****一 什么是webservice(用你的话描述webservice)?在什么时候用webservice（webservice能给我们解决什么样的问题）？****

一句话概括：****WebService是一种跨编程语言和跨操作系统平台的远程调用技术。****

所谓跨编程语言和跨操作平台，就是说服务端程序采用**[Java](http://lib.csdn.net/base/17" \o "Java EE知识库" \t "https://blog.csdn.net/qq_33326449/article/details/_blank)**编写，客户端程序则可以采用其他编程语言编写，反之亦然！跨操作系统平台则是指服务端程序和客户端程序可以在不同的操作系统上运行。

所谓远程调用，就是一台计算机a上的一个程序可以调用到另外一台计算机b上的一个对象的方法。譬如从天气预报系统中获取某个城市的天气数据在自己系统中进行展示；从证券交易系统中获取某只股票的交易信息在自己的系统中进行展示；又譬如一个商城系统中能够展示快递的跟踪信息，而这些信息就是通过webservice从具体的快递公司的系统中获取的数据。

其实可以从多个角度来理解WebService，从表面上看，WebService就是一个应用程序向外界暴露出一个能通过Web进行调用的API，也就是说能用编程的方法通过Web来调用这个应用程序。我们把调用这个WebService的应用程序叫做客户端，而把提供这个WebService的应用程序叫做服务端。从深层次看，WebService是建立可互操作的分布式应用程序的新平台，是一个平台，是一套标准。它定义了应用程序如何在Web上实现互操作性，你可以用任何你喜欢的语言，在任何你喜欢的平台上写Web service ，只要我们可以通过Web service标准对这些服务进行查询和访问。

****二．WSDL是什么，有什么作用？****

WSDL是web service definition language的缩写，即web service的定义（描述）语言。

怎样向别人介绍你的 web service 有什么功能，以及每个[函数](http://baike.baidu.com/view/15061.htm" \t "https://blog.csdn.net/qq_33326449/article/details/_blank)调用时的参数呢？你可能会自己写一套文档，你甚至可能会口头上告诉需要使用你的web service的人。这些非正式的方法至少都有一个严重的问题：当程序员坐到电脑前，想要使用你的web service的时候，他们的工具（如Visual Studio）无法给他们提供任何帮助，因为这些工具根本就不了解你的web service。解决方法是：用机器能阅读的方式提供一个正式的描述文档。web service描述语言(WSDL)就是这样一个基于XML的语言，用于描述web service及其[函数](http://baike.baidu.com/view/15061.htm" \t "https://blog.csdn.net/qq_33326449/article/details/_blank)、参数和返回值。因为是基于XML的，所以WSDL既是机器可阅读的，又是人可阅读的，这将是一个很大的好处。一些最新的开发工具既能根据你的web service生成WSDL文档，又能导入WSDL文档，生成调用相应web service的代码。

Webservice服务发布之后，通过浏览器访问发布的+?wsdl即可获得wsdl文档。

****三．WSDL文档主要有那几部分组成，分别有什么作用？****

一个WSDL文档的根元素是definitions元素，WSDL文档包含7个重要的元素：types, import, message, portType, operations, binding和service元素。

1、 definitions元素中一般包括若干个XML命名空间；

2、 Types元素用作一个容器，定义了自定义的特殊数据类型，在声明消息部分（有效负载）的时候，messages定义使用了types元素中定义的数据类型与元素；

3、 Import元素可以让当前的文档使用其他WSDL文档中指定命名空间中的定义；

4、 Message元素描述了Web服务的有效负载。相当于函数调用中的参数和返回值；

5、 PortType元素定义了Web服务的抽象接口，它可以由一个或者多个operation元素，每个operation元素定义了一个RPC样式或者文档样式的Web服务方法；

6、 Operation元素要用一个或者多个messages消息来定义它的输入、输出以及错误；

7、 Binding元素将一个抽象的portType映射到一组具体的协议（SOAP或者HTTP）、消息传递样式（RPC或者document）以及编码样式（literal或者SOAP encoding）；

8、 Service元素包含一个或者多个Port元素  
  
每一个Port元素对应一个不同的Web服务，port将一个URL赋予一个特定的binding，通过location实现。  
可以使两个或者多个port元素将不同的URL赋给相同的binding。

****四．SOAP是什么？****

  SOAP是simple object access protocal的缩写，即简单对象访问协议。 是基于XML和HTTP的一种通信协议。是webservice所使用的一种传输协议，webservice之所以能够做到跨语言和跨平台，主要是因为XML和HTTP都是独立于语言和平台的。Soap的消息分为请求消息和响应消息，一条SOAP消息就是一个普通的XML文档，包含下列元素：

1、 必需的 Envelope 元素，可把此XML文档标识为一条SOAP消息

2、 可选的 Header 元素，包含头部信息

3、 必需的 Body 元素，包含所有的调用和响应信息

4、 可选的 Fault 元素，提供有关在处理此消息所发生错误的信息

Soap请求消息

Soap响应消息

****五.怎么理解UDDI？****

UDDI是Universal Description Discovery and Integration的缩写，即统一描述、发现和整合规范。用来注册和查找服务，把web services收集和存储起来，这样当别人访问这些信息的时候就从UDDI中查找，看有没有这个信息存在。

****五.Webservice的SEI指什么？****

WebService EndPoint Interface（webservice终端[Server端]接口）

就是 WebService服务器端用来处理请求的接口

****六.说说你知道的webservice框架，他们都有什么特点？****

Webservice常用框架有JWS、Axis2、XFire以及CXF。

下面分别介绍一个这几种Web Service框架的基本概念

1、JWS是[Java](http://lib.csdn.net/base/17" \o "Java EE知识库" \t "https://blog.csdn.net/qq_33326449/article/details/_blank)语言对WebService服务的一种实现，用来开发和发布服务。而从服务本身的角度来看JWS服务是没有语言界限的。但是Java语言为Java开发者提供便捷发布和调用WebService服务的一种途径。  
2、Axis2是Apache下的一个重量级WebService框架，准确说它是一个Web Services / SOAP / WSDL 的引擎，是WebService框架的集大成者，它能不但能制作和发布WebService，而且可以生成Java和其他语言版WebService客户端和服务端代码。这是它的优势所在。但是，这也不可避免的导致了Axis2的复杂性，使用过的开发者都知道，它所依赖的包数量和大小都是很惊人的，打包部署发布都比较麻烦，不能很好的与现有应用整合为一体。但是如果你要开发Java之外别的语言客户端，Axis2提供的丰富工具将是你不二的选择。  
3、XFire是一个高性能的WebService框架，在Java6之前，它的知名度甚至超过了Apache的Axis2，XFire的优点是开发方便，与现有的Web整合很好，可以融为一体，并且开发也很方便。但是对Java之外的语言，没有提供相关的代码工具。XFire后来被Apache收购了，原因是它太优秀了，收购后，随着Java6 JWS的兴起，开源的WebService引擎已经不再被看好，渐渐的都败落了。  
4、CXF是Apache旗下一个重磅的SOA简易框架，它实现了ESB（企业服务总线）。CXF来自于XFire项目，经过改造后形成的，就像目前的Struts2来自WebWork一样。可以看出XFire的命运会和WebWork的命运一样，最终会淡出人们的视线。CXF不但是一个优秀的Web Services / SOAP / WSDL 引擎，也是一个不错的ESB总线，为SOA的实施提供了一种选择方案，当然他不是最好的，它仅仅实现了SOA**[架构](http://lib.csdn.net/base/16" \o "大型网站架构知识库" \t "https://blog.csdn.net/qq_33326449/article/details/_blank)**的一部分。  
注：对于Axis2与CXF之间的关系，一个是Axis2出现的时间较早，而CXF的追赶速度快。

如何抉择：   
1、如果应用程序需要多语言的支持，Axis2应当是首选了；  
2、如果应用程序是遵循 **[spring](http://lib.csdn.net/base/17" \o "Java EE知识库" \t "https://blog.csdn.net/qq_33326449/article/details/_blank)**哲学路线的话，Apache CXF是一种更好的选择，特别对嵌入式的Web Services来说；  
3、如果应用程序没有新的特性需要的话，就仍是用原来项目所用的框架，比如 Axis1，XFire，Celtrix或BEA等等厂家自己的Web Services实现，就别劳民伤财了。

****七．你的系统中是否有使用到webservice开发，具体是怎么实现的？****

如果你觉得自己掌握的不够好，对自己不够自信的可以回答为“我的系统中没有使用到webservice的开发，但是我掌握webservice开发的概念和流程”，然后可以给他讲讲相关的概念，也就是上面的这些问题的回答，这样可以绕过这个问题，因为并不是所有的系统都会涉及到webservice的开发。

另一种回答即是先给他介绍一种webservice开发框架，比如CXF，然后告诉他你做的是服务端开发还是客户端开发，如果你说你做的事服务端开发，那么你就告诉他怎么定义的webservice，使用了哪些注解，怎么跟spring进行的整合，怎么发布的服务等等；如果你告诉他你做的事客户端的开发，那么你可以告诉他你怎么生成的本地代码，然后又怎么通过本地代码去调用的webservice服务。