[**Local HTTPS**](https://gist.github.com/prof3ssorSt3v3/edb2632a362b3731274cfab84e9973f9)

**Facebook(META) API**

1강 수업소개  
연동 로그인 = Federated Identity  
Facebook Javascript SDK  
2강 인증과정과 Access Token  
Owner와 Client간의 정보 교류는 부담스러운 일  
federated Identity 사용  
Resource Token을 Resource Server와 교환하면서 Owner를 확인 및 Resource 정보 제어  
3강 앱등록  
서비스의 ID 찾기  
Meta Developers 로그인 -> 앱 만들기  
4강 구현계획  
Meta developers 문서 -> Login 문서클릭 - >   
SDK(Software Develope Kit)  
페이스북과 사이트간의 핸들(API)를 제어할 수 있도록 만든 핸들  
로그인 시스템을 구축하기 위한 과정  
1. SDK load  
2. SPK init(initialization) - Client id  
3. Login  
4. Logout  
5. isLogined 로그인 확인  
6. API (Facebook)  
  
5강 SDK설정과 로그인 여부 확인  
Load the JS SDK asynchronously 주석의 코드 따온다  
window.fbAsyncInit 객체를 가져온다   
fbAsyncInit는 약속되어 있는 변수로 초기화가 끝난 다음 함수로 만들 것  
FB.init을 통해 초기화  
메소드 사용  
appId (중요)  
cookie  
xfbml  
version (중요)  
  
1. 클라이언트 아이디 가져오기  
2. version 입력하기 facebook api change log 사이트 확인  
3. SDK init   
주요 코드  
- FB.getLoginStatus   
- initialized.statusChangeCallback  
  
/\* 원본  
            FB.getLoginStatus(function(response) {   // Called after the JS SDK has been initialized.  
            statusChangeCallback(response);        // Returns the login status.  
            });   
\*/  
var callback = function(response) {  
    statusChangeCallback(response);  
}  
FB.getLoginStatus(callback)  
  
  
6강. 로그인 상태를 버튼에 반영  
(SDK) B. JavaScript SDK 로그인 대화 상자를 사용한 로그인  
호출 샘플  
FB.login(function(response) {  
  if (response.status === 'connected') {  
    // Logged into your webpage and Facebook.  
  } else {  
    // The person is not logged into your webpage or we are unable to tell.   
  }  
});  
  
  
  
7강 로그인 로그아웃 구현  
SDK)  
B. JavaScript SDK 로그인 대화 상자를 사용한 로그인  
자체 로그인 버튼을 사용하려는 경우 FB.login()을 호출하면 로그인 대화 상자를 호출할 수 있습니다.  
  
FB.login(function(response){  
  // handle the response   
});  
  
로그인 상태를 확인하는 함수를 스크립트 상단으로 올리면 웹페이지의 전체가 사용 가능  
input onclick의 login 콜백으로 로그인 체크 함수를 호출한다.  
로그인 상태에 따라 버튼의 모양이 달라진다.  
  
8강 API 사용  
그래프 API

**구글 API**

* [[OAuth] 구글 소셜 로그인 구현 (2) - javascript로 연동](https://carpet-part1.tistory.com/635)

**1. API인증,** [**클라이언트 라이브러리 로드**](https://developers.google.com/identity/gsi/web/guides/client-library?hl=ko)

[**2. 로그인 버튼 렌더링**](https://developers.google.com/identity/gsi/web/guides/display-button?hl=ko#button_rendering)

- 자바스크립트를 사용하여 로그인 버튼을 렌더링하고 JWT를 브라우저의 자바스크립트 콜백 핸들러에 반환합니다.

[**3. 사용자 인증 정보 응답 처리**](https://developers.google.com/identity/gsi/web/guides/handle-credential-responses-js-functions?hl=ko#handle_credential_response)

* data-callback 속성에서 자바스크립트 함수를 설정하여 반환된 사용자 인증 정보 응답을 처리할 수 있습니다.
* 사용 중인 언어의 [JWT 디코딩 라이브러리](https://jwt.io/introduction)를 사용하여 JWT 사용자 인증 정보를 검증하고 디코딩할 수 있습니다.
  + [JWT 한글 설명](https://gorokke.tistory.com/181#JWT(JSON_Web_Token)_%EC%9D%B4%EB%9E%80), [요약글](https://velog.io/@jaehyeon23/JWT-Json-Web-Token)
  + [인증서 : 자격증 : 면허증 : 신임장](https://developers.google.com/identity/gsi/web/reference/js-reference?hl=ko#credential)
* 자바스크립트 메소드
  + [JSON.parse()](https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse)
  + [Window.prompt()](https://developer.mozilla.org/ko/docs/Web/API/Window/prompt)
  + [String.prototype.replace()](https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/String/replace)
  + [데이터 속성 사용하기 - dataset](https://developer.mozilla.org/ko/docs/Learn/HTML/Howto/Use_data_attributes)
  + [atob()](https://developer.mozilla.org/ko/docs/Web/API/atob)
  + [String.prototype.replace()](https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/String/replace)
    - [정규식](https://developer.mozilla.org/ko/docs/Glossary/Regular_expression)
    - [플래그를 활용한 고급 탐색](https://developer.mozilla.org/ko/docs/Web/JavaScript/Guide/Regular_expressions#%ED%94%8C%EB%9E%98%EA%B7%B8%EB%A5%BC_%ED%99%9C%EC%9A%A9%ED%95%9C_%EA%B3%A0%EA%B8%89_%ED%83%90%EC%83%89)