

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/onemail>

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

• SPRING

精尽 Spring 源码分析 —— 调试环境搭建 (Spring 5.3.1 版本)

今儿，我们来搭建一个 Spring 调试环境，目标是：**启动 Spring 容器，解析 XML 配置，创建 Bean 对象。**

为什么是 XML 配置，而不是注解呢？

使用 XML 配置来声明 Bean 的逻辑比较清晰，理解起来相对容易一些。

视频可见 B 站：<https://www.bilibili.com/video/BV1yh411q7jb/>

😬 芬芳比较腼腆，大家给个三连支持一下，感恩 1024~

1. 依赖工具

1.1 IDEA

当然是 JetBrains 出品的 IDEA 工具，相信绝大多数胖友都使用的它。

芬芳使用的是 **2020.3** 版本，胖友尽量保证不低于该版本哈。

友情提示：如果不知道怎么激活 IDEA 的胖友，可以访问 [《IDEA 激活指南》](#) 地址，花 5 分钟学习！支持所有版本~

1.2 JDK

需要使用 JDK 编译 Spring 的代码，这里芬芳使用的是 JDK **1.8** 版本

```
$ java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```

2. 源码拉取

使用 IDEA 从官方仓库 <https://github.com/spring-projects/spring-framework> 克隆项目。

友情提示：如果网络不是很好的胖友，可以选择和芳芳一样，使用 Gitee 提供的镜像仓库 <https://gitee.com/mirrors/Spring-Framework.git>

即使使用 Gitee 的镜像，克隆也要好几分钟，代码有 100M 左右！！

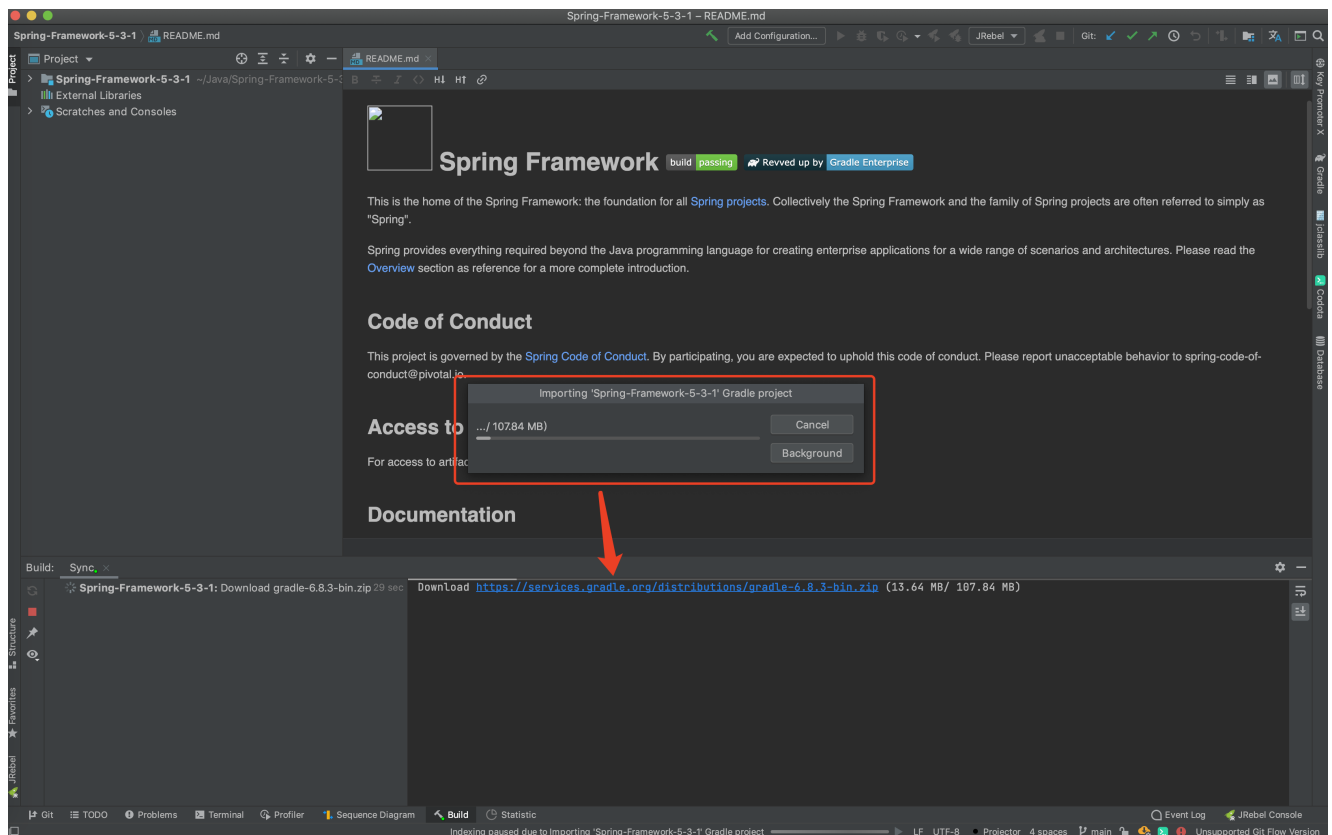
这里，我们使用的 Spring 版本是 **5.3.10-SNAPSHOT**。

友情提示：胖友可以考虑 Fork 下[官方仓库](#)，为什么呢？

既然开始阅读、调试源码，我们可能会写一些注释，有了自己的仓库，可以进行自由的提交。😁

3. 下载依赖

① 克隆完成 Spring 项目之后，IDEA 会自动下载需要的 Gradle 工具。如下图所示：

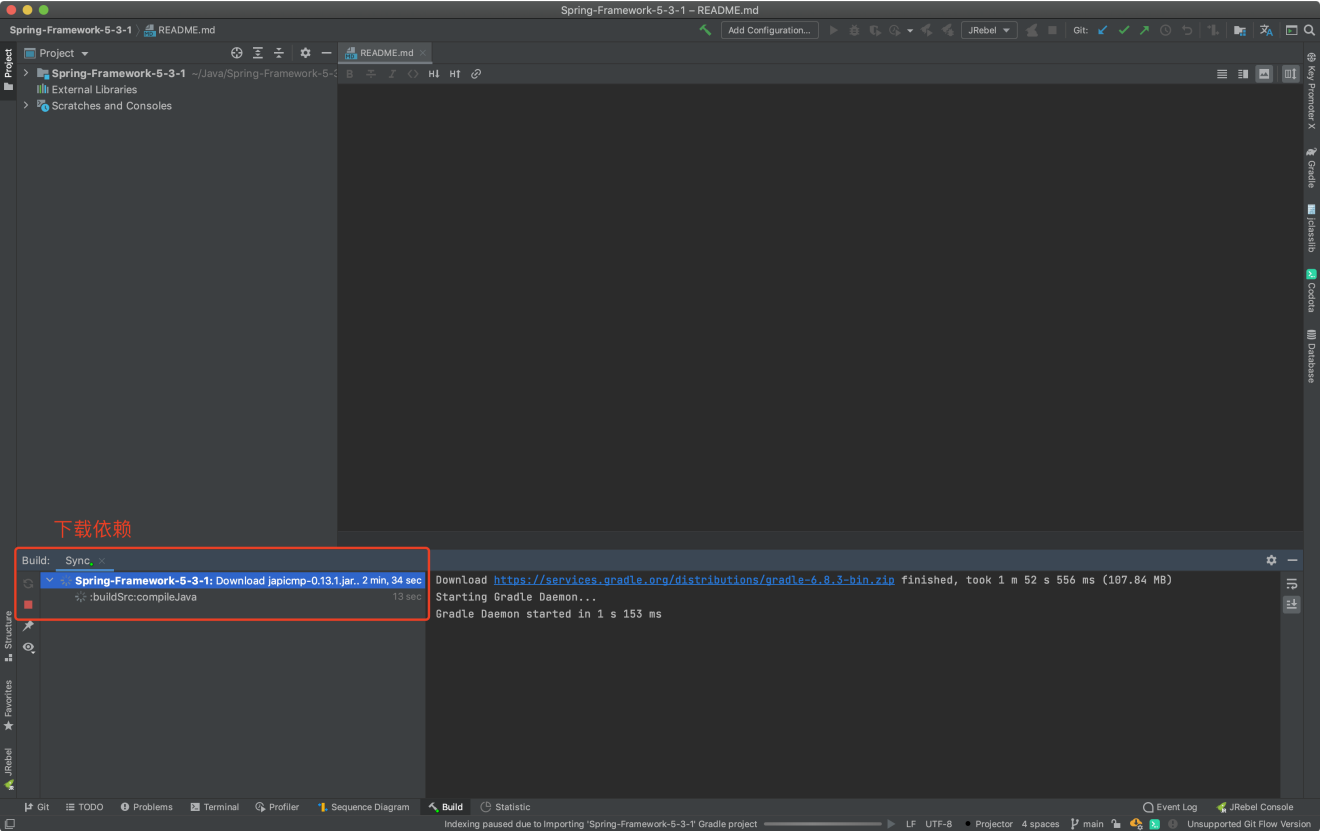


这里，我们使用的 Gradle 版本是 **6.8.3**。

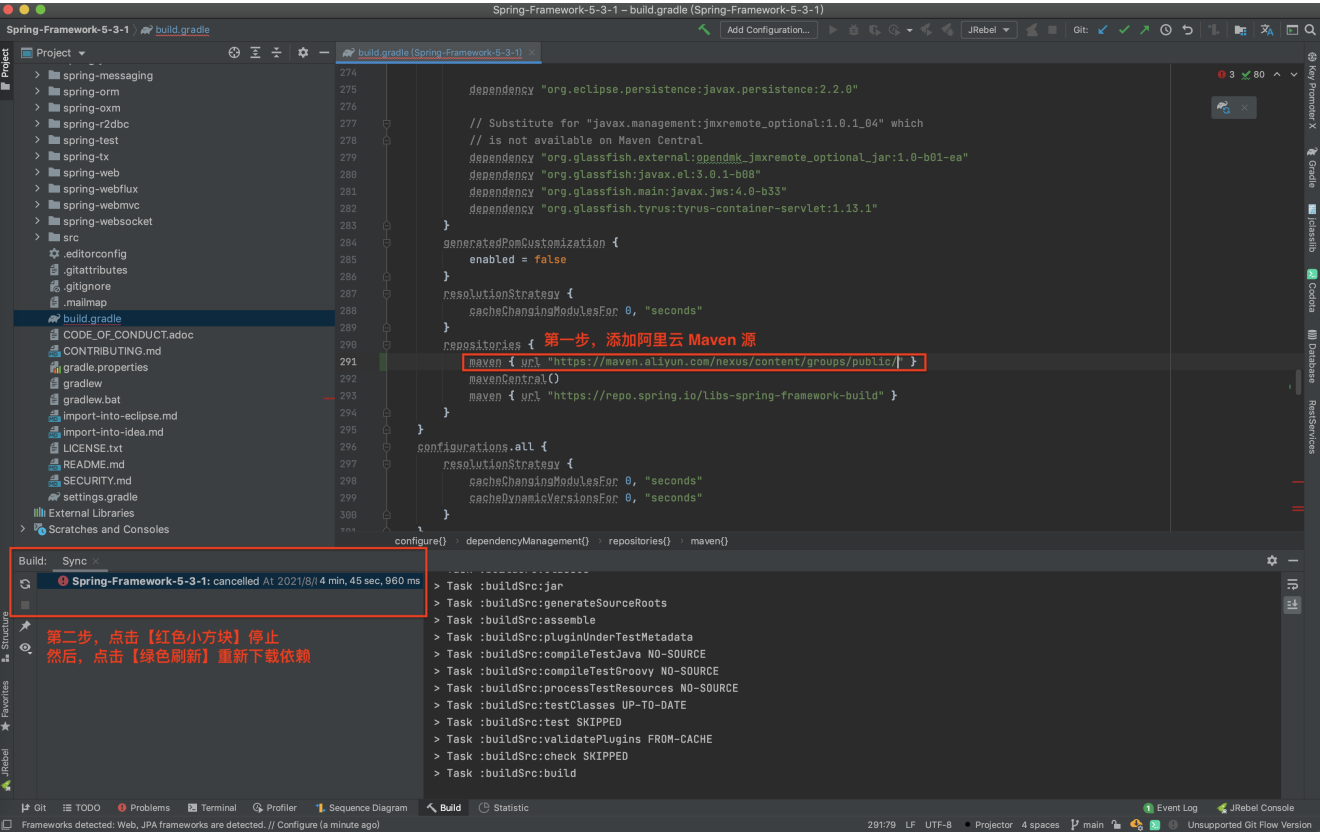
友情提示：由于是从国外的网址下载，所以需要耐心等待一会，芳芳花费了 2 分

多钟。

② 下载完 Gradle 工具之后，IDEA 就会使用它自动下载相关的依赖库。如下图所示：



因为 Gradle 支持使用 Maven 依赖，所以我们可以使用阿里云的 Maven 镜像 <https://maven.aliyun.com/nexus/content/groups/public/>。修改 build.gradle 文件，如下图所示：



效果非常不错，不过芳芳还是花了 10 分钟才下载完了依赖。主要原因是，Spring 对很多框架进行集成，导致引入了非常多的依赖。

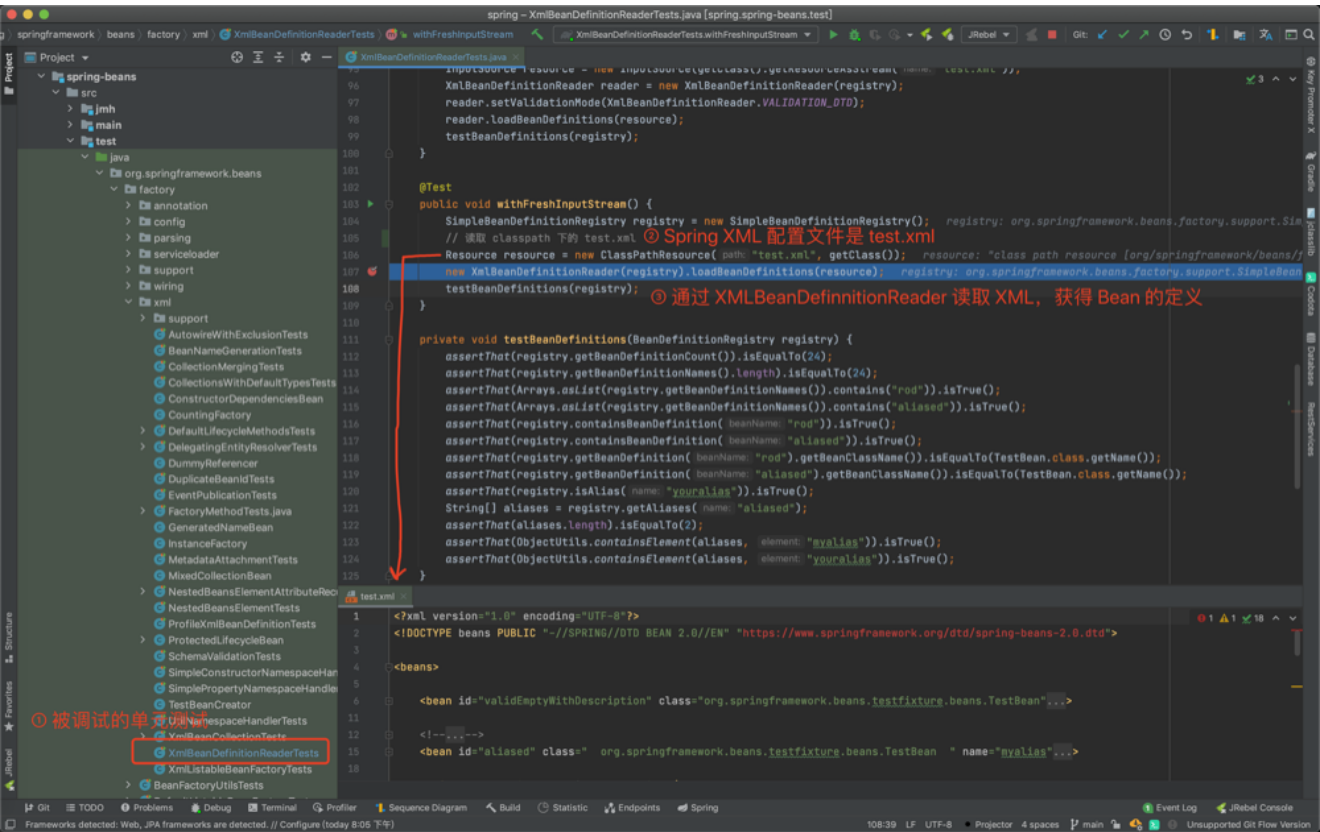
ps: 如果中间下载失败, 点击【绿色刷新】按钮, 继续下载依赖即可。

4. 调试 Spring 示例

依赖下载完, 我们就可以调试 Spring 的源码。虽然说 Spring 并没有直接提供 example 使用示例项目, 但是我们通过调试 Spring 提供的**单元测试类**, 了解 Spring 的执行流程。

例如说:

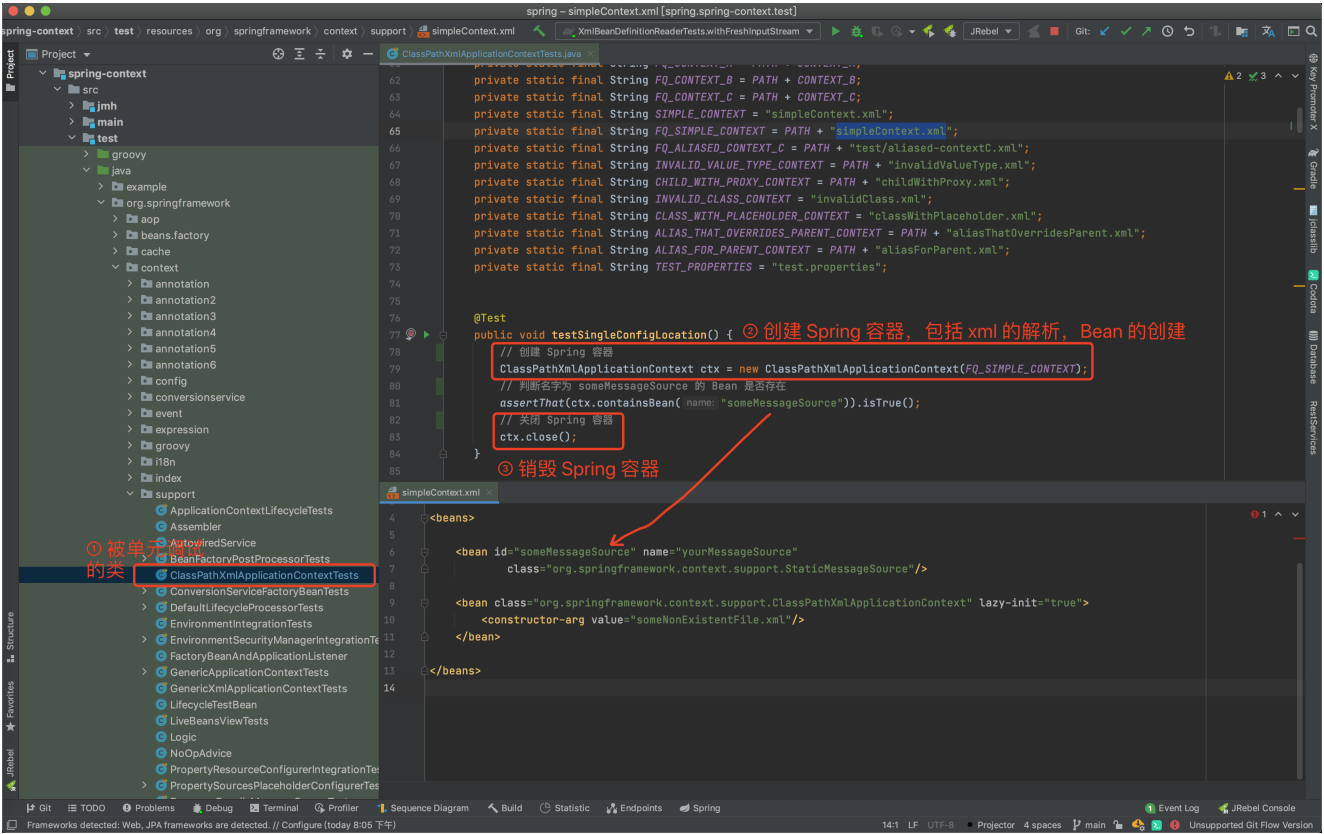
① 通过 Debug 运行 XmlBeanDefinitionReaderTests 类的 #withFreshInputStream() 的方法, 调试 Spring 解析 XML 配合, 获得 Bean 的**定义**。如下图所示:



后续, 胖友可以通过调试 XmlBeanDefinitionReader 类的源码, 了解更多。

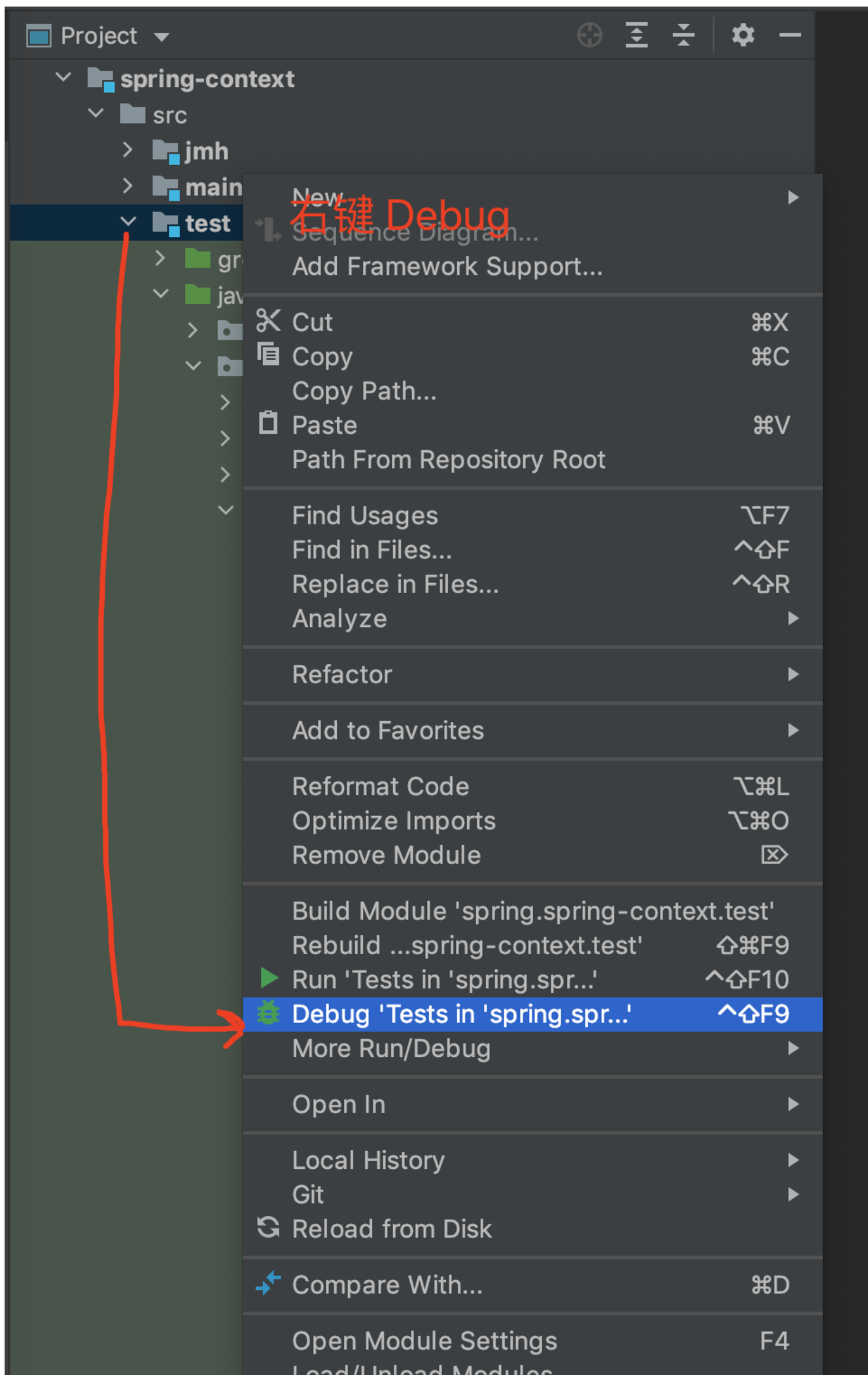
友情提示: Spring 是先解析到 Bean 的定义, 然后创建 Bean 对象。😊

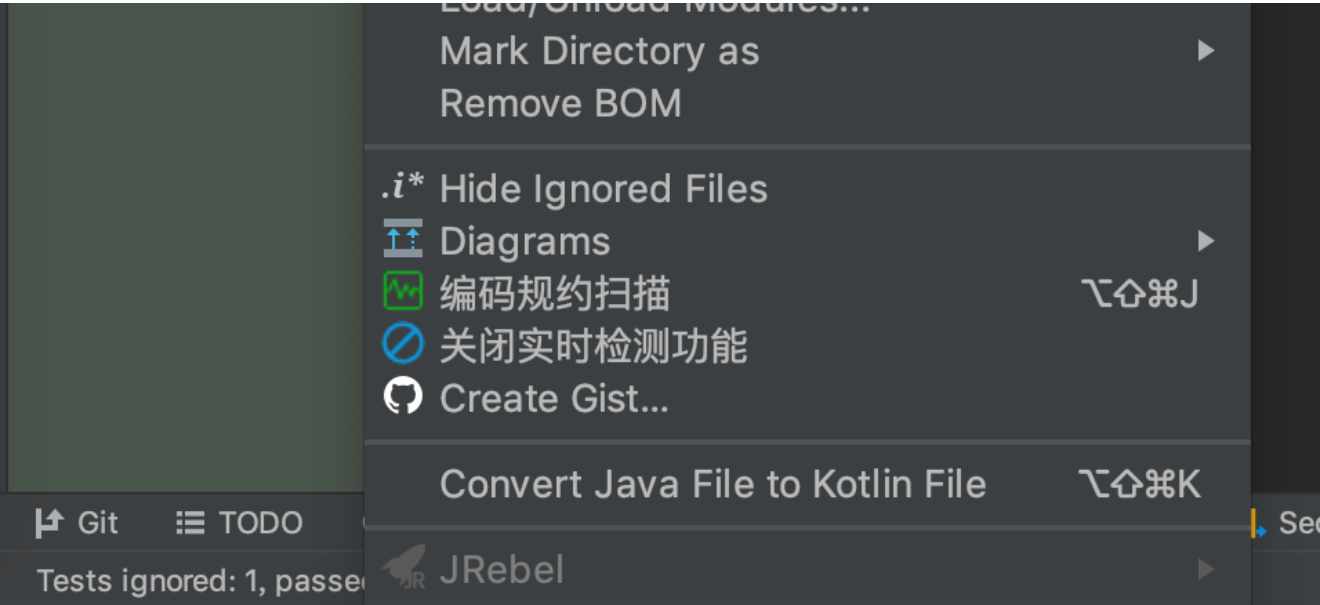
② 通过 Debug 运行 ClassPathXmlApplicationContextTests 类的 #testSingleConfigLocation() 的方法, 调试 Spring 容器的初始化过程, 包括 Bean 的**创建**。如下图所示:



后续，胖友可以通过调试 ClassPathXmlApplicationContext 类的源码，了解更多。

再额外分享一个小技巧，如果在阅读 Spring 源码的时候，发现一个逻辑的分支走不到，我们在该处打上断点，然后执行整个目录的单元测试。如下图所示：





5. 源码解析

Spring 的代码量，非常庞大，目前已经有 70W 行左右，自己直接死磕的话，基本在 3 天热情之后，就会选择放弃。

所以这里，茆茆来推荐下 Spring 相关的源码解析的书籍，帮助大家更好的阅读。

① 茆茆修改的《死磕 Spring 源码解析》，针对 Spring 5.1.1 版本。

如果大家在阅读中有碰到什么问题，欢迎星球给茆茆提问哈，大胆的，自己人！

② 《Spring源码深度解析（第 2 版）》，针对 Spring 5.X 版本，豆瓣评分 6.5 分，个人觉得分数有点偏低，7.5 分还是有的。

链接: <https://pan.baidu.com/s/1rxB3p5iR3JZFlwf92py9-w> 提取码: 81b1 复制这段内容后打开百度网盘手机App，操作更方便哦

写的还不错，茆茆最早读 Spring 源码也是看了这本书。当时还是第 1 版，现在都是第 2 版，真的是时光匆匆！哈哈哈，现在的书架上，还放了一本。

③ 《Spring 技术内幕（第 2 版）》，出版时间是 2012 年，针对的 Spring 版本可能相对老一点，豆瓣评分 6.5 分。

链接: <https://pan.baidu.com/s/1ad9BkvnvpvAoIU2DI3Dv5gg> 提取码: svyn 复制这段内容后打开百度网盘手机App，操作更方便哦

茆茆暂时没看过，胖友可以作为字典，在哪一块卡壳的时候，查一查有没讲到那块的源码。