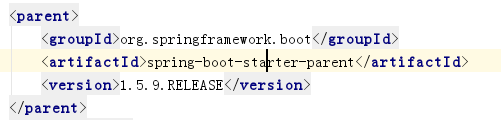
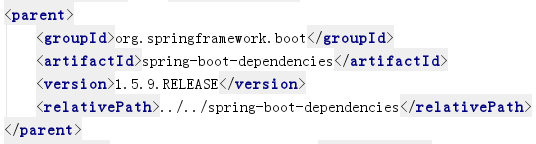
# POM文件

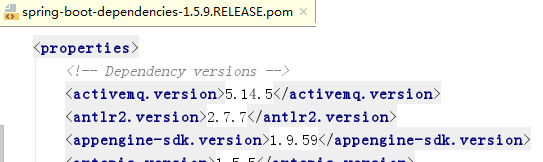
## 父项目



他的父项目是



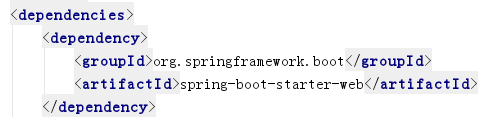
他来真正管理Spring Boot应用里面的所有依赖版本；



Spring Boot的版本仲裁中心；

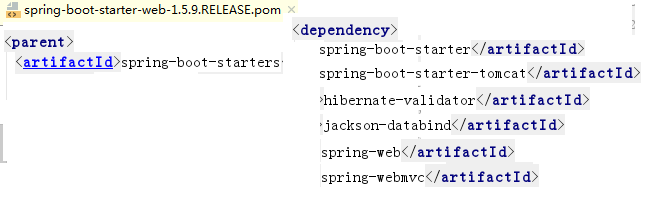
以后我们导入依赖默认是不需要写版本；（没有在dependencies里面管理的依赖自然需要声明版本号）

## 启动器



spring-boot-starter\*\*-==web==：

spring-boot-starter：spring-boot场景启动器；帮我们导入了web模块正常运行所依赖的组件；



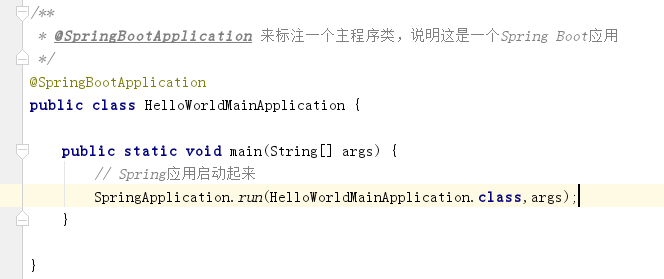
Spring Boot将所有的功能场景都抽取出来，做成一个个的starters（启动器），只需要在项目里面引入这些starter相关场景的所有依赖都会导入进来。要用什么功能就导入什么场景的启动器

官网：

https://docs.spring.io/spring-boot/docs/1.5.14.BUILD-SNAPSHOT/reference/htmlsingle/#using-boot-starter

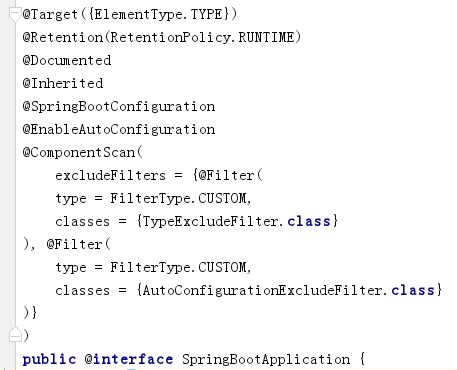
# 主程序类，主入口类

## @SpringBootApplication



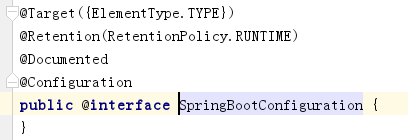
@SpringBootApplication\*\*: Spring Boot应用标注在某个类上说明这个类是SpringBoot的主配置类，SpringBoot就应该运行这个类的main方法来启动SpringBoot应用；

@SpringBootApplication源码

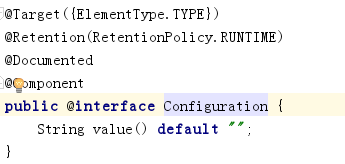


## @SpringBootConfiguration

@SpringBootConfiguration:Spring Boot的配置类：标注在某个类上，表示这是一个Spring Boot的配置类；



@Configuration:配置类上来标注这个注解；

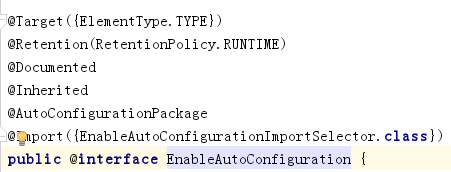


由以前配置文件替换成现在配置类；配置类也是容器中的一个组件；@Component

## @EnableAutoConfiguration

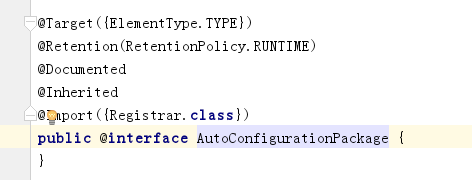
@ EnableAutoConfiguration：开启自动配置功能；

以前我们需要配置的东西，Spring Boot帮我们自动配置；@EnableAutoConfiguration告诉SpringBoot开启自动配置功能；这样自动配置才能生效；



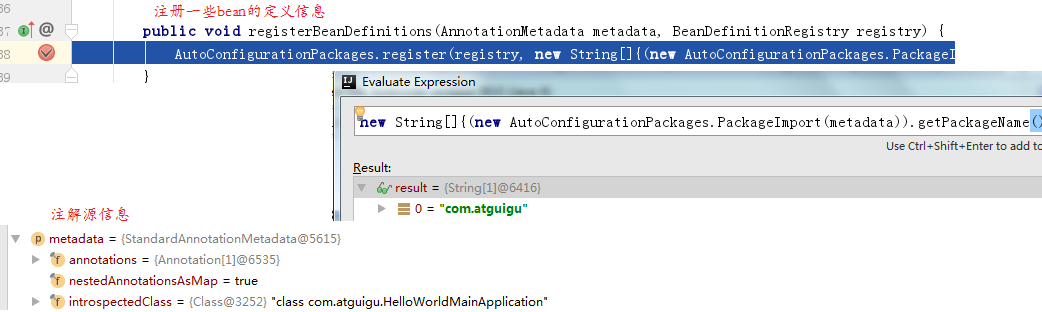
### @AutoConfigurationPackage

@AutoConfigurationPackage：自动配置包



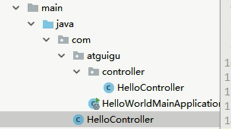
@Import(Registrar.class)：

Spring的底层注解@Import，给容器中导入一个组件；导入的组件由AutoConfigurationPackages.Registrar.class；断点



将主配置类（@SpringBootApplication标注的类）的所在包及下面所有子包里面的所有组件扫描到Spring容器

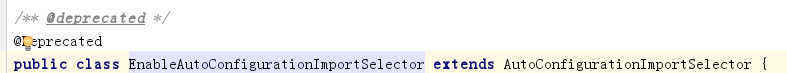
假如，不放在配置的包下面，就会报错



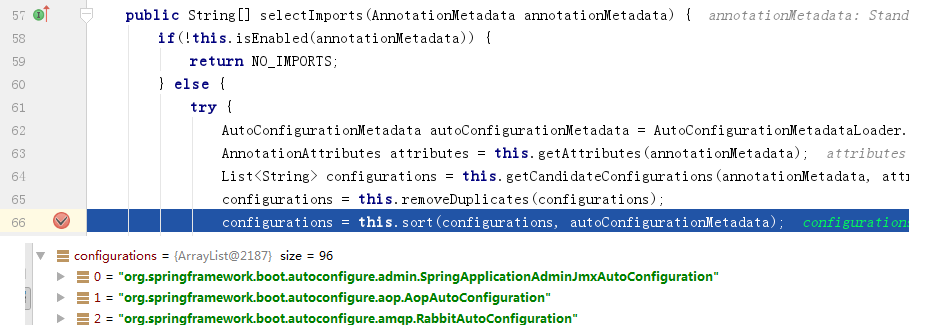
### @Import

@Import(EnableAutoConfigurationImportSelector.class)；给容器中导入组件？

EnableAutoConfigurationImportSelector：导入哪些组件的选择器；

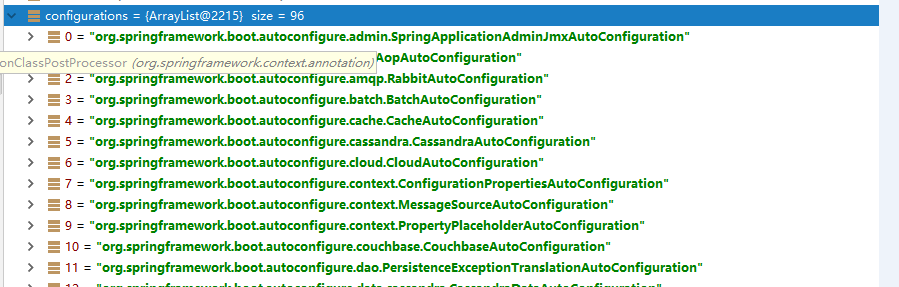


进入AutoConfigurationImportSelector，断点



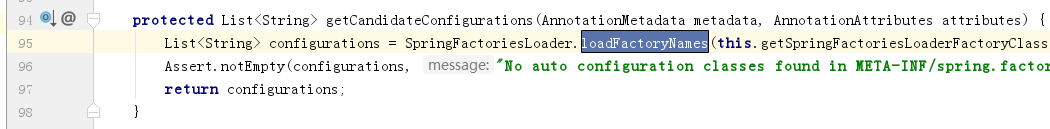
将所有需要导入的组件以全类名的方式返回；这些组件就会被添加到容器中；

会给容器中导入非常多的自动配置类（xxxAutoConfiguration）；就是给容器中导入这个场景需要的所有组件，并配置好这些组件；

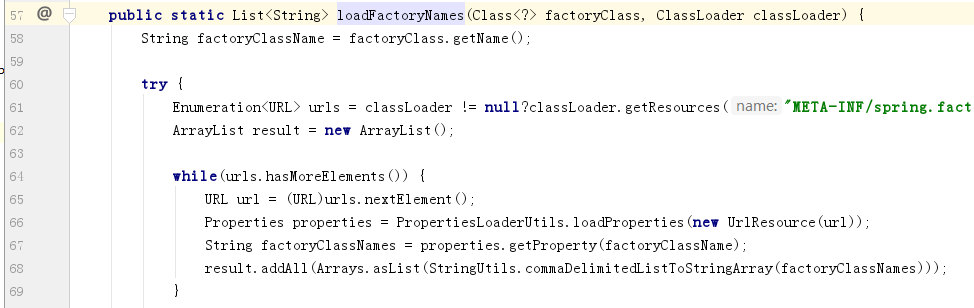


有了自动配置类，免去了我们手动编写配置注入功能组件等的工作；

找到方法loadFactoryNames；SpringFactoriesLoader.loadFactoryNames(EnableAutoConfiguration.class,classLoader)；



点击进入loadFactoryNames



Spring Boot在启动的时候从类路径下的META-INF/spring.factories中获取EnableAutoConfiguration指定的值，将这些值作为自动配置类导入到容器中，自动配置类就生效，帮我们进行自动配置工作；

以前我们需要自己配置的东西，自动配置类都帮我们；

J2EE的整体整合解决方案和自动配置都在spring-boot-autoconfigure-1.5.9.RELEASE.jar；