A quick note before you start: this lesson will also be listed on Module 6 (Lesson 5).

I chose to include the lesson in both Module 4 and Module 6 because it naturally fits into both - it's focused on both the OAuth2 security as well as the AngularJS front-end.

The reference code is the one on branch *module6* - so make sure you switch to that before going through the video material.

1. Goals

Learn how to consume a REST API secured with OAuth2 (the Password flow) and using JSON Web Tokens, from an AngularJS front-end.

2. Lesson Notes

The main disadvantage of **using the Password Flow** directly in the front end is that we're exposing the client credentials into the public front end.

A simple alternative would be **using the Implicit Flow** - which has the advantage of being design specifically for public clients. The disadvantage of this particular flow is that it provides a bad user experience, as it requires a redirect. That, coupled with the fact that it doesn't support Refresh Tokens makes it less than ideal as well.

Finally, the real solution is to use keep **using the Password Flow but also use a thin proxy** between the front-end and the Authorization Server. The proxy would simply add the client credentials and forward the requests - so we're no longer exposing client credentials to the front-end.

3. Resources

- Cookies vs Tokens. Getting auth right with Angular.JSCookies vs Tokens. Getting auth right with Angular.JS

- OAuth and Single Page JavaScript Web-Apps