

Class: UserController

UserController(UserService)

Class with User controllers functions

Constructor

```
new UserController(UserService)
```

Parameters:

Name	Type	Description
UserService	class	insert our user service in our class

Source: [src/controllers/user.controller.js, line 11](#)

Methods

```
(async) checkFollow(req, res) →  
{object|null}
```

check users relationship

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 113](#)

Returns:

user or null

Type

object | null

```
(async) delete(req, res) → {string}
```

Home

Classes

AdminController
AdminService
ApiHeroController
ApiHeroService
AuthController
AuthService
BaseService
DBConexion
Hero
HeroController
HeroService
Server
Team
TeamController
TeamService
User
UserController
UserHero
UserHeroController
UserHeroService
UserService

Global

_config
conexion
config
container
cors
DB
express
generateToken
isAdmin
swaggerUi

delete a user

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 61](#)

Returns:

message

Type

string

`(async) followUser(req, res) → {string}`

follow a user

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 75](#)

Returns:

message

Type

string

`(async) get(req, res) → {object}`

Get a user by id

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 21](#)

Returns:

user

Type

object

```
(async) getFollowersUsers(req, res) →  
{Array}
```

followers users

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 138](#)

Returns:

users followers

Type

Array

```
(async) getFollowUsers(req, res) →  
{Array|null}
```

followed users

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 126](#)

Returns:

user or null

Type

Array | null

```
(async) getUserByName(req, res) →  
{Array|null}
```

search user by name

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 101](#)

Returns:

user or null

Type

Array | null

```
(async) newImg(req, res) → {string}
```

update img

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 150](#)

Returns:

message

Type

string

```
(async) unFollowUser(req, res) → {string}
```

Unfollow a user

Parameters:

Name	Type	Description
req	Object	Express request object

Name	Type	Description
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 88](#)

Returns:

message

Type
string

`(async) update(req, res) → {string}`

update a user

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 35](#)

Returns:

message

Type
string

`(async) updatePass(req, res) → {string}`

update a user pass

Parameters:

Name	Type	Description
req	Object	Express request object
res	Object	Express response object

Source: [src/controllers/user.controller.js, line 48](#)

Returns:

message

Type
string