# OAuth2 Dökümanı

1. **OAuth2 Protokolü**

OAuth 2.0 uygulamaların birbirleri arasında veri erişimine olanak sağlayan açık bir yetkilendirme protokolüdür. 2006 yılında 1.0 dan sonra geliştirilmiştir. OAuth2 kullanıcı doğrulama amaçlı değil uygulama yetkilendirme amaçlı bir protokoldür. Yani bir uygulamanın başka bir uygulama tarafından doğrulanmış kullanıcılarına izin vermesidir. Örneğin bir web uygulamasının facebook, google veya github ile giriş yapılması gibi. Günümüzde birçok sosyal medya uygulaması OAuth2 desteklemektedir. OAuth sunucu istemci arası bir kütüphane değildir. Sadece uygulamalar arası yetkilendirmeyi sağlayan bir protokoldür.

OAuth2 protokülü kullanan bir çok kütüphane geliştirilmiştir. Bu kütüphaneleri kullanarak;

1. Sunucu tarafında yetkilendirme uygulaması geliştirebiliriz. Bu uygulama sayesinde doğrulama ve yetkilendirme sunucusu oluşturabiliriz. Başka uygulamalarda geliştirdiğimiz bu uygulamayı kullanarak yetkilendirme yapabilirler. (Tıpkı google, facebook gibi)
2. OAuth2 protokolüne uygun client uygulamalar geliştirebiliriz.

## OAuth Protokolü Rolleri

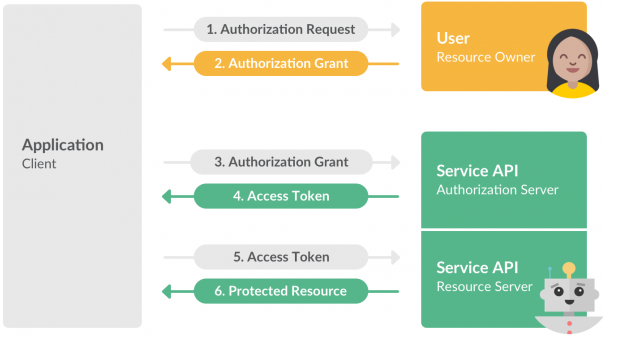
**Resource Owner**: Bir sistemde bulunan veriye sahip olan kişi veya uygulamadır. Örneğin Twitter uygulamasında hesabınıza girerek paylaştığınız tweet, resim veya videolar size ait birer veridir ve burada Resource Owner siz olursunuz.

**Client Application:** Resource Owner’ın sahip olduğu verilere erişen uygulamadır. Eğer kişisel blog sayfanızda size ait tweet’leri paylaşmak istiyorsanız blog sayfanızın Resource Server’a yani Twitter sunucularına erişmeniz gerekir. Bu durumda Client, sizin kişisel blog sayfanız olur.

**Authorization Server:** Client Application olarak belirlenen uygulamalara yetki veren sunuculardır. Size ait Tweet’leri kişisel blog sayfanıza çekmek istediğinizde Twitter sunucularının sizin blog sayfasını tanıması gerekmektedir. Bu durumda önce sayfanızın adres bilgilerini Twitter sunucularına kaydetmeniz gerekmektedir. Ardından sizin sayfanızdan Twitter sunucularına bir talep gittiğinde uygulamanız yetkilendirilecektir. Bu doğrulamayı ve yetkilendirmeyi yapan Twitter sunucuları OAuth 2.0 protokolündeki Application Server’dır.

**Resource Server:** Erişilmek istenen korunmuş verilerin bulunduğu yerdir. Örneğin kişisel blog sayfanızda yayınlamak üzere erişmek istediğiniz Tweet’leri sunan Api uçları (endpoints) Reource olarak adlandırılır.

## OAuth Akışı



OAuth akışının başlangıç noktası Resource Owner yani kaynak sahibidir. Reource Owner, yetkilendirme akışını Client Application üzerinden başlatır. Yetki verme (Authorizetion Grant) isteği uygulama üzerinden Authorization Server tarafına aktarılır. Kullanıcı doğrulandığında Client Application tarafına kaynağa erişmek amacıyla bir Access Token gönderilir. Artık Access Token ile Resource Server üzerindeki bilgilere erişmek mümkün hale gelmiştir.

## OAuth2 Authorization Grant Flow (Yetki Akışı )

Client Application bir Resource Server üzerindeki korunmuş bir veriye erişmek istediğinde ihtiyaç duyduğu yetkiyi Authorization Server üzerinden alır. Bu nedenle yetkilendirilecek olan her Client Application öncelikle Authorization Server üzerine kaydedilmelidir. Bu sayede hangi uygulamanın yetkilendirilecek uygulama olduğu Authorization Server tarafından bilinir ve belirlenen liste haricindeki uygulamalara geçit verilmez. Uygulamanın Authorization Server üzerine kaydı bir kez gerçekleştirilir ve kayıt silinene kadar uygulama sistemde geçerli kalır.

Bir uygulama Authorization Server üzerine kaydedildiğinde, uygulamaya özel bir Client ID ve Client Secret şeklinde benzersiz iki yetki belgesi verilir. Resource Server üzerinde korunmuş verilere erişmek isteyen her Client Application kendi Client ID ve Client Secret bilgisini Authorization Server üzerinden doğrular. Authorization Server üzerine kaydedilen Client Application için kaydedilen bir diğer parametre ise Redirect URI’dır. Bu parametre Resource Owner için başarılı bir şekilde yetkilendirme gerçekleştirildiğinde yönlendirme yapmak için kullanılır. Örneğin kullanıcı adı ve şifresini doğru giren bir kullanıcının ana sayfaya yönlendirilmesi bu şekilde gerçekleştirilebilir.

OAuth 2.0 teknolojisinde yetki verme işlemi Authorization Grant olarak adlandırılır.

#### Authorization Grant (Yetki Verme)

Client Application yetkilendirmesi sırasında Authorization Server ile Resource Server işbirliği içerisinde hareket eder.

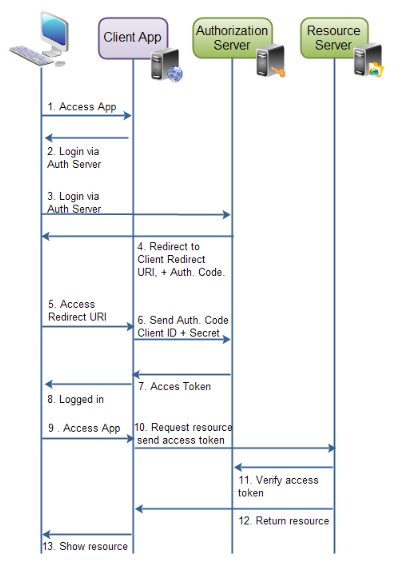
OAuth 2.0 protokolünde teknik olarak dört farklı yetkilendirme seçeneği bulunmaktadır.

* Authorization Code
* Implicit
* Resource Owner Password Credendials
* Client Credentials

Bir uygulamaya yetki verme işlemi sırasında birden fazla adımdan meydana gelen bir dolaşım söz konusudur. Bu dolaşım yada akış OAuth 2.0 teknolojisinde Flow olarak adlandırılır.Tüm akışlar (flow) her durumda kullanılabilir. Ancak her durum için kullanılması önerilen akışlar vardır. Uygulama türüne göre Web sitesi, mobil uygulama veya javascript uygulamaları için önerilen akışlar vardır. Örneğin native(mobil) ve javascript uygulamaları için implicit flow önerilir.

#### Authorization Code

Sunucu tabanlı çalışan uygulamalar için yetki verme işlemi Authorization Code tekniği ile gerçekleştirilir. Yani uygulama sunucusu (Client Application Server) ile veri kaynağının bulunduğu sunucu (Reource Server) arasında bir kimlik kontrolü gerçekleştirilir. Bu dolaşımda, Resource Owner tarafında kimlik bilgileri dolaşmaz. Authorization Code yetkilendirmesi Confidential Client türündeki uygulamalarda kullanılır.



1. Resource Owner(kullanıcı) korunmuş bir kaynağa erişmek üzere uygulamayı(Client Application) açar.
2. Uygulama kullanıcıya giriş yapması gerektiğini bildiren bir sayfaya yönlendirir. Bu sayfada giriş yapmak için Authorization Server seçenekleri bulunur. Örneğin Facebook, Gmail, Twitter gibi.
3. Bir seçeneği tercih eden kullanıcı giriş yapabilmesi için Authorization Server üzerinde bulunan login sayfasına yönlendirir. Örneğin Facebook, Gmail, Twitter giriş sayfası. Uygulama sunucusu arka planda Authorization Server tarafına Client ID bilgisini de gönderdiğinden dolayı Authorizaiton Server uygulamayı tanır.
4. Authorization Server sayfasından kullanıcı bilgileri girilerek login işlemi gerçekleştirilir. Örneğin Twitter ile login işlemi gerçekleştirilir. Login işlemi başarılı bir şekilde gerçekleştirildikten sonra uygulamanın Twitter hesabı ile yetki almasını kabul edip etmediği sorulur. Hatta profil bilgileri, resim bilgileri kullanılsın mı şeklinde işaretlenebilir kutucuklar halinde kullanıcıya seçenekler sunulur. Kullanıcı kabul ederse Client Application’a yönlendirilir.
5. Client Applicaiton sayfasına yönlendirildiğinde Authorization Server üzerine önceden kaydedilmiş olan Redirect URI adresine yönlendirilir. Yönlendirme ile birlikte Authorization Server tarafından yetkilendirmeyi temsil eden bir doğrulama kodu gönderir.
6. Redirect URI yönlendirmesi sonrasında Client Application doğrudan Authorization Server ile bağlantı kurar ve doğrulama kodu(authorization code) ile birlikte Client ID ve Client Secret bilgisini gönderir.
7. Eğer Authorization Server bilgilerin doğruluğunu kabul ederse geriye bir access token gönderir.
8. Kullanıcı login olduğunu görüntüleyen bir sayfaya yönlendirilir.
9. Korunmuş bir veri kaynağına erişim talebi yapılır.
10. CLient Applicaiton daha önce elde ettiği access token ile birlikte Resource Server üzerindeki kaynağı talep eder. Bu kez Resource Server access token doğrulamasını yapmak için Authorization Server ile bağlantı kurar. Token geçerliliği doğrulanırsa kaynaktaki veri uygulamaya gönderilir.

#### Implicit Grant

Implicit yetkilendirme Authorization Code yetkilendirmeye benzer adımlara sahiptir. Ancak Authorization Code ile yetkilendirme sırasında yetkilendirme kodu sunucu ile Authorization Server arasında dolaşırken Implicit yetkilendirmede yetkilendirmeyi temsil eden access token vardır ve bu access token User Agent(internet tarayıcı) ile Authorization Server arasında dolaşır. Yani yetkilendirme kodu olan access token açık bir ortamda dolaşmaktadır. Bu nedenle Imlicit yetkilendirmeler Public Client türündeki uygulamalarda kullanılır. Örneğin javascript uygulamaları gibi.

Client Application yetkilendirme sırasında sadece Client ID bilgisini Authorization Server’a gönderir. Client Secret bilgisini göndermez. Aksi taktirde bu bilgiler üçüncü şahısların eline geçebilir.

#### Resource Owner Password Credentials

Access token elde etmek için kullanıcı adı ve şifre gibi bilgilerinin doğrudan gönderilerek yetki alma işlemi sırasında kullanılır. Bu yöntem Resource Owner ile Client arasında çok güvenilir bir ortam olduğunda tercih edilmelidir. Aksi taktirde güvenlik riskleri çok yüksek olacaktır.

#### Client Credentials

Client Credentials yetkilendirme türü kullanılırken uygulama(Client Application) kendine ait kimlik bilgilerini (Client ID ve Client Secret) Authorization Server’a göndererek access token bilgisini elde eder.

Bu tür yetkilendirmeler kullanıcı bilgilerini gerektirmeyen makineler arası haberleşme(machine-to-machine) gibi uygulamalarda tercih edilebilir. Bir makinenin yerine getireceği görevleri bir API üzerinden okuması veya yazması buna örnek olabilir. Ya da kullanıcı iznini gerektirmeyen kaynak kullanımlarında tercih edilebilir.

Client Credentials yetkilendirmesi sadece Client ID ve Client Secret bilgilerini gerekli kılarken, Resource Owner Password Credentials yetkilendirmesi doğrudan kullanıcının şifresini istemektedir.

# OAuth2 Protocol Endpoints

OAuth 2.0 protokolünde tanımlanmış üç adet endpoint vardır. Bunlardan iki tanesi authorization server endpoint, bir tanesi ise client endpoint şeklindedir.

* Authorization Endpoint (Server)
* Token Endpoint (Server)
* Redirection Endpoint (Client)

Endpoint olarak adlandırılan kavram web server üzerinde tanımlanan bir URI adresidir. Protokole dair bağlantılar tanımlanan bu adresler üzerinden sağlanmaktadır.

#### Authorization Endpoint

Authorization Server üzerinde yetkilendirme amacıyla tanımlanan bir (URI) adrestir. Amacı kaynak sahibinden(Resource Owner) istemci uygulamanın(Client Applicaiton) kaynağa erişebilmesi için yetki alabilmesidir.

Authorization Server öncelikle kaynak sahibinin kimlik bilgilerini doğrular. Ancak kullanıcı kimlik doğrulama işlemi OAuth 2.0 protokolü kapsamının dışında bir işlemdir. Yani kullanıcılar Facebook, Twitter veya Google üzerindeki hesaplar olabileceği gibi şirketinize ait bir veri tabanında da bulunuyor olabilir. Bu durumda doğrulama yükü dış kaynaklı bir sürece dahildir.

#### Token Endpoint

Authorization Server üzerinde istemci uygulamalara(Client Applicaiton) access token sağlamak amacıyla tanımlanan (URI) adrestir. Uygulama eğer Confidential Client türünde ise doğrulama kodu (authorization code), Client ID, Client Secret gibi bilgiler karşılığında access token bilgisi elde edilir. Eğer Public Client türünde bir uygulama ise kullanıcı adı ve şifre bilgileri ile access token elde edilir.

Güvenli bir ortamda veri alışverişini sağlamak açısından Authorization Endpoint ve Token Endpoint için tanımlanan adresler için TLS(SSL) gerekliliği aranmalıdır.

#### Redirection Endpoint

Yetkilendirme işlemi gerçekleştiğinde, kaynak sahibinin (Resource Owner) yönlendirileceği bir aderstir (URI). Bu adres Client Applicaiton içerisinde tanımlanmaktadır.