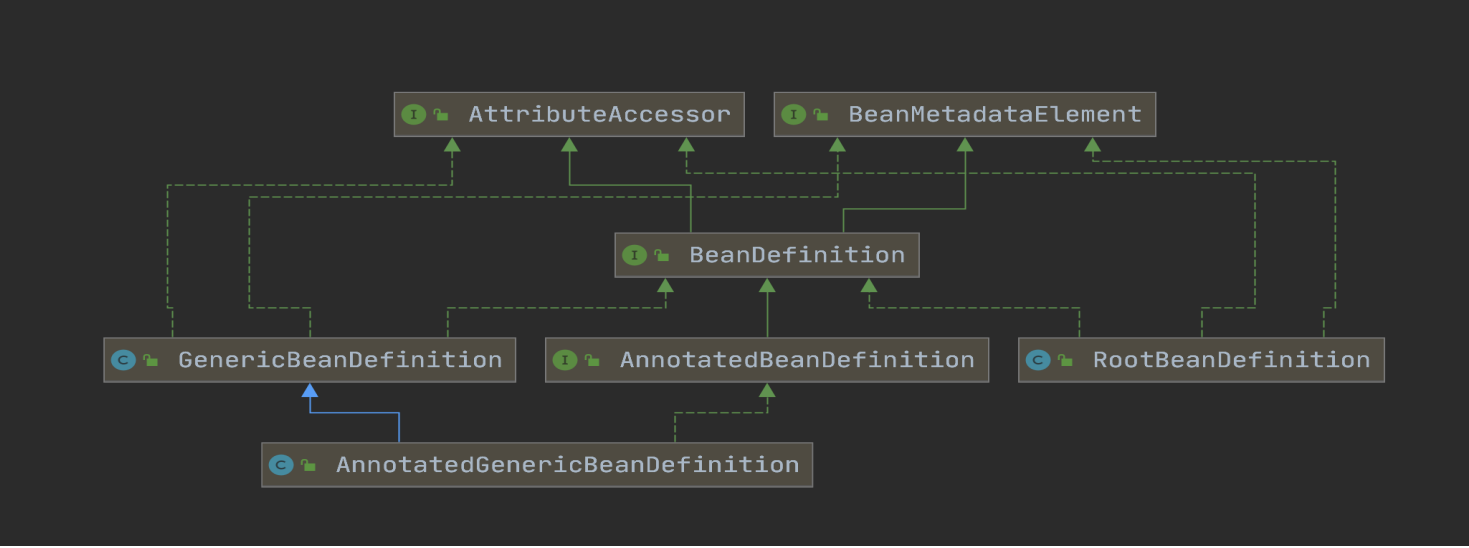


注解完成

```
@Component[ @Configuration @Service @Repostiiry @Contrller]
User{
}
```



当我们获得BD后，还是有机会，影响

Spring对于这个对象的Set注入 和 构造注入 非常重要的扩展点 BeanFactoryPostProcessor

```
MutablePropertyValues propertyValues = beanDefinition.getPropertyValues();
propertyValues.add( propertyName: "password", propertyValue: "99999");

ConstructorArgumentValues values = beanDefinition.getConstructorArgumentValues();
values.addArgumentValues( other: null);
```

```
<bean id="" class=""
<property name="name" value="12345"/> -----> PropertyValues
<constructor-arg> -----> ContrcutorArgumentValues
```

spring为什么要设计BDHolder 操作方便 传参方便

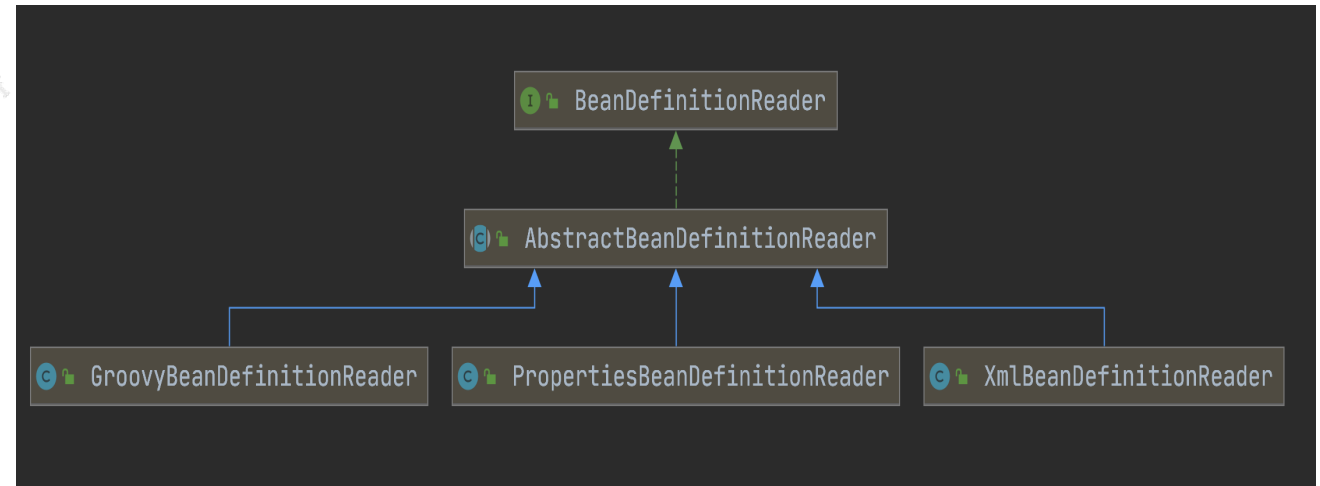
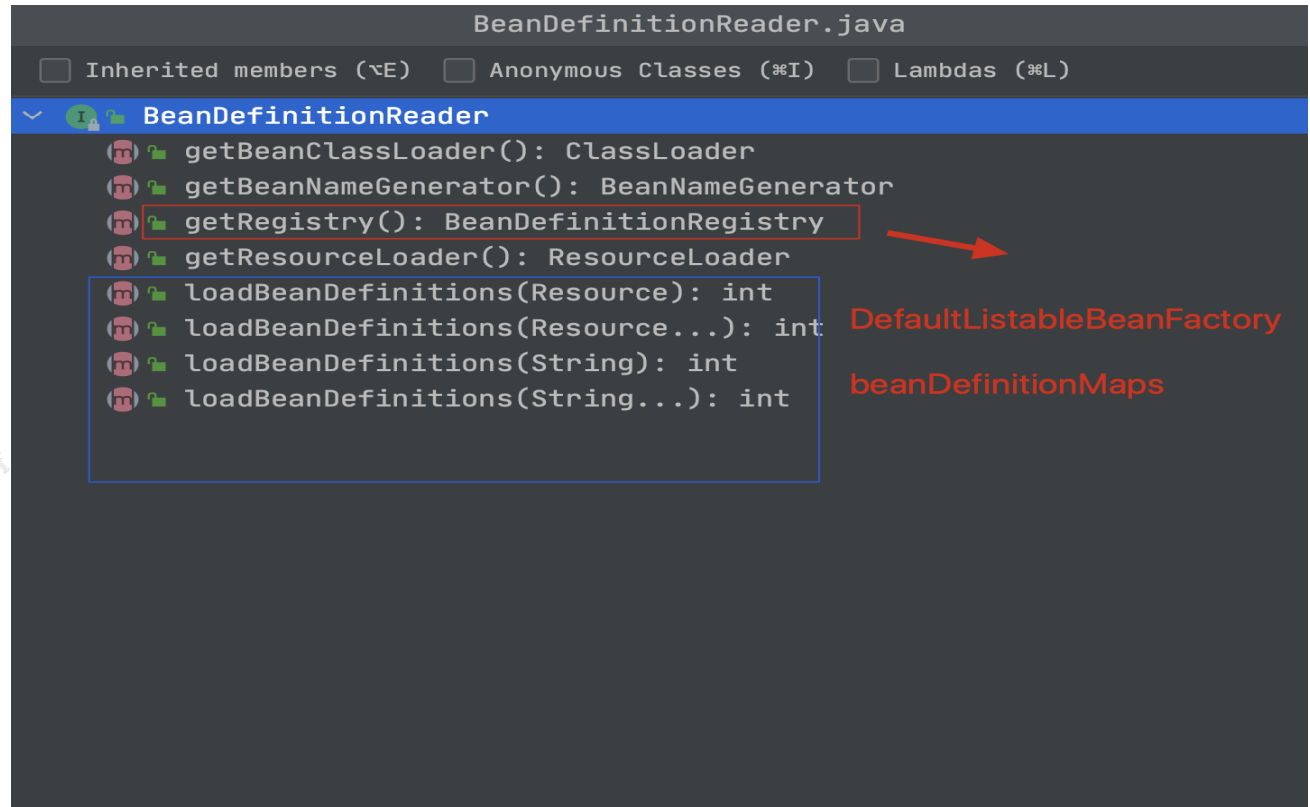
DB.getBeanName()---->

beanName DB

AnnotationGenericBeanDefinition 注解

RootBeanDefintion XML Spring 目前建议 废除 RootBeanDefintion 不支持父子bean

GenericBeanDefinition



1 结合Spring的实际使用情况 XmlBeanDefinitionReader

```
BeanDefinitionHolder bdHolder = delegate.parseBeanDefinitionElement(ele);
BeanDefinitionReaderUtils.registerBeanDefinition(bdHolder, getReaderContext().getRegistry());
```

2 疑问 基于注解的配置形式

```
AnnotatedBeanDefinitionReader
AnnotatedBeanDefinitionReader(BeanDefinitionRegistry registry)
doRegisterBean
1. 创建BeanDefinition
2. registry 注册
```

后续的开发过程中，往往所提供的操作 可能就叫做 Register 但是 内部 一定包含

1. BD创建 （3种方式）

2. DefaultListableBeanFactory.register ----> beanDefinitionMap