1. Components
2. HeroesDetailComponent

- Info
- Source
- Template
- Styles
- DOM Tree

### File

`src/app/components/heroes-detail/heroes-detail.component.ts`

### Description

Component for hero details

### Implements

`OnInit`

### Metadata

|  |  |
|---|---|
| selector | `app-heroes-detail` |
| styleUrls | `./heroes-detail.component.scss` |
| templateUrl | `./heroes-detail.component.html` |

### Index

###### Properties

###### Methods

### Constructor

```
constructor(_heroService: HeroService, _UserHeroService: UserHeroService, _UserService: UserService, dialog: MatDialog, _activatedRoute: ActivatedRoute)
```

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:76

Constructor in which we inject hero service, formBuilder and router service

**Parameters :**

| Name | Type | Optional |
|---|---|---|
| _heroService | `HeroService` | No |
| _UserHeroService | `UserHeroService` | No |
| _UserService | `UserService` | No |
| dialog | `MatDialog` | No |
| _activatedRoute | `ActivatedRoute` | No |

### Methods

**commentHeroDialog**

```
commentHeroDialog(status)
```

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:261

Open modal to comment hero

**Parameters :**

| Name | Optional |
|---|---|
| status | No |

**Returns :** `void`

**favorite**

```
favorite()
```

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:171

Add to favorite

**Returns :** `void`

**followHero**

`followHero()`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:141

Follow hero

**Returns :** `void`

**getHero**

`getHero(id)`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:119

Get hero info

**Parameters :**

| Name | Optional |
|---|---|
| id | No |

**Returns :** `void`

**getHeroComment**

`getHeroComment()`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:246

Get Hero comments

**Returns :** `void`

**getHeroUsu**

`getHeroUsu()`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:220

Check if the user follow the hero

**Returns :** `void`

**getIdentity**

`getIdentity()`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:107

Get user identity

**Returns :** `void`

**ngOnInit**

`ngOnInit()`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:95

Start when the component inits

**Returns :** `void`

**unfavorite**

`unfavorite()`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:186

Unfavorite hero

**Returns :** `void`

**unfollowHero**

`unfollowHero()`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:156

Unfollow hero

**Returns :** `void`

**voteHero**

`voteHero(hovered)`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:202

vote a hero

**Parameters :**

| Name | Optional |
|---|---|
| hovered | No |

**Returns :** `void`

## Properties

**Public allComments**

*Default value :* `[]`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:76

variable to store all comments about the hero

**Public comment**

*Type :* `string`

*Default value :* `''`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:68

variable to store the user comment about the hero

**Public dialog**

*Type :* `MatDialog`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:85

**Public favoriteHero**

*Type :* `boolean`

*Default value :* `false`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:48

variable to check if hero is favorited by the user

**Public followtHero**

*Type :* `boolean`

*Default value :* `false`

Defined in src/app/components/heroes-detail/heroes-detail.component.ts:52

variable to check if the hero is followed by the user

**Public hero**
*Type :* `Hero`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:36

variable to store hero info

**Public heroUsu**
*Type :* `UserHero`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:40

variable to store the relationships in user hero

**Public hovered**
*Type :* `any`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:56

variable for star ranking

**Public identity**
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:44

variable to store user identity

**Public idParams**
*Type :* `number`
*Default value :* `0`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:72

variable to store id from param

**page**
*Type :* `number`

*Default value :* `1`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:28

variable to page pagination

**pageSize**
*Type :* `number`
*Default value :* `4`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:32

variable for pagination size

**rate**
*Type :* `number`
*Default value :* `0`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:24

variable to store hero rating

**Public readonly**
*Type :* `any`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:60

variable for star ranking

**Public score**
*Type :* `number`
*Default value :* `0`
Defined in src/app/components/heroes-detail/heroes-detail.component.ts:64

variable to store star ranking points

```
import { Component, OnInit } from '@angular/core'
import { HeroService } from 'src/app/services/hero.service'
import { Hero } from '../../models/hero'
import { ActivatedRoute } from '@angular/router'
import { UserHeroService } from 'src/app/services/user-hero.service'
import { UserService } from 'src/app/services/user.service'
import { MatDialog } from '@angular/material/dialog'
import { ComentHeroDialogComponent } from '../modals/coment-hero-dialog/coment-hero-dialog.component'
import { UserHero } from 'src/app/models/userHero'

/**
 * Component for hero details
 */
@Component({
  selector: 'app-heroes-detail',
  templateUrl: './heroes-detail.component.html',
  styleUrls: ['./heroes-detail.component.scss']
})

export class HeroesDetailComponent implements OnInit {
  /**
   * variable to store hero rating
   */
  rate: number = 0;
  /**
   * variable to page pagination
   */
  page = 1;
  /**
   * variable for pagination size
   */
  pageSize = 4;
  /**
   * variable to store hero info
   */
  public hero: Hero
  /**
   * variable to store the relationships in user hero
   */
  public heroUsu: UserHero
  /**
   * variable to store user identity
   */
  public identity
  /**
   * variable to check if hero is favorited by the user
   */
  public favoriteHero: boolean = false
  /**
   * variable to check if the hero is followed by the user
   */
  public followtHero: boolean = false
  /**
   * variable for star ranking
   */
  public hovered: any
  /**
   * variable for star ranking
   */
  public readonly: any
  /**
   * variable to store star ranking points
   */
  public score: number = 0
  /**
   * variable to store the user comment about the hero
   */
  public comment: string = ''
  /**
   * variable to store id from param
   */
  public idParams: number = 0
  /**
   * variable to store all comments about the hero
   */
  public allComments: [] = []

  /**
   * Constructor in which we inject hero service, formBuilder and router service
   */
  constructor(
    private _heroService: HeroService,
    private _UserHeroService: UserHeroService,
    private _UserService: UserService,
    public dialog: MatDialog,
    private _activatedRoute: ActivatedRoute
  ) {
    this.hero = new Hero(0, '', '', 0, 0, 0, 0, 0, 0, '', '', '', '', '', '', '', '', '', '', '', '', '', null, 0)
    this.heroUsu = new UserHero(0, 0, 0, '', false, false)
  }

  /**
   * Start when the component inits
   */
  ngOnInit() {
    this._activatedRoute.params.subscribe(params => {
      this.idParams = params['id']
      this.getHero(this.idParams)
      this.getIdentity()
      this.getHeroComment()
    })
  }

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

  /**
   * Get hero info
   *  @param {number} id
   *
   */
  getHero(id) {
    window.scrollTo(0, 0);
    this._heroService.getHeroById(id).subscribe(
      res => {
        this.hero = res
      },
      error => {
        console.log(error)
      }
    )
    this._UserHeroService.getHeroRateScore(id).subscribe(
      res => {
        this.rate = res[0].rate;
      }, err => {
        console.log(err)
      }
    )
  }

  /**
   * Follow hero
   */
  followHero() {
    var ids = { idUsu: this.identity.id, idHero: this.hero.idHero }
    this._UserHeroService.followHero(ids).subscribe(
      res => {
        this.getHeroUsu()
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Unfollow hero
   */
  unfollowHero() {
    var ids = { idUsu: this.identity.id, idHero: this.hero.idHero }
    this._UserHeroService.unfollowHero(ids).subscribe(
      res => {
        this.getHeroUsu()
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Add to favorite
   */
  favorite() {
    var ids = { idUsu: this.identity.id, idHero: this.hero.idHero }
    this._UserHeroService.favorite(ids).subscribe(
      res => {
        this.getHeroUsu()
      },
      error => {
        console.log(error)
      }
    )
  }
```

```
  /**
   * Unfavorite hero
   */
  unfavorite() {
    var ids = { idUsu: this.identity.id, idHero: this.hero.idHero }
    this._UserHeroService.unfavorite(ids).subscribe(
      res => {
        this.getHeroUsu()
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * vote a hero
   *  @param {number} hovered
   */
  voteHero(hovered) {
    var ids = {
      score: hovered,
      idUsu: this.identity.id,
      idHero: this.hero.idHero
    }
    this._UserHeroService.voteHero(ids).subscribe(
      res => {
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Check if the user follow the hero
   */
  getHeroUsu() {
    this._UserHeroService.getHeroUsu(this.identity.id, this.idParams).subscribe(
      res => {
        if (res) {
          this.heroUsu = res
          if (this.heroUsu.follow['data'][0] === 1) {
            this.followtHero = true
          } else {
            this.followtHero = false
          }
          if (this.heroUsu.favorite['data'][0] === 1) {
            this.favoriteHero = true
          } else {
            this.favoriteHero = false
          }
        }
        // console.log(this.hero);
      },
      error => {
        console.log(error)
      }
    )
  }
  /**
   * Get Hero comments
   */
  getHeroComment() {
    this._UserHeroService.getHeroComments(this.idParams).subscribe(
      res => {
        this.allComments = res
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Open modal to comment hero
   * @param {string} status
   */
  commentHeroDialog(status): void {
    const dialogRef = this.dialog.open(ComentHeroDialogComponent, {
      data: {
        idUsu: this.identity.id,
        idHero: this.idParams,
        status: status,
        comment: this.comment
      }
    })
    dialogRef.afterClosed().subscribe(result => {
      console.log('The dialog was closed')
      this.getHeroUsu()
      this.getHeroComment()
    })
  }
}
```

```
<div id="HeroContainer" class="color1">
  <div class="cardHero">
    <div class="points">
      <p *ngIf="rate == 10" class="tenP">{{ rate }}</p>
      <p *ngIf="rate != 10" class="noTenP">{{ rate }}</p>
      <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 26 26">
        <g transform="translate(0 -1026.362)">
          <circle cx="13" cy="1039.362" r="13" fill="#ffa40b" />
          <path
            style="
              line-height: normal;
              text-indent: 0;
              text-align: start;
              text-decoration-line: none;
              text-decoration-style: solid;
              text-decoration-color: #000;
              text-transform: none;
              block-progression: tb;
              isolation: auto;
              mix-blend-mode: normal;
              marker: none;
            "
            fill="#ff7712"
            d="M 15.042969 25.830078 A 12.999989 12.999989 0 0 0 15.208984 25.810547 A 12.999989 12.999989 0 0 0 16.476562 25.525391 A 12.999989 12.999989 0 0 0 17.710938 25.117188 A
            color="#000"
            font-family="sans-serif"
            font-weight="400"
            transform="translate(0 1026.362)"
            white-space="normal"
          />
          <path
            style="
              line-height: normal;
              text-indent: 0;
              text-align: start;
              text-decoration-line: none;
              text-decoration-style: solid;
              text-decoration-color: #000;
              text-transform: none;
              block-progression: tb;
              isolation: auto;
              mix-blend-mode: normal;
              marker: none;
        "
        fill="#fff"
        d="m 78.795036,1015.7352 c 0.05089,0.074 0.09807,0.137 0.162364,0.2547 0.194225,0.3554 0.422345,0.868 0.647368,1.3973 0.225027,0.5294 0.448538,1.0769 0.652305,1.529 0.1018
        color="#000"
        font-family="sans-serif"
        font-weight="400"
        overflow="visible"
        transform="matrix(1.06795 0 0 .97117 -71.11 46.368)"
        white-space="normal"
      />
    </g>
  </svg>
</div>
<div class="nameAndFollow">
  <h1>{{ hero.heroName }}</h1>

  <div class="HButton" *ngIf="identity">
    <button
      *ngIf="!followtHero"
      type="button"
      class="followB"
      mat-raised-button
      (click)="followHero()"
    >
      <span>FOLLOW</span>
    </button>
    <button
      *ngIf="followtHero"
      type="button"
      class="UnfollowB"
      mat-raised-button
      (click)="unfollowHero()"
    >
      <span>UNFOLLOW</span>
    </button>
    <button
      *ngIf="followtHero && !favoriteHero"
      type="button"
      class="followB"
      mat-raised-button
      (click)="favorite()"
    >
      <span>FAVORITE</span>
    </button>
    <button
      *ngIf="followtHero && favoriteHero"
      type="button"
      class="UnfollowB"
      mat-raised-button
      (click)="unfavorite()"
    >
      <span>UNFAVORITE</span>
    </button>
  </div>
</div>
<div id="HeroDetail">
  <img
    [src]="hero.image"
    [alt]="hero.heroName"
    class="imgH"
    onError="this.src='../../../assets/img/nogender_2.jpg'"
  />
  <div class="HeroInfo">
    <div class="stats">
      <div class="statsC">
        <img
          class="icon"
          src="../../../assets/img/intelligence.svg"
          alt=""
        />
        <div>
          <p class="statName">INTELLIGENCE</p>
          <p class="progresB I">
            <ngb-progressbar
              type="warning"
              [value]="hero.intelligence"
              [striped]="true"
              >{{ hero.intelligence }}
            </ngb-progressbar>
          </p>
        </div>
      </div>
      <div class="statsC">
        <img class="icon" src="../../../assets/img/combat.svg" alt="" />
        <div>
          <p class="statName">COMBAT</p>
          <p class="progresB C">
            <ngb-progressbar
              type="warning"
              [value]="hero.combat"
              [striped]="true"
              >{{ hero.combat }}
            </ngb-progressbar>
          </p>
        </div>
      </div>
      <div class="statsC">
        <img class="icon" src="../../../assets/img/durability.svg" alt="" />
        <div>
          <p class="statName">DURABILITY</p>
          <p class="progresB D">
            <ngb-progressbar
              type="warning"
              [value]="hero.durability"
              [striped]="true"
              >{{ hero.durability }}
            </ngb-progressbar>
          </p>
        </div>
      </div>
      <div class="statsC">
        <img class="icon" src="../../../assets/img/power.svg" alt="" />
        <div>
          <p class="statName">POWER</p>
          <p class="progresB P">
            <ngb-progressbar
              type="warning"
              [value]="hero.power"
              [striped]="true"
              >{{ hero.power }}</ngb-progressbar
            >
          </p>
        </div>
      </div>
      <div class="statsC">
        <img class="icon" src="../../../assets/img/speed.svg" alt="" />
        <div>
          <p class="statName">SPEED</p>
          <p class="progresB S">
            <ngb-progressbar
              type="warning"
              [value]="hero.speed"
              [striped]="true"
              >{{ hero.speed }}</ngb-progressbar
            >
          </p>
        </div>
      </div>
```

```html
        <div class="statsC">
          <img class="icon" src="../../../assets/img/strenght.svg" alt="" />
          <div>
            <p class="statName">STRENGTH</p>
            <p class="progresB ST">
              <ngb-progressbar
                type="warning"
                [value]="hero.strength"
                [striped]="true"
                >{{ hero.strength }}
              </ngb-progressbar>
            </p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="cardHero2">
  <div class="cardHero3">
    <div class="infoCard3">
      <p><span>NAME:</span> {{ hero.fullName }}</p>
      <p><span>PLACE OF BIRTH:</span> {{ hero.placeOfBirth }}</p>
      <p><span>PUBLISHER:</span> {{ hero.publisher }}</p>
      <p><span>ALIGNMENT:</span> {{ hero.alignment }}</p>
      <p><span>FIRST APPERANCE: </span>{{ hero.firstApperance }}</p>
      <p><span>RACE:</span> {{ hero.race }}</p>
      <p><span>WORK:</span> {{ hero.work }}</p>
    </div>
    <div class="silueteI">
      <img
        *ngIf="hero.gender === 'Male'"
        src="./../../../assets/img/man.png"
        alt=""
      />
      <img
        *ngIf="hero.gender != 'Male' && hero.gender != 'Female'"
        src="./../../../assets/img/noGenderF.png"
        alt=""
      />
      <img
        *ngIf="hero.gender === 'Female'"
        src="./../../../assets/img/woman.png"
        alt=""
      />

      <p class="gender">{{ hero.gender }}</p>
      <p class="height">{{ hero.height }}</p>
      <p class="weight">{{ hero.weight }}</p>
      <p class="eyes">{{ hero.eyeColor }} eyes</p>
      <p class="hair">{{ hero.hairColor }} hair</p>
    </div>
  </div>
  <div *ngIf="hero.biography" class="Bio">
    <h5>BIOGRAPHY</h5>
    <p>{{ hero.biography }}</p>
  </div>
</div>

<div *ngIf="identity">
  <div class="YScore">
    <h5>YOUR SCORE</h5>
    <ngb-rating
      class="star"
      [(rate)]="heroUsu.score"
      (hover)="hovered = $event"
      (leave)="hovered = 0"
      [readonly]="readonly"
      [max]="10"
      (click)="voteHero(hovered)"
    ></ngb-rating>
  </div>
  <div class="Ycomment">
    <h5>YOUR COMMENT</h5>
    <img
      *ngIf="!heroUsu.comment"
      class="iconBuuton"
      (click)="commentHeroDialog('new')"
      src="../../../assets/img/message.svg"
      alt=""
    />
    <img
      *ngIf="heroUsu.comment"
      class="iconBuuton"
      (click)="commentHeroDialog('edit')"
      src="../../../assets/img/job.svg"
      alt=""
    />
    <img
      *ngIf="heroUsu.comment"
      class="iconBuuton"
      (click)="commentHeroDialog('delete')"
      src="../../../assets/img/trash.svg"
      alt=""
    />
  </div>
  <p class="comP">{{ heroUsu.comment }}</p>
</div>
<div class="comments">
  <h4>ALL COMMENTS</h4>
  <div class="tablaContainer">
    <table *ngIf="allComments" class="table">
      <thead>
        <tr>
          <th class="imgUser" scope="col"></th>
          <th scope="col">USER</th>
          <th scope="col">COMMENT</th>
        </tr>
      </thead>
      <tbody>
        <tr
          *ngFor="
            let comment of allComments
              | slice: (page - 1) * pageSize:(page - 1) * pageSize + pageSize;
            index as i
          "
        >
          <td class="usu imgUser">
            <div class="userAvatar2 img">
              <img
                class="imgUser"
                *ngIf="comment.photo"
                src="{{ comment.photo }}"
                alt="Photo a book"
              />
            </div>
          </td>
          <td
            class="textName"
            [routerLink]="['/publicProfile', comment.idUsu]"
          >
            <p>
              {{
                comment.alias.length > 20
                  ? (comment.alias | slice: 0:20) + "..."
                  : comment.alias
              }}
            </p>
          </td>
```

```
            <td class="comS">
              <ngb-rating
                class="star2"
                [(rate)]="comment.score"
                [readonly]="true"
                max="10"
              >
              </ngb-rating>
              <br />
              {{ comment.comment }}
            </td>
          </tr>
        </tbody>
      </table>
    </div>
    <div *ngIf="!allComments">
      <div class="noInfo">
        <p>There is not comments</p>
      </div>
    </div>
  </div>
  <div *ngIf="allComments" class="pag">
    <ngb-pagination
      [(page)]="page"
      [pageSize]="pageSize"
      [collectionSize]="allComments.length"
    >
    </ngb-pagination>
  </div>
</div>
```

./heroes-detail.component.scss

```
ngb-rating {
  outline: none;
}

.star {
  position: relative;
  display: inline-block;
  font-size: 2rem;
  color: #cc4224;
}

.cardHero {
  border: 2.5px solid #00a23d;
  box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
  padding: 2em;
  padding-top: 0.5em;
  background-color: #272727;
  color: white;
}
.cardHero2 {
  border: 2.5px solid #00a23d;
  box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
  padding: 2em;
  background-color: #272727;
  color: white;
  margin-top: 2em !important;
}
.cardHero3 {
  display: flex;
  font-family: "B612";
  flex-direction: column;
  justify-content: space-around;
  align-items: center;
}
.Bio {
  padding-top: 2vw;
  text-align: center;
  font-family: "B612";
}
.Bio h5 {
  color: #00a23d;
}
.cardHero h1 {
  text-align: center;
  padding-bottom: 1vw;
  color: #cc4224;
  font-family: "Bangers", cursive;
  text-decoration: underline;
  -webkit-text-decoration-color: #00a23d; /* Safari */
  text-decoration-color: #00a23d;
}
#HeroDetail {
  display: flex;
  flex-direction: column;
  justify-content: space-around;
}
.imgH {
  border-radius: 2%;
  width: 100%;
  height: 100%;
  box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
}

.HeroInfo {
  width: 100%;
}
.stats {
  padding-top: 2vw;
}
.statsC {
  display: flex;
  flex-direction: row;
  padding-bottom: 0.4vw;
  align-items: center;
}
.statsC div {
  padding-left: 2vw;
  width: 100%;
  font-family: "B612";
  color: rgb(223, 221, 221);
}
.icon {
  width: 10vw !important;
  height: 10vw !important;
  padding-top: 0 !important;
}
.progresB {
  width: 100%;
  font-size: 2vw;
}
::ng-deep.progress {
  font-size: 3.5vw !important;
  height: 4.5vw !important;
  border-radius: 0 !important;
  background-color: #272727 !important;
}
.statName {
  margin-bottom: 0;
}
.S ::ng-deep.progress-bar {
  background-color: #45c2f4 !important;
}

.I ::ng-deep.progress-bar {
  background-color: #fa681c !important;
}
```

```
.C ::ng-deep.progress-bar {
  background-color: #ff634d !important;
}

.D ::ng-deep.progress-bar {
  background-color: #b429f6 !important;
}

.P ::ng-deep.progress-bar {
  background-color: #ffc627 !important;
}

.ST ::ng-deep.progress-bar {
  background-color: #7bd74f !important;
}

.silueteI {
  color: #cc4224;
  position: relative;
}
.silueteI img {
  width: 63vw;
  height: 86vw;
}
.silueteI p {
  position: absolute;
  font-size: 14px;
}
.hair {
  bottom: 90%;
  right: 12%;
}
.eyes {
  bottom: 83%;
  right: 12%;
}
.gender {
  bottom: 94%;
  right: 73%;
}
.height {
  bottom: 43%;
  right: -24%;
}
.weight {
  bottom: -7%;
  right: 44%;
}

.infoCard3 {
  width: 100%;
}
.infoCard3 span {
  color: #00a23d;
}

.comments {
  margin-top: 2em !important;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-family: "B612";
}
.comments h4 {
  color: #cc4224;
  text-decoration: underline;
  -webkit-text-decoration-color: #00a23d; /* Safari */
  text-decoration-color: #00a23d;
}

///tablee
.tablaContainer {
  color: #00a23d;
}
.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);
}

.star2 {
  font-size: 15px;
  color: #cc4224;
  outline: none;
}

.pag {
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 1em 0 1em 0;
}
.textName {
  padding-top: 30px !important;
  cursor: pointer;
}
.textName:hover p {
  color: #cc4224;
}
.comS {
  width: 70%;
}

.noInfo {
  text-align: center;
  color: #00a23d;
}

//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;
}

.YScore {
  padding-top: 2vw;
  color: #00a23d;
  font-family: "B612";
}
.YScore p {
  color: rgb(223, 221, 221);
}
.iconBuuton {
  cursor: pointer;
  height: 8vw;
}

.Ycomment {
  display: flex;
  flex-direction: row;
  font-family: "B612";
  color: #00a23d;
}
.Ycomment h5 {
  padding-top: 0.5vw;
  padding-right: 0.5vw;
}
.comP {
  font-family: "B612";
  color: rgb(223, 221, 221);
}

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

.followB,
.UnfollowB {
  margin: 0 0.5vw 1.2vw 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: 0.7rem;
  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%);
  }
}

.UnfollowB {
  color: #2a75b3;
}

.nameAndFollow {
  display: flex;
  flex-direction: row;
  justify-content: center;
}

.HButton {
  padding-left: 2vw;
  padding-top: 0.5vw;
}
.imgUser {
  display: none;
}

.points {
  text-align: right;
  position: relative;
}
.points svg {
  height: 3.5em;
  position: relative;
}
.points p {
  position: absolute;
  color: black;
  z-index: 2;
  font-family: "B612";
}
.tenP {
  right: 1.1em;
  top: 1.2em;
}
.noTenP {
  right: 1.5em;
  top: 1.2em;
}

/* web */
@media (min-width: 992px) {
  .cardHero3 {
    flex-direction: row;
  }

  .silueteI img {
    width: 16vw;

    height: 27vw;
  }

  .tablaContainer {
    width: 70%;
  }
  .imgUser {
    display: table-cell;
  }

  .star2 {
    font-size: 27px;
  }
  .iconBuuton {
    height: 2vw;
  }

  .followB,
  .UnfollowB {
    font-size: 1rem;
  }

  #HeroDetail {
    flex-direction: row;
  }

  .imgH {
    width: 19.5vw;
  }

  .HeroInfo {
    width: 50%;
  }

  .progresB {
    width: 29vw;
  }

  ::ng-deep.progress {
    font-size: 0.9vw !important;
    height: 1.5vw !important;
  }
  .icon {
    width: 4vw !important;
    height: 4vw !important;
  }

  .infoCard3 {
    width: 40%;
  }

  .stats {
    padding-top: 0;
  }
}
```

**Legend**
Html element
Component
Html element with directive

# result-matching ""

# No results matching ""