

## 一、标签处理

因为都是通过spring的标签去解析dubbo配置文件，所以肯定有一个实现类是实现了spring的BeanDefinitionParser接口。

DubboDefinitionParser实现了BeanDefinitionParser接口，用来解析xml中的配置。

其实就是将dubbo-provider.xml文件中的标签解析成对应的config类，加载为springbean管理。那怎么处理标签对应的parser之间的映射关系呢？  
dubbo维护了一个DubboNameSpaceHandler 其实就是把对应的标签和对应的BeanDefinitionParser对应起来。

BeanDefinitionParser每次初始化时，都会穿进去要处理的config类和required属性。

```
@Override
public void init() {
    registerBeanDefinitionParser( elementName: "application", new DubboBeanDefinitionParser(ApplicationConfig.class, required: true)
    registerBeanDefinitionParser( elementName: "module", new DubboBeanDefinitionParser(ModuleConfig.class, required: true)
    registerBeanDefinitionParser( elementName: "registry", new DubboBeanDefinitionParser(RegistryConfig.class, required: true)
    registerBeanDefinitionParser( elementName: "monitor", new DubboBeanDefinitionParser(MonitorConfig.class, required: true)
    registerBeanDefinitionParser( elementName: "provider", new DubboBeanDefinitionParser(ProviderConfig.class, required: true)
    registerBeanDefinitionParser( elementName: "consumer", new DubboBeanDefinitionParser(ConsumerConfig.class, required: true)
    registerBeanDefinitionParser( elementName: "protocol", new DubboBeanDefinitionParser(ProtocolConfig.class, required: true)
    registerBeanDefinitionParser( elementName: "service", new DubboBeanDefinitionParser(ServiceBean.class, required: true)
    registerBeanDefinitionParser( elementName: "reference", new DubboBeanDefinitionParser(ReferenceBean.class, required: true)
    registerBeanDefinitionParser( elementName: "annotation", new AnnotationBeanDefinitionParser());
}
```

## 二、服务暴露过程