

1. Injectables
2. HeroService

- [Info](#)
- [Source](#)

File

`src/app/services/hero.service.ts`

Description

Service that contains all the routes to our api regarding heroes

Index

Properties

- Private [baseUrl](#)

Methods

- Public [allDCHeroes](#)
- Public [allMarvelHeroes](#)
- Public [getHeroById](#)
- Public [getHeroByName](#)
- Public [getWinner](#)
- Public [modifyHero](#)
- Public [mostCombat](#)
- Public [mostDurability](#)
- Public [mostIntelligence](#)
- Public [mostPower](#)
- Public [mostSpeed](#)
- Public [mostStrength](#)
- Public [newHero](#)
- Public [newHeros](#)
- Public [profileImgHeroes](#)

Constructor

constructor(http: [HttpClient](#))

Defined in [src/app/services/hero.service.ts:18](#)

Constructor in which we inject httpClient to make http requests

Parameters :

Name	Type	Optional
http	HttpClient	No

Methods

Public allDCHeroes

`allDCHeroes()`

Defined in [src/app/services/hero.service.ts:57](#)

Get all the DC heroes

Returns : [Observable<Hero\[\]>](#)

Array of heroes

Public allMarvelHeroes

allMarvelHeroes()

Defined in [src/app/services/hero.service.ts:49](#)

Get all the marvel heroes

Returns : [Observable<Hero\[\]>](#)

Array of heroes

Public getHeroById

getHeroById(id: [any](#))

Defined in [src/app/services/hero.service.ts:41](#)

Get a hero by id

Parameters :

Name Type Optional

id [any](#) No

Returns : [Observable<Hero>](#)

Hero

Public getHeroByName

getHeroByName(name: [string](#))

Defined in [src/app/services/hero.service.ts:30](#)

Get a hero by his name

Parameters :

Name Type Optional

name [string](#) No

Returns : [Observable<Hero\[\]>](#)

Array of heroes

Public getWinner

getWinner(idHero: [number](#), idHero2: [number](#))

Defined in [src/app/services/hero.service.ts:67](#)

Get the hero with the highest points

Parameters :

Name Type Optional

idHero [number](#) No

idHero2 [number](#) No

Returns : [Observable<Hero>](#)

Hero

Public modifyHero

modifyHero(idHero: [number](#), data)

Defined in [src/app/services/hero.service.ts:155](#)

Modify hero

Parameters :

Name	Type	Optional
------	------	----------

idHero	number	No
--------	------------------------	----

data		No
------	--	----

Returns : [any](#)

Public mostCombat

mostCombat ()

Defined in [src/app/services/hero.service.ts:117](#)

Get the hero with more combat stats

Returns : [Observable<Hero>](#)

Hero

Public mostDurability

mostDurability ()

Defined in [src/app/services/hero.service.ts:101](#)

Get the more resistant hero

Returns : [Observable<Hero>](#)

Hero

Public mostIntelligence

mostIntelligence ()

Defined in [src/app/services/hero.service.ts:77](#)

Get the smartest hero

Returns : [Observable<Hero>](#)

Hero

Public mostPower

mostPower ()

Defined in [src/app/services/hero.service.ts:109](#)

Get the more powerful hero

Returns : [Observable<Hero>](#)

Hero

Public mostSpeed

mostSpeed ()

Defined in [src/app/services/hero.service.ts:93](#)

Get the fastest hero

Returns : [Observable<Hero>](#)

Hero

Public mostStrength

mostStrength ()

Defined in [src/app/services/hero.service.ts:85](#)

Get the strongest hero

Returns : [Observable<Hero>](#)

Hero

Public newHero

newHero(hero)

Defined in [src/app/services/hero.service.ts:141](#)

Create a new hero

Parameters :

Name Optional

hero No

Returns : [any](#)

Public newHeros

newHeros()

Defined in [src/app/services/hero.service.ts:125](#)

Get the latest heroes added to the bd

Returns : [Observable<Hero\[\]>](#)

Array of heroes

Public profileImgHeroes

profileImgHeroes()

Defined in [src/app/services/hero.service.ts:133](#)

Get hero images for profile

Returns : Observable<any[]>

Array of heroes img

Properties

Private baseUrl

Type : [string](#)

Default value : environment.BASE_API_URL

Defined in [src/app/services/hero.service.ts:18](#)

Variable in which we store the url of our api

```
import { Injectable } from '@angular/core';
import { environment } from 'src/environments/environment';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Hero } from '../models/hero';
import { Observable } from 'rxjs';

/**
 * Service that contains all the routes to our api regarding heroes
 */
@Injectable({
  providedIn: 'root'
})

export class HeroService {
  /**
   * Variable in which we store the url of our api
   */
  private baseUrl: string = environment.BASE_API_URL;

  /**
   * Constructor in which we inject httpClient to make http requests
   */
  constructor(private http: HttpClient) { }
```

```

/**
 * Get a hero by his name
 * @param {string} name
 * @returns Array of heroes
 */
public getHeroByName(name: string): Observable<Hero[]> {
    return this.http.get<Hero[]>(`${this.baseUrl}/hero/searchHeroByName/${name}`);
}

/**
 * Get a hero by id
 * @param {any} id
 * @returns Hero
 */
public getHeroById(id: any): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/getByid/${id}`);
}

/**
 * Get all the marvel heroes
 * @returns Array of heroes
 */
public allMarvelHeroes(): Observable<Hero[]> {
    return this.http.get<Hero[]>(`${this.baseUrl}/hero/all/MarvelHeroes`);
}

/**
 * Get all the DC heroes
 * @returns Array of heroes
 */
public allDCHeroes(): Observable<Hero[]> {
    return this.http.get<Hero[]>(`${this.baseUrl}/hero/all/DCHeroes`);
}

/**
 * Get the hero with the highest points
 * @param {number} idHero
 * @param {number} idHero2
 * @returns Hero
 */
public getWinner(idHero: number, idHero2: number): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/getWinner/${idHero}/${idHero2}`);
}

/**
 * Get the smartest hero
 * @returns Hero
 */
public mostIntelligence(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/stats/mostIntelligence`);
}

/**
 * Get the strongest hero
 * @returns Hero
 */
public mostStrength(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/stats/mostStrength`);
}

/**
 * Get the fastest hero
 * @returns Hero
 */
public mostSpeed(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/stats/mostSpeed`);
}

/**
 * Get the more resistant hero
 * @returns Hero
 */
public mostDurability(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/stats/mostDurability`);
}

/**
 * Get the more powerful hero
 * @returns Hero
 */
public mostPower(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/stats/mostPower`);
}

```

```

}

/**
 * Get the hero with more combat stats
 * @returns Hero
 */
public mostCombat(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/hero/stats/mostCombat`);
}

/**
 * Get the latest heroes added to the bd
 * @returns Array of heroes
 */
public newHeros(): Observable<Hero[]> {
    return this.http.get<Hero[]>(`${this.baseUrl}/hero/new/heros`);
}

/**
 * Get hero images for profile
 * @returns Array of heroes img
 */
public profileImgHeroes(): Observable<any[]> {
    return this.http.get<any[]>(`${this.baseUrl}/hero/img/profileImgHeroes`);
}

/**
 * Create a new hero
 * @param {Hero} hero
 */
public newHero(hero) {
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    const tokenAuth = localStorage.getItem('token');
    headers = headers.set('Authorization', `${tokenAuth}`);
    return this.http.post(this.baseUrl + '/admin/newHero', hero, {
        headers
    });
}

/**
 * Modify hero
 * @param {number} idHero
 * @param {any} data
 */
public modifyHero(idHero: number, data) {
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    const tokenAuth = localStorage.getItem('token');
    headers = headers.set('Authorization', `${tokenAuth}`);
    return this.http.put(`${this.baseUrl}/admin/modifyHero/${idHero}`, data, {
        headers
    });
}
}

```

result-matching ""

No results matching ""