

Pesquise e escreva um texto sobre a API de autorização chamada OAuth.

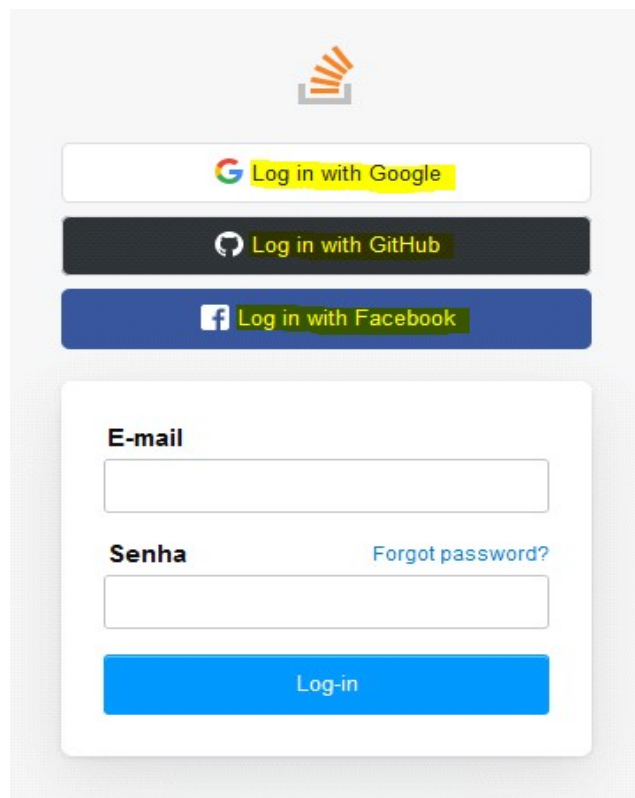
Resposta:

A API OAuth começou em 2006 devido ao Blaine Cook não conseguir identificar um padrão para acesso as APIs, OAuth 1.0 foi lançado oficialmente em 2007, ele é uma autorização que permite aos usuários acessar sites de maneira segura fazendo login com contas de outros sites, como exemplo, contas Google, Twitter, Facebook e etc. Basicamente o protocolo OAuth gera uma permissão para acesso aos recursos do servidor, tokens de acesso a terceiros com aprovação do servidor e proprietário dos recursos. Já o Protocolo OAuth 2.0 surgiu em 2012, ele serve para gerar acesso limitado a recursos sem expor suas credenciais.

1. Para que serve e para o que foi criado o protocolo OAuth?

Resposta:

É um protocolo de autorização que utiliza tokens para acessar aplicativos ou websites sem que você precise se cadastrar, basicamente você acessa com outra conta já logada, um exemplo é o próprio Stackoverflow, quando você acessa a página de login ele te dá opções de login com Google, GitHub e Facebook. Ele foi criado pois não existia um método de autenticação padronizado e universal e também para melhorar a segurança das autenticações.



The image shows a login interface for Stack Overflow. At the top, there is a logo consisting of four orange slanted lines. Below the logo, there are three buttons for social login: "Log in with Google" (white button with Google logo), "Log in with GitHub" (dark gray button with GitHub logo), and "Log in with Facebook" (blue button with Facebook logo). Below these buttons, there is a section for traditional login. It contains a label "E-mail" above a text input field, a label "Senha" above another text input field, and a link "Forgot password?" in blue text next to the password field. At the bottom of this section is a blue button labeled "Log-in".

2. Descreva o fluxo do protocolo OAuth na versão 2.0

Resposta:

O fluxo do protocolo OAuth 2.0 são 6 etapas, primeiro o client solicita o acesso as "funções", logo após temos a etapa de autorização do Resource Owner, se passar dessa etapa ele chega até o Authorization Server para solicitar o access token, se todas as etapas derem certo ele gera o access token e envia ao client, passando dessa parte ele solicita acesso a recursos protegidos ao Resource Server com o Access Token, caso o access token for válido o Resource Server libera acesso aos recursos.

"Quais os agentes envolvidos?"

Resource Owner: É o responsável por controlar o acesso aos recursos protegidos. Digamos que ele é o dono do recurso;

Resource Server: Digamos que é a própria "API", contém os dados dos usuários, hospeda os recursos que serão solicitados para acesso.

Client: É o responsável por solicitar o acesso aos recursos protegidos do Resource Owner;

Authorization Server: Responsável por gerar tokens de acesso, autoriza o Client a acessar os recursos permitidos pelo Resource Owner.

Qual o fluxo de informação entre estes agentes?

Primeiro o Client solicita autorização ao Resource Owner para acessar suas "funções";

Se o Resource Owner autorizar, o Client recebe um authorization grant. Isso é uma credencial que demonstra que a autorização foi cedida pelo Resource Owner;

Nessa etapa o Client solicita o access token ao Authorization Server, para isso ele envia o authorization grant que o Resource Owner "enviou";

Caso o Client for autorizado e o authorization grant for verdadeiro, o Authorization Server cria o access token e manda para o Client;

O Client solicita ao Resource Server um acesso a um recurso protegido, e realiza uma autenticação utilizando o access token gerado pelo Authorization Server;

Se o access token for válido, o Resource Server responde à solicitação do Client liberando o recurso solicitado.



3.Descreva como este serviço, se mal utilizado, pode trazer problemas de segurança para uma empresa.

Resposta:

Se o protocolo OAuth for mal utilizado pode ocorrer Ataques em Rede, captura de informações no processo de autorização, quando ocorre a solicitação do código de acesso para o Authorization Server se estiver alguém “atrás” da rede pode ter total controle da sessão associada ao Access Token, o envio de Urls maliciosas aos usuários quando solicitado o Access token, criando assim um método de “Phising”, isso causa diversos problemas de segurança nesse método de login se não for bem desenvolvido.

4.Cite pelo menos 10 serviços, de grandes empresas provedoras de autorização que utilizam, este protocolo.

Resposta:

Discord - OAuth 2.0

Facebook - OAuth 2.0

GitHub - OAuth 2.0

Instagram - OAuth 2.0

LinkedIn - OAuth 1.0, 2.0

Twitter - OAuth 1.0, 2.0

Amazon - OAuth 2.0

Dropbox - OAuth 1.0, 2.0

PayPal - OAuth 2.0

Reddit - OAuth 2.0

Twitch - OAuth 2.0

Spotify - OAuth 2.0