**OAuth 2.0 참조하기**

**https://developers.daum.net/services/apis/docs/oauth2\_0/reference**

**목차**

* [용어 정의](https://developers.daum.net/services/apis/docs/oauth2_0/reference#%EC%9A%A9%EC%96%B4-%EC%A0%95%EC%9D%98)
* [클라이언트 등록](https://developers.daum.net/services/apis/docs/oauth2_0/reference#%ED%81%B4%EB%9D%BC%EC%9D%B4%EC%96%B8%ED%8A%B8-%EB%93%B1%EB%A1%9D)
* [웹 어플리케이션에서 OAuth 2.0 사용하기](https://developers.daum.net/services/apis/docs/oauth2_0/reference#%EC%9B%B9-%EC%96%B4%ED%94%8C%EB%A6%AC%EC%BC%80%EC%9D%B4%EC%85%98%EC%97%90%EC%84%9C-oauth-2-0-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0)
* [그 외 어플리케이션에서 OAuth 2.0 사용하기](https://developers.daum.net/services/apis/docs/oauth2_0/reference#%EA%B7%B8-%EC%99%B8-%EC%96%B4%ED%94%8C%EB%A6%AC%EC%BC%80%EC%9D%B4%EC%85%98%EC%97%90%EC%84%9C-oauth-2-0-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0)
* [OAuth 2.0의 에러코드](https://developers.daum.net/services/apis/docs/oauth2_0/reference#oauth-2-0%EC%9D%98-%EC%97%90%EB%9F%AC%EC%BD%94%EB%93%9C)

**용어 정의**

설명에 들어가기 앞서 일단 사용하는 용어에 대해 알아봅시다.

1. **인증서버(Authorization Server)** OAuth 1.0a에서는 API서버가 인증을 하는 역할도 같이 했지만 OAuth 2.0은 인증 역할만 하는 서버를 따로 둘 수 있습니다.
2. **API서버(API Server)** OAuth를 통해서 접근을 허용하는 웹 애플리케이션. OAuth 1.0a에서 서비스 프로바이더(Service Provider)
3. **클라이언트(Client)** OAuth를 사용해서 리소스 오너를 대신해 API 서버에 접근하는 웹사이트나 애플리케이션. OAuth 1.0a에서 컨슈머(Consumer)
4. **리소스 오너(Resource Owner)** 인증서버에 계정을 가지고 있는 개인. OAuth 1.0a에서 사용자(User)

**클라이언트 등록**

클라이언트 개발자는 API 서버의 데이터를 사용하는 어플리케이션이나 웹사이트를 제작하기 위해 앱을 등록하고, 클라이언트 아이디와 클라이언트 시크릿 키를 발급받아야 합니다.

**웹 어플리케이션에서 OAuth 2.0 사용하기**

Implicit Grant라고하며, Javascript 등을 이용해 클라이언트 브라우저등에서만 모든 처리가 이루어지는 요청에 활용할수 있습니다.

**1. 사용자를 Daum계정에 접근하도록 리다이렉트합니다.**

**URL**

https://apis.daum.net/oauth2/authorize

**Parameters**

| **요청변수** | **값** | **설명** |
| --- | --- | --- |
| client\_id | string(필수) | 등록시 Daum에서 발급한 client ID |
| redirect\_uri | string(필수) | 등록한 redirect\_uri |
| response\_type | string(필수): "token" 이라고 입력 | Implicit Grant 인증을 위함 |

**예제 URL**

https://apis.daum.net/oauth2/authorize?client\_id=123412341234&amp;redirect\_uri=http%3A%2F%2Ftadoli.net&amp;response\_type=token

**2. Daum에서 redirect\_uri로 AccessToken 을 발급합니다.**

Javascript 등을 이용해 클라이언트 브라우저등에서만 모든 처리가 이루어지는 요청에 활용할수 있습니다.

**예**

http://{redirect.uri}/daum\_exam.html#access\_token=4905ae1e956e3d2a76a83f681fdb565d306e27e6406303526a763adc405

**그 외 어플리케이션에서 OAuth 2.0 사용하기**

Authorization Code Grant라고 부르며, JSP/Servlet, PHP등과 같은 Server-side 프로그래밍으로 인증을 구현할 경우 사용하기 적합한 인증 방식입니다.

**1. 인증 요청 단계**

**URL**

https://apis.daum.net/oauth2/authorize

**Parameters**

| **요청변수** | **값** | **설명** |
| --- | --- | --- |
| client\_id | string(필수) | 등록시 Daum에서 발급한 client ID |
| redirect\_uri | string(필수) | 등록한 redirect\_uri |
| response\_type | string(필수): "code" 이라고 입력 | Authorization code 인증을 위함 |

**예제 URL**

https://apis.daum.net/oauth2/authorize?client\_id=123412341234&redirect\_uri=http%3A%2F%2Ftadoli.net&response\_type=code

**인증 확인 및 Authorization code 발급**

리소스 오너를 로그인을 할수있도록 유도 한 후 로그인에 성공하면, redirect\_uri에 code를 붙혀서 다시 Redirect 시킵니다.

**예**

http://{redirect.uri}/?code={authorization\_code}

**2. 인증 허가 및 Access Token 발급 요청 단계**

content-type이 application/x-www-form-urlencoded인 HTTP POST 요청을 합니다.

**URL**

https://apis.daum.net/oauth2/token

**Parameters**

| **요청변수** | **값** | **설명** |
| --- | --- | --- |
| client\_id | string(필수) | 등록시 Daum에서 발급한 client ID |
| client\_secret | string(필수) | 등록시 Daum에서 발급한 client secret |
| redirect\_uri | string(필수) | 등록한 redirect\_uri |
| code | string(필수) | 1번 단계에서 발급 받은 authorization code |
| grant\_type | string(필수) | "authorization\_code" 입력 |

**Access Token 발급**

Access Token이 발급됩니다.

{

"scopes" : ["user"],

"token\_type" : "Bearer",

"refresh\_token" : "cc6f5a70b411c851cbbef9bd445a0f7ea9a67c075c7915a8d5fa95237880d5f7e3a7b0b30f6f12a592455fb8233cdd1f2dff98b18a1017f05fd852b35dd503dd",

"access\_token" : "156aabd8e8b2e3617045fc1330e98952857030b60901ba1d2242af1c766da03f",

"expires\_in" : 3600

}

**3. Access Token을 갱신하기**

Authorization Code Grant로 생성한 access\_token은 생성시 함께 받은 refresh\_token을 이용해 갱신할 수 있습니다.

content-type이 application/x-www-form-urlencoded인 HTTP POST 요청을 합니다.

**URL**

https://apis.daum.net/oauth2/token

**Parameters**

| **요청변수** | **값** | **설명** |
| --- | --- | --- |
| client\_id | string(필수) | 등록시 Daum에서 발급한 client ID |
| client\_secret | string(필수) | 등록시 Daum에서 발급한 client secret |
| redirect\_uri | string(필수) | 등록한 redirect\_uri |
| refresh\_token | string(필수) | 2번 단계에서 발급 받은 refresh token |
| grant\_type | string(필수) | "refresh\_token" 입력 |

**Access Token 재발급**

Access Token이 재발급됩니다.

{

"scopes" : ["user"],

"token\_type" : "Bearer",

"access\_token" : "456aabd8e8b2e3617045fc1220f98952853030b60901ba4d2242af1c766dd000",

"expires\_in" : 86400

}

**OAuth 2.0의 에러코드**

{

error: "invalid\_redirect\_uri"

}

| **이름** | **설명** |
| --- | --- |
| invalid\_request | 올바른 요청이 아닙니다. |
| invalid\_grant | 올바른 인증 방법이 아닙니다. |
| unauthorized\_client | 올바른 클라이언트가 아닙니다. |
| unsupported\_grant\_type | 지원하지 않는 인증 방법입니다. |
| unsupported\_response\_type | 지원하지 않는 응답 방식입니다. |
| invalid\_scope | 올바른 요청영역이 아닙니다. |
| server\_error | 오픈 API 서비스 내부 시스템 에러. |
| temporary\_unavailable | 오픈 API 서비스 연결 실패. 서비스 시스템 과부하 또는 장애로 인한 서비스 연결 실패. |