#))  
Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:34](#)

Constructor in which we inject our services and different elements

**Parameters :**

Name	Type	Optional
dialogRef	<a href="#">MatDialogRef&lt;selectHeroComponent&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/select-hero-dialog/select-hero-dialog.component.ts:56](#)

function get a hero by name

**Parameters :**

Name Optional

name No

**Returns :** [void](#)

**ngOnInit**

ngOnInit ()

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:50](#)

Start when the component init

**Returns :** [void](#)

**onNoClick**

onNoClick ()

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:105](#)

To close modal

**Returns :** [void](#)

**Async saveHeroStorage**

saveHeroStorage ()

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:84](#)

function to save hero in the storage

**Returns :** [any](#)

**selectHero**

selectHero (idHero)

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:71](#)

function to save idHero

**Parameters :**

Name Optional

idHero No

**Returns :** [void](#)

## Properties

**activeState**

Type : [number](#)

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

**Public data**

Type : [any](#)

**Decorators :**

@Inject (MAT\_DIALOG\_DATA)

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:42](#)

Type : MatDialogRef<selectHeroComponent>

## Public heroes

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:30](#)

**Public HeroSelectId**

*Default value : 0*

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:34](#)

**selectedIndex**

*Default value :* null

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

*Type* : Hero

Defined in [src/app/components/modals/select-hero-dialog/select-hero-dialog.component.ts:26](#)

```
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'
```

```
/**
 * Component to select a Hero
 */
@Component({
  selector: 'select-hero-dialog',
  templateUrl: './select-hero-dialog.component.html',
  styleUrls: ['./select-hero-dialog.component.scss']
})
```

```
export class selectHeroComponent implements OnInit {

    activeState: number;

    selectedIndex: number = null;

    // public setROW(_index:number){this.selectedIndex=_index}

    /**
     * variable to store session hero from session
     */
    public sessionHero: Hero
    /**
     * variable to store heros from the search
     */
    public heroes: Hero[]
    /**
     * variable to store selected hero id
     */
    public HeroSelectId: number = 0

    /**
     * Constructor in which we inject our services and different elements
     */
    constructor(
        public dialogRef: MatDialogRef<selectHeroComponent>,
        private _HeroService: HeroService,
        @Inject(MAT_DIALOG_DATA) public data: any
    ) {
        this.sessionHero = new Hero(0, '', '', 0, 0, 0, 0, 0, 0, 0, 0, 0, '', '', '', '', '', '', '', '', '', '', '', null, 0)
    }

    /**
     * Start when the component inits
     */
    ngOnInit() { }

    /**
     * function get a hero by name
     */
}
```

```

    * @param {string} teamName
    */
    getHeroByName(name) {
        this._HeroService.getHeroByName(name).subscribe(
            res => {
                this.heroes = res
            },
            error => {
                console.log(error)
            }
        )
    }
}

/**
 * function to save idHero
 * @param {number} idHero
 */
selectHero(idHero) {
    this.HeroSelectId = idHero
    this.activeState = idHero;

    // this.selectedIndex = index;
}

/**
 * function to save hero in the storage
 * @param {number} id
 */
async saveHeroStorage() {
    this._HeroService.getHeroById(this.HeroSelectId).subscribe(
        res => {
            this.sessonHero = res
            if (this.data.position === 1) {
                sessionStorage.setItem('Hero', JSON.stringify(this.sessonHero))
            } else {
                sessionStorage.setItem('Hero2', JSON.stringify(this.sessonHero))
            }
            this.dialogRef.close();
        },
        error => {
            console.log(error)
        }
    )
}

/**
 * To close modal
 */
onNoClick(): void {
    this.dialogRef.close()
}
}

<div id="containerModal">
<h1>Search Hero by name</h1>
<p>Chose the hero and click select</p>
<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>
</form>
<div mat-dialog-content>
    <div class="HeroContainer">
        <div
            class="heroByName"
            *ngFor="let hero of heroes; index as i"
            [ngClass]="{ red: activeState == hero.idHero }"
        >
            <p>{{ hero.heroName }}</p>
            <img
                class="imgc"
                *ngIf="data.action === 'edit'"
                (click)="selectHero(hero.idHero)"
                [src]="hero.image"
                [alt]="hero.heroName"
                onError="this.src='../.../assets/img/nogender_2.jpg'"
            />
            <img
                class="imgc"
                *ngIf="data.action === 'fight'"
                (click)="selectHero(hero.idHero)"
            />
        </div>
    </div>
</div>

```

```

        [src]="hero.image"
        [alt]="hero.heroName"
        onError="this.src='../.../assets/img/nogender_2.jpg'"
      />
    </div>
  </div>
</div>
<div class="buttContainer">
  <button
    *ngIf="data.action === 'edit'"
    mat-button
    [mat-dialog-close]="HeroSelectId"
    cdkFocusInitial
  >
    Select Hero
  </button>
  <button class="cancel" mat-button [mat-dialog-close]="">No Thanks</button>
  <button
    *ngIf="data.action === 'fight'"
    type="button"
    mat-raised-button
    (click)="saveHeroStorage()"
  >
    Select hero
  </button>
</div>
</div>

```

```
./select-hero-dialog.component.scss
```

```

.imgc {
  height: 2em;
}

#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;
}

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

.imgc {
  width: 100%;
  height: 14em;
}

.heroByName {
  width: 10em;
  height: 15em;
  position: relative;
  text-align: center;
  cursor: pointer;
}

.heroByName p {
  position: absolute;
  color: rgb(223, 221, 221);
  background-color: #00a23d;
  width: 100%;
}

.mat-dialog-content {

```

```
    max-height: 39vh !important;
    width: 40vh !important;
}

// .red{
//     color: #cc4224;
//     border-bottom: 5px solid #cc4224;
// }

// .red {
//     background-color: red;
//     border: 5px solid #cc4224;}

.red {
    border: 6px solid #cc4224;
}

@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 ""**