**Chapter 09. 패스포트로 사용자 인증하기**

* 페이스북 로그인
* 페이스북은 Open Standard Protocol 기반 OAuth 인증으로 계정 인증이 가능함.

1. /login 패스로(get) 로그인 화면 진입
2. 클라이언트 인증 요청 시 /auth/facebook 패스(post)로 요청
3. 페이스북 로그인 페이지 출력 🡪 페이스북 서버에서 인증 🡪 로그인 성공
4. 로그인 성공 시 내 웹 서버는 페이스북 서버로부터 결과를 알려주는 인증 토큰(Access Token)을 전달받는다.
5. 페이스북 서버에서 /auth/facebook/callback 패스로 인증 성공 여부와 함께 사용자 정보를 내 웹 서버로 전달하면, 패스포트 모듈은 이 정보를 DB에 저장한 후 클라이언트에 응답을 보내게 된다.

* 페이스북 developers page로 가서 앱 등록 후 App ID와 시크릿 코드를 확인한다.

545274919298556

e8b46f9d5aef0f2d099f5e01917e24e8

* [config/config.js] 내 facebook 객체 추가

facebook: {

clientID: '545274919298556',

clientSecret: 'e8b46f9d5aef0f2d099f5e01917e24e8',

callbackURL: '/auth/facebook/callback'

}

* [config/passport.js]

var facebook = require('./passport/facebook');

..

passport.use('facebook', facebook(app, passport));

추가

* [config/passport/facebook.js] 에서 스트래티지 정의

var FacebookStrategy = require('passport-facebook').Strategy;

var config = require('../config');

module.exports = function(app, passport){

return new FacebookStrategy({

clientID: config.facebook.clientID,

clientSecret: config.facebook.clientSecret,

callbackURL: config.facebook.callbackURL

}, function(accessToken, refreshToken, profile, done){

console.log('passport의 facebook 호출됨');

console.dir(profile);

var options = {

criteria: {'facebook.id': profile.id}

};

var database = app.get('database');

database.UserModel.load(options, function(err, user){

if(err) { return done(err); }

if(!user){

var user = new database.UserModel({

name: profile.displayName,

email: profile.emails[0].value,

provider: 'facebook',

facebook: profile.\_json

});

user.save(function(err){

if(err) { console.log(err); }

return done(err, user);

});

} else {

return done(err, user);

}

});

})

}

* [database/user\_schema.js] 수정

Provider, authToken, facebook 속성 추가하고 salt와 hashed\_password의 required를 삭제한다.

* [config/config.js]에서 컬렉션 명을 users6 로 수정한다.
* [routes/user\_passport.js] : 페이스북 라우팅 함수 추가

// 페이스북 인증 라우팅

router.route('/auth/facebook').get(passport.authenticate('facebook', {

scope: 'email'

}));

// 페이스북 인증 콜백 라우팅

router.route('/auth/facebook/callback').get(passport.authenticate('facebook', {

successRedirect: '/profile',

failureRedirect: '/'

}));

* [views/index.ejs] 에 페이스북 버튼 추가

<a href="/auth/facebook" class="btn btn-primary"><span class="fa fa-facebook"></span> 페이스북으로 로그인</a>

* [views/profile.ejs]

<http://147.46.109.80:9090/board/board-read.do?boardId=webqna&boardNo=150109243057582&command=READ&page=1&categoryId=-1>

수정한 오류 사항

var FacebookStrategy = require('passport-facebook').Strategy;

var config = require('../config');

module.exports = function(app, passport){

return new FacebookStrategy({

clientID: config.facebook.clientID,

clientSecret: config.facebook.clientSecret,

callbackURL: config.facebook.callbackURL,

profileFields: ['id', 'emails', 'name']

}, function(accessToken, refreshToken, profile, done){

console.log('passport의 facebook 호출됨');

console.dir(profile);

var options = {

criteria: {'facebook.id': profile.id}

};

var database = app.get('database');

database.UserModel.findOne({'facebook.id': profile.id}, function(err, user){

if(err) { return done(err); }

if(!user){

console.log('없던 계정이니 새로운 모델을 생성합니다.');

var user = new database.UserModel({

name: profile.displayName,

email: profile.emails[0].value,

provider: 'facebook',

authToken: accessToken,

facebook: profile.\_json

});

user.save(function(err){

if (err) console.log(err);

return done(err, user);

});

} else {

console.log('로그인 이력이 있는 사용자입니다.');

return done(err, user);

}

});

})

}