1. Module 3 Overview

The goal of the third module is simple security for the API.

We'll set up Basic Authentication and , but we'll also take a step back and look at the broader picture, with the goal of understanding the security ecosystem available to us.

We'll of course explore the more advanced parts of that ecosystem in Module 6.

2. Goal of this Lesson

Learn about the high level options for API security and secure the API with Basic Authentication.

3. Lesson Notes

API Security Options

The API Security ecosystem has quite a number of options:

- Basic Authentication (stateless)
- Digest Authentication (stateless)

- Form-based Authentication (stateful)
- OAuth 2 (stateful)
- OAuth 2 + JWT (stateless)
- Custom Token Implementation (stateful or stateless)

The Security Configuration - Java vs XML

For Spring Security, the XML configuration is quite mature and concise, while the Java config is still new and somewhat rough around the edges.

We are mainly going to use Java for configuration, but keep in mind that we can always fall back on XML if needed.

A quick note here is that we are disabling the *web.xml* and transitioning to a Java config.

Errata: Before this gets fixed in the video, a quick note here - the value of the create-session element should be *stateless* not *false*:

```
<http create-session="stateless">
```

Spring Security and Maven

We'll either need:

- the *spring-boot-starter-security* dependency

- or - if we cannot use Spring Boot - *spring-security-web*

The Java Security Configuration

Standard elements in the security config:

- the *@Configuration* annotation
- scan the security packages
- enable web security
- extend the *WebSecurityConfigurerAdapter*
- configure global security
- configure http security and enable basic authentication

Security - from the Client side

- with no auth header - consume the API - get prompted

- with the wrong authentication header - consume the API -

- with an incomplete auth header - 403

- with the right authentication header - 200 OK

Existing Users and Default Credentials

As I mentioned early in Module 1 - Lesson 3, when the system starts up, a simple setup (*SecuritySetup.java*) runs creating some default Privileges, Roles and Users that will immediately be ready to use.

The users created via this simple setup are important here because these are of course the default users we can consume the API with, once it's secured.

The default admin user has the following credentials: *admin@fake.com / adminpass*

4. Resources

- The Module 3 Branch of the project on Github