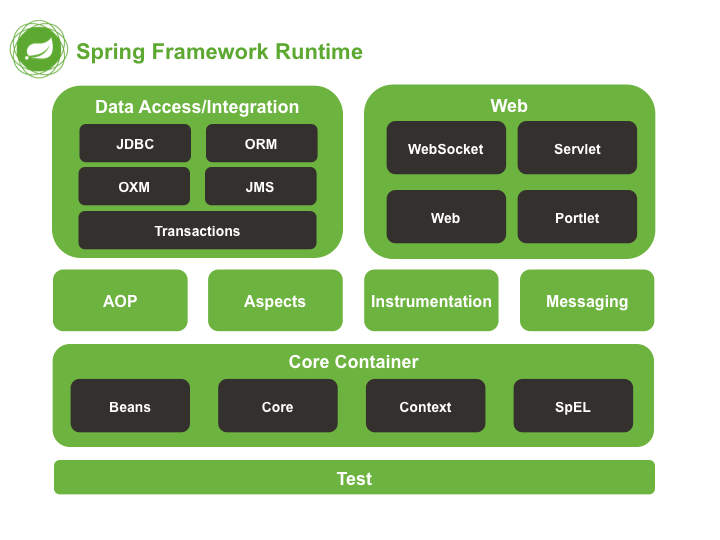
# spring源码分析

## 指南

记录一下这段时间看<spring源码深度解析>的记录吧

本次学习基于spring3.2来进行记录

## spring整体架构



### core container

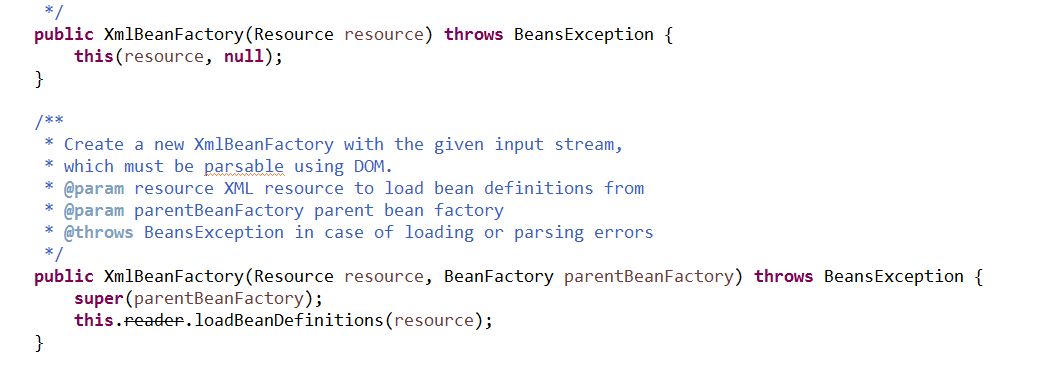
core container(核心容器)包含有core、beans、context和Spring Expression Language(spel)。

core和beans模块是框架的基础部分，提供IOC（Inversion of Control控制反转）和依赖注入等特性，这里的基础概念是beanfactory，它提供对factory模式的经典实现来消除对程序性单例模式的需要，并真正允许你从程序逻辑中分离出依赖关系和配置。

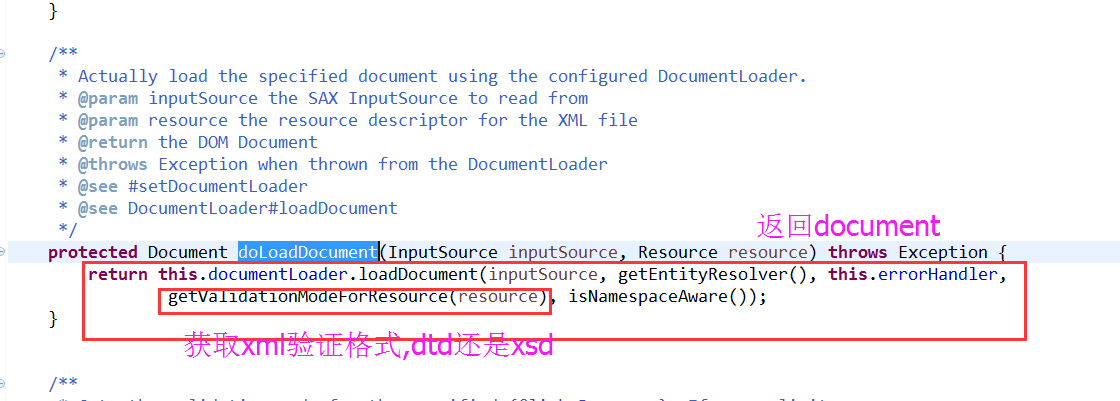
## 加载xml

BeanFactory bf =**new** XmlBeanFactory(new ClassPathResource("spring-mvc.xml"));

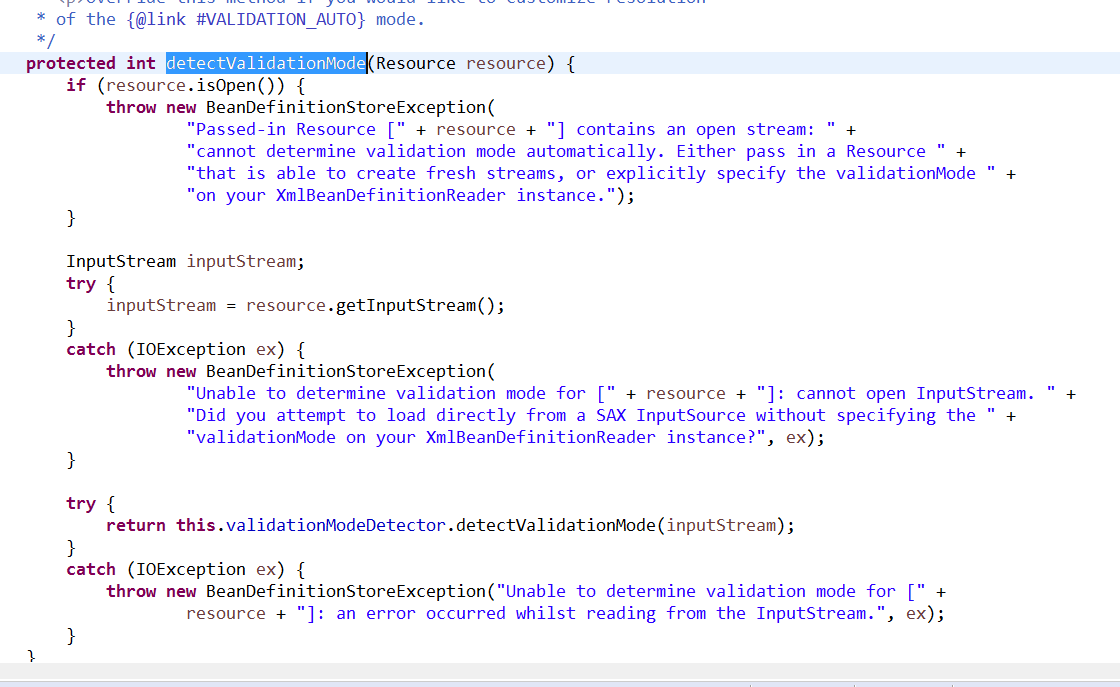
### 判断xml验证模式

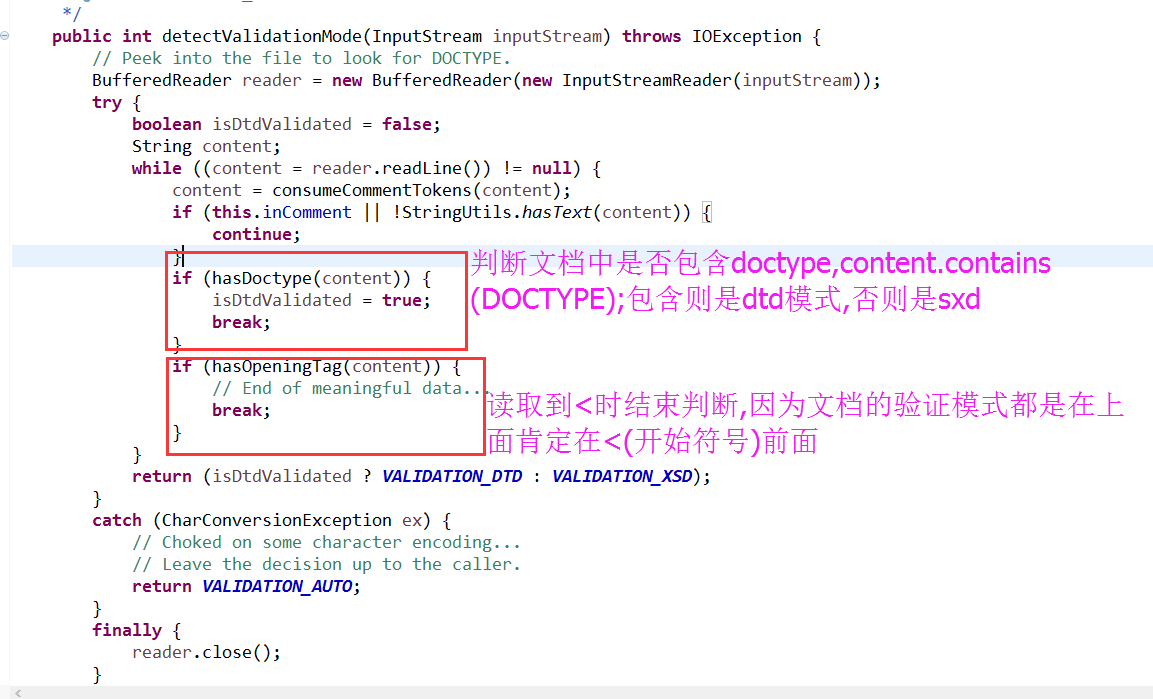




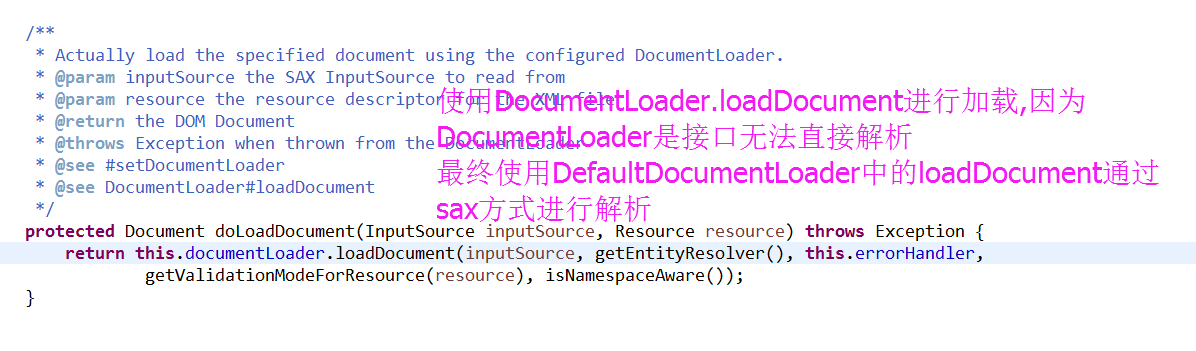


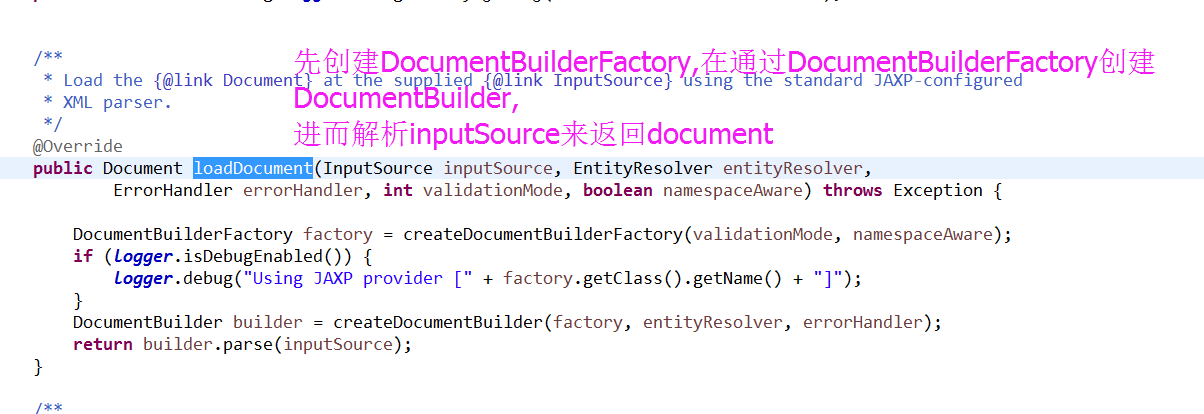






### 获取document





上图中需要**EntityResolver参数**

**何为** **EntityResolver :**

官方解释: 如果ＳＡＸ应用程序叙事实现自定义处理外部实体,则必须实现此接口,

并使用setEntityResolver方法向SAX 驱动器注册一个实例.

也就是说,对于解析一个xml,sax

首先会读取该xml文档上的声明,根据声明去寻找相应的dtd定义,以便对文档的进行验证,  
默认的寻找规则,(即:通过网络,实现上就是声明DTD的地址URI地址来下载DTD声明),  
并进行认证,下载的过程是一个漫长的过程,而且当网络不可用时,这里会报错,就是应为相应的dtd没找到,  
  
 **EntityResolver 的作用就是项目本身就可以提供一个如何寻找DtD 的声明方法**,

即:由程序来实现寻找DTD声明的过程,比如我们将ＤＴＤ放在项目的某处在实现时直接将此文档读取并返回个SAX即可,这样就避免了通过网络来寻找DTD的声明  
  
**首先看看EntityResolver 接口声明的方法.**

public abstract InputSource resolveEntity (String publicId,

String systemId)

throws SAXException, IOException;

这里,他接收2个参数,publicId ,systemId ,并返回一个InputStream对象,  
这里我们以特定的文件来讲解;

**如果我们在解析验证模式为xsd的配置文件,代码如下**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

...

</beans>

读取得到以下参数,  
 publicId : null  
 systemId : http://www.springframework.org/schema/beans/spring-beans-2.5.xsd

**如果我们解析的是DTD的配置文件;**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

...

</beans>

读取得到以下参数:  
publicId : -//SPRING//DTD BEAN//EN  
systemId : http://www.springframework.org/dtd/spring-beans.dtd

之前已经提到过,默认的寻找规则,(即:通过网络,实现上就是声明DTD的地址URI地址来下载DTD声明),这样才会造成延迟,用户体验也不好,一般的做法是将验证文件放在自己的工程里面,

那么怎么做才能将这个Url转换为自己工程里对应的文件呢?我们以加载DTD文件为例看看Spring中是如何通过getEntityResolver() 对  EntityResolver 的获取,

**DelegatingEntityResolver**

我们知道Spring中使用DelegatingEntityResolver 类为 EntityResolver的实现类,

 EntityResolver 的实现方法如下

