

Spring工厂加载机制

• 1. JDK SPI

SPI全称Service Provider Interface，是Java提供的一套用来被第三方实现或者扩展的接口，它可以用来启用框架扩展组件和替换组件，SPI的作用就是为这些被扩展的API寻找服务实现，简单来说就是可以使用SPI来解耦，实现插件的自由插拔。

• 2. Spring SPI

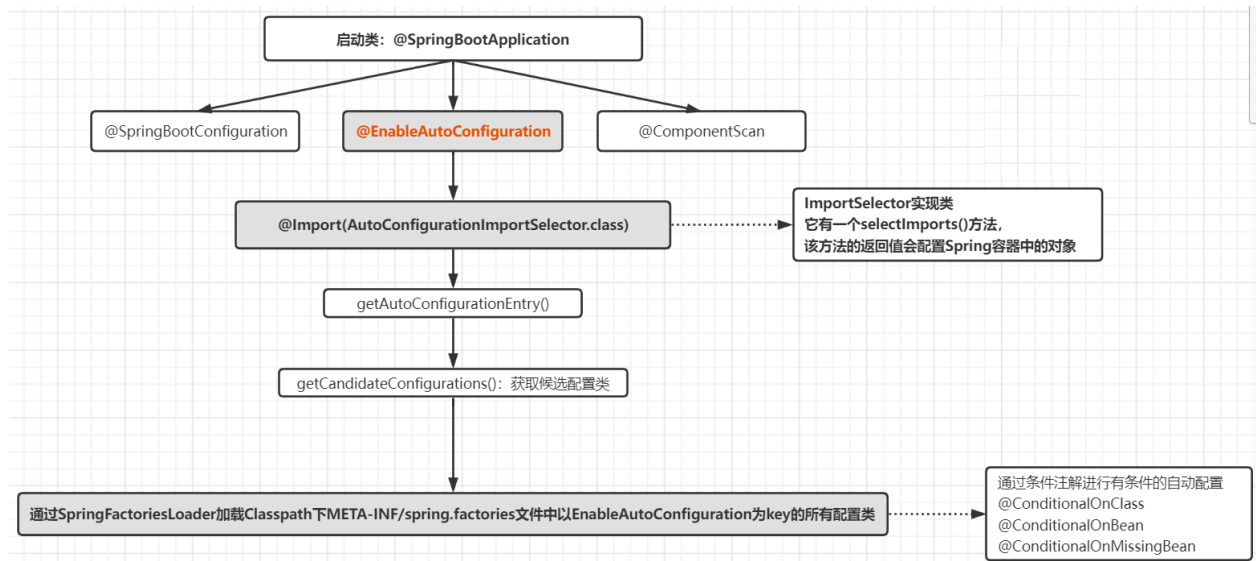
Spring框架从3.2版本开始提供了 `SpringFactoriesLoader` 工厂加载机制，类似于Java提供的SPI机制，Spring利用这种机制实现了Spring Bean的自动配置。

• 3. `SpringFactoriesLoader` 工厂的加载机制的默认约定

1. `SpringFactoriesLoader`是Spring框架内部的通用工厂加载机制（Spring3.2 之后提供的）
2. 它负责加载classpath下的 `META-INF/spring.factories` 文件，并实例化文件中的类型
3. 文件内容格式必须是Properties格式（k=v），
key必须是某个类型的全限定名称（包名+类名）
value可以是单个值，可以是使用逗号分隔的多个值
例如：`example.MyService=example.MyService1,example.MyService2`
key和value之间可以没有继承关系

• 4. Spring Boot的自动配置

1. Spring Boot自动配置主要是通过Spring工厂加载机制实现的；
2. 启动类上的`@SpringBootApplication`注解包含了三个注解：`@SpringBootConfiguration`、`@EnableAutoConfiguration`、`@ComponentScan`，其中最重要的注解就是`@EnableAutoConfiguration`；
3. `@EnableAutoConfiguration`注解导入一个 `AutoConfigurationImportSelector` 类型，这个类型实现了 `ImportSelector`接口 的 `selectImports()`方法，在这个方法中使用Spring工厂加载机制加载classpath下 `META-INF` 文件夹下的 `spring.factories` 文件中以`EnableAutoConfiguration`为key的所有Value值（自动配置类），对满足条件的类型利用Java Config方式进行自动配置；



• 5. Spring Boot 2.7.0 之后的自动配置实现方式

1. Spring Boot 2.7.0之后增加了加载Classpath下 `META-INF/spring` 文件夹下的 `org.springframework.boot.autoconfigure.AutoConfiguration.imports` 文件中的类型

