Type to search

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

File

```
src/app/components/fight/fight-hero/fight-hero.component.ts
```

Description

Component for Fight hero

Implements

[OnInit](#)

Metadata

selector	`app-fight-hero`
styleUrls	`./fight-hero.component.scss`
templateUrl	`./fight-hero.component.html`

Index

Properties

- Public [dialog](#)
- Public [hero](#)
- Public [sessonHero](#)
- Public [sessonHero2](#)

Methods

- [clearHerosSession](#)
- [getHeroWinner](#)
- [getSessionHeroes](#)
- [ngOnInit](#)
- [playAudio](#)
- [selectFightDialog](#)

Constructor

```
constructor(_hero: HeroService, dialog: MatDialog)
```
Defined in [src/app/components/fight/fight-hero/fight-hero.component.ts:28](#)

Constructor in which we inject our services and different elements

Parameters :

Name	Type	Optional
_hero	`HeroService`	No
dialog	`MatDialog`	No

Methods

clearHerosSession

`clearHerosSession()`

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:50

clear local storage

Returns : void

getHeroWinner

`getHeroWinner()`

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:73

Get winner hero

Returns : void

getSessionHeroes

`getSessionHeroes()`

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:57

Get heroes in session storage

Returns : void

ngOnInit

`ngOnInit()`

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:43

Start when the component inits

Returns : void

playAudio

`playAudio()`

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:65

Play audio

Returns : void

selectFightDialog

`selectFightDialog(position)`

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:90

Open modal to choose hero

Parameters :

Name	Optional
position	No

Returns : void

Properties

Public dialog

Type : `MatDialog`

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:33

Public hero

Type : Hero

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:20

variable to store the winner hero

Public sessonHero

Type : Hero

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:24

variable to store the 1st hero in local storage

Public sessonHero2

Type : Hero

Defined in src/app/components/fight/fight-hero/fight-hero.component.ts:28

variable to store the 2nd hero in local storage

```typescript
import { Component, OnInit } from '@angular/core'
import { Hero } from 'src/app/models/hero'
import { HeroService } from 'src/app/services/hero.service'
import { MatDialog } from '@angular/material'
import { selectHeroComponent } from '../../modals/select-hero-dialog/select-hero-dialog.component'

/**
 * Component for Fight hero
 */
@Component({
  selector: 'app-fight-hero',
  templateUrl: './fight-hero.component.html',
  styleUrls: ['./fight-hero.component.scss']
})
export class FightHeroComponent implements OnInit {
  /**
   * variable to store the winner hero
   */
  public hero: Hero
  /**
   * variable to store the 1st hero in local storage
   */
  public sessonHero: Hero
  /**
   * variable to store the 2nd hero in local storage
   */
  public sessonHero2: Hero

  /**
   * Constructor in which we inject our services and different elements
   */
  constructor(private _hero: HeroService, public dialog: MatDialog) {
    this.hero = new Hero(0, '', '', 0, 0, 0, 0, 0, 0, '', '', '', '', '', '', '', '', '', '', '', '', null, 0)
    this.sessonHero = new Hero(0, '', '', 0, 0, 0, 0, 0, 0, '', '', '', '', '', '', '', '', '', '', '', '', null, 0)
    this.sessonHero2 = new Hero(0, '', '', 0, 0, 0, 0, 0, 0, '', '', '', '', '', '', '', '', '', '', '', '', null, 0)

  }

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

  /**
   * clear local storage
   */
  clearHerosSession() {
    sessionStorage.clear()
  }

  /**
   * Get heroes in session storage
   */
  getSessionHeroes() {
    this.sessonHero = JSON.parse(sessionStorage.getItem('Hero'))
    this.sessonHero2 = JSON.parse(sessionStorage.getItem('Hero2'))
  }

  /**
   * Play audio
   */
  playAudio() {
    let audio = <HTMLAudioElement>document.getElementById("myAudio");
    audio.play();
  }

  /**
   * Get winner hero
   */
  getHeroWinner() {
    this._hero
      .getWinner(this.sessonHero.idHero, this.sessonHero2.idHero)
      .subscribe(
        res => {
          this.hero = res
        },
        error => {
          console.log(error)
        }
      )
  }

  /**
```

```
  * Open modal to choose hero
  * @param {number} position
  */
 selectFightDialog(position) {
   const dialogRef = this.dialog.open(selectHeroComponent, {
     data: {
       position: position,
       action: 'fight'
     }
   })
   dialogRef.afterClosed().subscribe(result => {
     this.getSessionHeroes()
   })
 }
}

<div class="color2 padCom">
  <div class="titleFi">
    <h2>FIGHT HERO: ONE VS ONE</h2>
    <h5>
      Select the first hero by clicking on the
      <span> Select first hero</span> button and searching among the many heroes
      we have below; then select a second hero whth the
      <span>Select second hero</span> button and then <span>click</span> on the
      fight button! We will see <span> WHO IS THE BEST!!!</span>.
    </h5>
    <p>The winner of the battle is calculated based on its stats.</p>
  </div>
  <div class="selectTeam">
    <div class="firstTEAM">
      <button
        type="button"
        class="UnfollowB"
        mat-raised-button
        (click)="selectFightDialog(1)"
      >
        <span>Select first Hero</span>
      </button>
      <div *ngIf="!sessonHero || sessonHero.idHero == 0">
        <h3 class="teamName">
          NO HERO SELECTED
        </h3>
        <div id="TeamContainer">
          <div class="card2">
            <img
              src="./../../../../assets/img/nomember.jpg  "
              alt="nohero"
              class="img-fluid img2"
            />
            <div class="overImg rela">
              <h4></h4>
            </div>
          </div>
        </div>
      </div>
      <div *ngIf="sessonHero && sessonHero.idHero > 0" class="sessooonHero">
        <div class="card2">
          <img
            *ngIf="sessonHero.image"
            [src]="sessonHero.image"
            [routerLink]="['/heroesDetail', sessonHero.idHero]"
            [alt]="sessonHero.heroName"
            onError="this.src='../../../assets/img/nogender_2.jpg'"
            class="img-fluid img2"
          />
          <div class="overImg rela">
            <h4>{{ sessonHero.heroName }}</h4>
          </div>
        </div>
      </div>
    </div>
    <div class="fightB">
      <audio id="myAudio">
        <source
          src="./../../../../assets/sound/276254__dynajinn__fight-voiceover-fight.wav"
          type="audio/wav"
        />
        Your browser does not support the audio element.
      </audio>
      <div class="BF">
        <img
          class="iconBuuton"
          (click)="getHeroWinner(); playAudio()"
          src="./../../../../assets/img/sword.svg"
          alt=""
        />
        <p class="text">
          FIGHT!!!!
        </p>
      </div>
    </div>
    <div class="firstTEAM">
```

```html
      <button
        type="button"
        class="UnfollowB"
        mat-raised-button
        (click)="selectFightDialog(2)"
      >
        <span>Select second hero</span>
      </button>
      <div *ngIf="!sessonHero2 || sessonHero2.idHero == 0">
        <h3 class="teamName">
          NO HERO SELECTED
        </h3>
        <div id="TeamContainer">
          <div class="card2">
            <img
              src="./../../../../assets/img/nomember.jpg  "
              alt="nohero"
              class="img-fluid img2"
            />
            <div class="overImg rela">
              <h4></h4>
            </div>
          </div>
        </div>
      </div>
      <div *ngIf="sessonHero2 && sessonHero2.idHero > 0" class="sessooonHero">
        <div class="card2">
          <img
            *ngIf="sessonHero2.image"
            [src]="sessonHero2.image"
            [routerLink]="['/heroesDetail', sessonHero2.idHero]"
            [alt]="sessonHero2.heroName"
            onError="this.src='../../../assets/img/nogender_2.jpg'"
            class="img-fluid img2"
          />
          <div class="overImg rela">
            <h4>{{ sessonHero2.heroName }}</h4>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div *ngIf="hero.idHero" class="teamwinner">
  <div class="contwinner">
    <h1 class="h1Teamwin">
      THE WINNER IS: <span>{{ hero.heroName }}</span>
    </h1>
    <div class="cardWin">
      <div class="card2">
        <img class="imgIconW" src="./../../../../assets/img/crown.svg" />
        <img
          *ngIf="hero.image"
          [src]="hero.image"
          [routerLink]="['/heroesDetail', hero.idHero]"
          [alt]="hero.heroName"
          onError="this.src='../../../assets/img/nogender_2.jpg'"
          class="img-fluid img2"
        />
        <div class="overImg rela">
          <h4>{{ hero.heroName }}</h4>
        </div>
      </div>
    </div>
  </div>
</div>


./fight-hero.component.scss

.selectTeam {
  display: flex;
  align-items: center;
  justify-content: space-between;
  flex-direction: column;
  align-items: center;
}

.firstTEAM {
  display: flex;
  width: 100%;
  align-items: center;
  flex-direction: column;
}
.team1 {
  width: 50%;
}

#TeamContainer,
.sessooonHero,
.cardWin {
  display: flex;
  width: 100% !important;
```

```scss
    flex-wrap: nowrap !important;
    justify-content: space-around !important;
}


.teamName {
    padding-bottom: 1vw;
    color: #cc4224;
    font-family: "B612";
    font-size: 0.75rem;
}


.teamName span {
    color: #00a23d;
}


.fightB {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    min-height: 6em;
}


.text {
    opacity: 0;
}
.BF:hover .text {
    transform: translate(35%, 10px);
    opacity: 1;
    font-family: "B612";
    color: rgb(223, 221, 221);
}


.iconBuuton {
    cursor: pointer;
    height: 12vw;
    background-color: #f3d403;
    border-radius: 2%;
    width: 56vw;
    border: 2.5px solid #00a23d;
}


.iconBuuton:hover {
    transform: scale(1.1);
    -ms-transform: scale(1.1);
    -moz-transform: scale(1.1);
    -webkit-transform: scale(1.1);
    -o-transform: scale(1.1);
    background-color: #cc4224;
}

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

.UnfollowB {
    margin: 0 0.5vw 3vw 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;
```

```scss
    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%);
  }
}

.h1Teamwin {
  color: #f3d403;
  font-family: "Bangers", cursive;
  -webkit-text-stroke: 2px #cc4224;
  padding: 0.5em;
  font-size: 2.5rem !important;
}
.h1Teamwin span {
  color: #00a23d;
}

.titleFi h2 {
  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;
}
.titleFi h5 {
  color: rgb(223, 221, 221);
}
.titleFi h5 span {
  color: #00a23d;
  font-family: "B612";
}
.titleFi p {
  color: #cc4224;
  font-family: "B612";
  padding-bottom: 2vw;
}

.teamwinner {
  padding: 2em;
  background-color: #272727;
  color: white;
  display: flex;
  align-items: center;
  justify-content: center;
  padding-bottom: 6em;
}
.teamwinner .contwinner {
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  text-align: center;
  background-color: #1a1a1a;
  border: 2.5px solid #00a23d;
  box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
}
.imgIconW {
  position: absolute;
  z-index: 3;
  right: 1em;
  top: 0em;
  transform: rotate(20deg);
  width: 25px;
  height: 25px;
}
.card2:hover .imgIconW {
  transform: scale(1.1);
  -ms-transform: scale(1.1);
  -moz-transform: scale(1.1);
  -webkit-transform: scale(1.1);
  -o-transform: scale(1.1);
}
::ng-deep .cardWin .card2 {
  width: 50%;
}
```

```
::ng-deep .cardWin .card2 .img2 {
  border: 8px dotted #00a23d !important;
  border-radius: 2%;
}

@media (min-width: 992px) {
  .imgIconW {
    top: -1em;
    width: 50px;
    height: 50px;
  }
  .BF:hover .text {
    transform: translate(35%, 20px);
  }
  .iconBuuton {
    height: 8vw;
    width: 10vw;
  }

  .firstTEAM {
    width: 40%;
  }
  .selectTeam {
    flex-direction: row;
  }

  #TeamContainer {
    flex-direction: row;
  }

  .img-fluid {
    width: 100%;
  }

  .h1Teamwin {
    font-size: 3.5rem !important;
  }

  .fightB {
    min-height: 20em;
  }

  ::ng-deep .sessooonHero .card2 {
    width: 37%;
  }

  .card2 {
    width: 37%;
  }
  ::ng-deep .cardWin .card2 {
    width: 19vw;
  }
  ::ng-deep .cardWin .card2 .img2 {
    height: 25vw;
  }

  .teamwinner .contwinner {
    width: 60%;
  }

  .teamName {
    font-size: 1.75rem;
  }
}

@media (min-width: 768px) and (max-width: 991px) {
  .teamName {
    font-size: 1.75rem;
  }

  .titleFi h5 {
    font-size: 1.7rem;
  }
}
```

Legend
Html element
Component
Html element with directive

result-matching ""

No results matching ""