Ciências ULisboa

Aplicações Distribuídas 2021/22

Guião de apoio 9
Aplicações OAuth 2 em Flask

---

1. Introdução ao tema

Neste guião exploraremos o protocolo OAuth. Será implementada uma aplicação através da *framework* de desenvolvimento WEB *Flask* com autenticação e autorização OAuth.

2. Introdução ao *OAuth 2*

O protocolo OAuth utiliza HTTPS e permite a aplicações clientes o acesso a recursos protegidos de utilizadores, sem que seja necessário a obtenção das credenciais do utilizador dono dos recursos. No entanto, é necessário que obtenha a autorização prévia do dono dos recursos. É necessário haver uma terceira-parte, o servidor de autorização, que fará a ponte entre os dois intervenientes, nomeadamente:

1) aceitar o pedido da aplicação cliente para acesso aos recursos;
2) pedir a autorização ao dono do recurso, obtendo deste o códido de autorização (authorization code);
3) criar o token de acesso com base no código de autorização e a entrega deste à aplicação cliente.

A aplicação cliente acede aos recursos utilizando o token de acesso. Para que todo o processo de OAuth 2 funcione, é necessário que a aplicação seja registada numa API de OAuth (p. ex., Github, Spotify, Google, Facebook, Twitter) para a obtenção de um Client_ID e um Secret_ID e a indicação do URI para onde será redireccionado o código de autorização.

A listagem a seguir é um exemplo de como conectar uma aplicação cliente em OAuth, registada na API da GitHub. Exemplos semelhantes para outros servidores de autorização podem ser encontrados em http://requests-oauthlib.readthedocs.io/en/latest/index.html.

Listagem 1 – Exemplo com OAuth 2

```
from requests_oauthlib import OAuth2Session
import os
os.environ['OAUTHLIB_INSECURE_TRANSPORT'] = '1'

# Credenciais da app cliente registada no Github (https://github.com/settings/applications/new)
client_id = '<o id obtido da github>'
client_secret = '<o secret obtido da github>'

# URIs do github para obtencao do authorization_code, do token, de callback e do recurso protegido
authorization_base_url = 'https://github.com/login/oauth/authorize'
token_url = 'https://github.com/login/oauth/access_token'
redirect_uri= 'http://localhost'
protected_resource = 'https://api.github.com/user'

github = OAuth2Session(client_id, redirect_uri=redirect_uri)
# Pedido do authorization_code ao servidor de autorização (e dono do recuro a aceder)
authorization_url, state = github.authorization_url(authorization_base_url)
print ('Aceder ao link (via browser) para obter a autorizacao,', authorization_url)

# Obter o authorization_code do servidor vindo no URL de redireccionamento
url_response = input(' insira o URL devolvido no browser e cole aqui:')

# Obtencao do token
github.fetch_token(token_url, client_secret=client_secret, authorization_response=url_response)

# Acesso a um recurso protegido
r = github.get(protected_resource)
print (r.content.decode())
```

3. Exercícios

1. Registe no GitHub (https://github.com/settings/applications/new) uma aplicação para obtenção de um `client_id` e `secret_id`. No registo da aplicação redirecione esta para o `http://localhost`.

2. Copie o programa apresentado na Listagem 1, configure-o com as credenciais que obteve e execute-o. Apoiando-se na documentação sobre o módulo *requests_oauthlib*, deverá entender o papel de algumas funções disponíveis nas classes disponibilizadas pelo módulo (em destaque na listagem). Para além disso, deverá perceber o funcionamento do programa e identificar os passos do protocolo (indicados na introdução).

3. Registe no Spotify (https://developer.spotify.com/dashboard/applications) uma aplicação, caso ainda não o tenha feito para os projetos da disciplina. Nas configurações da aplicação, acrescente o `http://localhost` como URI de redirecionamento. Modifique os dados nas variáveis `client_id` e `client_secret` e os links no programa anterior para agora contactarem o servidor de autorização do Spotify (ao invés do GitHub):

```
authorization_base_url = 'https://accounts.spotify.com/authorize'
token_url = 'https://accounts.spotify.com/api/token'
redirect_uri= 'http://localhost'
protected_resource = 'https://api.spotify.com/v1/me'
```

4. Acrescente no programa anterior as linhas necessárias para convertê-lo para uma aplicação Flask (ver PL07). Para testar se está a funcionar, pode criar um método para o recurso raíz que retorna apenas um 'Hello world!'.

5. Crie um recurso (p. ex., `/login`) com o Flask para que o utilizador possa iniciar o processo de autorização com o OAuth 2 (ver TP09). Este recurso obterá a URL de autorização e redirecionará o utilizador para a mesma. O utilizador poderá então autorizar a aplicação a aceder ao recurso protegido.

6. Crie um recurso (p. ex., `/callback`) com o Flask para ser utilizado como endereço de redirecionamento do protocolo de autorização. Este recurso receberá, na URL do request, o código de autorização e estado necessários para pedir o token de acesso automaticamente. Note que a variável `redirect_uri` e a URI de redirecionamento configurada na página do Spotify precisam mudar para `https://localhost:5000/callback`. Após obter o token de acesso, este recurso deverá redirecionar o utilizador para um outro recurso (ver o próximo exercício) que acederá ao recurso protegido.

7. Crie um recurso (p. ex., `/profile`) com o Flask para que a aplicação cliente utilize o token de acesso obtido no exercício anterior para aceder ao recurso protegido no lugar do utilizador. Este recurso protegido pode ser por exemplo a página de perfil do utilizador no serviço escolhido (p. ex., Github ou Spotify).

8. Modifique a aplicação Flask para que ela e o cliente se comuniquem de TLS com autenticação mútua. Para tal, copie e utilize no Flask as chaves e os certificados gerados na PL08.

---

4. Bibliografia e outro material de apoio

- Flask User's guide: https://flask.palletsprojects.com/
- Flask API: https://flask.palletsprojects.com/en/2.1.x/api/

- Módulo requests:
  http://docs.python-requests.org/en/master/
- Protocolo OAuth
  https://oauth.net
- Módulo requests-oauthlib:
  https://pypi.python.org/pypi/requests-oauthlib
  http://requests-oauthlib.readthedocs.io/en/latest/index.html