|  |  |
| --- | --- |
| **BACKEND - CONTROLLER** | |
| Vraag | Kopieren |
| **ACCES CONTROLLER** | |
| Destructure NA post | const { accessCode } = req.body; |
| JWT | // Stel de payload samen  const payload = {  sub: sessionId, // ID van het session object  role: accCode.role, // Role van het accCode object  code: accCode.code, // Code van het accCode object  };  // Stel de opties samen  const options = {  expiresIn: '1d', // Token vervalt na 1 dag  };  // Genereer de token  const secretKey = process.env.JWT\_SECRET;  const token = jwt.sign(payload, secretKey, options); |
| Server Cookie | return res              .cookie("acces\_token", token, {                httpOnly: true,                maxAge: 24 \* 60 \* 60 \* 1000, // 1 dag in milliseconden              })              .json({ message: -"Logged in succesfully " }); |
| Response terugsturen | return res.status(401).json({ msg: 'Toegangscode niet geldig' }); |
| Cookie verwijderen | return res        .clearCookie("acces\_token") // Naam cookie        .status(200); |
| **TODOS CONTROLLER** | |
| Gegevens in variabele steken uit object | const reqCourseId = req.acces.courseId;        const reqSessionId = req.acces.sessionId; |
| Const terug sturen als response | return res.json( todos); |
| **BACKEND - MIDDLEWARES** | |
| **ACCESS MIDDLEWARE** | |
| Uit cookies halen | const todoToken = req.cookies?.todo; |
| JSON web Token verfireeren | const secretKey = process.env.JWT\_SECRET;      const decoded = jwt.verify(todoToken, secretKey);      req.session = decoded.sub;      req.accessCode = decoded.code;      req.role = decoded.role; |
|  | req.access = {  sessionId: decoded.sub, // De sub property uit het decoded object  accessCode: decoded.code, // De code property uit het decoded object  role: decoded.role, // De role property uit het decoded object  courseId: decoded.courseId, // De id property uit het course object  }; |
| Volende middleware/request reponse handler | next(); |
| Cookie foutmelding middleware => naam staat in controller | return res        .clearCookie("acces\_token")        .status(403)        .json({ msg: "Ongeldige token" });    } |
| **EINDE ACCES CONTROLLER** | |
| Checken van role uit request object | const role = req?.role;    if (role && role === "CREATER") next();    else {      return res.sendStatus(403);    } |
| Checken of iets overdue is | if (todo) {        const now = new Date();        req.isOverDue = todo.deadline ? now > todo.deadline : false;        next();      } else {        res.sendStatus(404);      } |
| **ROUTES** | |
| Nieuwe get route toevoegen met methode vanuit controller | router.get("/", categoriesController.getAll); |
| Nieuwe post route | const accessController = require("../controllers/access\_controller"); => controller toevoegen  router.post("/access", accessController.gainAccess); |
| Nieuwe get route met methode vanuit controller | router.get("/verify", accessController.verifyAccess); |
| Nieuwe get route met meerdere middleware en methode uit controller | router.get(    "/verifyCreater",    accesMiddleware,    isCreatorMiddleware,    accessController.verifyCreatorAccess  ); |
| Nieuwe get route met 1 middleware en 1 methode uit controller | router.get("/closeSession", accesMiddleware, accessController.removeAccess); |
| Alle komende routers middleware laten gebruiken | router.use(accesMiddleware); |
| **APP.JS** | |
| Routes importeren | const indexRouter = require("./routes/index");  const todosRouter = require("./routes/todos");  const categoriesRouter = require("./routes/categories"); |

StatusCode Ok Verzoek succesvol aangemaatk 200

Created

Verzoek verwerkt en nieuwe recourse gemaakt 201

No content

Verzoek succesvol verwerkt, geeft niets terug 204

Bad Request

Verzoek onjuiste/ontbrekende gegevens 400

Unauthorized

Authenticatie vereist en/of mislukt 401

Forbidden Client

geen toegang tot resource 403

Not found

Gevraagde resource bestaat niet 404

Internal server error

Server heeft onverwachte fout aangetroffen 500

Yarn

.ENV file bekijken voor connectionstring

Npx prisma migrate deploy

Npx prisma generate

Yarn seed

Yarn start