#### Spring脱坑记

##### 之Spring IoC：Principles

问题描述

Spring IoC

解决方案

**什么是IoC(控制反转)**？

控制反转并不是Spring所特有的，Spring为我们所做的更多的可以说是为我们创建并管理Bean，然后在我们需要的时候再依赖注入（Dependency Injection）给我们。这也是为什么后来Spring IoC更多又叫Spring DI。Spring IoC是一种思想，而DI依赖注入是这种思想的具体实现方案。

Web容器是如何启动的？

一般来讲，Web容器（Tomcat等）启动WebApplication会有以下步骤：

1. Web容器（Tomcat等）在启动Web应用程序的时候会为每个WebApplication创建唯一的ServletContext对象。可以把ServletContext看作一个Web应用的服务器端组件的共享内存。Web应用的所有部分都可以使用该上下文ServletContext。
2. Web容器将<context-param></context-param>解析为key-value对，并交给ServletContext。
3. Web容器根据<listener></listener>中的类创建ServletContextListener监听实例，即启动监听。
4. 在ServletContext初始化的时候Web容器会回调ServletContextListener监听类中contextInitialized(ServletContextEvent servletContextEvent)初始化方法。这里就是我们的SpringWebApplication的Spring工厂开始创建的地方。
5. Web容器解析<filters><filters>，并启动拦截器。拦截器开始起作用，当有请求进入时，执行Filter的doFilter方法。
6. 初始化Servlet等

Web容器启动WebApplication的时候主要做了2件事：

1. Web容器会帮我们创建启动XMLWebApplicationContext工厂
2. 将我们配置的contextConfigLocation（即spring-application.xml）设置给我们的Spring工厂ApplicationContext。

ApplicationContext是Spring提供的一个高级的IoC容器，它除了能够提供IoC容器的基本功能外，还为用户提供了国际化等附加服务。

**Spring IoC容器有哪些特点**？

Import

Alias（别名）

Bean

Beans

FactoryBean 如果要获取FactoryBean对象，需要在bean的id或者name前面加一个&符号来获取

**项目中是如何使用Spring IoC容器的**？

项目中使用私有SpringBeanFactory代理Spring框架的ApplicationContext工厂类。

ModuleInitializer.java

private void createApplicationContext() {  
 SpringBeanFactory.*setApplicationContext*(ctx);  
}

cleanMetedataToolView = (CleanMetedataToolView) SpringBeanFactory.*getSpringBean*("cleanMetedataToolView");

□