# Uma introdução ao OAuth 2



## Introdução

O OAuth 2 é uma estrutura de autorização que permite que os aplicativos obtenham acesso limitado às contas de usuários em um serviço HTTP, tal como o Facebook, GitHub, e DigitalOcean. Ele funciona delegando a autenticação de usuário ao serviço que hospeda a conta do usuário, e autorizando aplicações de terceiros a acessar a conta do usuário. O OAuth 2 fornece fluxo de autorização para aplicações web e desktop, e para dispositivos móveis.

Este guia informativo é orientado para desenvolvedores de aplicativos, e fornece uma visão geral dos papéis OAuth 2, tipos de concessão de autorização, casos de uso, e fluxos.

Vamos começar com os papéis OAuth!

## Papéis OAuth

O OAuth define quatro papéis:

- Proprietário do Recurso
- Cliente
- Servidor de Recurso
- Servidor de Autorização

Vamos detalhar cada papel nas subseções a seguir.

## Proprietário do Recurso: *Usuário*

O proprietário do recurso é o usuário que autoriza uma aplicação a acessar sua conta. O acesso da aplicação à conta do usuário é limitado ao “escopo” da autorização concedida (por exemplo acesso para leitura ou escrita).

## Recurso / Servidor de Autorização: *API*

O servidor de recurso hospeda as contas de usuário protegidas, e o servidor de autorização verifica a identidade do usuário e então emite tokens de acesso para a *aplicação*.

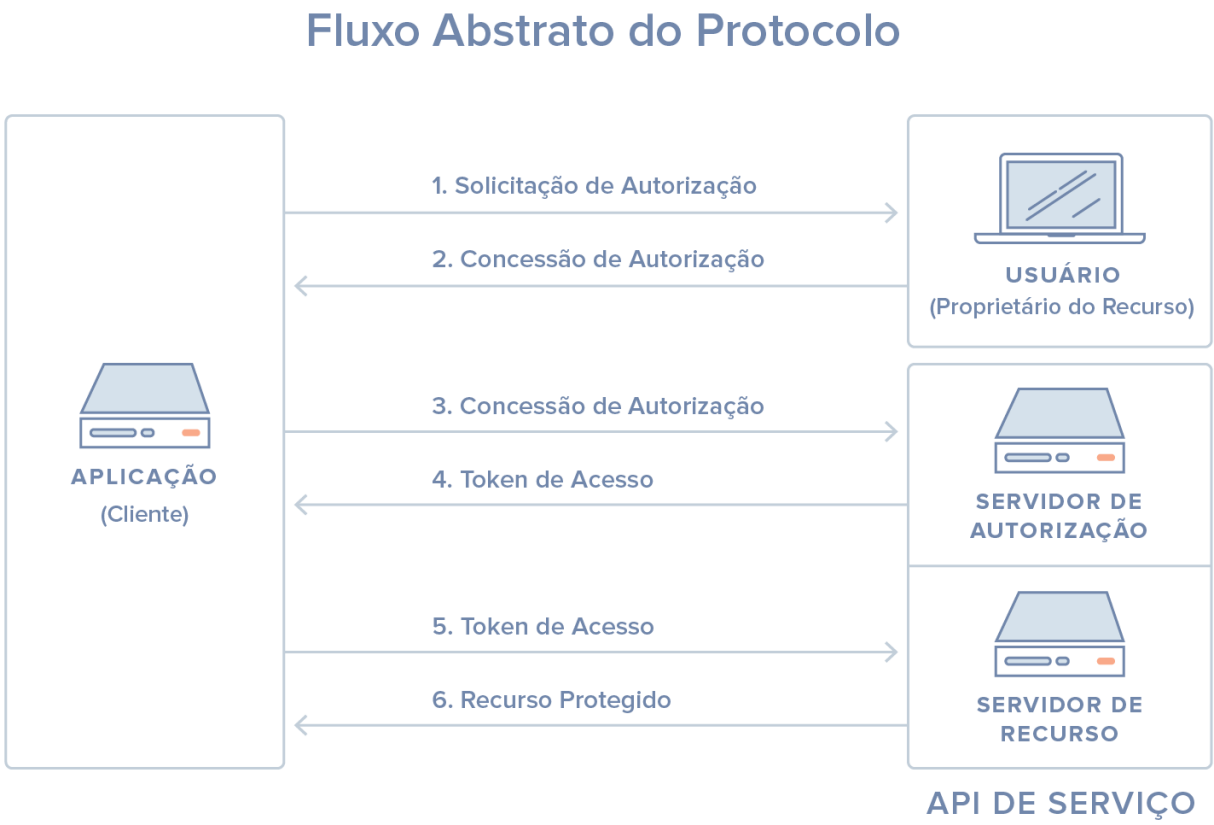
Pelo ponto de vista de um desenvolvedor de aplicações, a **API** de um serviço cumpre os papéis de servidor de recursos e de autorização. Vamos nos referir a ambos os papéis combinados, como papel de *Serviço* ou de *API*.

## Cliente: *Aplicação*

O cliente é a *aplicação* que quer acessar a conta do usuário. Antes de fazer isso, ela deve ser autorizada pelo usuário, e a autorização deve ser validada pela API.

## Fluxo Abstrato do Protocolo

Agora que você tem uma ideia do que são os papéis OAuth, vamos dar uma olhada em um diagrama de como eles geralmente interagem uns com os outros.




Aqui está uma explicação mais detalhada dos passos no diagrama:

1. A *aplicação* solicita autorização para acessar recursos do serviço do *usuário*
2. Se o *usuário* autorizar a solicitação, a *aplicação* recebe uma concessão de autorização
3. A *aplicação* solicita um token de acesso ao *servidor de autorização* (API) através da autenticação de sua própria identidade, e da concessão de autorização
4. Se a identidade da aplicação está autenticada e a concessão de autorização for válida, o *servidor de autorização* (API) emite um token de acesso para a aplicação. A autorização está completa.
5. A *aplicação* solicita o recurso ao servidor de recursos (API) e apresenta o token de acesso para autenticação

6. Se o token de acesso é válido, o *servidor de recurso* (API) fornece o recurso para a *aplicação*

O fluxo real desse processo será diferente dependendo do tipo de concessão de autorização em uso, mas essa é a ideia geral. Iremos explorar diferentes tipos de concessão em uma seção posterior.

## Registro da Aplicação

Antes de utilizar o OAuth com a sua aplicação, você deve registrar sua aplicação com o serviço. Isso é feito através de um formulário de registro na parte de "desenvolvedor" ou "API" do website do serviço, onde você irá fornecer as seguintes informações (e provavelmente detalhes sobre a sua aplicação):

- Nome da Aplicação
- Site da Aplicação
- URI de Redirecionamento ou URL de Retorno

A URI de redirecionamento é para onde o serviço irá redirecionar o usuário depois de autorizar (ou negar) a sua aplicação, e portanto, a parte do seu aplicativo que irá lidar com códigos de autorização ou tokens de acesso.

## ID do Cliente e Segredo do Cliente

Uma vez que sua aplicação esteja registrada, o serviço irá emitir "credenciais do cliente" na forma de *identificador do cliente* e *segredo do cliente*. O ID do Cliente é uma sequência exposta publicamente que é utilizada pelo serviço de API para identificar a aplicação, e também é utilizada para construir URLs de autorização que são apresentadas aos usuários. O Segredo do Cliente é utilizado para autenticar a identidade da aplicação para o serviço de API quando a aplicação solicita acesso à conta do usuário, e deve ser mantido em segredo entre a aplicação e a API.

## Concessão de Autorização

No *Fluxo Abstrato do Protocolo* acima, os primeiros quatro passos cobrem a obtenção de uma concessão de autorização e do token de acesso. O tipo da concessão de autorização depende do método usado pela aplicação para solicitar a autorização, e dos tipos de concessão suportados pela API. O OAuth 2 define quatro tipos de concessão, cada um dos quais é útil em diferentes casos:

- **Código de Autorização:** usado com aplicações do lado servidor
- **Implícito:** usado com Apps Móveis ou Aplicações Web (aplicações que rodam no dispositivo do usuário)

- **Credenciais de Senha do Proprietário do Recurso:** usado com aplicativos confiáveis, como os de propriedade do próprio serviço
- **Credenciais do Cliente:** usado no acesso de API das Aplicações

Agora vamos descrever os tipos de concessão em mais detalhes, seus casos de uso e fluxos, nas seções a seguir.

## Tipo de Concessão: Código de Autorização

O tipo de concessão **código de autorização** é o mais comumente usado porque ele é otimizado para *aplicações do lado servidor*, onde o código fonte não é publicamente exposto, e a confidencialidade do *Segredo do Cliente* pode ser mantida. Este é um fluxo baseado em redirecionamento, o que significa que a aplicação deve ser capaz de interagir com o *agente do usuário* (i.e. o navegador web do usuário) e receber códigos de autorização da API que são roteados através do agente do usuário.

Agora, descreveremos o fluxo do código de autorização:


## Passo 1: Link do Código de Autorização

Primeiro, o usuário recebe um link de código de autorização que se parece com o seguinte:

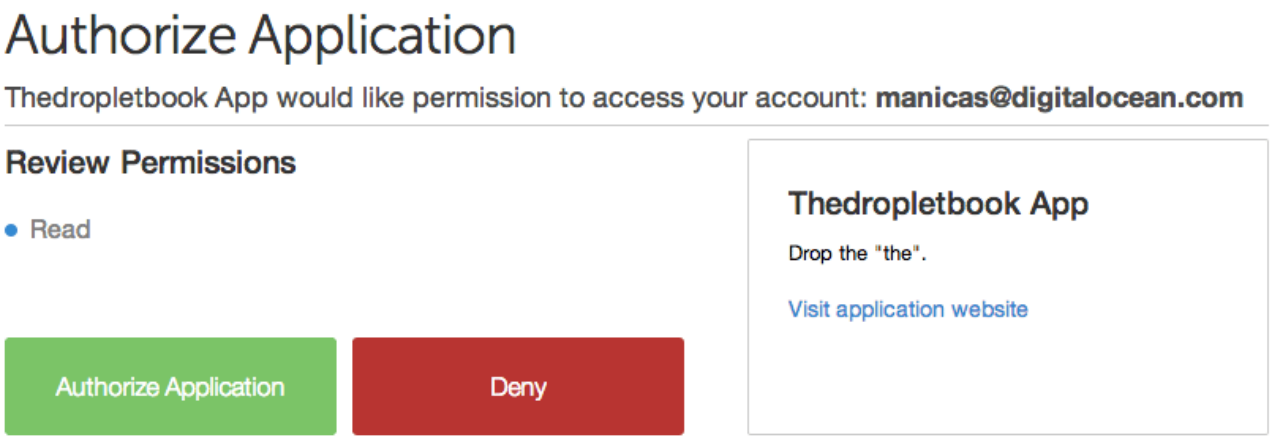
```
https://cloud.digitalocean.com/v1/oauth/authorize?
response_type=code&client_id=CLIENT_ID&redirect_uri=CALLBACK_URL&scope=read
```

Aqui está uma explicação sobre os componentes do link:

- **https://cloud.digitalocean.com/v1/oauth/authorize**: o endpoint de autorização da API
- **client_id=client_id**: o ID de Cliente da aplicação (como a API identifica a aplicação)
- **redirect_uri=CALLBACK_URL**: onde o serviço redireciona o agente do usuário depois que um código de autorização é concedido
- **response_type=code**: especifica que sua aplicação está solicitando um código de concessão de autorização
- **scope=read**: especifica o nível de acesso que a aplicação está solicitando

## Passo 2: O Usuário Autoriza a Aplicação

Quando o usuário clica no link, ele deve primeiro fazer login no serviço, para autenticar sua identidade (a menos que ele já esteja logado). Em seguida ele será solicitado pelo serviço a *autorizar* ou *negar* o acesso da aplicação à sua conta. Aqui está um exemplo de solicitação de autorização da aplicação:




Esta captura de tela em particular é a tela de autorização da DigitalOcean, e podemos ver que “Thedropletbook App” está solicitando autorização para acesso de leitura na conta de “manicas@digitalocean.com”.

## Passo 3: A Aplicação Recebe o Código de Autorização

Se o usuário clica em “Authorize Application”, o serviço redireciona o agente do usuário para a URI de redirecionamento da aplicação, que foi especificada durante o registro do ciente, juntamente com um *código de autorização*. O redirecionamento seria algo parecido com isto (assumindo que a aplicação é “dropletbook.com”):

```
https://dropletbook.com/callback?code=AUTHORIZATION_CODE
```

## Passo 4: A Aplicação Solicita o Token de Acesso

A aplicação solicita um token de acesso da API, passando o código de autorização juntamente com detalhes de autenticação, incluindo o *segredo do cliente*, para o endpoint de token da API. Aqui está um exemplo de solicitação POST ao endpoint de token da DigitalOcean:

```
https://cloud.digitalocean.com/v1/oauth/token?
client_id=CLIENT_ID&client_secret=CLIENT_SECRET&grant_type=authorization_code&
```

## Step 5: A Aplicação Recebe o Token de Acesso

Se a autorização é válida, a API irá enviar uma resposta contendo o token de acesso (e opcionalmente, um token de atualização) para a aplicação. A resposta inteira será parecida com isto:

```
{"access_token":"ACCESS_TOKEN","token_type":"bearer","expires_in":2592000,"ref
{"name":"Mark E. Mark","email":"mark@thefunkybunch.com"}}
```

Agora a aplicação está autorizada! Ela pode usar o token para acessar a conta do usuário através da API do serviço, limitada ao escopo de acesso, até que o token expire ou seja revogado. Se um token de atualização foi emitido, ele poderá ser utilizado para solicitar um novo token de acesso se o token original expirou.

## Tipo de Concessão: Implícito

O tipo de concessão **implícito** é usado para apps móveis e aplicações web (i.e. aplicações que rodam em um navegador web), onde a confidencialidade do *segredo do cliente* não é garantida. O tipo de concessão implícito é também um fluxo baseado em redirecionamento, mas o token de acesso é dado ao agente de usuário para encaminhar para a aplicação, por isso pode ser exposto ao usuário e a outros aplicativos no dispositivo do usuário. Adicionalmente, este fluxo não autentica a identidade da aplicação, e confia na URI de redirecionamento (que foi registrada com o serviço) para servir a esse propósito.

O tipo de concessão implícito não suporta tokens de atualização.

O fluxo de concessão implícito funciona basicamente da seguinte forma: o usuário é solicitado a autorizar a aplicação, a seguir o servidor de autorização repassa o token de acesso ao agente do usuário, que por sua vez o repassa à aplicação. Se você está curioso sobre os detalhes, continue lendo.

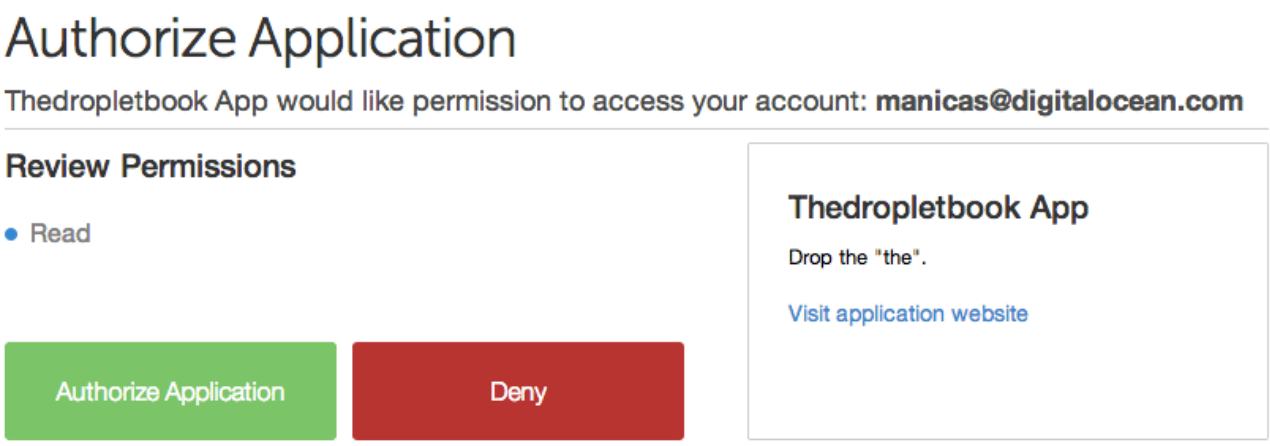

## Passo 1: Link de Autorização Implícita

Com o tipo de concessão implícito, o usuário recebe um link de autorização, que solicita um token da API. Este link se parece com o link de código de autorização, exceto por que ele está solicitando um *token* em vez de um código (observe o *tipo de resposta* "token"):

```
https://cloud.digitalocean.com/v1/oauth/authorize?
response_type=token&client_id=CLIENT_ID&redirect_uri=CALLBACK_URL&scope=read
```

## Passo 2: O Usuário Autoriza a Aplicação

Quando o usuário clica no link, ele deve primeiro fazer login no serviço, para autenticar sua identidade (a menos que ele já esteja logado). Em seguida ele será solicitado pelo serviço a *autorizar* ou *negar* o acesso da aplicação à sua conta. Aqui está um exemplo de solicitação de autorização da aplicação:




Você pode ver que “Thedropletbook App” está solicitando autorização para acesso de “leitura” na conta de “manicas@digitalocean.com”.

## Passo 3: O Agente do Usuário Recebe o Token de Acesso com a URI de Redirecionamento

Se o usuário clica em “Authorize Application”, o serviço redireciona o agente do usuário para a URI de redirecionamento da aplicação e inclui um fragmento de URI contendo o token de acesso. Seria algo assim:

```
https://dropletbook.com/callback#token=ACCESS_TOKEN
```

## Passo 4: O Agente do Usuário Segue a URI de Redirecionamento

O agente do usuário segue a URI de redirecionamento mas retém o token de acesso.

## Passo 5: A Aplicação Envia o Script de Extração do Token de Acesso

A aplicação retorna uma página web que contém um script que pode extrair o token de acesso da URI completa de redirecionamento que o agente do usuário havia retido.

## Passo 6: O Token de Acesso é repassado para a Aplicação

O agente do usuário executa o script fornecido e repassa o token de acesso extraído para a aplicação.

Agora a aplicação está autorizada! Ela pode usar o token para acessar a conta do usuário através da API do serviço, limitada ao escopo de acesso, até que o token expire ou seja revogado.

## Tipo de Concessão: Credenciais de Senha do Proprietário do Recurso

Com o tipo de concessão **credenciais de senha do proprietário do recurso**, o usuário fornece suas credenciais do serviço (nome de usuário e senha) diretamente para a aplicação, que utiliza as credenciais para obter um token de acesso do serviço. Este tipo de concessão deve ser ativado somente no servidor de autorização se os outros fluxos não forem viáveis. Além disso, ele só deve ser usado se o aplicativo for confiável para o usuário (por exemplo ele é de propriedade do serviço, ou do sistema operacional desktop do usuário).

## Fluxo de Credenciais de Senha

Após o usuário entregar suas credenciais para a aplicação, esta irá então solicitar um token de acesso ao servidor de autorização. A solicitação POST se pareceria com isto:

```
https://oauth.example.com/token?
grant_type=password&username=USERNAME&password=PASSWORD&client_id=CLIENT_ID
```

Se as credenciais do usuário forem verificadas, o servidor de autorização retorna um token de acesso para a aplicação. Agora a aplicação está autorizada!

**Nota**: A DigitalOcean atualmente não suporta o tipo de concessão de credenciais de senha, dessa forma o link aponta para um servidor de autorização imaginário em “oauth.example.com”.

## Tipo de Concessão: Credenciais do Cliente

O tipo de concessão **credenciais do cliente** fornece para a aplicação uma forma de acessar sua própria conta de serviço. Exemplos de quando isso pode ser útil inclui se uma aplicação quer atualizar sua descrição registrada ou URI de redirecionamento, ou acessar outros dados armazenados em sua conta de serviço via API.

## Fluxo de Credenciais do Cliente

A aplicação solicita um token de acesso enviando suas credenciais, seu ID do cliente e segredo do cliente, para o servidor de autorização. Um exemplo de solicitação POST poderia se parecer com o seguinte:

```
https://oauth.example.com/token?
grant_type=client_credentials&client_id=CLIENT_ID&client_secret=CLIENT_SECRET
```

Se as credenciais da aplicação forem verificadas, o servidor de autorização retorna um token de acesso para a aplicação. Agora a aplicação está autorizada a utilizar sua própria conta!

**Nota**: A DigitalOcean atualmente não suporta o tipo de concessão de credenciais do cliente, dessa forma o link aponta para um servidor de autorização imaginário em “oauth.example.com”.

## Exemplo de Uso do Token de Acesso

Uma vez que a aplicação tem um token de acesso, ela pode usar o token para acessar a conta do usuário via API, limitado ao escopo do acesso, até que o token expire ou seja revogado.

Aqui está um exemplo de uma solicitação de API, utilizando `curl`. Observe que ela inclui o token de acesso:

```
curl -X POST -H "Authorization: Bearer
ACCESS_TOKEN""https://api.digitalocean.com/v2/$OBJECT"
```

Assumindo que o token de acesso é válido, a API irá processar a solicitação de acordo com as suas especificações. Se o token de acesso estiver expirado ou mesmo inválido, a API irá retornar um erro de “solicitação inválida”.

## Fluxo do Token de Atualização

Depois que um token de acesso expira, utilizá-lo para realizar uma solicitação irá resultar em um erro “Token Inválido”. Nesse ponto, se um token de atualização foi incluído quando o token de acesso original foi emitido, ele pode ser usado para solicitar um token de acesso renovado do servidor de autorização.

Aqui está um exemplo de solicitação POST, utilizando um token de atualização para obter um novo token de acesso:

```
https://cloud.digitalocean.com/v1/oauth/token?
grant_type=refresh_token&client_id=CLIENT_ID&client_secret=CLIENT_SECRET&refre
```

## Conclusão

Isto conclui este guia OAuth 2. Agora você deve ter uma boa ideia de como o OAuth 2 funciona, e quando um fluxo de autorização em particular deve ser utilizado.

Se você quiser aprender mais sobre o OAuth 2, consulte esses valiosos recursos:

- How To Use OAuth Authentication with DigitalOcean as a User or Developer
- How To Use the DigitalOcean API v2
- The OAuth 2.0 Authorization Framwork