Admin token : eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2NWQxMjdmNmMyYjgwZjVjNDgzOGJkZGUiLCJpYXQiOjE3MDgyNjAwODIsImV4cCI6MTcwODI2MzY4Mn0.NdLlnxddTG4Hj3GcpIIzy1mdkVyYq6l12RRFi4szvJE

const jwt = require('jsonwebtoken');

const authenticate = (req, res, next)=>{

    const token = req.headers.authorization;

    if(!token){

        return res.status(401).json({ message: 'Unauthorized: Missing token'});

    }

    jwt.verify(token, 'your\_secret\_key', (err, decoded)=>{

        if(err){

            return res.status(401).json({ message: 'Unauthorized: Invalid token' });

        }

        req.user = decoded;

        next();

    });

};

module.exports = authenticate;

Exactement. Voici les étapes que vous devriez suivre :

1. Envoyez une requête POST à votre endpoint de connexion (par exemple, **/api/admins/login**) en fournissant les informations d'identification nécessaires (par exemple, email et mot de passe).
2. Si les informations d'identification sont correctes, votre endpoint de connexion renverra un jeton JWT.
3. Copiez ce jeton JWT et utilisez-le comme valeur de l'en-tête Authorization dans vos requêtes ultérieures.
4. Incluez le middleware authenticate.js dans vos routes sécurisées après le point d'authentification, comme vous l'avez déjà fait. Ce middleware vérifiera la validité du jeton JWT dans chaque requête en examinant l'en-tête Authorization.
5. Assurez-vous que le jeton JWT est bien inclus dans l'en-tête Authorization de chaque requête ultérieure pour lesquelles vous souhaitez vérifier l'authenticité de l'utilisateur.

En suivant ces étapes, vous pouvez utiliser le jeton JWT obtenu après une connexion réussie pour accéder aux routes sécurisées qui nécessitent une authentification.

Haut du formulaire