**Resource（接口）**

*从底层资源的实际类型（例如文件或类路径资源）中****抽象****出来的资源描述符的接口，本质是字符串*

**BeanDefinitionReader（接口）**

*bean 定义读取器的简单接口。使用资源Resource和字符串参数****指定加载方法****，用于读取配置信息，本质是接口，一般通过向上转型进行子类实例化实现自身功能*

**AbstractBeanDefinitionReader(抽象类)**

*和BeanDefinitionReader相似（是它的子类），可通过向上转型进行子类实例化实现自身功能，同时继承了EnvironmentCapable接口，获得了更多功能*

*以下是AbstractBeanDefinitionReader所拥有的功能*

*1、根据传入的bean factory 创建 AbstractBeanDefinitionReader*

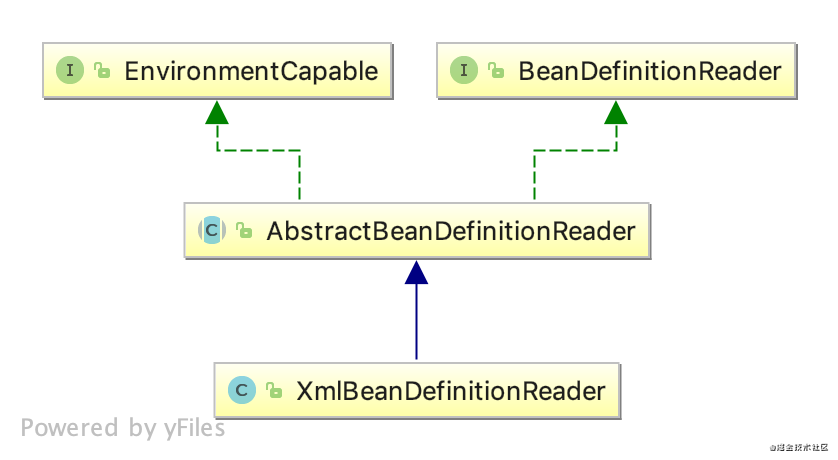
*将 registry 参数赋值给 AbstractBeanDefinitionReader 的成员变量 private final BeanDefinitionRegistry registry;。*

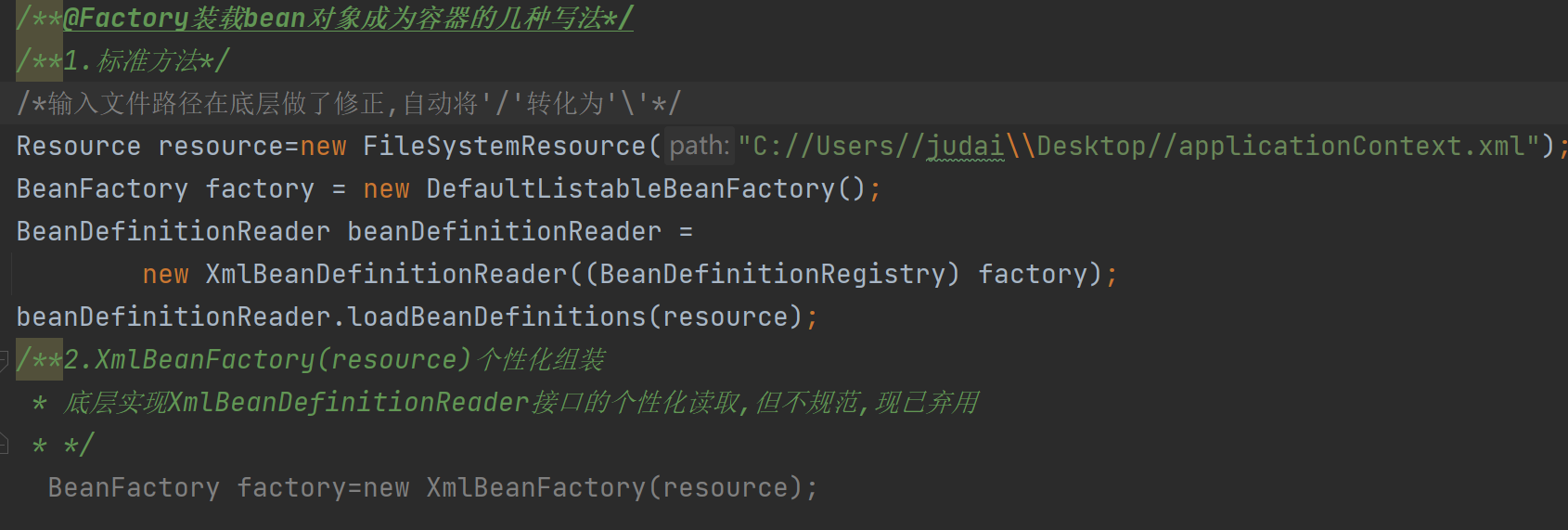
*2、判断传入的 bean factory 是否实现了 ResourceLoader 接口，如果 bean 工厂实现了 ResourceLoader 接口，则将它赋值给成员变量*

*private ResourceLoader resourceLoader;，否则将 resourceLoader 初始化为 PathMatchingResourcePatternResolver 对象。*

*3、判断传入的 bena factory 是否实现了 EnvironmentCapable 接口，如果 bean 工厂实现了 EnvironmentCapable 接口，则将它赋值给成员变量*

*private Environment environment;，否则将 environment 初始化为 StandardEnvironment 对象。*



**从微观上梳理它的核心组件**

**（；´д｀）ゞ分析BeanFactory在本例中的创建流程：**

1. **Resource resource=new FileSystemResource()…实例化类传入resource路径字符串参数**
2. **BeanFactory factory=new DefaultListableBeanFactory()…创建BeanFactory容器的大致框架**
3. **BeanDefinitionReader reader=new XmlBeanDfinitonReader((BeanDificitionRegistry)factory)…底层使用super修改父类属性以完善父类框架（就是BeanFactory）**
4. **reader.loadBeanDefinitions(resource)…将配置信息字符串resource对应的配置信息导入BeanDefinitionReader(再借由super将配置信息绑定给父类BeanFactory)**

## 深入解读资源（Resource）的创建

### ClassPathResource

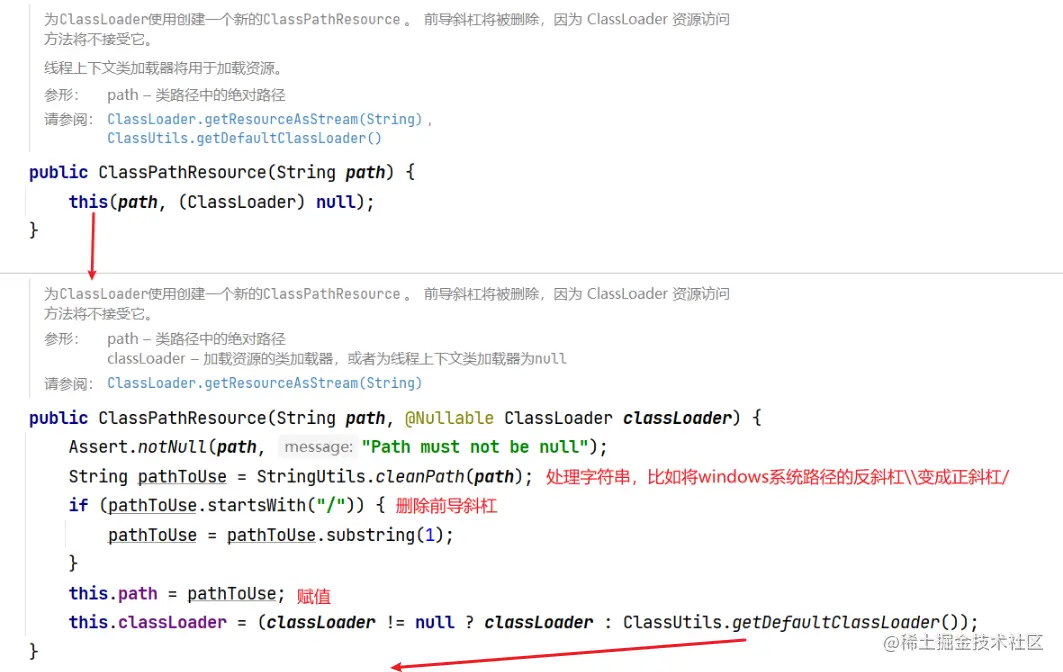
功能说明



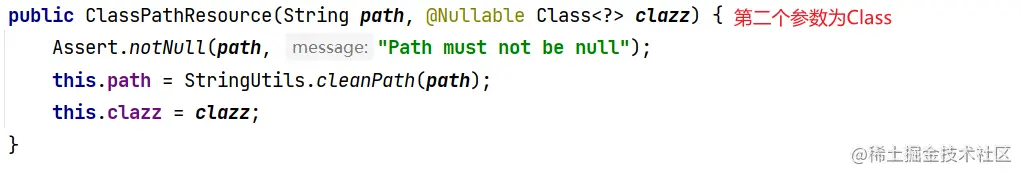
类路径资源的Resource的实现。 使用给定的ClassLoader***或***给定的Class***来加载资源***。

成员变量

功能说明已经指出：使用给定的ClassLoader**或**给定的Class**来加载资源**\*，所以有这些成员变量。

**构造方法是重点**

****



**小结**

* + path需要删除前导斜杠。
  + 成员变量Class和ClassLoader选其一即可。
  + **ClassPathResource类的源码看下来，其实它本质就是想调用ClassLoader类提供的getResource()也可以是getResourceAsStream(),又或者是，Class类提供的getResourceAsStream(),但是本质还是基于ClassLoader类提供的getResouce();**
  + 只要得到ClassLoader，差不多就可以得到Resource了。
  + **Resource其实就是URL。**