

1. Components
2. DeleteUserDialogComponent

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

## File

`src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts`

## Description

Component to delete a user

## Implements

[OnInit](#)

## Metadata

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

## Index

### Properties

- Public [correctdata](#)
- Public [data](#)
- Public [dialogRef](#)
- Public [message](#)
- Public [passForm](#)

### Methods

- [getErrorMessage](#)
- [ngOnInit](#)
- [openSnackBar](#)
- [submit](#)

## Constructor

constructor(dialogRef: [MatDialogRef](#), FormBuilder: [FormBuilder](#), \_snackBar: MatSnackBar, \_UserService: [UserService](#), \_router: [Router](#), data: [any](#))

Defined in [src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts:31](#)

Constructor in which we inject our services and different elements

#### Parameters :

Name	Type	Optional
dialogRef	<a href="#">MatDialogRef&lt;DeleteUserDialogComponent&gt;</a>	No
formBuilder	<a href="#">FormBuilder</a>	No
_snackBar	MatSnackBar	No
_UserService	<a href="#">UserService</a>	No
_router	<a href="#">Router</a>	No
data	<a href="#">any</a>	No

## Methods

### getErrorMessage

getErrorMessage (dato)

Defined in [src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts:62](#)

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 charact..."](#)

message

### ngOnInit

ngOnInit ()

Defined in [src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts:48](#)

Start when the component inits

Returns : [void](#)

### openSnackBar

openSnackBar (message: [string](#), action: [string](#))

Defined in [src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts:111](#)

function to open snackBars

#### Parameters :

Name Type Optional

message [string](#) No

action [string](#) No

Returns : [void](#)

### submit

submit (passForm)

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

function to submit form

#### Parameters :

Name Optional

passForm No

Returns : [void](#)

## Properties

Public correctdata

Type : [boolean](#)

Defined in [src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts:31](#)

variable to check if the function was ok

## Public data

Type : [any](#)

## Decorators :

@Inject (MAT\_DIALOG\_DATA)

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

## Public dialogRef

Type : [MatDialogRef<DeleteUserDialogComponent>](#)

Defined in [src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts:37](#)

## Public message

Type : [string](#)

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

variable to save message info

## Public passForm

Type : [FormGroup](#)

Defined in [src/app/components/modals/delete-user-dialog/delete-user-dialog.component.ts:23](#)

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 { UserService } from 'src/app/services/user.service';
import { Router } from '@angular/router';

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

export class DeleteUserDialogComponent implements OnInit {
  /**
   * to add FormGroup
   */
  public passForm: 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<DeleteUserDialogComponent>,
    private formBuilder: FormBuilder,
    private _snackBar: MatSnackBar,
    private _UserService: UserService,
    private _router: Router,
    @Inject (MAT_DIALOG_DATA) public data: any
  ) { }

  /**
   * Start when the component inits
   */
  ngOnInit() {
    this.passForm = this.formBuilder.group({
      password: [
```

```

        '',
        [Validators.required, Validators.minLength(6), Validators.maxLength(30)]
    ],
    });
}

/**
 * function to control error messages
 * @param {string} dato
 * @returns message
 */
getErrorMessage(dato) {
    var result: string;
    if (this.passForm.controls[dato].hasError('required')) {
        return (result = 'This information is required');
    } else if (this.passForm.controls[dato].hasError('minlength')) {
        return (result = 'You must enter at least 6 characters');
    } else if (this.passForm.controls[dato].hasError('maxlength')) {
        return (result = 'The maximum of characters is 30');
    } else {
        return (result = '');
    }
}

/**
 * function to submit form
 * @param {any} passForm
 */
submit(passForm) {
    var deleteObj = { email: this.data.email, password: passForm.value.password };

    this._UserService.deleteUser(this.data.userId, deleteObj).subscribe(
        res => {
            this.openSnackBar('YOUR ACOUNT HAD BEEN DELETED', 'Close')
            this.correctdata = true;
            this.dialogRef.close("Close modal!");
            localStorage.clear();
            this._router.navigate(['/']);
        },
        err => {
            this.correctdata = false;
            if (err.status === 400) {
                this.message = 'Invalid Password';
                console.log(err.status);
                console.log(this.message);
            } else {
                console.log(err.status);
                this.message = 'Error deleting your account';
                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">
  <h5>{{ data.alias }} Are you sure about deleting your account?</h5>
  <div id="form">
    <h1>You have to send your password</h1>
    <form [formGroup]="passForm" (ngSubmit)="submit(passForm)">
      <mat-form-field appearance="outline">
        <mat-label>Password</mat-label>
        <input
          matInput
          type="password"
          formControlName="password"
          placeholder="Password"
        />
        <mat-error *ngIf="!passForm.controls['password'].valid">
          {{ getErrorMessage("password") }}</mat-error>
        </mat-form-field>
      </div>
    </div>
  </div>

```

```

<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]="!passForm.valid"
  >
    Delete account
  </button>
</div>
</form>
</div>
</div>

```

```
./delete-user-dialog.component.scss
```

```

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

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

#containerModal button {
  background-color: #f3d403;
  border: none;
  border-radius: 10px;
  border: 2.5px solid #00a23d;
  padding: 1.5vw;
  font-size: 0.8rem;
}

.buttContainer {
  display: flex;
  justify-content: space-between;
}

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

#containerModal button .cancel {
  background-color: #2a75b3;
}

@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 ""