

Global

Members

`_config`

to store the server conf

Source: [src/startup/index.js, line 7](#)

`(constant) config`

we bring the route to the configuration

Source: [src/startup/container.js, line 10](#)

`(constant) container`

we create our container where we define our elements We create our container where we define our elements and inject the different elements that make up our API to be able to work with them from where we want In the container we indicate if what we bring is a class, a function or a value and that it is a singleton type to make sure that there is only one instance of that element

Source: [src/startup/container.js, line 45](#)

`(constant) cors`

cors module

Source: [src/routes/index.js, line 10](#)

`(constant) DB`

we bring the route to the DB connexion

Source: [src/startup/container.js, line 22](#)

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`

(constant) `express`

express module

Source: [src/routes/index.js, line 5](#)

`swaggerUi`

swagger conf

Source: [src/routes/index.js, line 16](#)

Methods

`conexion() → {void}`

Function to start the server and conection with the DB

Source: [index.js, line 10](#)

Returns:

Type
void

`generateToken(user) → {object}`

function in which we return the token along with its duration

Parameters:

Name	Type	Description
user	object	the user with whom we make the token

Source: [src/helpers/jwt.helper.js, line 15](#)

Returns:

Authentication token encrypting the user object with our secret password

Type
object

`isAdmin(req) → {any}`

Function to check if you are an administrator

Parameters:

Name	Type	Description
req	object	Element with the user info

Source: [src/middlewares/adminAuth.middleware.js](#),
[line 6](#)

Returns:

res if the user is not admin else next

Type
any