

auth.middleware.js

```
import createHttpError from 'http-errors'
import jwt from 'jsonwebtoken'
import config from '../config/index.js'
import AuthService from '../services/auth.service.js'

const AuthMiddleware = {}

AuthMiddleware.authenticate = async (req, res, next) => {
  try {
    let { accessToken, refreshToken } = req.cookies

    if (accessToken) {
      let decoded = jwt.verify(accessToken, config.secretKey)
      req.user = decoded
      next()
      return
    }

    if (!accessToken && refreshToken) {
      let decodedRef = jwt.verify(refreshToken, config.secretKey)
      let session = await AuthService.checkSession(decodedRef.sessionId)
      let { user, _id } = session

      req.user = { ...user, session: _id }

      let access = jwt.sign({ ...user, session: _id }, config.secretKey, {
        expiresIn: config.acctoken,
      })

      let refresh = jwt.sign({ session: _id }, config.secretKey, {
        expiresIn: config.reftoken,
      })
    }
  }
}
```

```

    res.cookie('accessToken', access, {
      maxAge: config.acctoken,
      httpOnly: true,
    })

    res.cookie('refreshToken', refresh, {
      maxAge: config.reftoken,
      httpOnly: true,
    })

    next()
    return
  }

  next(createHttpError(401, 'Authentication Failed!'))
} catch (error) {
  console.log({ error })
  next(createHttpError(401, 'Authentication Failed!'))
}
}

export default AuthMiddleware

```