OAuth2.0认证学习资料

一、什么是OAuth模型

OAuth（开放授权）一个开放标准

允许第三方网站在用户授权的前提下访问在用户在服务商那里存储的各种信息。

而这种授权无需将用户提供用户名和密码提供给该第三方网站。

OAuth允许用户提供一个令牌给第三方网站，一个令牌对应一个特定的第三方网站，同时该令牌只能在特定的时间内访问特定的资源。

理解：相当于网站登录时的第三方登录认证。

OAuth2.0协议的参与者

RO (resource owner): 资源所有者，对资源具有授权能力的人。

RS (resource server): 资源服务器，它存储资源，并处理对资源的访问请求。

Client: 第三方应用，它获得RO的授权后便可以去访问RO的资源。

AS (authorization server): 授权服务器，它认证RO的身份，为RO提供授权审批流程，并最终颁发授权令牌(Access Token)。

运行流程：

（A）用户打开客户端以后，客户端要求用户给予授权。

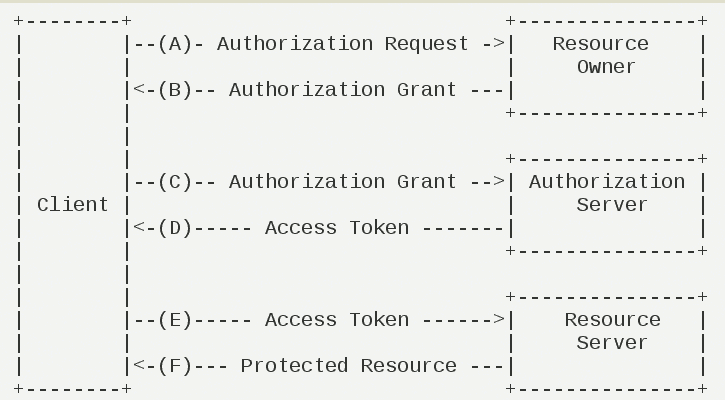
（B）用户同意给予客户端授权。

（C）客户端使用上一步获得的授权，向认证服务器申请令牌。

（D）认证服务器对客户端进行认证以后，确认无误，同意发放令牌。

（E）客户端使用令牌，向资源服务器申请获取资源。

（F）资源服务器确认令牌无误，同意向客户端开放资源。



五、客户端的授权模式

客户端必须得到用户的授权（authorization grant），才能获得令牌（access token）。OAuth 2.0定义了四种授权方式。

授权码模式（authorization code）

简化（隐式）模式（implicit）

密码（RO凭证）模式（resource owner password credentials）

客户端模式（client credentials）

授权码模式请求流程：

(1) Client请求RO的授权，请求中一般包含：要访问的资源路径，操作类型，Client的身份等信息。

(2) RO批准授权，并将“授权证据”发送给Client。至于RO如何批准，这个是协议之外的事情。典型的做法是，AS提供授权审批界面，让RO显式批准。这个可以参考下一节实例化分析中的描述。

(3) Client向AS请求“访问令牌(Access Token)”。此时，Client需向AS提供RO的“授权证据”，以及Client自己身份的凭证。

(4) AS验证通过后，向Client返回“访问令牌”。访问令牌也有多种类型，若为bearer类型，那么谁持有访问令牌，谁就能访问资源。

(5) Client携带“访问令牌”访问RS上的资源。在令牌的有效期内，Client可以多次携带令牌去访问资源。

(6) RS验证令牌的有效性，比如是否伪造、是否越权、是否过期，验证通过后，才能提供服务。

OAuth的作用就是让"客户端"安全可控地获取"用户"的授权，与"服务商提供商"进行互动。

开发者需要根据各自的应用场景，选择适用的OAuth2.0授权流程：

网站或者站外Web应用，请参考：Authorization Code流程；

桌面和无线客户端应用，请参考：Authorization Code流程，无server端的也可以使用Implicit Grant流程，无线客户端可以直接使用官方SDK，通过WebView方式使用授权页；

针对云平台的API授权，只是针对开发者，参考：百度OAuth2.0对外提供的开发者授权方式Developer Credentials。

以下介绍了三种应用场景方便开发者熟悉授权流程。

Server端使用百度OAuth2.0授权调用开放API流程

1. 引导用户到如下地址进行授权：

http://openapi.baidu.com/oauth/2.0/authorize?

response\_type=code&

client\_id=YOUR\_CLIENT\_ID&

redirect\_uri=YOUR\_REGISTERED\_REDIRECT\_URI&

scope=email&

display=popup

2. 如果用户同意授权,页面跳转至 YOUR\_REGISTERED\_REDIRECT\_URI/?code=CODE 。

3. 换取Access Token。

https://openapi.baidu.com/oauth/2.0/token?

grant\_type=authorization\_code&

code=CODE&

client\_id=YOUR\_CLIENT\_ID&

client\_secret=YOUR\_CLIENT\_SECRET&

redirect\_uri=YOUR\_REGISTERED\_REDIRECT\_URI

返回值

{

"access\_token": "1.a6b7dbd428f731035f771b8d15063f61.86400.1292922000-2346678-124328",

"expires\_in": 86400,

"refresh\_token": "2.385d55f8615fdfd9edb7c4b5ebdc3e39.604800.1293440400-2346678-124328",

"scope": "basic email",

"session\_key": "ANXxSNjwQDugf8615OnqeikRMu2bKaXCdlLxn",

"session\_secret": "248APxvxjCZ0VEC43EYrvxqaK4oZExMB",

}

4. 使用获得的OAuth2.0 Access Token调用API