

1. Components
2. PublicProfileComponent

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

File

`src/app/components/public-profile/public-profile.component.ts`

Description

Component that bring public user profile

Implements

[OnInit](#)

Metadata

selector `app-public-profile`
styleUrls `./public-profile.component.scss`
templateUrl `./public-profile.component.html`

Index

Properties

- Public [followActive](#)
- Public [FollowedUsers](#)
- Public [FollowersUsers](#)
- Public [heroesFav](#)
- Public [identity](#)
- Public [idParams](#)
- [page](#)
- [page2](#)
- [pageSize](#)
- Public [showName](#)
- Public [user](#)

Methods

- [checkFollow](#)
- [favorites](#)
- [followUser](#)
- [getFollowedUsers](#)
- [getFollowersUsers](#)
- [getIdentity](#)
- [getUsuInfo](#)
- [ngOnInit](#)
- [unFollowUser](#)

Constructor

constructor(_UserHero: [UserHeroService](#), _UserService: [UserService](#), _activatedRoute: [ActivatedRoute](#))
Defined in [src/app/components/public-profile/public-profile.component.ts:50](#)

Constructor in which we inject user service, user Hero service, and service to control the router params

Parameters :

Name	Type	Optional
_UserHero	UserHeroService	No
_UserService	UserService	No
_activatedRoute	ActivatedRoute	No

Methods

checkFollow

checkFollow()

Defined in [src/app/components/public-profile/public-profile.component.ts:145](#)

Check if you follow the user

Returns : [void](#)

favorites

favorites()

Defined in [src/app/components/public-profile/public-profile.component.ts:102](#)

Get user favorites heroes

Returns : [void](#)

followUser

followUser()

Defined in [src/app/components/public-profile/public-profile.component.ts:114](#)

Get user followed by the user

Returns : [void](#)

getFollowedUsers

getFollowedUsers()

Defined in [src/app/components/public-profile/public-profile.component.ts:159](#)

Get user followed by the user

Returns : [void](#)

getFollowersUsers

getFollowersUsers()

Defined in [src/app/components/public-profile/public-profile.component.ts:173](#)

Get user followers

Returns : [void](#)

getIdentity

getIdentity()

Defined in [src/app/components/public-profile/public-profile.component.ts:81](#)

Get user identity

Returns : [void](#)

getUsuInfo

getUsuInfo()

Defined in [src/app/components/public-profile/public-profile.component.ts:88](#)

Get user info

Returns : [void](#)

ngOnInit

ngOnInit()

Defined in [src/app/components/public-profile/public-profile.component.ts:66](#)

Start when the component inits

Returns : [void](#)

unFollowUser

unFollowUser()

Defined in [src/app/components/public-profile/public-profile.component.ts:130](#)

Unfollow the user

Returns : [void](#)

Properties

Public followActive

Type : [any](#)

Default value : ''

Defined in [src/app/components/public-profile/public-profile.component.ts:41](#)

variable to store active follow

Public FollowedUsers

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

Default value : []

Defined in [src/app/components/public-profile/public-profile.component.ts:45](#)

variable to store info about users followeds

Public FollowersUsers

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

Default value : []

Defined in [src/app/components/public-profile/public-profile.component.ts:50](#)

variable to store info about users followers

Public heroesFav

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

Defined in [src/app/components/public-profile/public-profile.component.ts:33](#)

variable to save array of favorite heroes

Public identity

Defined in [src/app/components/public-profile/public-profile.component.ts:37](#)

variable to store user identity

Public idParams

Type : [number](#)

Default value : 0

Defined in [src/app/components/public-profile/public-profile.component.ts:25](#)

variable to save the id from params

page

Type : [number](#)

Default value : 1

Defined in [src/app/components/public-profile/public-profile.component.ts:18](#)

page2

Type : [number](#)

Default value : 1

Defined in [src/app/components/public-profile/public-profile.component.ts:19](#)

pageSize

Type : [number](#)

Default value : 4

Defined in [src/app/components/public-profile/public-profile.component.ts:20](#)

Public showName

Type : [boolean](#)

Default value : true

Defined in [src/app/components/public-profile/public-profile.component.ts:21](#)

Public user

Type : [User](#)

Defined in [src/app/components/public-profile/public-profile.component.ts:29](#)

variable to save user

```
import { Component, OnInit } from '@angular/core'
import { ActivatedRoute } from '@angular/router'
import { User } from 'src/app/models/user'
import { UserService } from 'src/app/services/user.service'
import { UserHeroService } from 'src/app/services/user-hero.service'
import { Hero } from 'src/app/models/hero'

/**
 * Component that bring public user profile
 */
@Component({
  selector: 'app-public-profile',
  templateUrl: './public-profile.component.html',
  styleUrls: ['./public-profile.component.scss']
})

export class PublicProfileComponent implements OnInit {
  page = 1;
  page2 = 1;
  pageSize = 4;
  public showName: boolean = true;
  /**
   * variable to save the id from params
   */
  public idParams: number = 0
  /**
   * variable to save user
   */
  public user: User
  /**
   * variable to save array of favorite heroes
   */
  public heroesFav: Hero[]
  /**
   * variable to store user identity
   */
  public identity
  /**
   * variable to store active follow
   */
  public followActive: any = ''
  /**
   * variable to store info about users followeds
   */
  public FollowedUsers: User[] = []

  /**
   * variable to store info about users followers
   */
  public FollowersUsers: User[] = []

  /**
   * Constructor in which we inject user service, user Hero service, and service to control the router params
   */
  constructor(
    private _UserHero: UserHeroService,
    private _UserService: UserService,
    private _activatedRoute: ActivatedRoute
  ) {
    this.user = new User(0, '', '', '', '', '', new Date(0), '', false)
  }
}
```

```

}

/**
 * Start when the component inits
 */
ngOnInit() {
  this._activatedRoute.params.subscribe(params => {
    this.idParams = params['user']
    this.getUsuInfo()
    this.favorites()
    this.getIdentity()
    this.checkFollow()
    this.getFollowersUsers()
    this.getFollowedUsers()
  })
}

/**
 * Get user identity
 */
getIdentity() {
  this.identity = this._UserService.getIdentity()
}

/**
 * Get user info
 */
getUsuInfo() {
  this._UserService.getUser(this.idParams).subscribe(
    res => {
      this.user = res
    },
    error => {
      console.log(error)
    }
  )
}

/**
 * Get user favorites heroes
 */
favorites() {
  this._UserHero.allHerosFav(this.idParams).subscribe(res => {
    this.heroesFav = res
  }),
  error => {
    console.log(error)
  }
}

/**
 * Get user followed by the user
 */
followUser() {
  var ids = { idUsu: this.identity.id, idUsuFollow: this.idParams }
  this._UserService.followUser(ids).subscribe(
    res => {
      this.checkFollow()
      this.getFollowersUsers()
    },
    err => {
      console.log(err)
    }
  )
}

/**
 * Unfollow the user
 */
unFollowUser() {
  this._UserService.unFollowUser(this.identity.id, this.idParams).subscribe(
    res => {
      this.checkFollow()
      this.getFollowersUsers()
    },
    err => {
      console.log(err)
    }
  )
}

/**
 * Check if you follow the user
 */
checkFollow() {
  this._UserService.checkFollow(this.identity.id, this.idParams).subscribe(
    res => {

```

```

        this.followActive = res
    },
    err => {
        console.log(err)
    }
}
)
}

/**
 * Get user followed by the user
 */
getFollowedUsers() {
    this._UserService.getFollowUsers(this.idParams).subscribe(
        res => {
            this.FollowedUsers = res
        },
        error => {
            console.log(error)
        }
    )
}

/**
 * Get user followers
 */
getFollowersUsers() {
    this._UserService.getFollowersUsers(this.idParams).subscribe(
        res => {
            this.FollowersUsers = res
        },
        error => {
            console.log(error)
        }
    )
}
}

<div class="color1">
    <div class="cardHero">
        <div class="TitleF">
            <h1>PUBLIC PROFILE</h1>
            <button
                *ngIf="!followActive && identity.id != idParams"
                type="button"
                class="followF"
                mat-raised-button
                (click)="followUser()"
            >
                <span>FOLLOW USER</span>
            </button>
            <button
                *ngIf="followActive && identity.id != idParams"
                type="button"
                class="followUN"
                mat-raised-button
                (click)="unFollowUser()"
            >
                <span>UNFOLLOW USER</span>
            </button>
        </div>
        <div class="UserInfo">
            <div class="imgUserInfo">
                <img [src]="user.photo" [alt]="user.alias" class="" />
                <br />
            </div>

            <div class="infoCard3">
                <p><span>ALIAS:</span> {{ user.alias }}</p>
                <p><span>NAME:</span> {{ user.userName }}</p>
                <p><span>DATE OF BIRTH: </span>{{ user.dateOfBirth | date }}</p>
                <p><span>SURNAME:</span> {{ user.surname }}</p>
            </div>
            <div class="tableMat">
                <mat-tab-group>
                    <mat-tab label="FOLLOW">
                        <div class="tablaContainer">
                            <table *ngIf="FollowedUsers" class="table">
                                <tbody>
                                    <tr>
                                        *ngFor="
                                            let user of FollowedUsers
                                            | slice
                                            : (page - 1) * pageSize
                                            : (page - 1) * pageSize + pageSize;
                                        index as i
                                        "
                                    >

```

```

        <td class="usu imgUser">
            <div class="userAvatar2 img">
                
            </div>
        </td>
        <td
            class="textName"
            [routerLink]="['/publicProfile', user.idUsu]"
        >
            <p>
                {{
                    user.alias.length > 20
                    ? (user.alias | slice: 0:20) + "..."
                    : user.alias
                }}
            </p>
        </td>
    </tr>
</tbody>
</table>
</div>
<div *ngIf="FollowedUsers" class="pag">
    <ngb-pagination
        [(page)]="page"
        [pageSize]="pageSize"
        [collectionSize]="FollowedUsers.length"
    >
    </ngb-pagination>
</div>
</mat-tab>

<mat-tab label="FOLLOWERS">
    <div class="tablaContainer">
        <table *ngIf="FollowersUsers" class="table">
            <tbody>
                <tr>
                    <td>
                        *ngFor="
                            let user of FollowersUsers
                                | slice
                                    : (page2 - 1) * pageSize
                                    : (page2 - 1) * pageSize + pageSize;
                                index as i
                        "
                    >
                        <td class="usu imgUser">
                            <div class="userAvatar2 img">
                                
                            </div>
                        </td>
                        <td
                            class="textName"
                            [routerLink]="['/publicProfile', user.idUsu]"
                        >
                            <p>
                                {{
                                    user.alias.length > 20
                                    ? (user.alias | slice: 0:20) + "..."
                                    : user.alias
                                }}
                            </p>
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
        <div *ngIf="FollowedUsers" class="pag">
            <ngb-pagination
                [(page)]="page2"
                [pageSize]="pageSize"
                [collectionSize]="FollowersUsers.length"
            >
            </ngb-pagination>
        </div>
    </mat-tab>
</mat-tab-group>
</div>
</div>

```

```

    </div>
</div>
<div class="color1">
    <app-team [idUsu]="idParams" [showName]="showName"></app-team>
</div>

```

```

./public-profile.component.scss

```

```

.cardHero {
    border: 2.5px solid #00a23d;
    box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
    padding: 2em;
    padding-bottom: 0em !important;
    background-color: #272727;
    color: white;
}

.cardHero h1 {
    text-align: center;

    color: #cc4224;
    font-family: "Bangers", cursive;
    text-decoration: underline;
    -webkit-text-decoration-color: #00a23d; /* Safari */
    text-decoration-color: #00a23d;
}

.TitleF {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    padding-bottom: 2vw;
}

.UserInfo {
    display: flex;
    font-family: "B612";
    flex-direction: column;
    justify-content: space-around;
}

.imgUserInfo img {
    border-radius: 2%;
    width: 17em;
    height: 87%;
    box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
}

.imgUserInfo {
    height: 25em;
}

.infoCard3 {
    padding-top: 1em;
}

.infoCard3 span {
    color: #00a23d;
}

// button
$bg: #272727;
$fg: #f3d403;
$border-width: 0.2rem;
$corner-size: 3rem;
$dur: 0.3s;

.followB,
.followB2,
.followF,
.followUN {
    margin: 1vw 0.5vw 0 0.5vw;
    outline: none;

    font-family: "B612";
    letter-spacing: 0.02rem;
    cursor: pointer;
    background: transparent;
    border: $border-width solid currentColor;
    padding: 0.1rem 0.5rem;
    font-size: 1rem;
    color: $fg;
    position: relative;
    transition: color $dur;

    &:hover {
        color: #cc4224;
        &::before {
            width: 0;

```



```

    }
    &::after {
        height: 0;
    }
}
&:active {
    border-width: $border-width / 2;
}
span {
    position: relative;
    z-index: 2;
}
&::before,
&::after {
    content: "";
    position: absolute;
    background: $bg;
    z-index: 1;
    transition: all $dur;
}
&::before {
    width: calc(100% - #{$corner-size});
    height: calc(101% + #{$border-width * 2});
    top: -$border-width;
    left: 50%;
    transform: translateX(-50%);
}
&::after {
    height: calc(100% - #{$corner-size});
    width: calc(101% + #{$border-width * 2});
    left: -$border-width;
    top: 50%;
    transform: translateY(-50%);
}
}
.followB2,
.followUN {
    color: #2a75b3;
}

.followF,
.followUN {
    margin: 0vw 0.5vw 0 0.5vw;
}

.pad {
    margin-top: 2%;
}

.userAvatar2 {
    height: 4em;
    width: 4em;
    background-repeat: no-repeat;
    background-position: 50%;
    border-radius: 50%;
    background-size: 100% auto;
    overflow: hidden;
    border: #00a23d 2px solid;
    height: auto;
    width: 4.1em;
    height: 4.1em;
}
.imgUser {
    height: auto;
    width: 4.1em;
    height: 4.1em;
}
.table tr {
    color: #00a23d;
}
.table td {
    color: rgb(223, 221, 221);
}

.imgUser img {
    width: 4.1em !important;
    height: 5.1em !important;
}

.textName {
    font-family: "B612";
    padding-top: 30px !important;
    cursor: pointer;
}
.textName:hover p {
    color: #cc4224;
}

```

```
::ng-deep.mat-tab-label .mat-tab-label-content {
  font-family: "B612" !important;
  color: #00a23d !important;
}

.pag {
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 1em 0 1em 0;
}
//pagination

ngb-pagination ::ng-deep a {
  color: black !important;
}
ngb-pagination ::ng-deep .page-item.active .page-link {
  color: white !important;
  border-color: black !important;
  background-color: #00a23d;
}
ngb-pagination ::ng-deep .page-item.active .page-link:focus,
.page-link:not(:disabled):not(.disabled):active:focus {
  box-shadow: 0 0 0 0.1rem black;
}
ngb-pagination ::ng-deep .page-link:not(:disabled):not(.disabled):active:focus {
  box-shadow: 0 0 0 0.1rem black;
}

.tableMat {
  height: 100%;
}

@media (min-width: 992px) {
  .TitleF {
    flex-direction: row;
  }
  .imgUserInfo {
    height: 31em;
  }

  .imgUserInfo img {
    width: 20em;
    height: 83%;
  }

  .UserInfo {
    flex-direction: row;
  }

  .UnfollowB {
    padding: 0.5rem 10.5rem;
  }
}
```

Legend

Html element

Component

Html element with directive

result-matching ""

No results matching ""