前端开发核心知识进阶：50 讲从夯实基础到突破瓶颈
来自 Lucas ... · 盐选专栏

查看详情 >

不可忽视的前端安全 - 单页应用鉴权设计

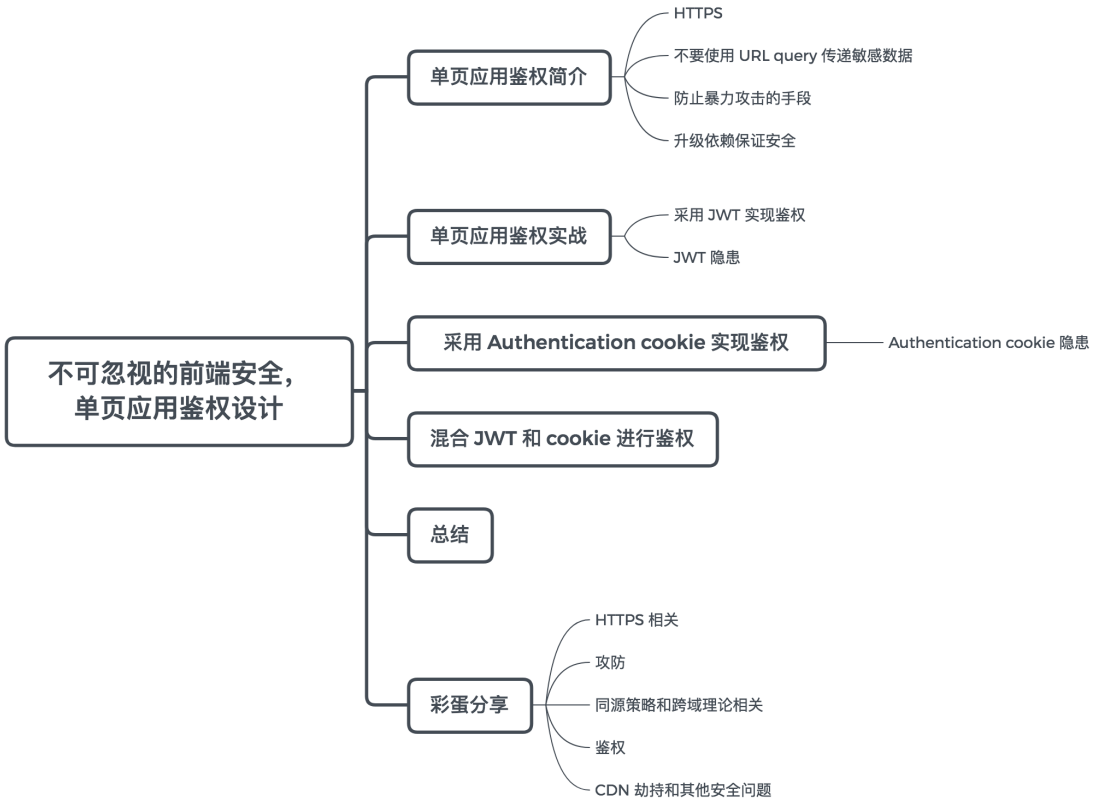
安全是计算机科学永远无法忽视的话题。随着互联网的发展，安全问题越来越突出，也越来越重要：它是一个程序可用性、健壮性的基础。这个话题可大可小，大到系统的设计，小到一行代码的写法，都可能影响系统的安全。

毫不例外，安全与前端开发的结合也持续走热。不管是经验丰富的程序员，还是尚在打基础的学生，也许都对 HTTPS、XSS、CSRF 等前端相关的安全问题不陌生。然而，这其中每一个主题都可以非常深入，都能系统地做一节课。但是，我认为面面俱到、走马观花地梳理这些内容，讲解这些概念价值不大。毕竟，这方面知识都已经比较成熟，社区上资料很多。

本讲我想从一个大部分产品都要涉及的登录鉴权入手，结合单页面应用，从这个角度，管中窥豹，尽可能多地涉及一些常见的安全知识，帮助大家了解前端安全。

接下来，让我们从应用场景入手，从前后端交互切入，以单页面应用为基础，呈现「鉴权」这个安全领域重要话题的全貌，并尽力覆盖到 XSS 和 CSRF 等攻击手段以及最佳实践。

关于这个主题的知识点如下：



单页应用鉴权简介

首先，我们要分清单页应用鉴权与传统鉴权方式有所不同：

单页应用采用前后端分离的设计方式，路由由前端管理，前后端遵循一定规范（如 REST、GraphQL），通过 AJAX 进行通信。在这种情况下，用户对页面请求时，后端经常无法获取用户身份信息，更无法确定返回的数据。

同时一次鉴权完毕后，如何在单页应用的体验当中，保持这个鉴权状态也值得思考。一般来说，单页应用鉴权采用下面的步骤实现。

> Step 1：前端根据用户交互，发送数据请求之前，需要准备用户信息，同数据请求一起发给后端处理。

> Step 2-1：后端按照约定好的规则，根据请求中带有的用户身份信息，进行验证。如果验证不通过，返回 403 或者 401 相关状态码或其他状态，以表示鉴权失败。

Step 2-2：如果鉴权成功，后端返回相关数据。

Step 3：前端根据数据渲染视图。

基本结构非常简单清晰：

在这个结构背后，隐藏的技术方案和安全细节非常值得我们思考，请继续阅读，我们将剖析几个重要概念和安全实践。

HTTPS

鉴权过程中，如果使用 HTTP 协议来传输敏感数据（用户昵称、用户密码、token……），那么很容易被中间人拦截获取。现代通信中，我们都使用 HTTPS 协议来对传输内容进行加密。关于 HTTPS 的应用及其原理，又是一个超级话题。这里由于内容的限制，不过多展开，给大家分享一下我收藏的关于 HTTPS 好的文章：

https 连接的前几毫秒发生了什么

完全图解 HTTPS

更安全的 Web 通信 HTTPS

图解基于 HTTPS 的 DNS

看图学 HTTPS

http 与 https 的区别我真的知道吗

深入揭秘 HTTPS 安全问题&连接建立全过程

HTTPS 系列干货（一）：HTTPS 原理详解

HTTPS 为什么更安全，先看这些

不要使用 URL query 传递敏感数据

URL query 会通过服务端日志、浏览器日志、浏览器历史记录查到。不要使用 URL query 传递敏感数据，这当然是最基本的准则之一。如果敏感数据在 URL query 中，这就给了恶意用户轻松获取数据的机会。同时，URL query 的长度也有限制，这也是其传递数据的弊端之一。

防止暴力攻击的手段

攻击者可以通过暴力手段，尝试攻破用户的密码等信息。因此后端服务要时刻注意加入频率限制，限制一个用户短时间尝试密码的次数；也可以限制可疑用户（比如触发了过多服务端错误用户）的访问。另外，需要注意的是不要给任何人暴露服务端的技术细节信息，比如要记得关闭 X-Powered-By（服务器响应头隐藏）；Node 端在使用 express.js 的情况下，强烈建议使用 Helmetjs。

Helmet 帮助 Node.js 开发者通过设置合理的 HTTP header，预防一些常见的 Web 漏洞，比如上面提到的关闭 X-Powered-By。实际上它就是一组灵活的中间件函数，增强以下 HTTP header 的安全性：

Content-Security-Policy 响应头，它可以设置应用是否可以引用某些来源内容，进而防止 XSS

关闭 X-Powered-By 响应头，以避免暴露服务端信息

增加 Public Key Pinning 响应头，预防中间人伪造证书

设置 Strict-Transport-Security 响应头，这样浏览器只能通过 HTTPS 访问当前资源

为 IE8+ 设置 X-Download-Options 响应头，目前只有 IE8+ 支持这个 header，用来预防下载内容的安全隐患

设置 Cache-Control 和 Pragma header 以关闭浏览器端缓存

设置 X-Content-Type-Options 响应头，以禁用浏览器内容嗅探

设置 X-Frame-Options 响应头，以预防 clickjacking，这个响应头给浏览器指示是否允许在 或者