Class: User

User(DB)

User class

Constructor

`new User(DB)`

Parameters:

Name	Type	Description
DB	class	insert our db connection class

Source: src/models/user.model.js, line 13

Methods

`(async) allUsers() → {array}`

all users from db

Source: src/models/user.model.js, line 210

Returns:

if the function is successful it will return one or more elements from the DB, if not will be returned indicating that there has been a failure

Type
array

`(async) checkFollow(idUsu, idUnfollow) → {object}`

Check if a user is following another

Parameters:

Name	Type	Description

Name	Type	Description
idUsu	number	id user
idUnfollow	number	id user2

Source: src/models/user.model.js, line 166

Returns:

if the function is successful it will return one elements from the DB, if not will be returned indicating that there has been a failure

Type
object

(async) comparePasswords(UserPas, password) → {boolean}

Compare encrypted passwords

Parameters:

Name	Type	Description
UserPas	string	User pass
password	string	user pass from db

Source: src/models/user.model.js, line 33

Returns:

The result of compareSync function

Type
boolean

(async) create(entity) → {*}

Insert a user in the db

Parameters:

Name	Type	Description
entity	object	body of the element that brings the path

Source: src/models/user.model.js, line 88

Returns:

if the function has been successful, a resolve will be returned indicating that it could be created, otherwise will be returned indicating the opposite

Type
*

(async) deleteUser(idUsu) → {*}

Delete user

Parameters:

Name	Type	Description
idUsu	number	User id

Source: src/models/user.model.js, line 123

Returns:

if the function has been successful, a resolve will be returned indicating that it could be created, otherwise will be returned indicating the opposite

Type
*

(async) followUser(entity) → {*}

follow a user

Parameters:

Name	Type	Description
entity	object	body of the element that brings the path

Source: src/models/user.model.js, line 132

Returns:

if the function has been successful, a resolve will be returned indicating that it could be created, otherwise will be returned indicating the opposite

Type
*

(async) get(id) → {object}

Get a user by id

Parameters:

Name	Type	Description
id	number	

Source: src/models/user.model.js, line 70

Returns:

if the function is successful it will return one element from the DB, if not will be returned indicating that there has been a failure

Type
object

(async) getFollowersUsers(idUsu) → {array}

Get if user is following you

Parameters:

Name	Type	Description
idUsu	number	

Source: src/models/user.model.js, line 177

Returns:

if the function is successful it will return one or more elements from the DB, if not will be returned indicating that there has been a failure

Type
array

(async) getFollowUsers(idUsu) → {array}

Get the people I follow

Parameters:

Name	Type	Description
idUsu	number	

Source: src/models/user.model.js, line 188

Returns:

if the function is successful it will return one or more elements from the DB, if not will be returned indicating that there has been a failure

Type
array

(async) getUserByemail(email) → {object}

Get a user by email

Parameters:

Name	Type	Description
email	string	

Source: src/models/user.model.js, line 79

Returns:

if the function is successful it will return one element from the DB, if not will be returned indicating that there has been a failure

Type
object

(async) getUserByName(userName) → {array}

Get user by name

Parameters:

Name	Type	Description
userName	string	user name

Source: src/models/user.model.js, line 154

Returns:

if the function is successful it will return one or more elements from the DB, if not will be returned indicating that there has been a failure

Type

array

(async) hasPass(password)

To hash pass

Parameters:

Name	Type	Description
password	string	

Source: src/models/user.model.js, line 57

Returns:

hash password

(async) newImg(entity, idUsu) → {*}

upgrade to user img

Parameters:

Name	Type	Description
entity	object	body of the element that brings the path
idUsu	number	User id

Source: src/models/user.model.js, line 200

Returns:

if the function has been successful, a resolve will be returned indicating that it could be created, otherwise will be returned indicating the opposite

Type

*

(async) pre(user) → {object}

check before saving the user that the password is encrypted

Parameters:

Name	Type	Description
user	object	

Source: src/models/user.model.js, line 42

Returns:

user with encrypted pass

Type
object

(async) unFollowUser(entity) → {*}

Unfollow a user

Parameters:

Name	Type	Description
entity	object	body of the element that brings the path

Source: src/models/user.model.js, line 143

Returns:

if the function has been successful, a resolve will be returned indicating that it could be created, otherwise will be returned indicating the opposite

Type
*

(async) update(entity, idUsu) → {*}

upgrade to user

Parameters:

Name	Type	Description
entity	object	body of the element that brings the path
idUsu	number	User id

Source: src/models/user.model.js, line 100

Returns:

if the function has been successful, a resolve will be returned indicating that it could be created, otherwise will be returned indicating the opposite

Type
*

(async) updatePass(entity, idUsu) → {*}

upgrade to user pass

Parameters:

Name	Type	Description
entity	object	body of the element that brings the path
idUsu	number	User id

Source: src/models/user.model.js, line 112

Returns:

if the function has been successful, a resolve will be returned indicating that it could be created, otherwise will be returned indicating the opposite

Type

*
