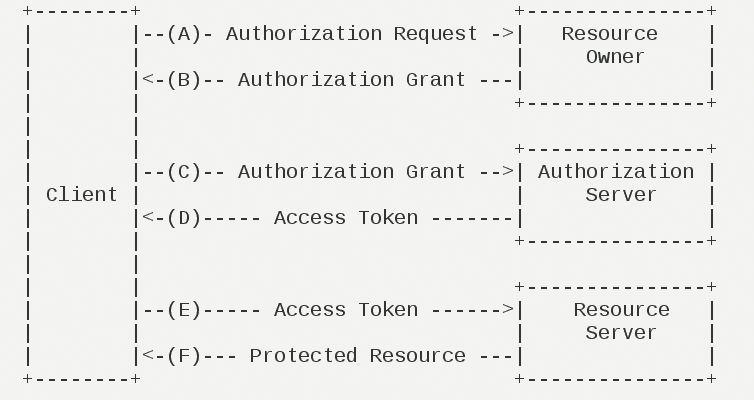
提到第三方登录，相信大家都已经耳熟能详了，比如什么QQ登陆、微信登陆、微博登陆等相信大家都用过，下图的摆在登录按钮下的一排第三方登录的按钮相信大家基本每次登陆的时候都能看见。



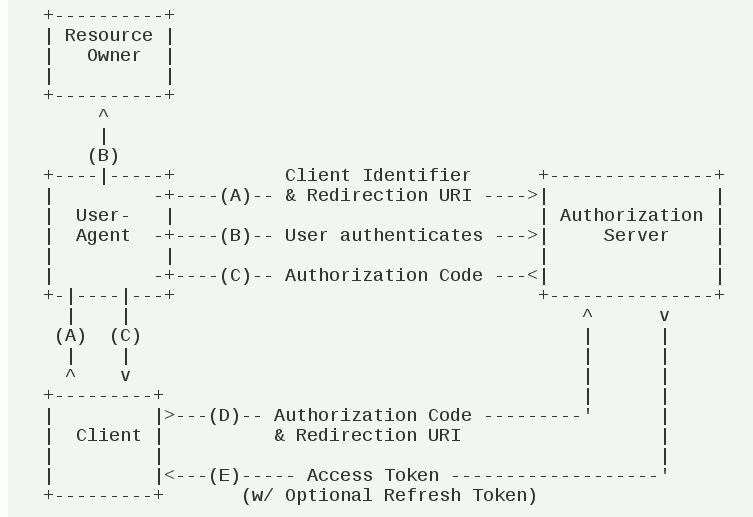
通过第三方登录，可以让我们非常迅速地登陆到我们想要进入的软件平台上，而省去了再输入一次用户名和密码的繁琐操作，甚至都省去了注册此平台用户的操作。而软件平台也是通过加入第三方登录的机制，来简化用户的登陆操作，并且可以利用大的第三方平台的用户资源，来提高对自己软件平台的推广，所以在这样的环境下，越来越多的软件开始支持一些大的第三方平台的登录了，那么你知道其中的原理吗？

下面就来简单讲讲第三方登录的原理。

说到原理，就不得不提到OAuth2.0了。OAuth2.0是什么？其实他就是一个协议，定义了一种用户验证和授权的标准，听到2.0，相信大家一定就会想到是不是还有个1.0的版本，确实，OAuth协议有1.0的版本，但是2.0的版本跟1.0的版本的区别很大，所以2.0并不兼容1.0，可以当做是两个不同的协议，现在的大部分的开放平台都用的是OAuth2.0的协议。这个协议就是第三方登录的基础，所以了解了OAuth2.0的授权机制，也就对第三方登录有了初步的了解，协议的运行流程如下图，下图摘自RFC 6749。



在OAuth2.0协议中，定义了多种途径获取访问令牌的方式，比如：授权码，客户端私有证书，资源拥有者密码证书，刷新令牌，断言证书等。在这里就只说第三方登录常用到的授权码模式。通过上图的流程可以看出，第三方登录总共需要三步，第一步：获取授权码code；第二步：通过授权码code获取令牌accessToken，获取到令牌其实就已经算是完成了授权；第三部：通过令牌获取openId。openId为此用户的唯一标示，只需将openId存入库中，下次登录就能通过openId来进行关联登录了。授权码模式的图解如下，摘自RFC6794。



看了原理大家是不是觉得很简单，其实当你了解到了原理之后会发现很多东西都没有那么的神秘。

上面讲述了认证的步骤，下面就为大家讲讲具体的流程。

首先你进入第三方开放平台对外开放的申请界面，进去之后，填写好你的网站的一些基本信息，以及回调地址，为什么要把回调地址单独拿出来说，因为他很重要，他是第三方网站授权完了之后要返回你的网站的时候用的地址。申请好了之后第三方平台会给你一个ID和KEY。

当你拿到这个ID和KEY后，相当于第三方平台已经对你的应用进行了一个授权。这时候你可以在第三方平台上找到对应的API或者是SDK，建议是直接用SDK，因为SDK一般都是封装好的，使用起来比较方便，当然为了平台的统一性，也可以使用API。

这时候你就可以执行上面的步骤了，首先就是获取code，你要通过ID和KEY对第三方平台给出的接口进行请求，一般还含有请求哪些权限，回调地址以及状态码等参数，状态码是用来防止受到CSRF攻击的，需要后端生成一个随机码。当你发出这个请求的时候就会达到第三方的授权登录页面，这个页面是第三方平台的，用于用户的登录以及对将要授予权限的确认，当用户确认并登陆后，第三方平台会直接通过回调地址将code给发过来，一般都是通过GET请求直接在地址后面加上参数，此code仅可使用一次，使用后就会作废。这样就完成了第一步的操作。

然后就是第二步，获取令牌accessToken。这里还是需要访问第三方平台给出的地址，此时的参数一般有：授权类型（这个授权类型上面说了是授权码模式，所以这个类型是authorization\_code），ID，KEY，code，回调地址。（这些参数根据不同的平台可能会有些许不同），然后第三方平台会返回的JSON中封装好令牌和过期时间，看到过期时间，大家应该就会知道令牌并不是永久有效的了，由于令牌的时效性，有些第三方平台给出了时效性更长甚至不会过期的刷新令牌refreshToken，通过上述获取令牌方式可以看到通过刷新令牌就能获取到令牌，这样能让你保证令牌一直不会过期，但是并不是所有的第三方平台都提供了refreshToken，比如现在微博就没有提供。在OAuth2.0的授权中，理论到了这一步也就算是授权完成了，即拿到令牌就算是完成了授权，而为了用户的登录，我们必须要存对用户有唯一的标识才行，刚才也说了，令牌是会过期的，过期了就会改变，所以它并不能作为唯一标识，有些平台在这一步会同时返回openId，那么就不用下一步再去获取了，当没有返回openId时，就需要我们的下一步了。

最后一步就是获取到用户的唯一编号openId了。这里直接调用第三方平台给出的地址，一般只需要传入accessToken参数就行，就能直接拿到openId了，然后将openId与自己平台的用户进行绑定，就可以实现第三方登录了。

上面讲到的是第三方登录的整个流程，在实际使用中，会更简单，比如假如你使用了第三方平台给出的SDK，这套SDK将上述的流程进行了封装，你只需要调用它所给出的几个接口就可以完成上述的所有操作，实现起来非常简单。

好了，说到这里相信你已经对第三方登录有了一定的了解了，并知道了该怎么去集成到你的系统中。想要更进一步，还是需要你自己动手去实现。

最后说下，第三方登录不仅仅是便于用户登录这么简单，OAuth2.0协议是一个授权协议，授权了取得OpenId只是最简单也是最基础的操作，在实际运用中，这个协议授权所获得到的令牌的权限远远不止这些，比如获取第三方平台的用户基本信息，调用第三方平台的一些API实现第三方平台的一些功能，如：分享微博、分享到朋友圈等等都是可以实现的，将这些第三方平台的功能整合到你的系统中，将会为你的用户带来更多地便利，也会让你的系统更加吸引用户。