

1. Components

2. SelectTeamDialogComponent

- [Info](#)

• [Source](#)

• [Template](#)

• [Styles](#)

• [DOM Tree](#)

File

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

Description

Component to select a team

Implements

[OnInit](#)

Metadata

selector app-select-team-dialog
styleUrls ./select-team-dialog.component.scss
templateUrl ./select-team-dialog.component.html

Index

Properties

- [activeState](#)

• Public [data](#)

• Public [dialogRef](#)

• Public [sessionTeam](#)

• Public [teams](#)

• Public [TeamSelectId](#)

Methods

- [getTeamByName](#)

• [ngOnInit](#)

• [onNoClick](#)

• Async [saveTeamStorage](#)

• [selectTeam](#)

Constructor

constructor(_TeamService: [TeamService](#), data: [any](#), dialogRef: [MatDialogRef](#))
Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:28](#)

Constructor in which we inject our services and different elements

Parameters :

Name	Type	Optional
_TeamService	TeamService	No
data	any	No
dialogRef	MatDialogRef<SelectTeamDialogComponent>	No

Methods

getTeamByName

getTeamByName (teamName)

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

function get a team by name

Parameters :

Name	Optional
teamName	No

Returns : [void](#)

ngOnInit

ngOnInit ()

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:44](#)

Start when the component inits

Returns : [void](#)

onNoClick

onNoClick ()

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:96](#)

To close modal

Returns : [void](#)

Async saveTeamStorage

saveTeamStorage ()

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:76](#)

function to save team in the storage

Returns : [any](#)

selectTeam

selectTeam (idTeam)

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:67](#)

function to save idteam

Parameters :

Name	Optional
idTeam	No

Returns : [void](#)

Properties

activeState

Type : [number](#)

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:16](#)

Public data

Type : [any](#)

Decorators :

@Inject (MAT_DIALOG_DATA)

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:35](#)

Public dialogRef

Type : [MatDialogRef<SelectTeamDialogComponent>](#)

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:36](#)

Public sessionTeam

Type : [Team](#)

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:24](#)

variable to store teams from sessions

Public teams

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

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:20](#)

variable to store teams

Public TeamSelectId

Type : [number](#)

Default value : 0

Defined in [src/app/components/modals/select-team-dialog/select-team-dialog.component.ts:28](#)

variable to store team selec id

```
import { Component, OnInit, Inject } from '@angular/core'
import { TeamService } from 'src/app/services/team.service'
import { MatDialogRef, MAT_DIALOG_DATA } from '@angular/material'
import { Team } from 'src/app/models/team'

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

export class SelectTeamDialogComponent implements OnInit {
  activeState: number;
  /**
   * variable to store teams
   */
  public teams: Team[]
  /**
   * variable to store teams from sessions
   */
  public sessionTeam: Team
  /**
   * variable to store team selec id
   */
  public TeamSelectId: number = 0

  /**
   * Constructor in which we inject our services and different elements
   */
  constructor(
    private _TeamService: TeamService,
    @Inject(MAT_DIALOG_DATA) public data: any,
    public dialogRef: MatDialogRef<SelectTeamDialogComponent>
  ) {
    this.sessionTeam = new Team(0, 0, '', '', '', '', '', '')
  }

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

  /**
   * function get a team by name
   * @param {string} teamName
   */
}
```

```

*/
getTeamByName(teamName) {
  this._TeamService.searchTeam(teamName).subscribe(
    res => {
      // console.log(res.idHero)
      this.teams = res
      // console.log(this.hero);
    },
    error => {
      console.log(error)
    }
  )
}

/**
 * function to save idteam
 * @param {number} idTeam
 */
selectTeam(idTeam) {
  this.TeamSelectId = idTeam
  this.activeState = idTeam
}

/**
 * function to save team in the storage
 * @param {number} idUsu
 */
async saveTeamStorage() {
  this._TeamService.getTeamInfo(this.TeamSelectId).subscribe(
    res => {
      this.sessonTeam = res
      if (this.data.position === 1) {
        sessionStorage.setItem('Team1', JSON.stringify(this.sessonTeam))
      } else {
        sessionStorage.setItem('Team2', JSON.stringify(this.sessonTeam))
      }
      this.dialogRef.close();
    },
    error => {
      console.log(error)
    }
  )
}

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

<div id="containerModal">
  <h1>Search Team by name</h1>
  <form>
    <mat-form-field appearance="outline">
      <mat-label>Team name</mat-label>
      <input
        matInput
        type="search"
        placeholder="Search"
        aria-label="Search"
        #searchQuery
        (keyup)="getTeamByName(searchQuery.value)"
      />
    </mat-form-field>
  </form>
  <div mat-dialog-content>
    <div>
      *ngFor="let team of teams"
      [ngClass]="{ red: activeState == team.idUsu }"
    >
      <h5 (click)="selectTeam(team.idUsu)">
        {{ team.teamName }} <span>By: {{ team.alias }}</span>
      </h5>
    </div>
  </div>
  <div class="buttContainer">
    <button class="cancel" mat-button [mat-dialog-close]="">No Thanks</button>
    <button type="button" mat-raised-button (click)="saveTeamStorage()">
      Select Team
    </button>
  </div>
</div>

./select-team-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;
}

h5 {
  cursor: pointer;
}

h5:hover {
  color: #cc4224;
}

span {
  color: Black !important;
}

.red {
  color: #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 ""