

Class: Hero

Hero(DB)

hero class

Constructor

new Hero(DB)

Parameters:

| Name | Type | Description |
|------|-------|--------------------------------|
| DB | class | insert our db connection class |

Source: [src/models/hero.model.js, line 16](#)

Methods

(async) allDCHeroes() → {array}

Get all DC heroes in db

Source: [src/models/hero.model.js, line 167](#)

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) allMarvelHeroes() → {array}

Get all marvel heroes in db

Source: [src/models/hero.model.js, line 157](#)

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

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**) **get(idTeam)** → {object}

Get a hero by id

Parameters:

| Name | Type | Description |
|--------|--------|-------------|
| idTeam | number | |

Source: [src/models/hero.model.js, line 49](#)

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**) **getHeroById(id)** → {object}

Get hero by id from api

Parameters:

| Name | Type | Description |
|------|--------|-------------|
| id | number | |

Source: [src/models/hero.model.js, line 261](#)

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) getHeroByName(name) → {array}`

get heroes by name from api

Parameters:

| Name | Type | Description |
|------|--------|-------------|
| name | string | |

Source: [src/models/hero.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) getWinner(idHero, idHero2) → {object}`

Get the winner hero

Parameters:

| Name | Type | Description |
|---------|--------|-------------|
| idHero | number | |
| idHero2 | number | |

Source: [src/models/hero.model.js, line 69](#)

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) insertHero(entity) → {*}`

insert a new hero in db

Parameters:

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

Source: [src/models/hero.model.js, line 226](#)

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`) `modifyHero(idHero, entity)` → `{*}`

update hero info

Parameters:

| Name | Type | Description |
|--------|--------|--|
| idHero | number | |
| entity | object | body of the element that brings the path |

Source: [src/models/hero.model.js, line 250](#)

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`) `mostCombat()` → `{object}`

Get the hero with highest combat stats

Source: [src/models/hero.model.js, line 137](#)

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*) mostDurability() → {object}

Get the hero with most durability

Source: [src/models/hero.model.js, line 117](#)

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*) mostIntelligence() → {object}

Get the most intelligent hero

Source: [src/models/hero.model.js, line 87](#)

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*) mostPower() → {object}

Get the most powerful hero

Source: [src/models/hero.model.js, line 127](#)

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) mostSpeed() → {object}`

Get the fastest hero

Source: [src/models/hero.model.js, line 107](#)

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) mostStrength() → {object}`

Get the strongest hero

Source: [src/models/hero.model.js, line 97](#)

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) newHero(entity) → {*}`

create new hero

Parameters:

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

Source: [src/models/hero.model.js, line 238](#)

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) newHeros() → {array}

Get new heroes in db

Source: [src/models/hero.model.js, line 147](#)

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) profileImgHeroes() → {array}

Get select img from heroes for user profile

Source: [src/models/hero.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) searchAllHeroes()

search all heroes in api

Source: [src/models/hero.model.js, line 323](#)

(async) searchHeroByName(name) → {array}

get heroes by name from db

Parameters:

| Name | Type | Description |
|------|--------|-------------|
| name | string | |

Source: [src/models/hero.model.js, line 334](#)

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