# GOOGLE-AUTH02-PASSPORT SETUP

* Dependencies :
* "scripts": {
* "start": "cross-env NODE\_ENV=production node app",
* "dev": "cross-env NODE\_ENV=development nodemon app"
* },
* "author": "Dougie",
* "license": "MIT",
* "dependencies": {
* "connect-mongo": "^4.6.0",
* "dotenv": "^16.0.1",
* "express": "^4.18.1",
* "express-handlebars": "^6.0.6",
* "express-session": "^1.17.3",
* "method-override": "^3.0.0",
* "moment": "^2.29.3",
* "mongoose": "^6.3.4",
* "morgan": "^1.10.0",
* "passport": "^0.6.0",
* "passport-google-oauth20": "^2.0.0"
* },
* "devDependencies": {
* "cross-env": "^7.0.3",
* "nodemon": "^2.0.16"
* }
* Go to google console
* Menu>>>API’s & Services>>> Credentials>>>Create Credentials>>0Auth Client ID>>callback URI: <http://localhost:3000/callback>
* Generate Client ID and Secret
* Copy and paste these into .env file
* Now app.js setup up like this
* const passport = require('passport')
* const session = require('express-session')
* // Sessions
* app.use(session({
* secret: 'secret',
* resave: false,
* saveUninitialized: false
* }))
* // Passport Middleware
* app.use(passport.initialize())
* app.use(passport.session())
* Now in passport.js (which is a config file we create) have the following code:
* const GoogleStrategy = require('passport-google-oauth20').Strategy
* const mongoose = require('mongoose')
* const User = require('../models/User')
* module.exports = function (passport) {
* passport.use(new GoogleStrategy({
* clientID: process.env.GOOGLE\_CLIENT\_ID,
* clientSecret: process.env.GOOGLE\_CLIENT\_SECRET,
* callbackURL: '/auth/google/callback'
* },
* async (accessToken, refreshToken, profile, done) => {
* console.log(profile)
* }))
* passport.serializeUser( (user, done) => {
* done(null, user.id)
* })
* passport.deserializeUser( (id, done) => {
* User.findById(id, (err, user) => done(err, user))
* })
* }
* Create a user model to accept google credentials from the user
* const mongoose = require('mongoose')
* const UserSchema = new mongoose.Schema({
* googleId: {
* type: String,
* required: true
* },
* displayName: {
* type: String,
* required: true
* },
* firstName: {
* type: String,
* required: true
* },
* lastName: {
* type: String,
* required: true
* },
* image: {
* type: String
* },
* createdAt: {
* type: Date,
* default: Date.now
* }
* })
* module.exports = mongoose.model('User', UserSchema)
* in Routes/Routes.js have this:
* // Google Auth
* app.get('/auth/google', AuthController.google)
* // Google Auth Callback
* app.get('/auth/google/callback', AuthController.googleCallback)
* Finally in AuthController have this:
* const passport = require('passport');
* // Auth with Google - Get /auth/google
* exports.google = passport.authenticate('google', { scope: ['profile']})
* // Google auth Callback - Get /auth/google/callback
* exports.googleCallback = passport.authenticate('google', { failureRedirect: '/'}), (req, res) => {
* res.redirect('/dashboard')
* }
* This will determine what will happen if the login request is successful and what should happen in it fails