

FICHERO MIDDLEWARE.JS → COMPROBACIÓN DE TOKEN Y SI TIENE ACCESO EL USUARIO DE 'ADMIN', 'STUDENT' O 'TEACHER'

```
/ AQUI ESTAN LAS FUNCIONES DE CONTROL DE ACCESO A CADA RUTA
const jwt = require('jsonwebtoken');
const dayjs = require('dayjs');
const userModel = require('../models/user');
const app = require('../app');
require('dotenv').config();
```

```
var payload; // es el user_id y user_name extraído del token
```

```
// el MIDDLEWARE CHECKTOKEN: se envía, se verifica con la frase y no ha caducado
const checkToken = async (req, res, next) => {
  console.log('----- COMPROBANDO TOKEN-----##### TOKEN ---
  ACCESO #####-----');
  console.log(process.env.CONTROL_RUTAS, process.env.VIDA_TOKEN);
  if (process.env.CONTROL_RUTAS) {
```

```
// 1 - Comprobamos si la cabecera existe
if (!req.headers['authorization']) {
  // Si la cabecera no existe la 'authorization', devolvemos el error
  console.log('-----NO EXISTE AUTHORITY-----##### TOKEN
  --- ACCESO #####-----');
  return res.status(401).send({ error: "«No autorizado» o «Se requiere
  autorización» -> Necesitas la cabecera de autorización" });
}
```

```
// 2 - Comprobamos si el token enviado es correcto
const token = req.headers['authorization'];
//console.log('EL ..... TOKEN ES:', token)
```

```
try {
  payload = jwt.verify(token, 'frase muy secreta');
  // console.log('EL PAYLOAD DESPUES DE VERIFICAR EL TOKEN ES:', payload)
} catch (err) {
  console.log('-----ERROR EN VERIFICACION DE TOKEN -----#####
  TOKEN --- ACCESO #####-----');
  return res.status(401).send({ error: "«No autorizado» o «Se requiere
  autorización» -> El token es incorrecto" });
}
```

```
// 3 - Comprobamos la fecha de caducidad
if (dayjs().unix() > payload.exp) {
  console.log('-----EL TOKEN HA EXPIRADO -----##### TOKEN ---
  ACCESO #####-----');
  return res.status(401).send({ error: "«No autorizado» o «Se requiere
  autorización» -> El token está caducado, pide otro" });
}
```

```
// req.user SIRVE PARA TODOS LOS MIDDLEWARE QUE SE ENCADENAN HARA LA RUTA
// USUARIO LOGEADO PARA TODA LA APLICACION req.user
req.user = await userModel.getUserById(payload.user_id);
console.log('...login del usuario: todos los datos del usuario', req.user[0]);
}
// Las comprobaciones son correctas y se puede acceder a la ruta on este token.
next();
}
```

```
const accesoTeachers = (req, res, next) => {
  console.log('-----CONTROL ACCESO SOLO A PROFESORES
  -----##### TOKEN --- ACCESO
```

```

#####-----');
console.log(process.env.CONTROL_RUTAS,process.env.VIDA_TOKEN);
if (process.env.CONTROL_RUTAS) {
//console.log('-----usuario-----',req.user[0][0].user_type)
if (req.user[0][0].user_type != "TEACHER") return res.status(401).send({ error:
"«No autorizado» o «Se requiere autorización» -> El usuario no tiene permiso en
esta ruta" });
// pasa la pelota a las rutas.
}
next();
}
const accesoAdministrator = (req, res, next) => {
console.log('-----CONTROL ACCESO SOLO A ADMINISTRADORES
##### TOKEN --- ACCESO
#####-----');
console.log(process.env.CONTROL_RUTAS,process.env.VIDA_TOKEN);
if (process.env.CONTROL_RUTAS) {
//console.log('-----usuario-----',req.user[0][0].user_type)
if (req.user[0][0].user_type != "ADMIN") return res.status(401).send({ error:
"«No autorizado» o «Se requiere autorización» -> El usuario no tiene permiso en
esta ruta" });
// pasa la pelota a las rutas.
}
next();
}
const accesoTeachersORAdministrator = (req, res, next) => {
console.log('-----CONTROL ACCESO SOLO A PROFESORES Y ADMINISTRADORES
##### TOKEN --- ACCESO
#####-----');
console.log(process.env.CONTROL_RUTAS,process.env.VIDA_TOKEN);
if (process.env.CONTROL_RUTAS) {
//console.log('-----usuario-----',req.user[0][0].user_type)
if (req.user[0][0].user_type != "ADMIN" || req.user[0][0].user_type != "TEACHER")
return res.status(401).send({ error: "«No autorizado» o «Se requiere
autorización» -> El usuario no tiene permiso en esta ruta" });
// pasa la pelota a las rutas.
}
next();
}
const accesoStudents = (req, res, next) => {
console.log('-----CONTROL ACCESO SOLO A ESTUDIANTES
##### TOKEN --- ACCESO
#####-----');
console.log(process.env.CONTROL_RUTAS,process.env.VIDA_TOKEN);
if (process.env.CONTROL_RUTAS) {
//console.log('-----usuario-----',req.user[0][0].user_type)
if (req.user[0][0].user_type != "STUDENT") return res.status(401).send({ error:
"«No autorizado» o «Se requiere autorización» -> El usuario no tiene permiso en
esta ruta" });
// pasa la pelota a las rutas.
}
next();
}

module.exports = {
checkToken,
accesoTeachers,
accesoAdministrator,
accesoStudents,
accesoTeachersORAdministrator
}

```