# Modules 17, 18 and 19 - the new OAuth material

As I'm sure you know, the Spring Security team has decided to re-write and consolidate the OAuth2 support in the framework.

That's both rare (it almost never happens in the Spring ecosystem) and also very good. OAuth is a critical part of the framework and the current stack is somewhat fragmented and harder to understand.

**The new stack will, naturally result in a lot of new material here, in Learn Spring Security as well.**

The material is actually not yet announced, but, of course, I've been working on it for a while now :)

With that in mind - you'll see the new text-only lessons appear in these next 3 modules - Module 17, 18 and 19.

There are two main reasons **I'm first releasing the lessons as text-only**.

One - the implementation is still ongoing so, while a lot of functionality is already GA, things might still change, so recording video is a bit early.

And two - it gives you access earlier to the lessons.

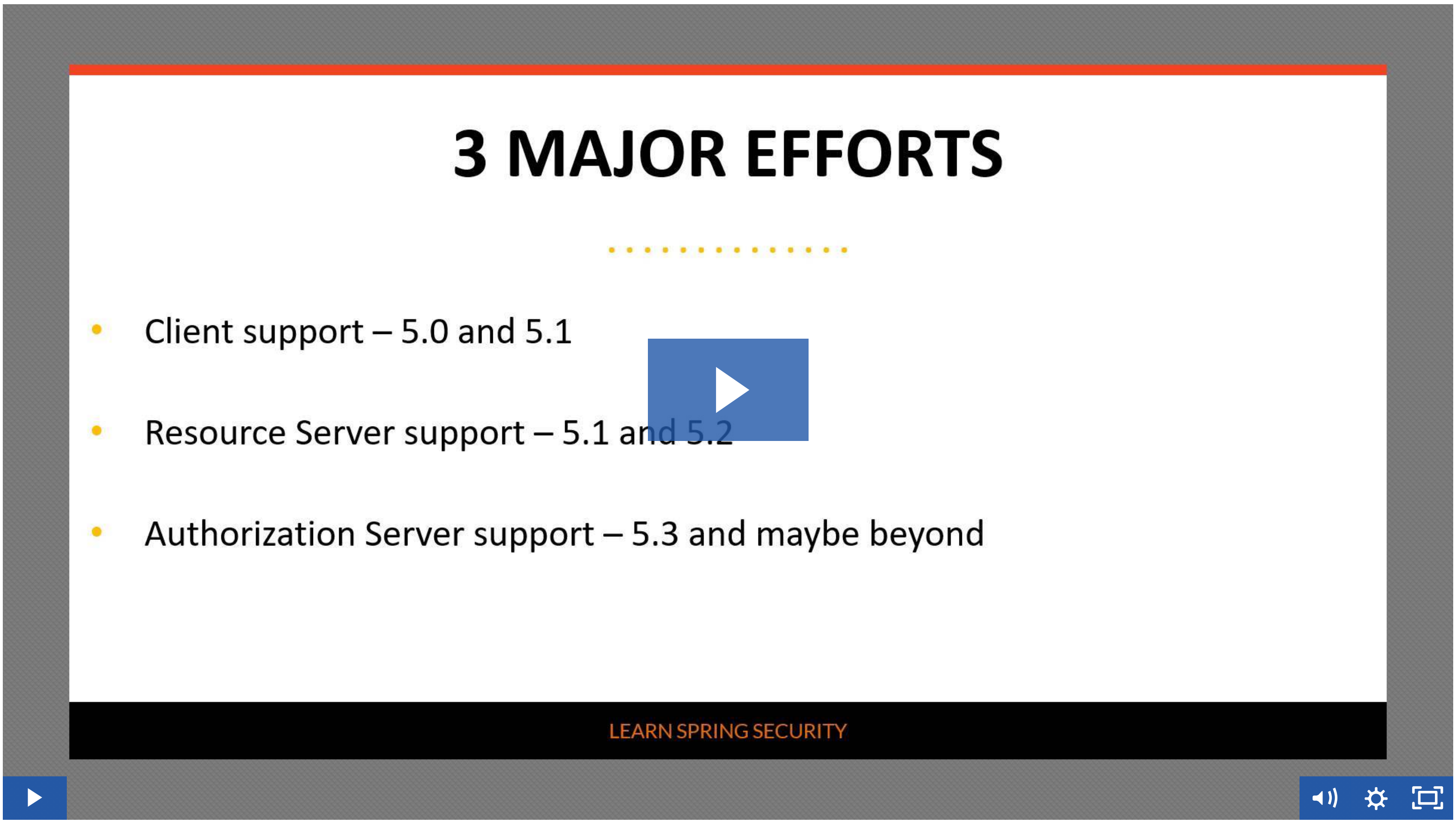
Of course, videos are coming :)

As always, ping me if you have any questions.

Cheers,

Eugen.

## Lesson 3: The State of OAuth2 in Spring Security 5 (NEW)



3 MAJOR EFFORTS

- Client support – 5.0 and 5.1
- Resource Server support – 5.1 and 5.2
- Authorization Server support – 5.3 and maybe beyond

LEARN SPRING SECURITY

### 1. Goal

In this lesson, we'll summarize the current state of the OAuth2 support in Spring and the work around it.

### 2. Lesson Notes

Before we look into any practical examples, let’s first understand the current state of OAuth in Spring Security 5.1 and beyond.

At the very start of 2018, the core team announced a really bold new direction in the development of OAuth support in Spring Security, which is basically a re-write of the functionality.

In the Spring ecosystem, rewrites are highly uncommon. So the fact that the team took this step means one thing: the OAuth2 support in Spring Security is getting a lot better.

This will modernize the implementation, as well as set the ground work for a lot of changes at the specification level. To have this as the reference implementation in Java, there's a need to be in sync with the OAuth specification. To be clear - this is a multi-year process.

Broadly speaking, the effort is split into 3 major areas:

- Client support
- Resource Server support
- Authorization Server support

Of course, there’s plenty of functionality that cuts across two or all three of these aspects.

The Client Support has been released with 5.0 and 5.1. Resource Server support was 5.1 and now 5.2.

And finally, the Authorization Support is coming after that, in Spring Security 5.3 and possibly beyond.

The existing OAuth stack is still fully supported and will be for a while. However, this is now in maintenance mode, meaning it’s getting security updates and minor features, but nothing major.

But the focus of the core team is now on the new functionality.

So which stack do we use?

At this point, there's not yet full parity between the existing and the new functionality. Therefore, if you’re working with OAuth functionality that’s already supported in the new GA releases of Spring Security, then use that. If it’s not yet supported, then use the old functionality.

Finally, there's a very useful feature matrix that shows exactly what's supported and where between the 2 stacks: [OAuth2 Feature Matrix](). This is updated as new functionality is being developed.

### 3. Resources

- [Next Generation OAuth 2.0 Support with Spring Security]()