Who am I?

Adrien Baudhuin

mail: adrien+uni[at]baudhuin.fr

Software Engineer at **AODocs**

All resources are available on **GitHub**

What are we going to learn?

- What is Spring
- Create a Spring Boot app
- Create REST API with Spring
- In next class: Connection to a database

Pre-requisites for this class

- Java
- Maven
- HTTP (Documentation)
- REST API (Documentation)

How to Spring

First came JEE (Java Enterprise Edition)

- Set of standards to build enterprise applications.
- A lot of libraries to solve common problems.

Quite old (1999) and not much used anymore.

Then came Spring

- A web framework to build web applications.
- Follows most of the JEE standards.
- Fixes modularity issues of JEE.
- Used everywhere (Netflix, Amazon, etc.)

More recently

New frameworks, specialized in different areas (microservices, serverless, etc.)

Most notable in the Java ecosystem:

- Micronaut
- Quarkus

Spring Web Framework

- A standardized structure.
- Highly configurable and customizable
- A set of libraries to solve common problems
    - Spring Web MVC (Create HTTP APIs)
    - Spring Data (Connect to Databases)
    - Spring Security (Secure and authenticate users)
    - Spring Cloud (Connect to Cloud Providers)

Spring Boot

- Spring is complex and hard to setup because of its flexibility.
- Spring boot is a preconfigured Spring application
- Easy to start a new project

Creating a Spring Boot app

Spring Initializer

Multi-Layer Architecture (Wikipedia)

> A multi-tier architecture is a client–server architecture in which presentation, application processing, and data management functions are physically separated.

Often, we have 3 layers:

- Controllers (Presentation)
- Services (Business logic)
- Repositories (Data)

Inversion of Control (Wikipedia)

> The application is in charge of creating all the objects it needs.

Dependency Injection (Wikipedia)

> The application injects the dependencies each component needs.

Aspect Oriented Programming (Wikipedia)

> Allows to add cross-cutting behaviours to an application without modifying the code.

First HTTP endpoint

- Create a controller
- Create a endpoint method
- Annotations
- Serialization

First service

- Create a service
- Adding logic
- Inject in the controller

Testing

- Unit testing Sercices
- Integration testing Controllers

Production

Lab

- Create a TODO app
- Multiple endpoints CRUD
- A service with business logic
- Tests