



链滴

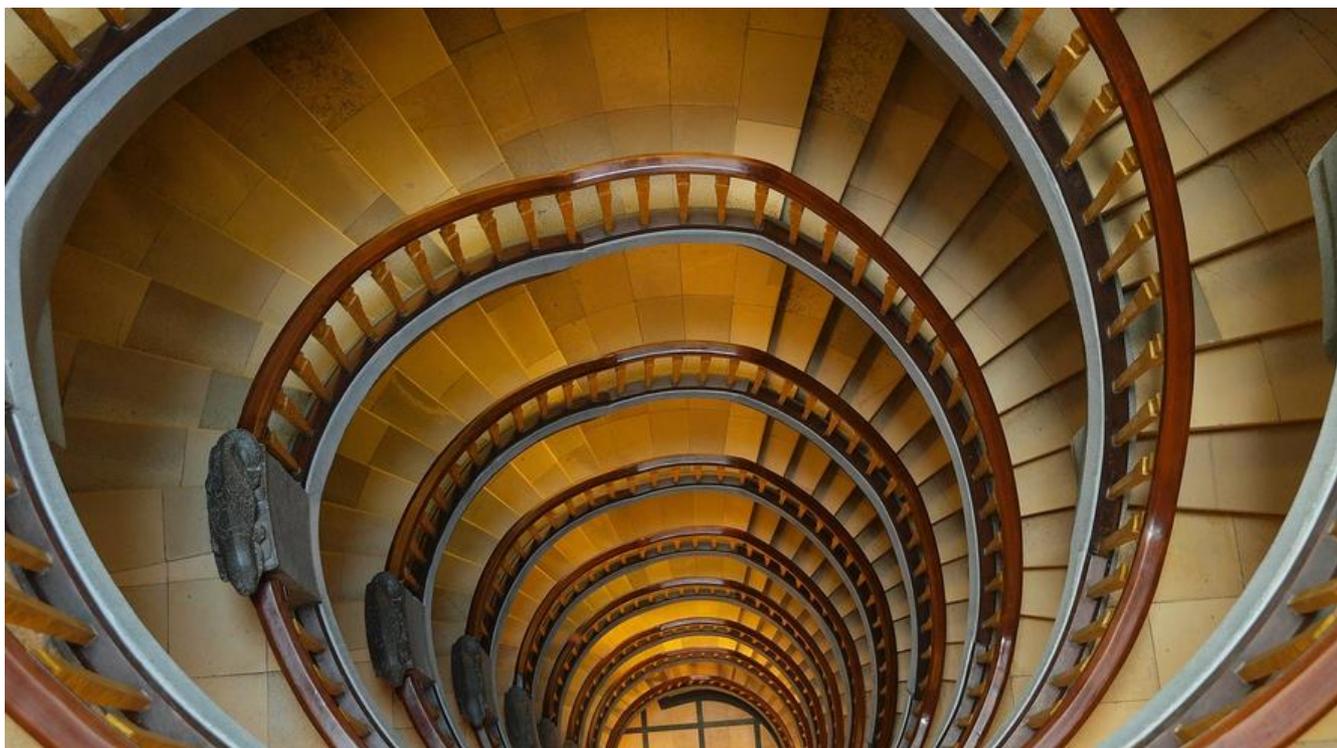
SpringBoot 初探：创建运行与文件说明

作者：[vcjmhg](#)

原文链接：<https://ld246.com/article/1592469691643>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



概述

本系列文章主要讲的是本人在阅读《Spring Boot In Action》过程中的总结的要点或者说是笔记，其本文是第一篇，主要讲述的是Spring Boot项目的创建、运行以及文件说明。

项目创建

一般来说，初始化一个SpringBoot项目方式好多种：

1. 通过SpringBoot 官方提供的Web页面：<http://start.spring.io>
2. 借助IDE（如IntelliJ IDEA）来创建
3. 通过使用curl命令下载一个官方的Demo程序
4. 通过Spring Tool Suite来创建

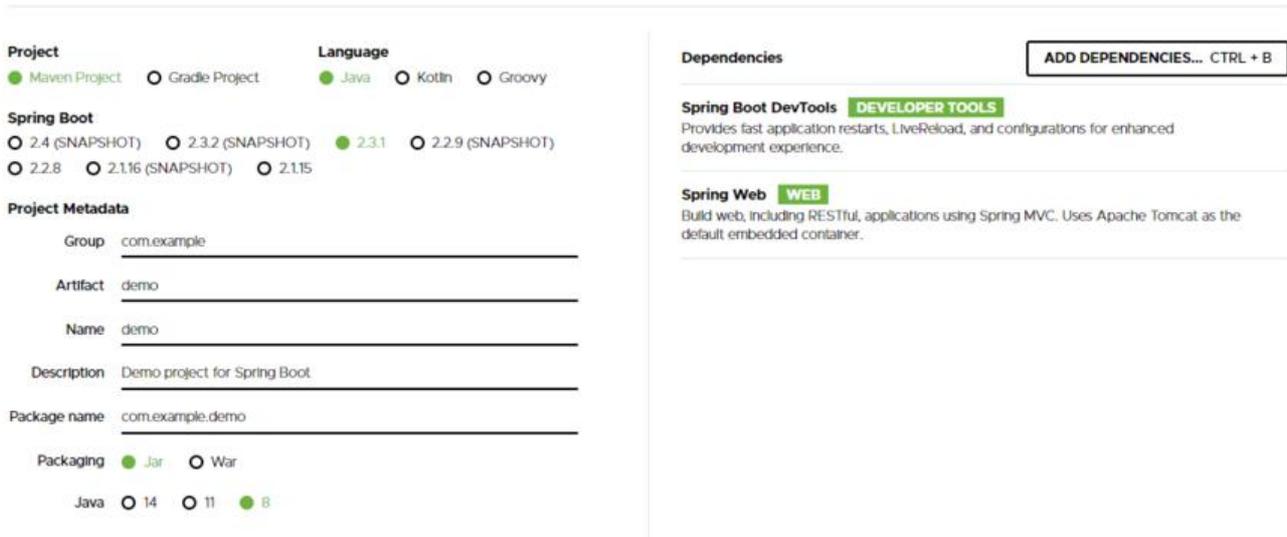
本文主要介绍前两种常用的创建方式

通过官方Web页面创建SpringBoot项目

1. 在线创建是SpringBoot官方提供的一种创建方式，浏览器中打开如下链接：

<http://start.spring.io>

2. 在打开的页面中我们可以选择项目的构建工具是 Maven 还是 Gradle、语言是 Java 还是其它、要用的 Spring Boot 版本号、项目的组织 Id（包名）、模块名称以及项目的依赖。

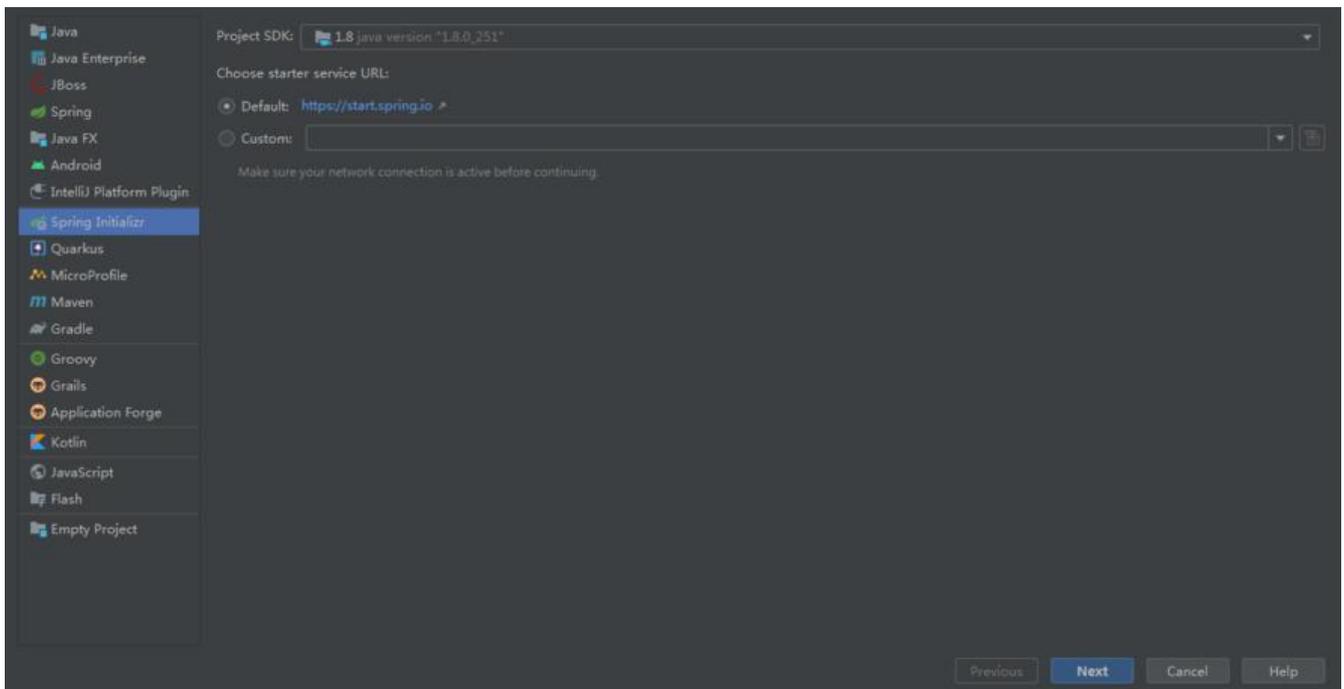


由于我开发的是SpringWeb应用因此，要选择Web。而Spring Boot DevTools是Spring官方提供的种开发工具，可以实现热部署，持续集成等功能，建议也勾选上。

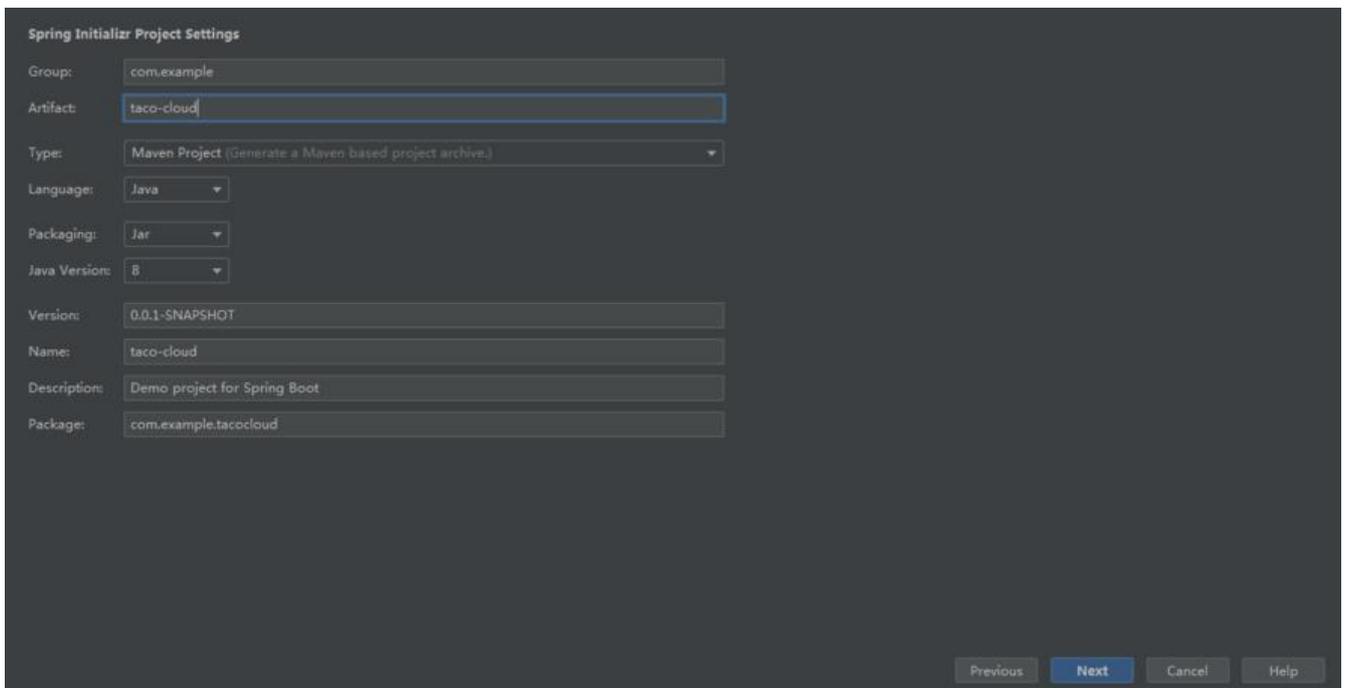
3. 填写完成之后点击 **CTRL+ENTER**会自动下载一个Maven项目压缩包。
解压后倒入到制定的IDE即可。

通过IntelliJ IDEA来创建一个SpringBoot项目

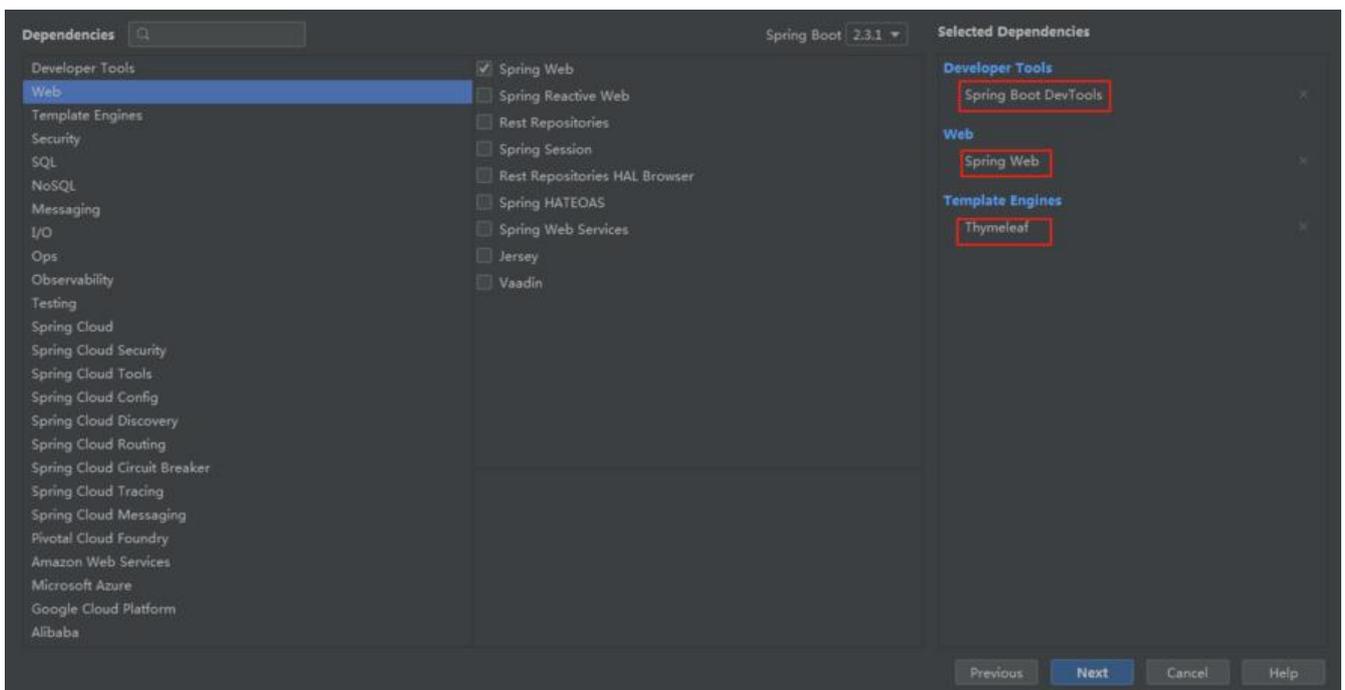
1. 创建项目时选择 **Spring Initializr**



2. 接着输入项目的基本信息，包括组织 Id、模块名称、项目构建类型、最终生成包的类型、Java 的本、开发语言、项目版本号、项目描述以及项目的包。



3. 接着选择项目所需要的依赖，之后IDEA会自动在pom.xml文件中添加对应的依赖。



4. 最后一步选择项目的路径以及名称，点击 **Finish**后 一个SpringBoot项目便创建成功了。

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

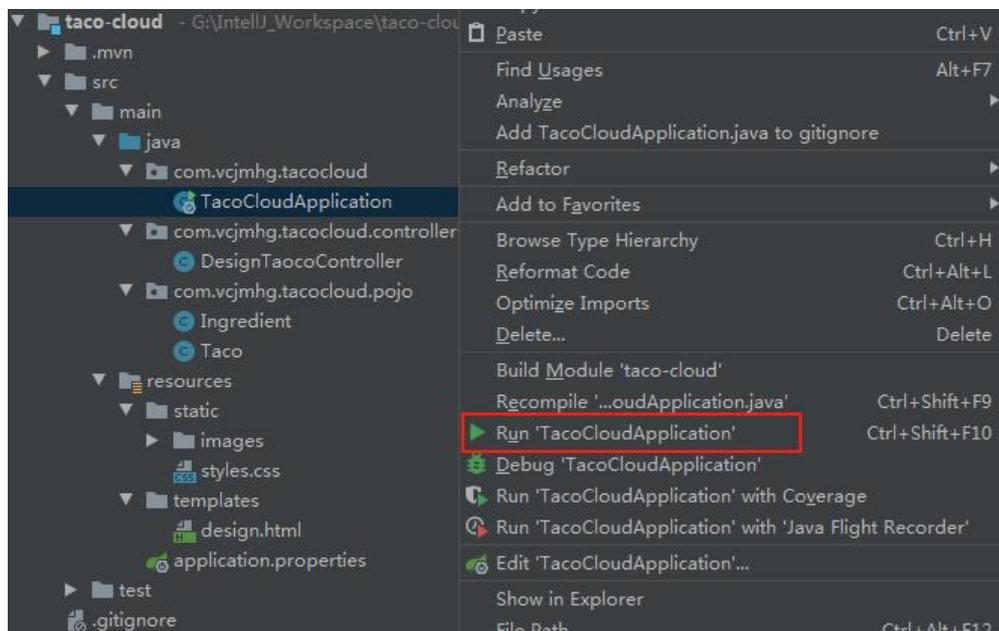
Thu Jun 18 15:26:08 CST 2020

There was an unexpected error (type=Not Found, status=404).

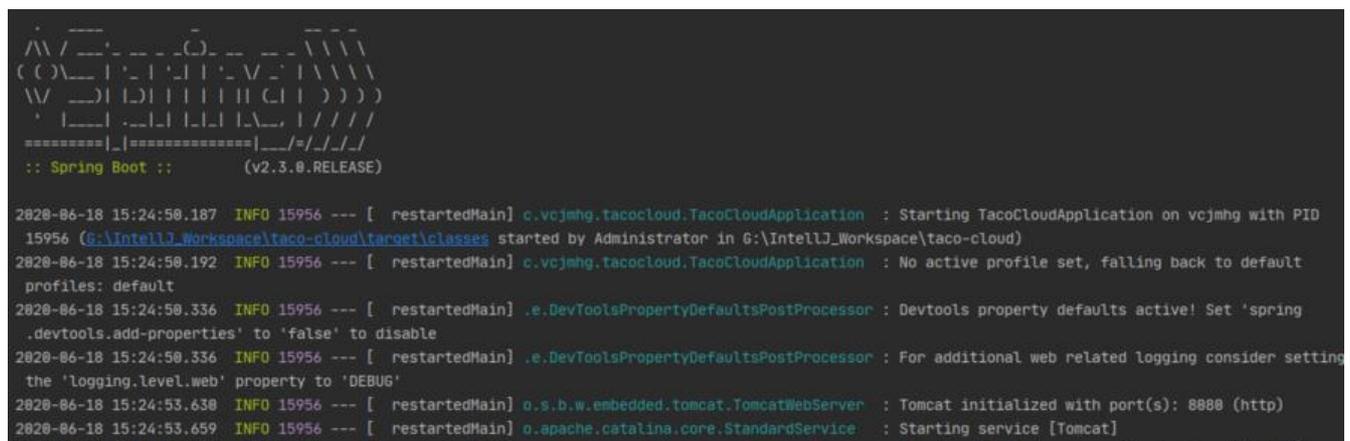
No message available

运行main函数

1. 在 IntelliJ IDEA 中右键点击 **App 类**, 然后点击 **run** 按钮即可启动项目。



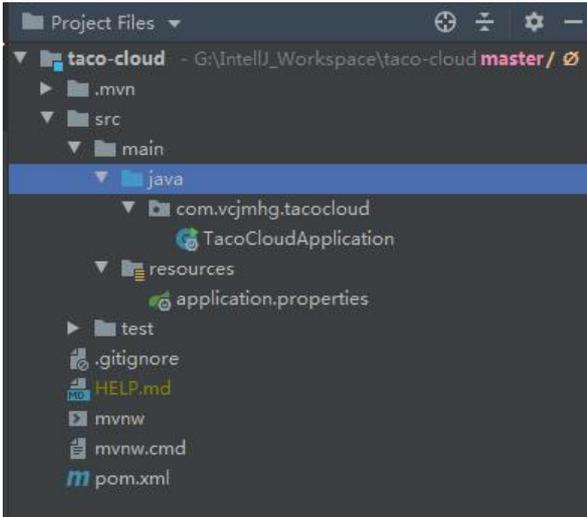
2. 启动完成后终端会显示如下信息:



各部分文件说明

项目结构

项目创建完成之后，可以看到一个SpringBoot项目具有如下结构：



简单来说我们可以将SpringBoot项目与一个典型的Maven项目或者Gradle项目结构非常类似，其中java源代码放到src/main/java目录下边，测试代码放到src/test/java目录下面，资源文件（如html文件）放置到/src/main/resources目录下。

除了上边三项内容我们可以发现，项目中还有如下内容：

1. **mvnw** 和 **mvnw.cmd**---Maven Wrapper的缩写。因为我们安装Maven时，默认情况下，系统所项目都会使用全局安装的这个Maven版本。但是，对于某些项目来说，它可能必须使用某个特定的Maven版本，这个时候，就可以使用Maven Wrapper，它可以负责给这个特定的项目安装指定版本的Maven，而其他项目不受影响。
2. **pom.xml**---maven项目的依赖文件，用来描述项目的依赖。
3. **TacoCloudApplication.java**---启动Spring boot项目的主类，通过调用该类的主方法可以启动Spring Boot项目。
4. **application.properties**---该文件在项目创建之初是空的，但是我们可以通过在该文件中设置某些特殊属性值，从而达到项目的特殊要求。
5. **static**---该目录用来放置任何的静态文件内容（图片，css文件，js文件等），项目初始化的时候为。
6. **templates**---该目录用来防止模板文件。
7. **TacoCloudApplicationTests.java**---一个简单的测试类保证 TacoCloudApplication可以成功运行。