|  |  |
| --- | --- |
| TOPIC/TITLE  Using Passport.js to add Cookies and sessions | |
| Keywords/Questions  Npm install passport.js  Express session  Passport  Passport-local-mongoose  App.post  Secret route  App.post login  App.getlogin  Log out | Notes : |
|  |
|  |
|  |
| Easy way to checking if you ar eusing the good package |
| When you search your package to the npmjs.com there are rating jjust near to the results |
|  |
| const session = require('express-session');  const passport = require("passport");  const passportLocalMongoose = require("passport-local-mongoose"); |
| Don’t need to require passport-local |
|  |
| https://www.npmjs.com/package/express-session |
|  |
| Then you have to place the code just above mongoose.connect and bellow the other app.use |
| app.use(session({    secret**:** 'keyboard cat',    resave**:** false,    saveUninitialized**:** true,    cookie**:** { secure**:** true }  })) |
| app.use(express.static("public"));  app.set('view engine', 'ejs');  app.use(bodyParser.urlencoded({  extended: true  }));  app.use(session({  secret: 'Our little secret.',  resave: false,  saveUninitialized: false,  cookie: { secure: false }  }))  mongoose.connect("mongodb://localhost:27017/userDB",{useNewUrlParser:true,useUnifiedTopology: true}); |
|  |
| http://www.passportjs.org/docs/ |
| app.use(session({  secret: 'Our little secret.',  resave: false,  saveUninitialized: false,  cookie: { secure: false }  }))  app.use(passport.initialize());  app.use(passport.session());  mongoose.connect("mongodb://localhost:27017/userDB", |
|  |
| https://www.npmjs.com/package/passport-local-mongoose |
|  |
| To use a plugin you userSchema have to look like this (not the standard one)  const userSchema = new mongoose.Schema ({  email : String,  password : String  }); |
|  |
| userSchema.plugin(passportLocalMongoose); |
|  |
| Now right under you user Model |
| Passport locale mongoose allows us to simplify the code we have to use with passport.js |
| app.post("/register", function(req,res){  User.register({username: req.body.username}, req.body.password, function(err, user){  if(err){  console.log(err);  res.redirect("/register")  }else{  passport.authenticate("local")(req,res function(){  res.redirect("/secrets");  })  }  })  }); |
|  |
| app.get("/secrets", function(req, res){  if (req.isAuthenticated()){  res.render("secrets")  }else{  res.redirect("login")  }  }) |
| app.post("/login",function(req,res){  const user = new User({  username:req.body.username,  password:req.body.password  })  req.login(user, function(err){  if(err){  console.log(err);  }else{  passport.authenticate("local")(req,res,function(){  res.redirect("/secrets")  })  }  })  }); |
| app.get("/login", function(req,res){  res.render("login");  }); |
|  |
| app.get("/logout", function(req,res){  req.logout();  res.redirect('/');  }) |
|  |
| http://www.passportjs.org/docs/ | |