Spring的ApplicationContext接口介绍

# ApplicationContext应用上下文

ApplicationContext翻译为 **应用上下文。**

Spring框架主要是一个容器框架，主要功能之一就是管理bean。

ApplicationContext对象对应的就是Spring的配置文件，配置文件的默认路径为**WEB-INF/applicationContext.xml**文件；如何在web.xml中配置spring的配置文件见单独文档介绍。

代码加载**applicationContext.xml**文件方式：

**ApplicationContext ac = new ClassPathXmlApplicationContext(“path”);**

下面就是利用ac来获取想要的bean。

对于配置为singleton的bean，这种方式在生成ac的时候就已经创建了bean对象。

但是对于，工厂方式的，singleton的bean是延迟加载的，在获取bean的时候才创建bean。

# ApplicationContext是Spring容器最常用的接口。

其有很多父接口，也有很多子接口，也有很多实现类。

父接口如**BeanFactory**。

有时候也可以利用**BeanFactory**代表**Spring容器**。不过一般都建议使用**ApplicationContext作为Spring容器**，（又称为**Spring应用上下文**）。

实现类主要由两个：**ClassPathXmlApplicationContext**和**FileSystemXmlApplicationContext**类。

两者的区别很明显：

* 前者是从类类加载路径下搜索配置文件，并根据配置文件创建Spring容器，
* 后者是根据文件系统的相对路径或绝对路径下去搜索配置文件，并根据配置文件来创建Spring容器。对于Java项目而言，类加载路径是稳定的，显然使用前者更好。

# 从Spring容器中获取Bean对象。

ApplicationContext ac =new ClassPathXmlApplicationContext("config/SpringAop.xml");

**ac.getBean(“”); //获取Bean对象**

## Object getBean（String id）：需要进行强制类型转换。

## T getBean（String name，Class<T> requiredType）:带有泛型参数，无须强制类型转换。