

npm install password

npm install password-google-oauth20

Setup password.js

const passport = require('passport');

const GoogleStrategy = require('passport-google-oauth20').Strategy;

const keys = require('./keys');

const User = require('../models/user-model');

passport.serializeUser((user, done) => {

    /\*

        function này được auto gọi sau khi GoogleStrategy hoàn thành done() method

        - ở đây mình chỉ ra rằng chỉ lưu user.id vào trong Cookie Session

    \*/

    done(null, user.id);

});

passport.deserializeUser((id, done) => {

    /\*

        Được gọi sau khi middleware

        passport.authenticate('google', { scope: ['profile'] })

        hoàn thành

        cái param id ở đây là do mình set ở cookie đó (serialize process)

})

    \*/

    User.findById(id).then((user) => {

        done(null, user);

    });

});

passport.use(

    new GoogleStrategy({

        // options for google strategy

        clientID: keys.google.clientID,

        clientSecret: keys.google.clientSecret,

        callbackURL: '/auth/google/redirect' // Đồng thời nhớ specify redirectURL y chang cho bên Google API

    }, (accessToken, refreshToken, profile, done) => {

        // check if user already exists in our own db

        User.findOne({googleId: profile.id}).then((currentUser) => {

            if(currentUser){

                // already have this user

                console.log('user is: ', currentUser);

                done(null, currentUser);

            } else {

                // if not, create user in our db

                new User({

                    googleId: profile.id,

                    username: profile.displayName,

                    thumbnail: profile.\_json.image.url

                }).save().then((newUser) => {

                    console.log('created new user: ', newUser);

                    done(null, newUser);

                });

            }

        });

    })

);

Auth Router

const router = require('express').Router();

const passport = require('passport');

// auth login

router.get('/login', (req, res) => {

    res.render('login', { user: req.user });

});

// auth logout

router.get('/logout', (req, res) => {

    req.logout();

    res.redirect('/');

});

// auth with google+

router.get('/google', passport.authenticate('google', {

    scope: ['profile']

}));

// callback route for google to redirect to

// hand control to passport to use code to grab profile info

router.get('/google/redirect', passport.authenticate('google'), (req, res) => {

    // res.send(req.user);

    res.redirect('/profile');

});

module.exports = router;

Profile Router chỉ là route bình thường + check user loggedIn thôi

app.js

// set up session cookies

app.use(cookieSession({

    maxAge: 24 \* 60 \* 60 \* 1000,

    keys: [keys.session.cookieKey]

}));

// initialize passport

app.use(passport.initialize());

app.use(passport.session());

// connect to mongodb

mongoose.connect(keys.mongodb.dbURI, () => {

    console.log('connected to mongodb');

});

// set up routes

app.use('/auth', authRoutes);

app.use('/profile', profileRoutes);

// create home route

app.get('/', (req, res) => {

    res.render('home', { user: req.user });

});

app.listen(3000, () => {

    console.log('app now listening for requests on port 3000');

});