구글 로그인

Passport로 구글 로그인만들기

설치

Npm install -s passport-google-oauth

1. 구글 클라우드 플랫폼에 프로젝트를 새로 만든다
2. 만든 프로젝트의 OAuth동의화면을 설정한다.
3. 그 후 사용자 인증 정보를 만든다. 이름과 리다이렉션 주소 등
4. 그후 id와 비밀번호를 받는다

var GoogleStrategy = require('passport-google-oauth').OAuth2Strategy;

혹은 var은 생략가능. 모듈을 불러온다.

passport.use(new GoogleStrategy({

        clientID: 17619446972-kvujjm08pa5th004vk4jf8iuut82a89v.apps.googleusercontent.com,

        clientSecret: jyZWOZ8hmX-CT\_FtTK88x-12,

        callbackURL: "http://www.example.com/auth/google/callback"

      },

      function(accessToken, refreshToken, profile, done) {

           User.findOrCreate({ googleId: profile.id }, function (err, user) {

             return done(err, user);

           });

      }

    ));

클라이언트 아이디와 시크릿에 구글로부터 받은 아이디와 비밀번호를 입력한다.

Client callback url 도 마찬가지로 바꾼다.

다만 소스코드에 정보를 저장하는 것은 좋지 않으므로(깃허브 등의 버전관리에 올라가게 되므로)

별도의 파일에 구글클라우드플랫폼에서 받은 json파일의 내용을 저장한다.

그리고 해당 파일을 모듈로 불러와 사용하고 해당 파일은 ignore한다.

별도의 파일을 복사하여 빈 내용의 파일을 만들어 해당 파일을 버전관리한다.

app.get('/auth/google',

        passport.authenticate('google', {

            scope: ['https://www.googleapis.com/auth/plus.login']

        }));

리소스 서버에서 제시하는 형식으로 url을 자동으로 만들어주는 passport기능

현재는 로그인 기능만을 구현할 것이기 때문에 scope가 한 개임.

app.get('/auth/google/callback',

        passport.authenticate('google', {

            failureRedirect: '/auth/login'//로그인 실패시 직접 로그인

        }),

        function (req, res) {

            res.redirect('/');//성공시 홈으로

        });

인증 완료 후 임시비밀번호(authorization code)를 callback으로 받는다.

예제에서는 기존 사용자와 구글로그인 사용자의 이메일이 같은 경우 같은 사용자로 id를 공유시킨다.

하지만 기존의 scope에는 email이 포함되지 않으므로 추가해야한다.

scope: ['https://www.googleapis.com/auth/plus.login','email']

emils라는 배열에 value객체로 들어있다.

var googleCredentials = require('../config/google.json');

    passport.use(new GoogleStrategy({

        clientID: googleCredentials.web.client\_id,

        clientSecret: googleCredentials.web.client\_secret,

        callbackURL: googleCredentials.web.redirect\_uris[0]

    },

        function (accessToken, refreshToken, profile, done) {

            console.log("googlestartage",accessToken, refreshToken, profile);

            var email = profile.emails[0].value;//이메일을 프로파일로 부터 가져온다

            var user = db.get('users').find({email:email}).value();//유저정보에서 같은 이메일 값을 갖는 유저를 찾는다

            user.googoleId=profile.id;//해당 유저의 정보에 구글 아이디를 추가한다.

            db.get('users').find({id:user.id}).assign(user).write();//해당 정보를 덮어 씌운다

            done(null,user);//해당 유저정보를 넘긴다. 그 후 로그인된다.

        }

    ));

다음과 같이 코드를 수정하면 중복아이디에 대한 로그인이 가능해진다.

 if (user) {//유저가 있는 경우

                user.googoleId = profile.id;//해당 유저의 정보에 구글 아이디를 추가한다.

                db.get('users').find({ id: user.id }).assign(user).write();//해당 정보를 덮어 씌운다

            }

            else {//없는 경우 새로 유저를 등록한다.

                user = {

                    id:shortid.generate(),

                    email:email,

                    nickname:profile.displayName,

                    googoleId:profile.id

                }

                db.get('users').push(user).write();

            }

            done(null, user);//해당 유저정보를 넘긴다. 그 후 로그인된다.

유저정보가 없는 경우 구글아이디를 기반으로 아이디를 생성하여 로그인한다.