|  |  |
| --- | --- |
| TOPIC/TITLE  implement Oauth using passport | |
| Keywords/Questions  Create a account for Google developers console  Create Oauth client ID  Require our strategy | Notes : |
|  |
| http://www.passportjs.org/packages/ |
| Passport-google-oauth20 |
| http://www.passportjs.org/packages/passport-google-oauth20/ |
|  |
| https://console.developers.google.com/projectselector2/apis/dashboard?supportedpurview=project |
| Then click on project and newproject |
| Only name the project and let the rest as it is => create |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| You have to put them in your .env file |
| 596684776513-022ivia2dl9u4dmq417jt0k77ele6v2a.apps.googleusercontent.com |
| \_iXBMGIiVZ2GOjbaeZ8eG0G5 |
|  |
| **var** GoogleStrategy = **require**('passport-google-oauth20').Strategy; |
| passport.use(**new** GoogleStrategy({  clientID: GOOGLE\_CLIENT\_ID,  clientSecret: GOOGLE\_CLIENT\_SECRET,  callbackURL: "http://www.example.com/auth/google/callback"  },  **function**(accessToken, refreshToken, profile, cb) {  User.findOrCreate({ googleId: profile.id }, **function** (err, user) {  **return** cb(err, user);  });  }  ));  You have to put it bellow in your code |
| passport.serializeUser(User.serializeUser());  passport.deserializeUser(User.deserializeUser()); |
|  |
| An before all the route |
|  |
| passport.use(new GoogleStrategy({  clientID: process.env.CLIENT\_ID,  clientSecret: process.env.CLIENT\_SECRET,  callbackURL: "http://localhost:3000/auth/google/secrets"  },  function(accessToken, refreshToken, profile, cb) {  User.findOrCreate({ googleId: profile.id }, function (err, user) {  return cb(err, user);  });  } |
| One emore thing to add due to deprecation google+ service is |
| https://github.com/jaredhanson/passport-google-oauth2/pull/51 |
| passport.use(new GoogleStrategy({  clientID: process.env.CLIENT\_ID,  clientSecret: process.env.CLIENT\_SECRET,  callbackURL: "http://localhost:3000/auth/google/secrets",  userProfileURL: "https://www.googleapis.com/oauth2/v3/userinfo"  }, |
| User.finor create is pseudo code in the passport documentation but you can install a package npm install mongoose-findorcreate |
| https://www.npmjs.com/package/mongoose-findorcreate |
| You have to require it : const findOrCreate **=** require('mongoose-findorcreate')  Then add as a plugin in our schema  // const secret = process.env.SECRET;  userSchema.plugin(passportLocalMongoose);  userSchema.plugin(findOrCreate); |
|  |
|  | |