

1. Injectables
2. UserHeroService

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

File

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

Description

Service that contains all the routes to our api about the relationship of users and heroes

Index

Properties

- Private [baseUrl](#)

Methods

- Public [allHerosFav](#)
- Public [allHerosFoll](#)
- Public [bestDCHero](#)
- Public [bestMarverHero](#)
- Public [bestOTHero](#)
- Public [commentHero](#)
- Public [deleteCHero](#)
- Public [favorite](#)
- Public [followHero](#)
- Public [getHeroComments](#)
- Public [getHeroRateScore](#)
- Public [getHeroUsu](#)
- Public [mostFollowHeros](#)
- Public [unfavorite](#)
- Public [unfollowHero](#)
- Public [voteHero](#)

Constructor

constructor(http: [HttpClient](#))

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

Constructor in which we inject httpClient to make http requests

Parameters :

Name	Type	Optional
http	HttpClient	No

Methods

Public allHerosFav

`allHerosFav(idUsu: number)`

Defined in [src/app/services/user-hero.service.ts:112](#)

Get all the favorite heroes

Parameters :

Name Type Optional

idUsu [number](#) No

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

hero[]

Public allHerosFoll

allHerosFoll(idUsu: [number](#))

Defined in [src/app/services/user-hero.service.ts:121](#)

Get all the follow heroes

Parameters :

Name Type Optional

idUsu [number](#) No

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

hero[]

Public bestDCHero

bestDCHero()

Defined in [src/app/services/user-hero.service.ts:145](#)

Get top rated DC hero

Returns : [Observable<Hero>](#)

hero

Public bestMarverHero

bestMarverHero()

Defined in [src/app/services/user-hero.service.ts:129](#)

Get top rated marvel hero

Returns : [Observable<Hero>](#)

hero

Public bestOTHero

bestOTHero()

Defined in [src/app/services/user-hero.service.ts:137](#)

Get top rated other hero

Returns : [Observable<Hero>](#)

hero

Public commentHero

commentHero(data)

Defined in [src/app/services/user-hero.service.ts:76](#)

Comment a hero

Parameters :

Name Optional

data No

Returns : [any](#)

Public deleteCHero

`deleteCHero(ids)`

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

Delete comment

Parameters :

Name Optional

ids No

Returns : [any](#)

Public favorite

`favorite(ids)`

Defined in [src/app/services/user-hero.service.ts:47](#)

Favorite a hero

Parameters :

Name Optional

ids No

Returns : [any](#)

Public followHero

`followHero(ids)`

Defined in [src/app/services/user-hero.service.ts:29](#)

Follow a hero

Parameters :

Name Optional

ids No

Returns : [any](#)

Public getHeroComments

`getHeroComments(id: number)`

Defined in [src/app/services/user-hero.service.ts:162](#)

Get all the comments on a hero

Parameters :

Name Type Optional

id [number](#) No

Returns : Observable<any>

any

Public getHeroRateScore

`getHeroRateScore(id: number)`

Defined in [src/app/services/user-hero.service.ts:171](#)

Get the hero's average score

Parameters :

Name Type Optional

id [number](#) No

Returns : Observable<any>

any

Public getHeroUsu

`getHeroUsu(idUsu: number, idHero: number)`

Defined in [src/app/services/user-hero.service.ts:97](#)

Get the relationship of the hero and the user

Parameters :

Name Type Optional

idUsu [number](#) No

idHero [number](#) No

Returns : [Observable<UserHero>](#)

UserHero

Public mostFollowHeros

[mostFollowHeros\(\)](#)

Defined in [src/app/services/user-hero.service.ts:153](#)

Get the most followed heroes

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

hero[]

Public unfavorite

[unfavorite\(ids\)](#)

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

Unfavorite a hero

Parameters :

Name Optional

ids No

Returns : [any](#)

Public unfollowHero

[unfollowHero\(ids\)](#)

Defined in [src/app/services/user-hero.service.ts:38](#)

Unfollow a hero

Parameters :

Name Optional

ids No

Returns : [any](#)

Public voteHero

[voteHero\(data\)](#)

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

vote a hero

Parameters :

Name Optional

data No

Returns : [any](#)

Properties

Private baseUrl

Type : [string](#)

Default value : [environment.BASE_API_URL](#)

Defined in [src/app/services/user-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 { UserHero } from '../models/userHero';
import { Hero } from '../models/hero';
import { Observable } from 'rxjs';

/**
 * Service that contains all the routes to our api about the relationship of users and heroes
 */
@Injectable({
  providedIn: 'root'
})
export class UserHeroService {
  /**
   * 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) { }

  /**
   * Follow a hero
   * @param {any} ids
   */
  public followHero(ids) {
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    return this.http.put(`${this.baseUrl}/userHero/followHero`, ids, { headers: headers });
  }

  /**
   * Unfollow a hero
   * @param {any} ids
   */
  public unfollowHero(ids) {
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    return this.http.put(`${this.baseUrl}/userHero/unfollowHero`, ids, { headers: headers });
  }

  /**
   * Favorite a hero
   * @param {any} ids
   */
  public favorite(ids) {
    let params = JSON.stringify(ids);
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    return this.http.put(`${this.baseUrl}/userHero/favorite`, params, { headers: headers });
  }

  /**
   * Unfavorite a hero
   * @param {any} ids
   */
  public unfavorite(ids) {
    let params = JSON.stringify(ids);
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    return this.http.put(`${this.baseUrl}/userHero/unfavorite`, params, { headers: headers });
  }

  /**
   * vote a hero
   * @param {any} data
   */
  public voteHero(data) {
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    return this.http.put(`${this.baseUrl}/userHero/voteHero`, data, { headers: headers });
  }

  /**
   * Comment a hero
   * @param {any} data
   */
  public commentHero(data) {
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    return this.http.put(`${this.baseUrl}/userHero/commentHero`, data, { headers: headers });
  }

  /**
   * Delete comment
   * @param {any} ids
   */
  public deleteCHero(ids) {
    let params = JSON.stringify(ids);
    let headers = new HttpHeaders().set('Content-Type', 'application/json');
    return this.http.put(`${this.baseUrl}/userHero/deleteCHero`, params, { headers: headers });
  }
}

```

```

/**
 * Get the relationship of the hero and the user
 * @param {number} idUsu
 * @param {number} idHero
 * @returns UserHero
 */
public getHeroUsu(idUsu: number, idHero: number): Observable<UserHero> {
    return this.http.get<UserHero>(`${this.baseUrl}/userHero/getHeroUsu/${idUsu}/${idHero}`);
}

// // //obtener el comentario de un heroe de un usuario
// // router.get('/commentHero/:idUsu/:idHero', UserHeroController.getCommentHero);
// public getcommentHero(idUsu: number, idHero: number) {
//     return this.http.get<UserHero>(`${this.baseUrl}/userHero/commentHero/${idUsu}/${idHero}`);
// }

/**
 * Get all the favorite heroes
 * @param {number} idUsu
 * @returns hero[]
 */
public allHerosFav(idUsu: number): Observable<Hero[]> {
    return this.http.get<Hero[]>(`${this.baseUrl}/userHero/allHerosFav/${idUsu}`);
}

/**
 * Get all the follow heroes
 * @param {number} idUsu
 * @returns hero[]
 */
public allHerosFoll(idUsu: number): Observable<Hero[]> {
    return this.http.get<Hero[]>(`${this.baseUrl}/userHero/allHerosFoll/${idUsu}`);
}

/**
 * Get top rated marvel hero
 * @returns hero
 */
public bestMarverHero(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/userHero/bestMarverHero`);
}

/**
 * Get top rated other hero
 * @returns hero
 */
public bestOTHero(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/userHero/bestOTHero`);
}

/**
 * Get top rated DC hero
 * @returns hero
 */
public bestDCHero(): Observable<Hero> {
    return this.http.get<Hero>(`${this.baseUrl}/userHero/bestDCHero`);
}

/**
 * Get the most followed heroes
 * @returns hero[]
 */
public mostFollowHeros(): Observable<Hero[]> {
    return this.http.get<Hero[]>(`${this.baseUrl}/userHero/mostFollowHeros`);
}

/**
 * Get all the comments on a hero
 * @param {number} id
 * @returns any
 */
public getHeroComments(id: number) : Observable<any>{
    return this.http.get<any>(`${this.baseUrl}/userHero/getHeroComments/${id}`);
}

/**
 * Get the hero's average score
 * @param {number} id
 * @returns any
 */
public getHeroRateScore(id: number): Observable<any> {
    return this.http.get<any>(`${this.baseUrl}/userHero/getHeroRateScore/${id}`);
}

// public modifyCHero( data) {
//     // para convertir el objeto en un string

```

```
// // let params = JSON.stringify(data);  
// let headers = new HttpHeaders().set('Content-Type', 'application/json');  
// return this.http.put(`${this.baseUrl}/userHero/modifyCHero`, data, { headers: headers });  
// }  
}
```

result-matching ""

No results matching ""