0Auth 2.0

OAuth é basicamente uma API de protocolo que permite aos usuários acesso limitado e específico a recursos de uma aplicação sem expor as credenciais, usando um token de acesso gerado por um servidor de autorização. Uma aplicação de terceiro pode ser utilizada para conceder acesso também.

No OAuth versão 2.0 existem basicamente quatro papéis:

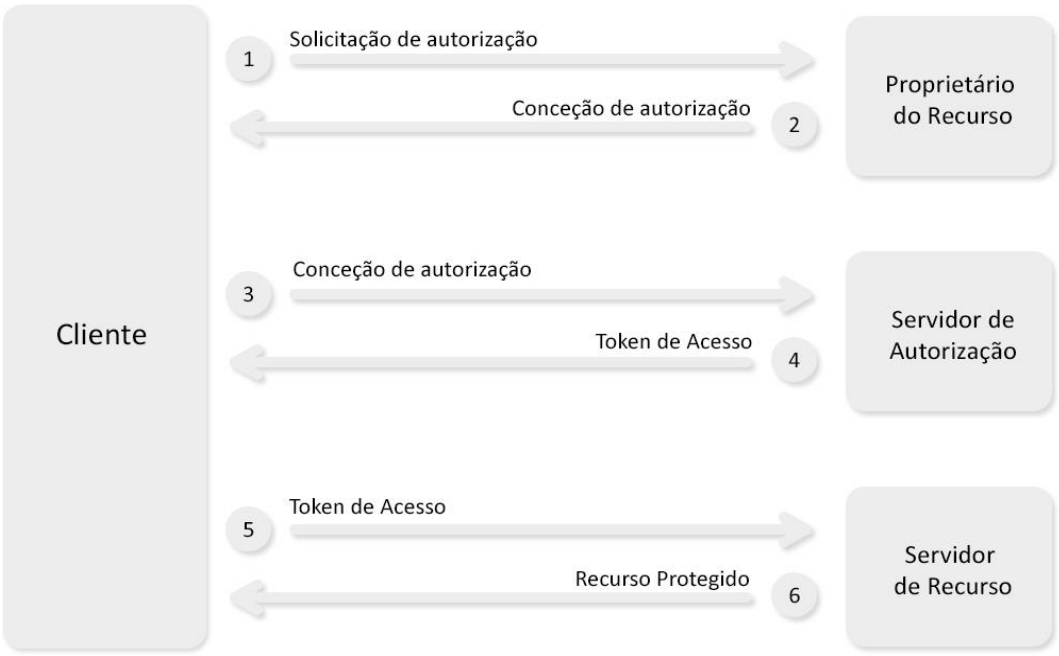
Resource Owner: Como o próprio nome diz, “Dono do recurso” é a entidade que controla os recursos que são protegidos. Basicamente o Usuário.

Resource Server: É o servidor que hospeda os recursos. Ele quem recebe as requisições.

Client: Entidade que solicita acesso aos recursos protegidos.

Authorization Server: É o servidor que gera tokens de acesso para o *Client.*

O Fluxo de informações entre as entidades acontece da seguinte forma:



O **Cliente** pede autorização ao **Proprietário do Recurso** para acessar os recursos do servidor do usuário;

Se for autorizada, O **Cliente** recebe uma Conceção de autorização informando que a autorização foi de fato concedida;

Em seguida o **Cliente** pede um Token de acesso ao **Servidor de autorização**;

Se o **Cliente** for autorizado e a Conceção de autorização valida, o **Servidor de autorização** gera um token de acesso e envia ao **Cliente**;

O **Cliente** em seguida pede acesso a um recurso que está protegido pelo **Servidor de recurso** e se identifica usando o token recebido pelo **Servidor de autenticação**.

E por fim, assumindo que o Token seja válido, o **Servidor de recurso** envia o recurso protegido ao cliente.

Existem quatro tipos de fluxo de informações:

Authorization Code: É o grant type mais utilizado e comum, é usado quando queremos obter determinados recursos de usuários por uma plataforma de terceiros. Ex.: Login com o google, Login com Facebook.

Implicit: É utilizado por aplicações de apenas uma página (SPAs).

Resource Owner Password Credentials: Utilizado quando o cliente solicita as credenciais de acesso diretamente, é utilizado em aplicações confiáveis (Trusted Apps).

Client Credentials: São utilizadas em integrações de sistemas, máquina para máquina.

O OAuth 2.0 é muito disperso pelo mundo atual estando presentes em muitas empresas pequenas e grandes, aqui algumas das mais conhecidas que já usam essa API:

- Amazon
- Apple
- Battle.net
- Discord
- Dropbox
- Facebook
- GitHub
- Google,
- Instagram
- Netflix
- Paypal

Finalizando, por melhor que seja essa API, se mal implementada podemos ter sérias falhas de segurança, sendo um exemplo o vazamento do Token de acesso, podendo autorizar um usuário que não é válido a acessar os recursos protegidos do Servidor de Recurso.