

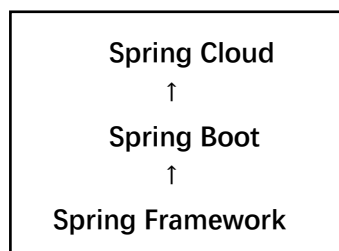
19 年上半年出版的书籍。 Spring Boot 核心思想。
(配套慕课网视频录制时间：2017.10.15 周日下午)

Spring Boot 2.0 ----初遇 Spring Boot

第一章：课程介绍

- Spring Boot 介绍
Spring 5.0 依赖了很多新型的技术， 必须依赖于 Java8 ，
是基于模块化 也就是 Maven 和 Gradle 的方式引入。
Web Flux ， Web MVC 整合。
- 环境： Java 8， Maven 3.2.5+ ，
- Helloworld 项目， 基于 Spring MVC 和 Web Flux 两种方式构建。
- 多模块构建： 模型层 (model) 持久层， 表示层。
- 打包和运行：
 - War 包， 放在 Tomcat 或者 Jetty 中运行。
 - Jar 包 通过 java-jar 等命令加载启动。
 - 采用本地或者开发环境的调试， 命令行启动。

Spring Boot 的角色：



(Spring Cloud 构建 Spring Boot 的分布式环境， 云应用。)

Spring Boot 2.0 新特性。

- 语言必须 Java 8+ ， 对函数式编程语言 如 Kotlin 支持
- 底层框架： Spring Framework 5.0.x ； (Spring 5.0 只支持 Java8 ， 亦支持 Kotlin， 源码中 很多是用 Kotlin 写的。)
- 全新特性：Web Flux （是 Spring Framework 5.0 的新特性 ， 是一种新的编程模型。
对 SpringMVC 做出了补充）
 - ◆ 传统的 SpringMVC 是用标签或者注解来描述请求， 映射， 响应 返回结果。
 - ◆ 兼容注解标签， 同时提出了新型 声明式函数式编程。

为什么选择 Web Flux

函数式编程： Java 8 Lambda 表达式。(一定要会 Lambda 表达式， 这个有点像前端的方式。)

响应式编程：Reactive Streams。

异步编程：Servlet 3.1 支持 非阻塞式 IO （NIO） Spring3.0 支持了异步的 NIO。

第二章： 第一个 SpringBoot 项目

环境准备

JDK 8

MAVEN 最新版即可{操蛋，maven3.6.2 和 IDEA 不兼容 还是不能搞最新的啊！！ 搞 3.6.0}

IDEA 最新的社区版即可。（日常开发完全够用了，还可以提高自己的能力。）

JDK：

下载 jdk-8u144-windows-x64

安装时候 不要 公共 JRE，记得新电脑要磁盘分区。

C：200（系统盘） D：300（软件盘，存所有 code 相关） E：500（数据盘，存游戏，文档等）。

新电脑来了之后，要完完全全的记录自己的环境配置，一分不拉。

旧电脑可以安装 Docker ， 学习数据可以放在 Dell 电脑里。

JDK 环境变量配置：

建议放在 Zhang Hao 的用户变量里面。

JAVA_HOME：jdk1.8 _144 文件夹目录

PATH： %PATH%;%JAVA_HOME%\bin

注意配置之后，要重启 cmd 才能检测。

java -version

echo %JAVA_HOME%

很多程序包括 IDEA 都自动装有 java 了，但是很多老程序还是需要本地的 Java，所以还是要配置环境变量（JAVA_HOME 和 PATH）的。

Maven 配置：

主要就两部分 setting.xml（D:\Apache\apache-maven-3.6.3\conf\settings.xml）

```
■ <localRepository>D:\Apache\maven-repository</localRepository>
■ <mirror>
    <id>alimaven</id>
    <name>aliyun maven</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public</url>
    <mirrorOf>central</mirrorOf>
</mirror>
```

Maven 环境变量配置：

建议放在 Zhang Hao 的用户变量里面。

M2_HOME：apache-maven 文件夹目录。

PATH：;% M2_HOME %\bin

检测是否成功，重启 cmd，输入：

mvn -version

`echo %M2_HOME%`

安装 IDEA 最新版本

IDEA 下好后自动有 32 位和 64 位，因为他可以自动提供 jre，这里不下载和安装 IDEA 的 JRE，用本地的即可。

可以从网上找相关的视频，进行 IDEA 的配置。

配置 maven

IDEA 内部嵌入了一个自带的 maven3. 不用，用本地的公用 maven 导入的时候同时导入源码，方便自我学习

Maven

Importing

Automatically download 将 ☒ Sources

配置 SDK

第一个 Spring Boot 应用（一）

编写 REST 程序

运行

用 HTTP 工具 postman 加以验证。

场景说明：

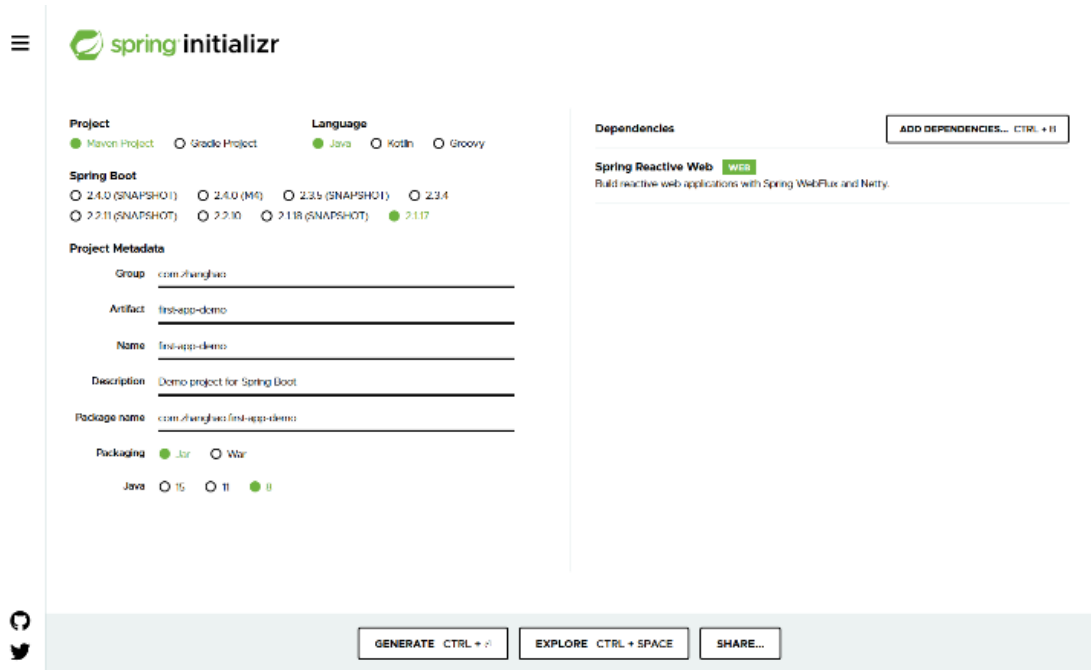
定义用户模型：属性（用户 ID 和名称）

客户端发送 POST 请求，创建用户，（Spring MVC）

客户端发送 GET 请求，获取所有用户（Web Flux）

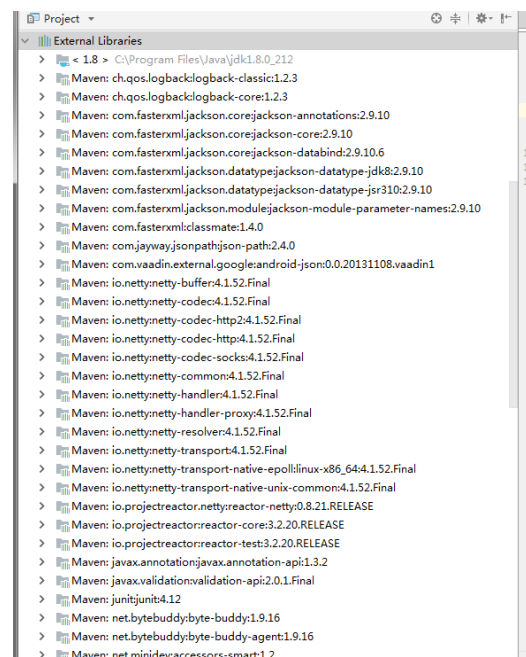
服务端用简单的 Map 的方式进行存储用户信息。就不用数据库了。

1. 下载 Postman 软件。
2. 进入 start.spring.io



选择相应的选项， 可以下载 demo， 然后 IDE import 即可。 很方便快捷的创建项目。
本项目 要选择 web Flux ， 即 Spring Reactive Web。

注意 IDEA 和 maven 有版本兼容性的问题。 IDEA 使用之前， 一定要配置好相关的配置



可以在 Maven 中看到是用 flux 的方式, 同时之前

Spring MVC 的注解方式也可以用

=====省略编程部分=====

运行程序

```
"C:\Program Files\Java\jdk1.8.0_212\bin\java.exe" ...
Connected to the target VM, address: '127.0.0.1:8278', transport: 'socket'

  ____  _
 / ___|| | | |
| |___| |_| |
 \___ \|  __/
      |_|_|_|

:: Spring Boot :: (v2.1.17.RELEASE)

2020-10-26 16:21:42.433 INFO 12208 --- [main] c.z.f.FirstAppDemoApplication : Starting FirstAppDemoApplication on SC-202007251559 with PID 12208 (D:\DATA\II
2020-10-26 16:21:42.439 INFO 12208 --- [main] c.z.f.FirstAppDemoApplication : No active profile set, falling back to default profiles: default
2020-10-26 16:21:44.620 INFO 12208 --- [main] o.s.b.web.embedded.netty.NettyWebServer : Netty started on port(s): 8080
2020-10-26 16:21:44.627 INFO 12208 --- [main] c.z.f.FirstAppDemoApplication : Started FirstAppDemoApplication in 2.693 seconds (JVM running for 5.477)
```

第一行的 pid 就是 该程序的进程号。

第三章：项目构建方式，多模块， 运行方式

Spring Boot 构建项目：

- 图形化方式 (<http://start.spring.io>)
- 命令行方式(Maven)
 - 暂时先不学，日后需要了再学

多模块 Spring Boot 项目

不同的模块履行不同的职责，不同的单元履行不同的使命。

重构：

- ◆ 调整主（父）工程类型（<packaging>）
- ◆ 创建子模块工程（<module>）
 - 模型层 model
 - 持久层 persistence
 - 表示层 web
- ◆ 子模块依赖管理（<dependency Management>）

其实就是在 pom 文件中 添加 module 依赖。一层包一层。

Web 层间接依赖所有，model 层无需依赖。

项目打包

Jar 作为第三方依赖或者是第二方依赖。

War 放在 Tomcat 等容器中运行。

Ear 包 企业中用 少见。

打成 jar 包

用 `java -jar web-0.0.1-SNAPSHOT.jar`

打成 war 包

都是在命令行来进行。 很吊！ 要慢慢来学习。

运行模式

IDEA 方式 （开发环境）

Jar/War 方式 （服务器线上部署）

Maven 插件方式 （当没有图形化，且有 Maven 插件 （一般很少见））

课堂总结

自动装配

嵌入式容器（Tomcat Jetty 等）

更好运维