



[返回首页](#)

芋道源码 —— 知识星球

我是一段不羁的公告！

记得给芬芳这 3 个项目加油，添加一个 STAR 噢。

<https://github.com/YunaiV/SpringBoot-Labs>

<https://github.com/YunaiV/onemall>

<https://github.com/YunaiV/ruoyi-vue-pro>

[2021-01-04](#)

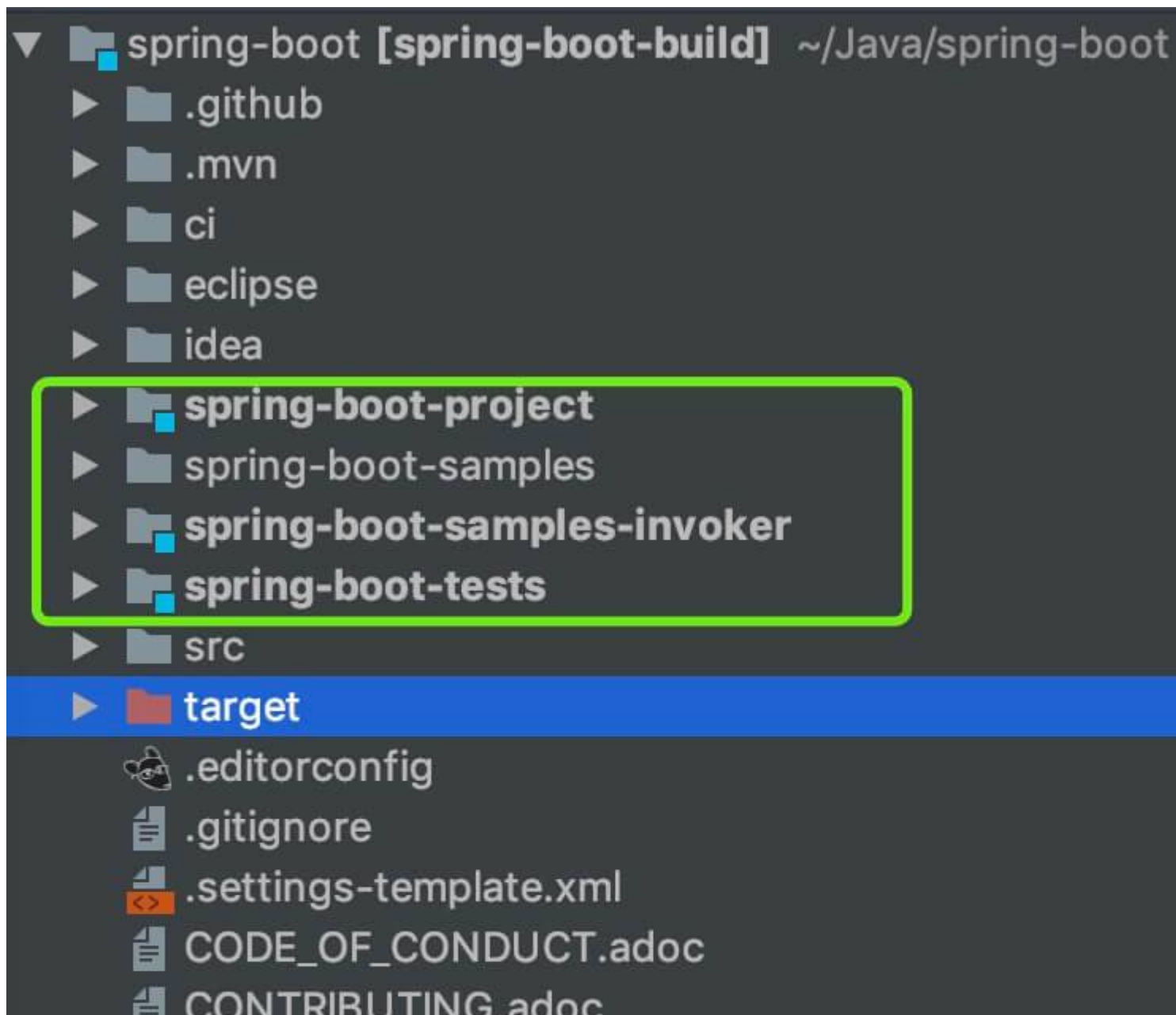
[Spring Boot](#)

精尽 Spring Boot 源码分析 —— 项目结构一览

1. 概述

本文主要分享 Spring Boot 的项目结构。

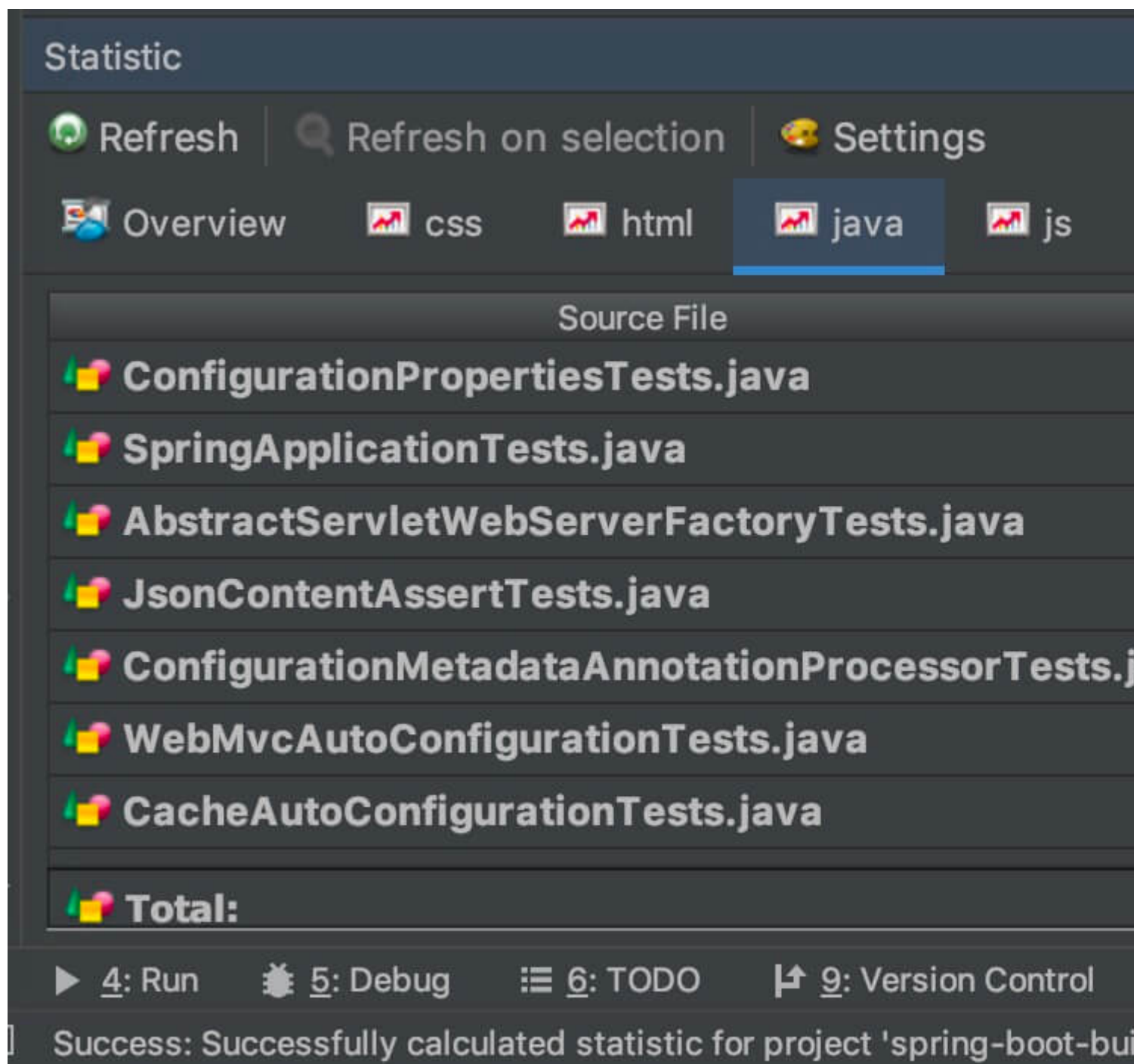
希望通过本文能让胖友对 Spring Boot 的整体项目有个简单的了解。



2. 代码统计

这里先分享一个小技巧。笔者在开始源码学习时，会首先了解项目的代码量。

第一种方式，使用 [IDEA Statistic](#) 插件，统计整体代码量。



我们可以粗略的看到，总的代码量在 268485 行。这其中还包括单元测试，示例等等代码。所以，不慌。特别是 Spring 项目的代码，单元测试覆盖是超级全面的。

第二种方式，使用 [Shell 脚本命令逐个 Maven 模块统计](#)。

一般情况下，笔者使用 `find . -name "*.java"|xargs cat|grep -v -e ^$ -e ^\s*\\/\./.*$|wc -l`。这个命令只过滤了部分注释，所以相比 [IDEA Statistic](#) 会偏多。

当然，考虑到准确性，胖友需要手动 cd 到每个 Maven 项目的 src/main/java 目录下，以达到排除单元测试的代码量。

项目	子项目
spring-boot-project	
	spring-boot
	spring-boot-actuator
	spring-boot-actuator
	spring-boot-autoconf
	spring-boot-cli
	spring-boot-dependen
	spring-boot-devtools
	spring-boot-docs
	spring-boot-parent
	spring-boot-properti
	spring-boot-starters
	spring-boot-test
	spring-boot-test-aut
	spring-boot-tools
spring-boot-samples	
spring-boot-samples-invoker	
spring-boot-tests	
合计	

统计完后，芴芴有点慌。哈哈哈哈哈。

3. spring-boot-project 项目

spring-boot-project 项目，Spring Boot 的代码实现，大概在 20W 代码左右。当然，实际上我们并不需要看完全部。芴芴自己的话，也只会选择关注部分的看看，并且写成源码解析。所以啊，尾随芴芴，不要方。

3.1 spring-boot 模块

spring-boot 模块，Spring Boot 的核心实现，大概在 4W 代码左右。提供了如下功能：

在 `org.springframework.boot.SpringApplication` 类，提供了大量的静态方法，可以很容易运行一个独立的 Spring 应用程序。

是不是超级熟悉。

带有可选容器的嵌入式 Web 应用程序（Tomcat、Jetty、Undertow）的支持。

在 `org.springframework.boot.web` 包下实现。

边界的外部配置支持。

… 省略其它。

感兴趣的胖友，可以自己先简单翻翻每个 package 包，基本每个包下，都是对每个功能的支持。例如说，web 支持 Web 服务器，jdbc 支持 JDBC 功能，task 支持调度任务，以及等等。

3.2 spring-boot-autoconfigure 模块

spring-boot-actuator-autoconfigure 模块，大概 4W 代码左右。spring-boot-autoconfigure 可以根据类路径的内容，自动配置大部分常用应用程序。通过使用

`org.springframework.boot.autoconfigure.EnableAutoConfiguration` 注解，会触发 Spring 上下文的自动配置。

这里的大部分，指的是常用的框架。例如说，Spring MVC、Quartz 等等。也就是说，如果 spring-boot-actuator-autoconfigure 模块，暂未提供的框架，需要我们去实现对应框架的自动装配。

这个模块的代码，必须要看，没得商量。

所以到此为止，我们已经看到对我们来研究 Spring Boot 最最最中航要的两个模块：spring-boot 和 spring-boot-autoconfigure，一共是 9W 行代码左右。

3.3 spring-boot-actuator 模块

spring-boot-actuator 模块，大概 2W 行代码左右。正如其模块的英文 actuator，它完全是一个用于暴露应用自身信息的模块：

提供了一个监控和管理生产环境的模块，可以使用 http、jmx、ssh、telnet 等管理和监控应用。

审计（Auditing）、健康（health）、数据采集（metrics gathering）会自动加入到应用里面。

一般情况下，我们可以不看这块代码的代码。

如果没有使用过 spring-boot-actuator 的胖友，可以看看下面两篇文章：

[《一起来学 SpringBoot 2.x | 第十四篇：强大的 Actuator 服务监控与管理》](#)

[《一起来学 SpringBoot 2.x | 第十五篇：actuator 与 spring-boot-admin 可以说的秘密》](#)

3.4 spring-boot-actuator-autoconfigure 模块

spring-boot-actuator-autoconfigure 模块，大概 1W7 行代码左右。它提供了 spring-boot-actuator 的自动配置功能。

一般情况下，我们可以不看这块代码的代码。

3.5 spring-boot-starter 模块

spring-boot-starter 模块，它不存在任何的代码，而是提供我们常用框架的 Starter 模块。例如：

spring-boot-starter-web 模块，提供了对 Spring MVC 的 Starter 模块。

spring-boot-starter-data-jpa 模块，提供了对 Spring Data JPA 的 Starter 模块。

而每个 Starter 模块，里面只存在一个 pom 文件，这是为什么呢？简单来说，Spring Boot 可以根据项目中是否存在指定类，并且是否未生成对应的 Bean 对象，那么就自动创建 Bean 对象。因为有这样的机制，我们只需要使用 pom 文件，配置需要引入的框架，就可以实现该框架的使用所需要的类的自动装配。

当然，正如我们在 [\[spring-boot-autoconfigure 模块\]](#) 所提到的，如果不支持的框架，需要自己实现对应的 autoconfigure 功能。举个例子，Dubbo 框架并未在 spring-boot-autoconfigure 模块实现自动装配，所以 Dubbo 团队提供了 [dubbo-spring-boot-project](#)。

如果觉得神奇的胖友，不烦可以跟着 [《快速开发一个自定义Spring Boot Starter》](#) 文章，来干一个自己的 Starter 模块。

3.6 spring-boot-cli 模块

spring-boot-cli 模块，大概 1W 行代码左右。它提供了 Spring 项目相关的命令行功能。它是 Spring Boot 的命令行界面。

它可以用来快速启动 Spring。

它可以运行 Groovy 脚本，开发人员不需要编写很多样板代码，只需要关注业务逻辑。

Spring Boot CLI 是创建基于Spring的应用程序的最快方法。

想要详细了解的胖友，可以看看 [《Spring Boot 文档 —— Part VII. Spring Boot CLI》](#) 文档。

一般情况下，我们可以不看这块代码的代码。

3.7 spring-boot-test 模块

spring-boot-test 模块，大概 1W 行代码左右。Spring Boot 提供测试方面的支持，例如说：

SpringBootTestRandomPortEnvironmentPostProcessor 类，提供随机端口。

org.springframework.boot.test.mock.mockito 包，提供 Mockito 的增强。

一般情况下，我们可以不看这块代码的代码。

3.8 spring-boot-test-autoconfigure 模块

spring-boot-test-autoconfigure 模块，大概 1W 行代码不到。它提供了 spring-boot-test 的自动配置功能

一般情况下，我们可以不看这块代码的代码。

3.9 spring-boot-devtools 模块

spring-boot-devtools 模块，大概 8000 行代码左右。通过它，来使 Spring Boot 应用支持热部署，提高开发者的开发效率，无需手动重启 Spring Boot 应用。

没有使用过的胖友，赶紧开始用啦。具体杂用，可以看看 [《Spring Boot 项目中使用 spring-boot-devtools 模块进行代码热部署，避免重新启动 web 项目》](#) 文章。

一般情况下，我们可以不看这块代码的代码。

3.10 spring-boot-tools 模块

spring-boot-tools 模块，大概 3W 行代码左右。它是 Spring Boot 提供的工具箱，所以在其内有多个子 Maven 项目。

注意哟，我们这里说的工具箱，并不是我们在 Java 里的工具类。困惑？我们来举个例子：spring-boot-maven-plugin 模块：提供 Maven 打包 Spring Boot 项目的插件。

关于 spring-boot-tools 模块的其它子模块，我们就暂时不多做介绍落。

一般情况下，我们可以不看这块代码的代码。

3.11 其它

spring-boot-project 项目的其它子模块如下：

spring-boot-properties-migrator 模块：500 行代码左右，帮助开发者从 Spring Boot 1 迁移到 Spring Boot 2。

spring-boot-dependencies 模块：无代码，只有所有依赖和插件的版本号信息。

spring-boot-parent 模块：无代码，该模块是其他项目的 parent，该模块的父模块是 spring-boot-dependencies。

spring-boot-docs 模块：1000 行代码左右，貌似是提供 Spring Boot 文档里的一些示例。不太确定，也并不重要。

4. spring-boot-samples 项目

spring-boot-samples 项目，2W 行代码左右。丧心病狂，提供了超级多的示例，简直良心无敌啊。

一般情况下，我们可以不看这块代码的代码。如果真的需要某个 Spring Boot 对某个框架的示例，大多数情况下，我们还是 Google 检索文章居多。

5. spring-boot-samples-invoker 项目

spring-boot-samples-invoker 项目，无代码，有点不造用户。当然，也并不重要。

6. spring-boot-tests

spring-boot-tests 项目，3000 行代码，主要是 Spring Boot 的集成测试、部署测试。

一般情况下，我们可以不看这块代码的代码。

666. 彩蛋

并没有什么实质性内容的一篇文章。感谢老田 [《Spring Boot 2.0系列文章\(五\): Spring Boot 2.0 项目源码结构预览》](#) 文章，让自己更加方便的了解 Spring Boot 2 的项目结构。

继续怼。有点偷懒的一个周六。

文章目录

1. [1. 1. 概述](#)
2. [2. 2. 代码统计](#)
3. [3. 3. spring-boot-project 项目](#)
 1. [3.1. 3.1 spring-boot 模块](#)
 2. [3.2. 3.2 spring-boot-autoconfigure 模块](#)
 3. [3.3. 3.3 spring-boot-actuator 模块](#)
 4. [3.4. 3.4 spring-boot-actuator-autoconfigure 模块](#)
 5. [3.5. 3.5 spring-boot-starters 模块](#)
 6. [3.6. 3.6 spring-boot-cli 模块](#)
 7. [3.7. 3.7 spring-boot-test 模块](#)
 8. [3.8. 3.8 spring-boot-test-autoconfigure 模块](#)
 9. [3.9. 3.9 spring-boot-devtools 模块](#)
 10. [3.10. 3.10 spring-boot-tools 模块](#)
 11. [3.11. 3.11 其它](#)
4. [4. 4. spring-boot-samples 项目](#)
5. [5. 5. spring-boot-samples-invoker 项目](#)
6. [6. 6. spring-boot-tests](#)
7. [7. 666. 彩蛋](#)