

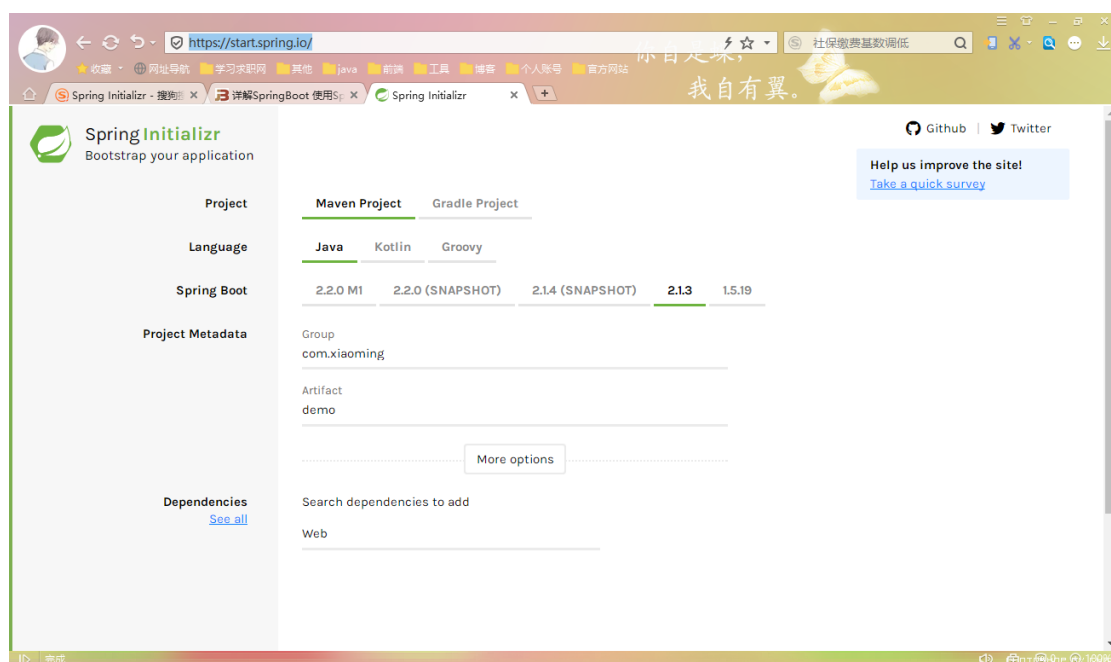
为什么使用 Spring Boot 构建微服务？

Spring Boot 除了是 Spring Cloud 的基础之外，也由于其自身的各项优点，如自动化配置、快速开发、轻松部署等，非常适合用作微服务架构中各项具体微服务开发框架。

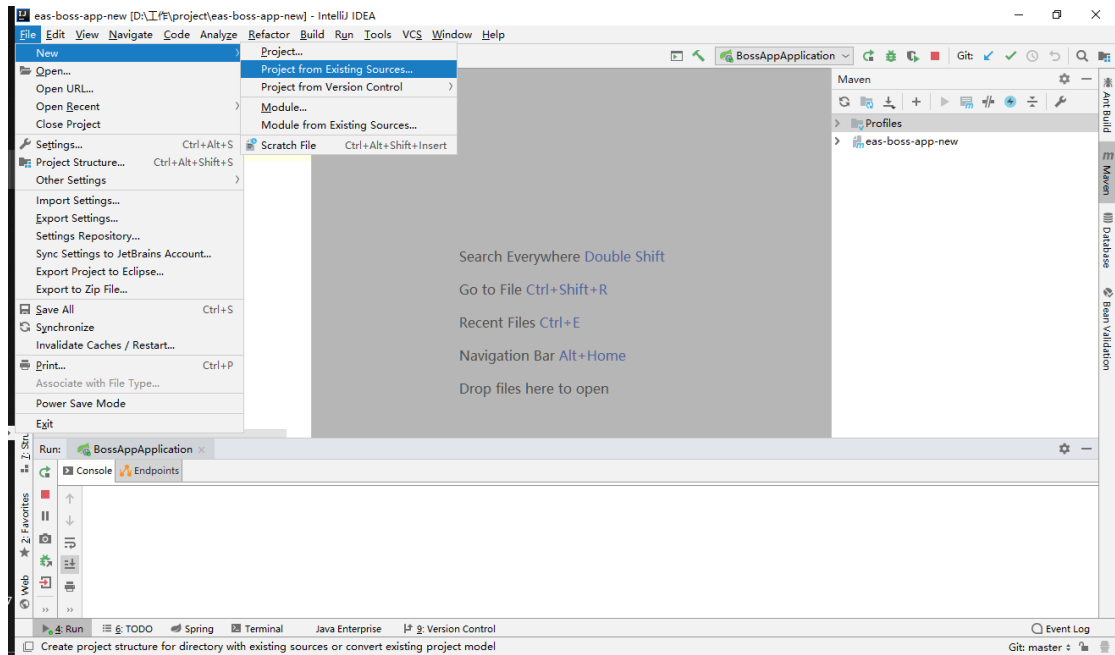
Spring Boot 除了可以很好融入 Docker 之外，其自身就支持嵌入式的 Tomcat、Jetty 等容器。

构建 Maven 项目

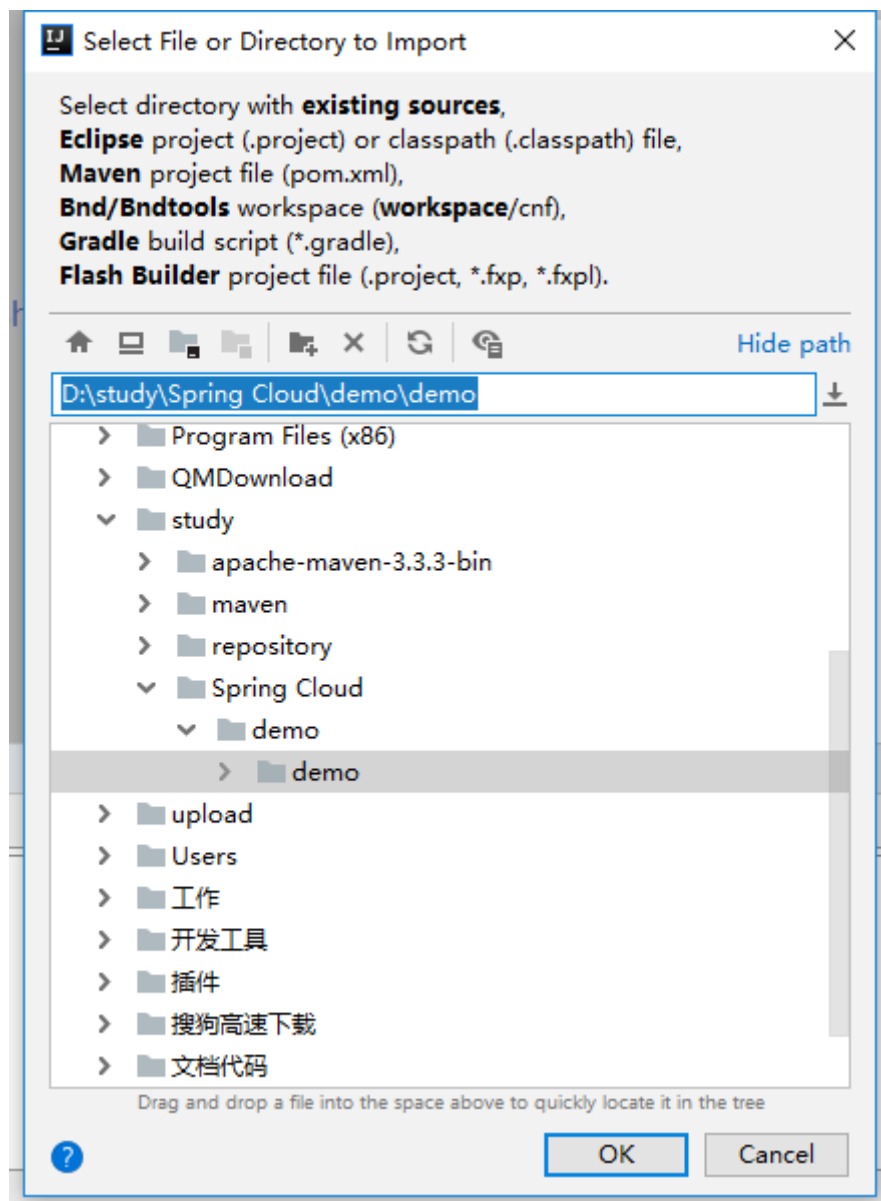
1. 通过官方的 Spring Initializr 工具来产生基础项目，访问 <https://start.spring.io/>
2. 选择构建工具，填写基本信息



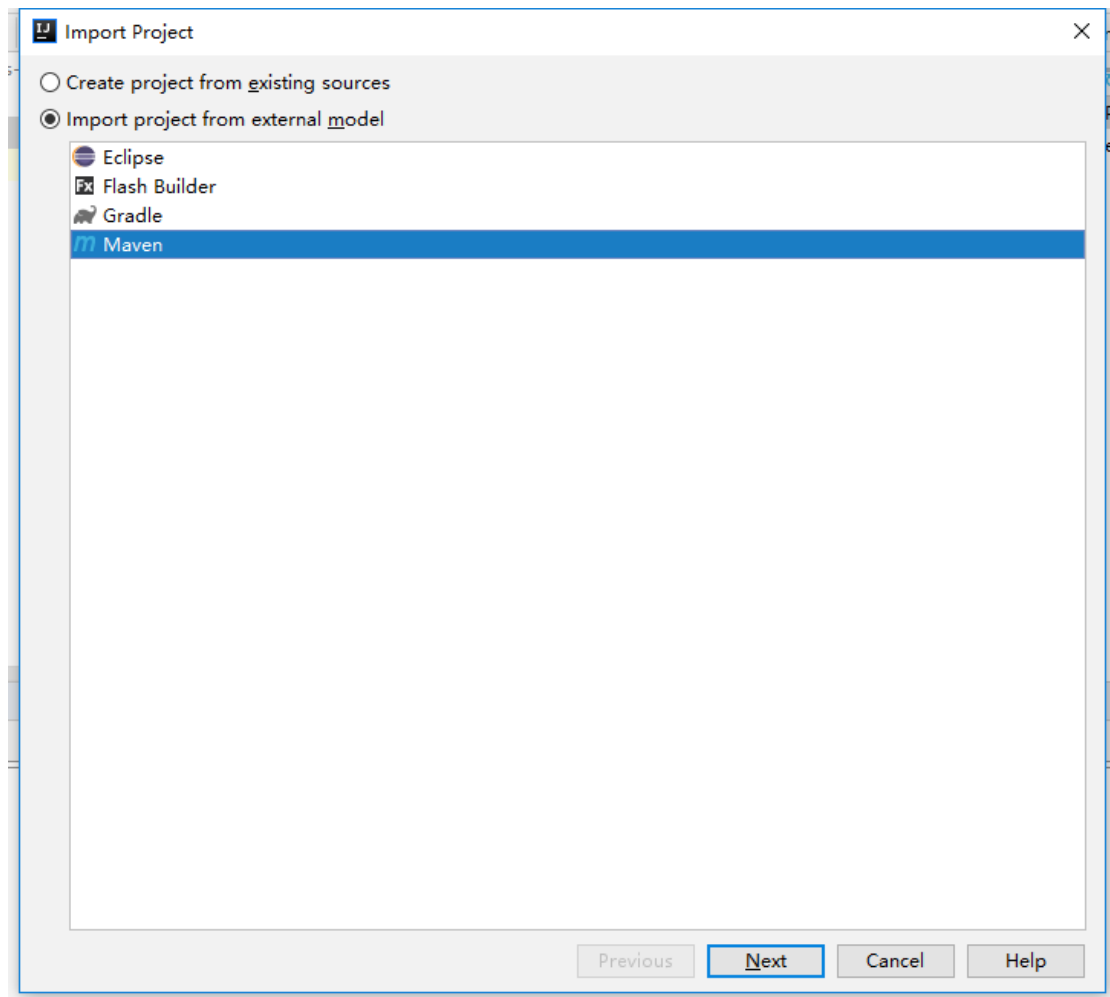
3. 单击 Generate Project 按钮下载项目压缩包
4. 解压项目包，使用 IDEA 以 Maven 项目导入
5. 选择 File-->New-->Project from Existing Sources...



6. 选择解压后的项目文件夹，单机 OK 按钮。



7. 单机 Import project from external model 并选择 maven，一直单机 Next 按钮



## 工程结构解析

- **src/main/java**: 主程序入口 DemoApplication，可以通过直接运行该类来启动 Spring Boot 应用。
- **src/main/resources**: 配置目录，该目录用来存放应用的一些配置信息，比如应用名、服务端口、数据库连接等。引入了 web 模块，会产生 static 目录与 templates 目录，static 用于存放静态资源，如图片、CSS、JavaScript 等；templates 用于存放 Web 页面的模板文件。
- **src/test/**: 单元测试目录，通过 JUnit4 实现，可以直接运行 Spring Boot 应用的测试。

## Maven 配置分析 (pom.xml 文件)

父项目中定义了 Spring Boot 版本的基础依赖以及一些默认配置内容，比如配置文件 application.properties 的位置等。

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.1.3.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
```

项目依赖:

spring-boot-starter-web: 全栈 Web 开发模块, 包含嵌入式 Tomcat、SpringMvc

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

spring-boot-starter-test: 通用测试模块, 包含 JUnit、Hamcrest、Mockito

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
```

这里的 web 和 test 模块, 在 Spring Boot 生态中被称为 StarterPOMs。StarterPOMs 是一系列轻便的依赖包, 是一套一站式的 Spring 相关技术的解决方案。开发者在使用和整合模块时, 不必再去搜寻样例代码中的配置来复制使用, 只需要引入对应的模块包即可。

Spring Boot 的 Starter POMs 采用 spring-boot-starter-\*的命名方式, \*代表一个特别的应用功能模块。