# 简介

*OAuth* 是针对访问授权的一个开放标准，它正通过许多实现（包括针对[Spring Security](http://www.oschina.net/p/spring+security)的一个实现）而不断获得动力。

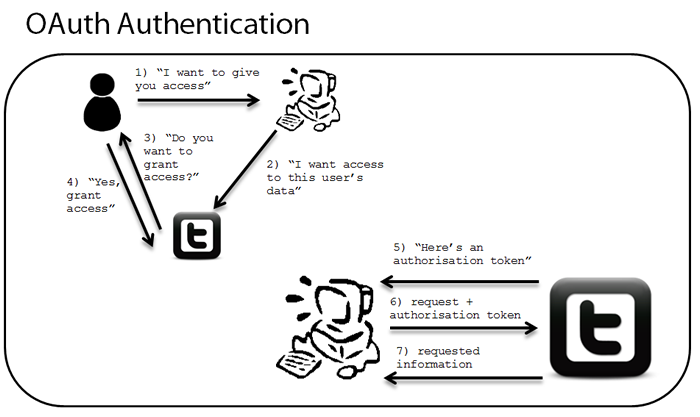
OAuth是由Blaine Cook、Chris Messina、Larry Halff 及David Recordon共同发起的，目的在于为API访问授权提供一个开放的标准。OAuth讨论组于2007年4月建立，以便向这个小组的实现者提供一种机构 来编写协议草案。Eran Hammer-Lahav与Google的DeWitt Clinton在开发过程中做出了巨大的贡献。规范的1.0版于2007年12月4日发布。

OAuth 协议为用户资源的授权提供了一个安全的、开放而又简易的标准。与以往的授权方式不同之处是OAUTH的授权不会使第三方触及到用户的帐号信息（如用户名与 密 码），即第三方无需使用用户的用户名与密码就可以申请获得该用户资源的授权，因此OAUTH是安全的。同时，任何第三方都可以使用OAUTH认证服务，任 何服务提供商都可以实现自身的OAUTH认证服务，因而OAUTH是开放的。业界提供了OAUTH的多种实现如 PHP，JavaScript，Java，Ruby等各种语言开发包，大大节约了程序员的时间，因而OAUTH是简易的。目前互联网很多服务如Open API，很多大头公司如Google，Yahoo，Microsoft等都提供了OAUTH认证服务，这些都足以说明OAUTH标准逐渐成为开放资源授权 的标准。

从高层次来看，OAuth按以下方式工作：

1. 你的站点已与不同的验证服务供应商建立了关系。
2. 你共享一个密码短语或者公钥，这样你就能使用它访问web联系人。
3. 你 根据验证服务供应商将用户重定向到登录页面。
4. 该用户登录然后告诉验证服务供应商你的站点访问其地址薄是没问题的。

# Twitter的Oauth授权过程



# Oauth的callback

还需要注意的是，因为OAuth是需要跳到主站的网页上去让用户授权，当用户授权完后，需要跳转回原网页，所以，一般来说，OAuth授权页都会带 一个 redirect\_url的参数，用于指定跳转回原来的网页。Github使用的这个跳转参数是redirect\_uri参数。一般来 说，redirect\_uri这个参数需要在服务器端进行验证。

你想一下，如果有人可以控制这个redirect\_uri这个参数，那么，你就可以让其跳转到别的网页上（可能会是个有恶意的网页）。如果你觉得跳 转到别的网页上也无所谓，那么你就错了。别忘了，当你对这个第三方的应用授权通过后，服务方会给第三方应用返回一个授权token，这个token会被加 到那个redirect\_uri参数后面然后跳转回去，如果这个redirect\_uri被别有用心的人改一个恶意的网址后，这个token也就被转过去 了，于是授权token也就被泄漏过去了。