

Java中关于XML的API惊鸿一瞥 PDF转换可能丢失图片或格式
， 建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Java_E4_B8_AD_E5_85_B3_c104_144530.htm 简单介绍一下Java关于xml的API，这样大家看到了缩写就知道是干什么的了。水平有限，多多包涵。 1、JAXP(Java API for XML Parsing) 2、JAXB(Java API for XML Binding) 3、JAXM(Java API for XML Messaging) 4、JAX-RPC (Java API for XML-RPC) 1、JAXP定义了在使用DOM, SAX, XSLT的通用的接口。这样在你的程序中你只要使用这些通用的接口，当你需要改变具体的实现时候也不需要修改代码。比如，你用的XSLT处理器太慢了，你想换一个，你不需要修改你以前的代码，只要修改一下JAXP的相关配置。（在后面我将详细地介绍）作为一个共同的接口，JAXP也有所谓的“最小公分母”效应，也就是说它支持的东西很有限。JAXP1.0支持XML1.0，XML Namespace1.0，SAX1.0以及DOM level 1。而JAXP1.1增加了对SAX2.0，DOM level 2以及XSLT1.0的支持。很明显如果你想使用Xalan的XPath相关的接口，JAXP就没有支持，你也只能将代码绑定到特定的Xalan的API上了。这里还要提一下JDOM，虽然它没有实现JAXP，但是由于它使用的简单性，还是很受欢迎，并且成为了JCP正式推荐的API。它也是一种树状的结构表现XML，在使用方法上要比w3c的dom标准简单易用的多。最新版本的JDOM在其内部已经开始使用JAXP的API，它会尽可能的去调用JAXP的API，如果不行就使用自己的默认XML解析器Xerces，XSLT处理器Xalan。 2、JAXB定义了Java数据对象和xml结构之间的一种双向映射关系。这样你就可以

很方便地将一个Java对象存储为一个xml文档，也可以从一个xml文档实例化一个Java对象。它的结构是这样子的：首先要有xml的dtd以及binding schema（这个不是xml的schema，而是一个定义Java对象和xml结构之间映射关系xml文档），通过这两个文件JAXB就可以生成与xml文档结构一致的Java源文件，编译之后就可以很方便地通过具体的xml文档得到与xml结构一致的Java类（就是生成的那些类）unmarshalling，反过来marshalling也可以。它的缺点也很明显，一旦xml的结构发生了改变，就要重新写binding schema以及重新生成编译Java类。sun的动作总是一如既往地慢，在JAXB出台之前已经有了一些用于xml data binding的框架，我们再来看看同样也是做xml databinding但是并没有实现JAXB的框架：一、Castor Castor不仅仅支持对XML的绑定，它还支持对LDAP对象，用OQL将SQL查询映射为对象，以及对JDO的支持。与JAXB不同的是，它需要的仅仅是xml的Schema。通过xml的Schema来生成相应的Java源代码，编译之后就可以marshalling和unmarshalling了。二、Zeus Zeus与Castor和JAXB相比，在class generation方面多做了些步骤，因此它可以支持多种的约束关系，包括对DTD，XML Schema以及TREX等等的支持。不过目前该项目好像已经不做了。三、Quick Quick也是一个非常灵活的框架，详细的情况可以google一下。3、JAXM JAXM是为SOAP通信提供访问方法和传输机制的API。目前它支持SOAP1.1规范以及同步和异步通信。JAXM定义了大量服务，JAXM的实现产品将会提供这些服务，使得开发者不用面对复杂的通信系统。JAXM体系结构中包括两个重要的组件：JAXM Client和Provider。Client通常是作为J2EE web或EJB容

器的一部分，以提供你所写的程序访问JAXM服务的能力。而Provider可以以不同的方式实现，主要负责发送和接收SOAP消息。这样你就可以直接地使用JAXM的API直接发送和接收SOAP消息。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com