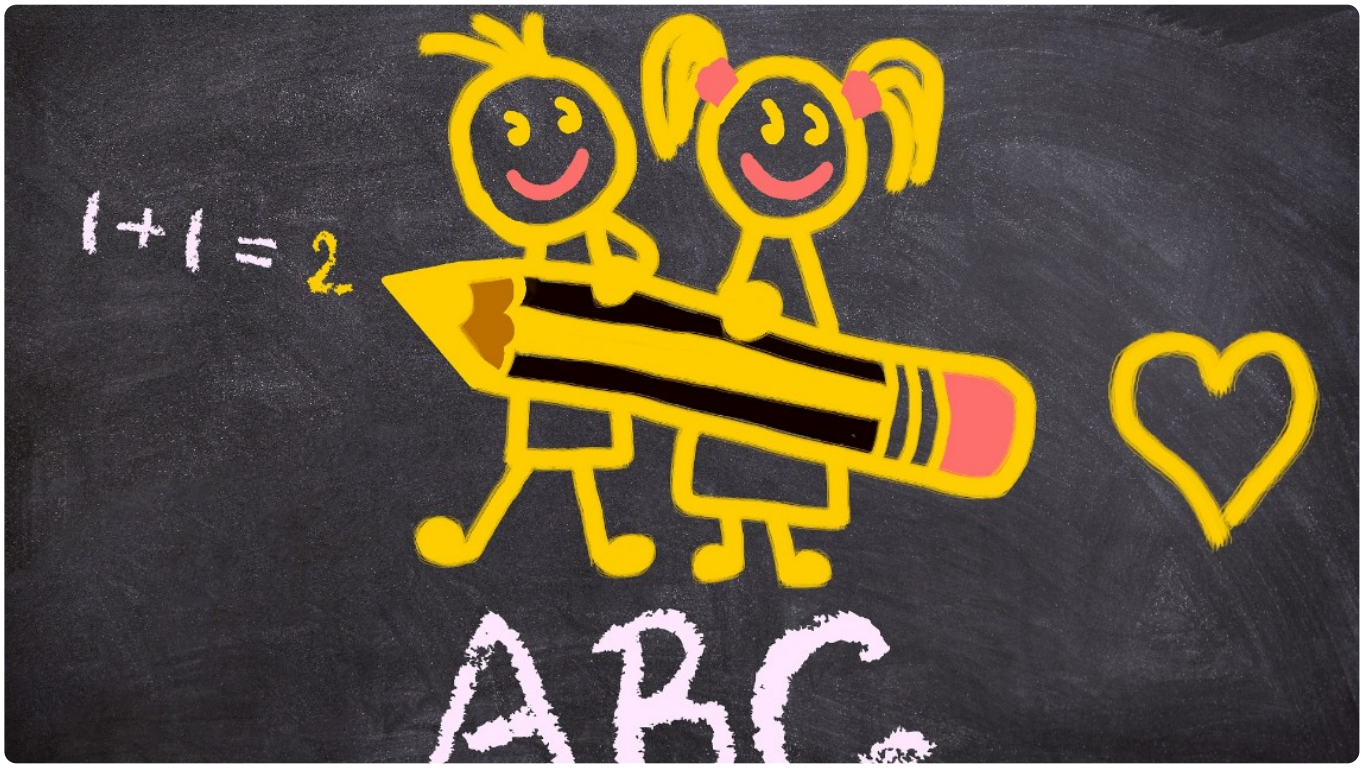
14 | 查漏补缺：OAuth 2.0 常见问题答疑

2020-07-30 王新栋

OAuth 2.0实战课 进入课程 >



讲述：李海明

时长 13:51 大小 12.69M

你好，我是王新栋。

从 6 月 29 日这门课上线，到现在已经过去一个多月了。我看到了很多同学的留言，有思考，也有提出的问题。那我首先，在这里要感谢你对咱们这门课的支持、鼓励和反馈。

在回复你们的留言时，我也把你们提出的问题记了下来。在梳理今天这期答疑的时候，我又从头到尾看了一遍这些问题，也进一步思考了每个问题背后的元认知，最后我归纳出了 6 个问题：

1. 发明 OAuth 的目的到底是什么？
2. OAuth 2.0 是身份认证协议吗？

3. 有了刷新令牌，是不是就可以让访问令牌一直有效了？
4. 使用了 HTTPS，是不是就能确保 JWT 格式令牌的数据安全？
5. ID 令牌和访问令牌之间有联系吗？
6. PKCE 协议到底解决的是什么问题？

接下来，我们就一一看看这些问题吧。

发明 OAuth 的目的到底是什么？

OAuth 协议的设计初衷，就是让最终用户也就是资源拥有者（小明），将他们在受保护资源服务器（京东商家开放平台）上的部分权限（查询当天订单）**委托**给第三方应用（小兔打单软件），使得第三方应用（小兔）能够代表最终用户（小明）执行操作（查询当天订单）。

这便是 OAuth 协议设计的目的。在 OAuth 协议中，通过为**每个第三方软件和每个用户的组合**分别生成对受保护资源具有**受限的访问权限的凭据，也就是访问令牌**，来代替之前的用户名和密码。**而生成访问令牌之前的登录操作，又是在用户跟平台之间进行的，第三方软件根本无从得知用户的任何信息。**

这样第三方软件的逻辑处理就大大简化了，它今后的动作就变成了请求访问令牌、使用访问令牌、访问受保护资源，同时在第三方软件调用大量 API 的时候，**不再传输用户名和密码，从而减少了网络安全的攻击面。**

从安全的角度来讲，**为每个第三方软件和每个用户的组合来生成一个访问令牌**的方式，可以减少对平台更多用户造成的危害。因为这样一来，单个第三方软件被攻破而带来的危害，仅仅会让这一个第三方软件的用户受到影响。

那么有的同学就要会问了，这样攻击的对象就会转移到授权服务身上。这个想法没错，但保护一个授权服务肯定要比保护成千上万个、由不同研发人员开发的第三方软件容易得多。

OAuth 2.0 是身份认证协议吗？

在这门课中，我其实一直在强调，**OAuth 2.0 是一种授权协议，“它一心只专注于干好授权这件事儿”，OAuth 2.0 不是身份认证协议。**但实际上，我在刚开始学习 OAuth 2.0 的时候，也曾错误地认为它是身份认证协议。

因为我当时觉得，有用户参与其中，比如小明在使用小兔打单软件之前，要向授权服务进行登录操作从而进行身份认证 ，那 OAuth 2.0 就应该是一个身份认证协议啊。

但是，小明必须登录之后才能进行授权，是一个额外的需求，登录跟授权体系是独立的。虽然登录操作看似“内嵌”在了 OAuth 2.0 的流程中，但生产环境中登录和授权还是两套独立存在的系统。所以说，**像这种“内嵌”的身份认证行为，并不是说 OAuth 2.0 自身承担起了身份认证协议的责任。**

同时，身份认证会告诉第三方软件当前的用户是谁，但实际上 OAuth 2.0 自始至终都没有向第三方软件透露过关于用户的任何信息。这一点，我们在讲发明 OAuth 协议的目的时也提到过。我们可以再想想小兔打单软件的例子，看是不是这样：小兔打单软件永远也不会知道小明的任何信息，它仅仅是请求访问令牌，使用访问令牌并最终调用查询订单的 API。

有了刷新令牌，是不是就可以让访问令牌一直有效了？

要回答这个问题，我们先复习下访问令牌和刷新令牌相关的几个知识点吧。

第一，OAuth 2.0 的核心是授权，授权的核心是令牌，也就是我们说的访问令牌。

第二，在第 3 讲中我们提到，为了提高用户的体验，OAuth 2.0 提供了刷新令牌的机制，使得访问令牌过期后，第三方软件在无需用户再次授权的情况下，可以重新请求一个访问令牌。

第三，在使用上，**刷新令牌只能用在授权服务上，而访问令牌只能用在受保护资源服务上。**

有了这些知识做基础，我们可以继续分析“有了刷新令牌，是不是就可以让访问令牌一直有效”这个问题了。

当访问令牌被 “递给” 受保护资源服务的时候，受保护资源服务需要对访问令牌进行验证，还要对访问令牌关联的权限和第三方软件的请求进行权限匹配校验。当访问令牌过期的时候，我们使用刷新令牌请求到的访问令牌，是授权服务重新生成的，而不是延长了原访问令牌的有效期。

当前的这个刷新令牌被使用之后，授权服务可以自行决定是颁发一个新的刷新令牌，还是仍然给第三方软件返回上一个刷新令牌。安全起见，我们的建议是**返回一个新的刷新令牌**。这时，你可能就有一个疑问了：第三方软件已经换了一个访问令牌了，刷新令牌又一直存在，那是不是就可以一直使用刷新令牌来获取访问令牌了呢？

要解决这个疑问，我们要知道的是，**刷新令牌也有有效期。**尽管生成了新的刷新令牌，但它的有效期不会改变，有效期的时间戳仍然是上一个刷新令牌的。刷新令牌的有效期到了，就不能再继续用它来申请新的访问令牌了。

使用了 HTTPS，是不是就能确保 JWT 格式令牌的数据安全？

OAuth 2.0 的使用从来都不应该脱离 HTTPS。因为访问令牌、应用密钥敏感信息要在网络上传输，都离不开 HTTPS 的保护。但是，HTTPS 也只是保证了访问令牌等重要信息在网络传输上的安全。

在 OAuth 2.0 的规范中，访问令牌对第三方软件是不透明的，从来都不应该被任何第三方软件解析到。由于 JWT 格式的令牌自包含了用户相关的信息，比如用户标识，因此仅仅对它进行签名还不够。要避免第三方软件有机会获取访问令牌所包含的信息，那我们在与第三方软件交互的环境下使用 JWT 格式的令牌时，还要对它进行加密来保障令牌的安全，而不是仅仅依靠 HTTPS。

ID 令牌和访问令牌之间有联系吗？

在第 9 讲中，我们在用 OAuth 2.0 实现一个 OpenID Connect 身份认证协议的时候，讲到了 ID 令牌。在这一讲的后面，有同学还是不太清楚 ID 令牌和访问令牌是啥关系，当时我就在留言区做了回复。现在，我重新整理了思路再和你解释一下，因为认识到 ID 令牌和访问令牌的联系与区别，对我们利用 OAuth 2.0 搭建一个身份认证协议来说太重要了。

我们先来总结下 ID 令牌和访问令牌的作用：

ID 令牌，也就是 ID_TOKEN，代表的是用户身份令牌，可以说是一个单独的身份认证结果，永远不会像访问令牌那样作为一个参数，去传递给其它外部服务；

访问令牌，也就是 ACCESS_TOKEN，就是一个令牌，是要被第三方软件用来作为凭证，从而代表用户去请求受保护资源服务的。

你看，这两种令牌是截然不同的。接下来，我们就分析下，它们的区别都体现在哪些方面吧。

第一，ID 令牌是对访问令牌的补充，而不是要替换访问令牌。之所以采用这样双令牌的方式，就是想让早先存在的访问令牌，可以在 OAuth 2.0 中继续保持对第三方软件的不透明性，而让后来新增的 ID 令牌要能够被解析，目的就是方便应用到身份认证协议中。

第二，ID 令牌和访问令牌有不同的生命周期，ID 令牌的生命周期相对来说更短些。因为 ID 令牌的作用就是代表一个单独的身份认证结果，它的使命就是用来标识用户的。而这个标识并不是用户名，用户登录的时候用的是用户名而不是这个 ID 令牌，所以如果用户注销或者退出了登录，ID 令牌的生命周期就随之结束了。

访问令牌可以在用户离开后的很长时间内，继续被第三方软件用来请求受保护资源服务。比如，小明使用了小兔打单软件的批量导出订单功能，如果耗时相对比较长，小明不必一直在场。

PKCE 协议到底解决的是什么问题？

我们在第 7 讲中学习 PKCE 协议时，我看到了大家对这个协议的很多留言，有的是自己的思考，有的是问题的进一步讨论。我们要理解 PKCE 协议到底解决了什么问题，就要先看一下它被推出的背景。

2012 年 10 月 OAuth 2.0 的正式授权协议框架，也就是官方的 RFC 6749 被正式发布，2015 年 9 月增补了 PKCE 协议，也就是官方的 RFC 7636。从时间上来看，从正式发布 OAuth 2.0 授权协议到增补发布了 PKCE 协议，整整间隔了三年，而这三年恰恰是移动应用蓬勃发展的时期。

同时，在原生的移动客户端应用保存秘钥又存在特殊的安全问题，使用 OAuth 2.0 授权码许可类型的客户端又容易受到授权码窃听的攻击。

所以，PKCE 被增补发布的背景是，移动应用大力发展，同时原生客户端使用 OAuth 2.0 面临着安全风险。这样我们就能理解了，发布 PKCE 协议的目的，主要就是缓解针对公开客户端的攻击，提高授权码使用的安全性。

总结

今天，我们专门用一节课来统一回答了 OAuth 2.0 的共性问题。我再来总结下你需要掌握的知识点：

1. OAuth 协议被发明的目的，就是用令牌代替用户名和密码。
2. OAuth 2.0 不能被直接用来“从事”身份认证协议的“工作”。虽然 OAuth2.0 的使用要求是在 HTTPS 的环境下，但这并不能解决 JWT 令牌对第三方软件“不透明”的问题，还需要进行加密。
3. 有了刷新令牌也不能让访问令牌一直有效下去，因为刷新令牌也有有效期。
4. ID 令牌是对访问令牌的补充，而不是要替代访问令牌。
5. PKCE 是 OAuth 2.0 的一个增补协议，主要用来缓解授权码被窃听的安全风险。

或许你在学习和实践 OAuth 2.0 时还会遇到其他问题，但不用担心，我们的留言区一直在，我也继续在留言区等着你，来回复你关心的、遇到的问题。

欢迎你在留言区分享你的观点，也欢迎你把今天的内容分享给其他使用 OAuth 2.0 的朋友，我们一起精进。

提建议

上一篇 13 | 各大开放平台是如何使用OAuth 2.0的？

下一篇 期末测试 | 一套习题，测试你的掌握程度

精选留言 (4)

写留言

Geek_bb8d16
2020-08-02

第三方登陆，比如微信登陆这个就很迷惑，这个是用OAuth2来实现登陆流程

作者回复: 微信的联合登录，比如极客时间用微信登录，这是利用了OAuth2.0的流程。

1

inrtyx
2020-07-31

能借问下吗？像雪花算法之类的，不一定是顺序插入吧？有可能先申请的反而后面才插入。这样的话插入性能鬼扯uuid好？

展开

一步

2020-07-30

ID 令牌要能够被解析 : 这句话要怎么理解的？ ID 要怎么进行解析呢？

作者回复: ID令牌 用作身份认证，什么是身份认证？就是“你是谁”，它包含了一个用户标识（注意不是用户名），所以要能够被解析。那访问令牌，不可以用作这个吗，不可以，咱们这篇答疑和之前的文章也都有提到，在用户“离开”后，第三方软件仍然可以使用访问令牌获取用户的信息，甚至访问令牌过期之后，还可以使用刷新令牌再换一个访问令牌。如果将访问令牌用作用户登录标识显然不合理。

陶陶

2020-07-30

没理解OIDC的使用场景是什么？因为就算没有id令牌，直接使用访问令牌也是可以通过请求用户信息接口获取当前登录人信息的

展开

6