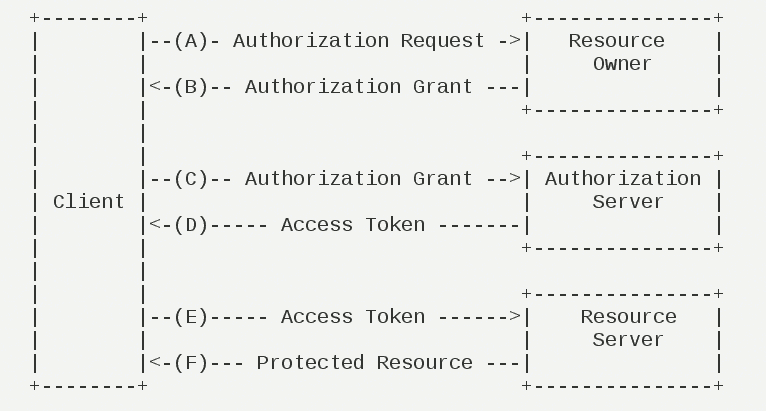
###### OAuth 2.0的运行流程如下图（摘自RFC 6749）：



图一

###### 名词解释：

（1） **Third-party application**：第三方应用程序，本文中又称"客户端"（client），图一左边的Client。

（2）**HTTP service**：HTTP服务提供商，本文中简称"服务提供商"。比如google。

（3）**Resource Owner**：资源所有者，本文中又称"用户"（user）。比如在google存储了图片的用户小明。

（4）**User Agent**：用户代理，本文中就是指浏览器。

（5）**Authorization server**：认证服务器，即服务提供商专门用来处理认证的服务器。

（6）**Resource server**：资源服务器，即服务提供商存放用户生成的资源的服务器。它与认证服务器，可以是同一台服务器，也可以是不同的服务器。比如google的图片存储服务器

###### 如何工作

让我们描述下这样一个场景，“某App 希望获得 Tom在 google上相关的数据”。

OAuth 2 在整个流程中有四种角色：

* 资源拥有者(Resource Owner) - 这里是 Tom
* 资源服务器(Resource Server) - 这里是google图片存储服务器
* 授权服务器(Authorization Server) - 这里当然还是google，因为google有相关数据
* 客户端(Client) - 这里是某App

参看图一，当 Tom 试图登录 google，打开了某App， 某App 将他重定向到 google的授权服务器，当 Tom 登录成功，并且许可自己的Email和个人信息被 某App 获取。这两个资源被定义成一个 Scope（权限范围） ，一旦准许， 某App 的开发者就可以申请访问权限范围中定义的这两个资源（Tom的Email和个人信息）。Tom 允许了权限请求，再次通过重定向返回 某App ，重定向返回时携带了一个 Access Token（访问令牌） ，接下来 某App 就可以通过这个 Access Token 从 google直接获取相关的授权资源（也就是Email和个人信息），而无需重新做 Tom 相关的鉴权。而且每当 Tom 登录了 某App ，都可以通过之前获得的 Access Token ，直接获取相关授权资源。