

```
BeanFactory beanFactory = new XmlBeanFactory(new ClassPathResource("beanFactoryTest.xml"));
MyTestBean myTestBean = (MyTestBean) beanFactory.getBean(s: "myTestBean");
```

```
public ClassPathResource(String path, ClassLoader classLoader) {
    Assert.notNull(path, message: "Path must not be null");
    String pathToUse = StringUtils.cleanPath(path);
    if (pathToUse.startsWith("/")) {
        pathToUse = pathToUse.substring(1);
    }

    this.path = pathToUse;
    this.classLoader = classLoader != null ? classLoader : ClassUtils.getDefaultClassLoader();
}
```

路径整理

```
public int loadBeanDefinitions(Resource resource) throws BeanDefinitionStoreException {
    return this.loadBeanDefinitions(new EncodedResource(resource));
}
```

资源文件编码

bean加载核心方法

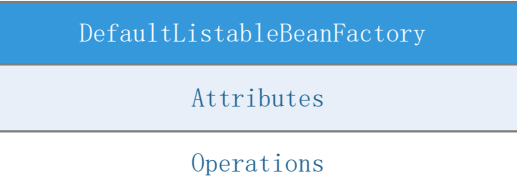
```
public int loadBeanDefinitions(EncodedResource encodedResource) throws BeanDefinitionStoreException {
```

```
InputStream inputStream = encodedResource.getResource().getInputStream();

try {
    org.xml.sax.InputSource 接收资源文件为流对象
    InputSource inputSource = new InputSource(inputStream);
    if (encodedResource.getEncoding() != null) {
        inputSource.setEncoding(encodedResource.getEncoding());
    }

    文件处理完毕，开始bean加载
    var5 = this.doLoadBeanDefinitions(inputSource, encodedResource.getResource());
}
```

XmlBeanFactory简单实现Default
并加入自己的XML读取Reader



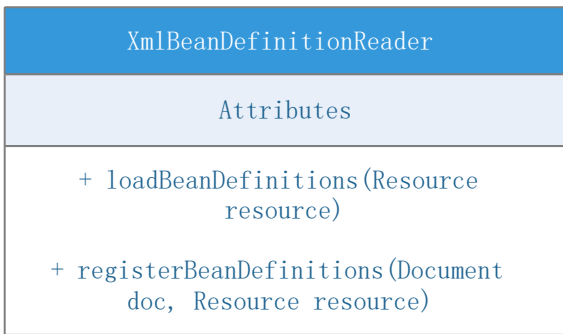
bean加载核心

XmlValidationModeDetector
格式发现器

extends

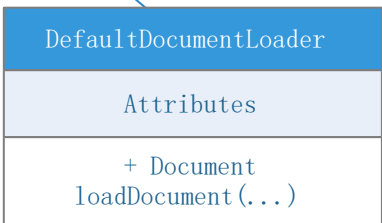


XML读取个性化
BeanDefinitionReader



Xml文件加载
bean的核心

受委托



```
int validationMode = this.getValidationModeForResource(resource);
Document doc = this.documentLoader.loadDocument(inputSource, this.getValidationModeForResource(resource));
return this.registerBeanDefinitions(doc, resource);
```

重头戏：注册bean

- 1) 获取对XML文件验证模式：（DTD还是XSD）
- 2) 加载XML文件，并得到相应的Document；
- 3) 根据返回的Document注册Bean信息。

```
public int registerBeanDefinitions(Document doc, Resource resource) throws BeanDefinitionStoreException {
    BeanDefinitionDocumentReader documentReader = this.createBeanDefinitionDocumentReader();
    documentReader.setEnvironment(this.getEnvironment());
    int countBefore = this.getRegistry().getBeanDefinitionCount();
    documentReader.registerBeanDefinitions(doc, this.createReaderContext(resource));
    return this.getRegistry().getBeanDefinitionCount() - countBefore;
}
```

doc是通过上一步的loadDocument获取到的；
this.createBeanDefinitionDocumentReader() 其实已经将documentReader实例化了DefaultBeanDefinitionDocumentReader的对象；
DefaultBeanDefinitionDocumentReader的目的之一就是提取root；

```
if (delegate.isDefaultNamespace(ele)) {
    this.parseDefaultElement(ele, delegate);
} else {
    delegate.parseCustomElement(ele);
}
```