

# 2022 부트캠프 041

## - Google Dev Console 새 API생성 / OAuth2.0 ( Init )

### !Tips & Links

- ※ 강의 <https://www.udemy.com/course/the-complete-web-development-bootcamp/>
- ※ Passport <https://www.passportjs.org/> / Google Dev Console <https://console.cloud.google.com/apis/dashboard>
- ※ 구글 APIs <https://console.cloud.google.com/apis/library?project=striking-gadget-351806&supportedpurview=project>

<b>OAuth</b>	Open Authorization / Open Standard for token based authorization (Third Party Login > like login with Twitter, Google, Facebook, etc...)						
(passport-google-oauth20)	<ol style="list-style-type: none"> <li>1) <b>Granular Access Levels</b>: Determining whether certain data is required or not</li> <li>2) <b>Read/Read+Write access</b>: Can write data back to third party, POST~</li> <li>3) <b>Revoke Access</b>: user can de-authorize app's access on third party(any point)</li> </ol>						
	<table> <tr> <td>① Set up your app</td><td>④ User Grants Permission</td></tr> <tr> <td>② Redirect to Authenticate</td><td>⑤ Receive Authorization code</td></tr> <tr> <td>③ User Logs In</td><td>⑥ Exchange AuthCode for Access Token</td></tr> </table>	① Set up your app	④ User Grants Permission	② Redirect to Authenticate	⑤ Receive Authorization code	③ User Logs In	⑥ Exchange AuthCode for Access Token
① Set up your app	④ User Grants Permission						
② Redirect to Authenticate	⑤ Receive Authorization code						
③ User Logs In	⑥ Exchange AuthCode for Access Token						

<b>Passport Google OAuth2.0</b> : Strategy DOC <a href="https://www.passportjs.org/packages/passport-google-oauth20/">https://www.passportjs.org/packages/passport-google-oauth20/</a>		
Init	npm install passport-google-oauth20	
Google Dev Console  새 플젝 생성&설정	▷ 새 프로젝트, New Project	새 프로젝트, API명 설정
	▷ OAuth동의 화면, OAuth consent screen	앱 이름 / 개발자 메일
	▷ 범위, Scope // 개인정보제공 동의 범위 설정	기본: email / profile / openid
	▷ 사용자 인증 정보 Credentials	[ API키, 클라이언트ID, 서비스계정 ]
	▷ OAuth 클라이언트 ID, OAuth Client ID // form작성 애플리케이션 유형, type: web application 이름, name: "~~~" 승인된 자바스크립트 원본 URI: <a href="http://localhost:3000">http://localhost:3000</a> 승인된 리디렉션 URI: <a href="http://localhost:3000/auth/google/secrets">http://localhost:3000/auth/google/secrets</a>	클라이언트 ID 보안 비밀번호 (client secret) 생성됨 (보안!)
.env저장	CLIENT_ID=~~~~~ CLIENT_SECRET=~~~~	env에서 쓸 때 스펠링, 대소문자 구분 주의 require('dotenv').config();는 최상단~~!
Config Strategy  DOC 붙여넣기  (+.env데이터) process.env.	<pre>const GoogleStrategy = require('passport-google-oauth20').Strategy;  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" }), function(accessToken, refreshToken, profile, cb) {   console.log(profile); // 구글 로그인한 profile log   User.findOrCreate({ googleId: profile.id }, function (err, user) {     return cb(err, user);   }); });</pre>	

```
/* -----<lang: en>----- */
Delegate[deləˈɡeɪt]대리, delegate the task to someone // Granular[ˈɡræn.jə.lər]세분화, granular access level
seems sketchy and insecure. // business is sunsetting
developer can determine what kind of data do they need from third party, and request those accordingly.
```