

48 Spring展望及后续(未完)

更新时间：2020-09-09 11:46:15



“

成功的奥秘在于目标的坚定。——迪斯雷利

”

Spring 历史与展望

Spring历史

时间回到 2002 年，当时正是 Java EE 和 EJB 大行其道的时候，很多知名公司都是采用此技术方案进行项目开发。这时候有一个美国的小伙子Rod Johnson认为 EJB 太过臃肿，并不是所有的项目都需要使用 EJB 这种大型框架，应该会有更好的方案来解决这个问题。



为了证明他的想法是正确的，于 2002 年 10 月甚至写了一本书《Expert One-on-One J2EE》，介绍了当时 Java 企业应用程序开发的情况，并指出了 Java EE 和 EJB 组件框架中存在的一些主要缺陷。在这本书中，他提出了一个基于普通 Java 类和依赖注入的更简单的解决方案。

在这本书发布后，一对一的 J2EE 设计和开发一炮而红。这本书免费提供的大部分基础架构代码都是高度可重用的。2003 年 Rod Johnson 和同伴在此框架的基础上开发了一个全新的框架命名为 Spring，据 Rod Johnson 介绍 Spring 是传统 J2EE 新的开始。随后 Spring 发展进入快车道。

- 2004 年 03 月，1.0 版发布。
- 2006 年 10 月，2.0 版发布。
- 2007 年 11 月发布了 Spring 2.5。
- 2009 年 12 月，Spring 3.0 发布。
- 2013 年 12 月，Pivotal 宣布发布 Spring 框架 4.0。
- 2017 年 09 月，Spring 5.0 发布。

Spring 5.x 新特性

Spring 与时俱进，充分吸收新的营养，加入自己的体系。Spring 的新特性主要如下：



总体上来说，Spring 5.x 的特性可以分为以下几类：

- 支持更高版本的 **JDK（8及以上）**：将不支持 JDK8 以下版本；
- **核心框架的修订**：由于 JDK8 反射的增强，Spring 5 支持有效获取 Method 的参数支持 @Nullable 和 @NotNull 注解提供了基于 Java 8 默认方法构建的选择性声明；
- **核心容器更新**：支持候选组件索引，可以替代类路径扫描。该支持已被添加到类路径扫描器的候选组件标识步骤的快捷方式中。GenericApplicationContext 和 AnnotationConfigApplicationContext 中实现函数式编程风格。对接口方法上的事务、缓存和异步注释的一致检测。XML 配置名称空间简化为无版本模式；
- 使用 **Kotlin** 进行函数式编程：引入了对 JetBrains Kotlin 语言的支持；
- 反应式编程模型。

展望：你精通Spring了吗？

通过前面的章节，我们已经对spring核心spring-core有了深入的了解。

- **IoC Container**：包括 BeanFactory，ApplicationContext 容器；

- **Events:** 事件;
- **Resources:** 资源使用;
- **i18n:** 国际化;
- **Validation:** 数据校验;
- **Data Binding:** 数据绑定;
- **Type Conversion:** 类型转换;
- **SpEL:** 表达式计算;
- **AOP:** 面向切面编程。

经过一番痛苦折磨，可能有同学会自我感觉到：我已经精通 Spring 了！我已经攀越到 Spring 的山顶。



不好意思的告诉你：一山还有一山高，你仅仅刚达到山脚下哦。

后续-致愿意继续攀登高峰的小伙伴们



“知道的越多，才发现自己不知道的更多”，当你越来越深入的时候，你会发现不知道的越来越多，对于愿意继续攀登高峰的小伙伴们，我将和大家一起，继续攀登 Spring5 的另外的山峰，期待和大家继续在一起：

实例揭秘spring5 mvc核心源码原理

实例揭秘spring-data-access核心源码原理

实例揭秘spring integration核心源码原理

实例揭秘spring5 测试Test 核心源码原理

实例揭秘spring5 和其它语言集成源码原理

最后的福利

不以规矩，不能成方圆

----《孟子》的《离娄章句上》

任何一门语言的问世和流传，以及取得这样大的影响力都离不开厂商、组织、开发者与用户们的共同参与，而 Sun 公司为了发展和更新这门语言组成了一个开放性国际组织**JCP**（Java Community Process），任何想要提议加入 Java 功能或特性都必须以**JSR**正式文件（Java Specification Request）进行提交然后经过 JCP 执行委员会投票，通过即成为最终标准文件，然后必须根据这个 JSR 做出免费且开发原始码的参考实现**RI**（Reference Implementation），并提供技术兼容性测试工具包**TCK**（Technology Compatibility Kit），厂商可以根据 JSR 实现产品。

现在 Java 无疑已经成为了业界共同制定的一个标准，每一个标准也代表着业界面临的一些问题，而一个**JSR**规范标准可以有多种技术解决方案。对于初学者的我们，学习 JSR 必然过于艰涩，但我们很有必要了解这个规范过程。

jcp官方地址: <https://jcp.org/>

Java EE 8技术对应的JSR标准

Java EE 8 基于Java EE 7。下面是 Java EE 8 在 Java EE7 之上的更新或者新增 JSR:

- JSR 366 – Java EE 8 Platform
- JSR 365 – Contexts and Dependency Injection (CDI) 2.0
- JSR 367 – The Java API for JSON Binding (JSON-B) 1.0
- JSR 369 – Java Servlet 4.0
- JSR 370 – Java API for RESTful Web Services (JAX-RS) 2.1
- JSR 372 – JavaServer Faces (JSF) 2.3
- JSR 374 – Java API for JSON Processing (JSON-P)1.1
- JSR 375 – Java EE Security API 1.0
- JSR 380 – Bean Validation 2.0
- JSR 250 – Common Annotations 1.3
- JSR 338 – Java Persistence 2.2
- JSR 356 – Java API for WebSocket 1.1
- JSR 919 – JavaMail 1.6

完整的信息可以参考<https://www.oracle.com/java/technologies/java-ee-glance.html>

}

