File

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

Description

Component to create or modify a team

Implements

`OnInit`

Metadata

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

Index

Constructor

```
constructor(dialogRef: MatDialogRef, formBuilder: FormBuilder, _snackBar: MatSnackBar, _TeamService:
TeamService, data: any)
```

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

Constructor in which we inject our services and different elements

Parameters :

Name	Type	Optional
dialogRef	`MatDialogRef<TeamDialogComponent>`	No
formBuilder	`FormBuilder`	No
_snackBar	`MatSnackBar`	No
_TeamService	`TeamService`	No
data	`any`	No

Methods

deleteTeam

`deleteTeam()`

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:122

To delete a team

Returns : `void`

getErrorMessage

`getErrorMessage(dato)`

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:64

function to control error messages

Parameters :

Name	Optional
dato	No

Returns : `"This information is required" | "You must enter at least 6 characters" | "" | "The maximum of ch...`

message

ngOnInit

`ngOnInit()`

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:46

Start when the component inits

Returns : `void`

openSnackBar

`openSnackBar(message: string, action: string)`

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:144

function to open snackBars

Parameters :

Name	Type	Optional
message	`string`	No
action	`string`	No

Returns : `void`

submit

`submit(teamForm)`

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:81

function to submit form

Parameters :

Name	Optional
teamForm	No

Returns : void

Properties

Public correctdata

Type : boolean

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

variable to check if the function was ok

Public data

Type : any

Decorators :

`@Inject(MAT_DIALOG_DATA)`

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:40

Public dialogRef

Type : MatDialogRef<TeamDialogComponent>

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:36

Public message

Type : string

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:26

variable to save message info

Public teamForm

Type : FormGroup

Defined in src/app/components/modals/team-dialog/team-dialog.component.ts:22

to add FormGroup

```
import { Component, OnInit, Inject } from '@angular/core'
import { MatDialogRef, MAT_DIALOG_DATA } from '@angular/material/dialog'
import { FormBuilder } from '@angular/forms'
import { Validators } from '@angular/forms'
import { FormGroup, FormControl, AbstractControl } from '@angular/forms'
import { MatSnackBar } from '@angular/material'
import { TeamService } from 'src/app/services/team.service'

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

export class TeamDialogComponent implements OnInit {
  /**
   * to add FormGroup
   */
  public teamForm: FormGroup
  /**
   * variable to save message info
   */
  public message: string
  /**
  * variable to check if the function was ok
  */
  public correctdata: boolean

  /**
   * Constructor in which we inject our services and different elements
 */
constructor(
  public dialogRef: MatDialogRef<TeamDialogComponent>,
  private formBuilder: FormBuilder,
  private _snackBar: MatSnackBar,
  private _TeamService: TeamService,
  @Inject(MAT_DIALOG_DATA) public data: any
) { }

/**
 * Start when the component inits
 */
ngOnInit() {
  this.teamForm = this.formBuilder.group({
    teamName: [
      '',
      [
        Validators.required,
        Validators.minLength(6),
        Validators.maxLength(300)
      ]
    ]
  })
}

/**
 * function to control error messages
 * @param {string} dato
 * @returns message
 */
getErrorMessage(dato) {
  var result: string
  if (this.teamForm.controls[dato].hasError('required')) {
    return (result = 'This information is required')
  } else if (this.teamForm.controls[dato].hasError('minlength')) {
    return (result = 'You must enter at least 6 characters')
  } else if (this.teamForm.controls[dato].hasError('maxlength')) {
    return (result = 'The maximum of characters is 300')
  } else {
    return (result = '')
  }
}

/**
 * function to submit form
 * @param {any} teamForm
 */
submit(teamForm) {
  if (this.data.status === 'new') {
    var data = { idUsu: this.data.idUsu, teamName: teamForm.value.teamName }

    this._TeamService.createTeam(data).subscribe(
      res => {
        this.openSnackBar('YOUR TEAM HAS BEEN CREATE', 'Close')
        this.correctdata = true
        this.dialogRef.close('Close modal!')
        window.location.reload();
      },
      err => {
        this.correctdata = false
        console.log(err.status)
        this.message = 'Error creating your Team'
        console.log(this.message)
      }
    )
  } else {
    var team = { teamName: teamForm.value.teamName }
    console.log(this.data.teamInfo.idTeam)
    this._TeamService.changeName(this.data.teamInfo.idTeam, team).subscribe(
      res => {
        this.openSnackBar('YOUR TEAM HAS BEEN UPDATE', 'Close')
        this.correctdata = true
        this.dialogRef.close('Close modal!')
        window.location.reload();
      },
      err => {
        this.correctdata = false
        console.log(err.status)
        this.message = 'Error modifying your Team'
        console.log(this.message)
      }
    )
  }
}

/**
 * To delete a team
```

```
   */
  deleteTeam() {
    this._TeamService.delete(this.data.teamInfo.idTeam).subscribe(
      res => {
        this.openSnackBar('YOUR TEAM HAS BEEN DELETE', 'Close')
        this.correctdata = true
        this.dialogRef.close('Close modal!')
        window.location.reload();
      },
      err => {
        this.correctdata = false
        console.log(err.status)
        this.message = 'Error deleting your Team'
        console.log(this.message)
      }
    )
  }

  /**
   * function to open snackBars
   *  @param {string} message
   *  @param {string} action
   */
  openSnackBar(message: string, action: string) {
    this._snackBar.open(message, action, {
      duration: 8000,
      panelClass: ['blue-snackbar']
    })
  }
}

<div id="containerModal">
  <div *ngIf="data.status === 'new' || data.status === 'modify'" id="form">
    <h1 *ngIf="data.status === 'new'">Create Team</h1>
    <h1 *ngIf="data.status === 'modify'">Modify Team Name</h1>
    <form [formGroup]="teamForm" (ngSubmit)="submit(teamForm)">
      <mat-form-field appearance="outline">
        <mat-label>Name of your team</mat-label>
        <input
          *ngIf="!data.teamInfo"
          matInput
          type="text"
          formControlName="teamName"
          placeholder="teamName"
        />
        <input
          *ngIf="data.teamInfo"
          matInput
          type="text"
          formControlName="teamName"
          placeholder="{{ data.teamInfo.teamName }}"
        />
        <mat-error *ngIf="!teamForm.controls['teamName'].valid">{{
          getErrorMessage("teamName")
        }}</mat-error>
      </mat-form-field>
      <div *ngIf="!correctdata">
        <p class="mensajeError">{{ this.message }}</p>
      </div>
      <div class="buttContainer">
        <button class="cancel" mat-button [mat-dialog-close]="">
          No Thanks
        </button>
        <button
          type="submit"
          class="LoginButton i"
          [disabled]="!teamForm.valid"
        >
          Send teamName
        </button>
      </div>
    </form>
  </div>
  <div *ngIf="data.status === 'delete'">
    <h1>Delete Team</h1>
    <h4>Are you sure about deleting your team {{ data.teamInfo.teamName }}</h4>
    <div *ngIf="!correctdata">
      <p class="mensajeError">{{ this.message }}</p>
    </div>
    <div class="buttContainer">
      <button class="cancel" mat-button [mat-dialog-close]="">No Thanks</button>
      <button type="button" mat-raised-button (click)="deleteTeam()">
        Delete Team
      </button>
    </div>
  </div>
</div>
```

```
./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;
}

h5 {
  cursor: pointer;
}
span {
  color: Black !important;
}

.red {
  color: #cc4224;
}

@media (min-width: 992px) {
  #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 ""