

# Class: Team

## Team(DB)

team class

## Constructor

new Team(DB)

### Parameters:

Name	Type	Description
DB	class	insert our db connexion class

Source: [src/models/team.model.js, line 12](#)

## Methods

(*async*) addMember(idTeam, entity) → {*\**}

add a member

### Parameters:

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

Source: [src/models/team.model.js, line 72](#)

### 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*) bestTeam() → {object}

## 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

Get the team with highest stats

Source: [src/models/team.model.js, line 102](#)

### 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) create(entity) → { * }`

create a team

### Parameters:

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

Source: [src/models/team.model.js, line 29](#)

### 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) delete(idTeam) → { * }`

delete team

### Parameters:

Name	Type	Description
idTeam	number	

Source: [src/models/team.model.js, line 95](#)

### 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) deleteMember(idTeam, entity) → {*}
```

delete team member

#### Parameters:

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

Source: [src/models/team.model.js, line 84](#)

#### 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(idTeam) → {object}
```

Get a team by id

#### Parameters:

Name	Type	Description
idTeam	number	

Source: [src/models/team.model.js, line 44](#)

#### 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) getTeamInfo(idUsu) → {object}
```

Get the team info

### Parameters:

Name	Type	Description
idUsu	number	

Source: [src/models/team.model.js, line 113](#)

### 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) getTeamWinner(idTeam1, idTeam2) → {object}

Get the winner team

### Parameters:

Name	Type	Description
idTeam1	number	
idTeam2	number	

Source: [src/models/team.model.js, line 123](#)

### 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) searchTeam(teamName) → {array}

search team by name

### Parameters:

Name	Type	Description
teamName	string	

---

Source: [src/models/team.model.js, line 143](#)

### 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*) `update(idTeam, entity) → {*}`

update team name

### Parameters:

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

Source: [src/models/team.model.js, line 60](#)

### 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  
\*