手机应用平台软件开发

11、Web服务开发

刘宁
Email：liuning78@mail.sysu.edu.cn

Web服务（web services）

- 基于网络的、分布式的模块化组件，它执行特定的任务，遵守具体的技术规范，这些规范使得Web Service能与其他兼容的组件进行互操作。
- Web Services 主要利用 HTTP 和 SOAP 协议使商业数据在 Web 上传输，SOAP通过 HTTP 调用商业对象执行远程功能调用，Web 用户能够使用 SOAP 和 HTTP通过 Web 调用的方法来调用远程对象。

Web服务的核心技术及其规范

SUN YAT-SEN UNIVERSITY

- Web服务一般是由企业发布的，具有特定，商业需求的在线应用服务。应用软件能够通过互联网来访问和使用这项服务。
- Web服务的主要目标是在不同平台下的可操作性。
  - XML：描述数据的标准方法
  - SOAP：表示信息交换的协议
  - WSDL：Web服务描述语言
  - UDDI (Universal Description, Discovery and Integration)：通用描述、发现与集成协议，它是一种独立于平台的，基于XML语言的用于在互联网上描述商务的协议

Web服务的核心技术及其规范

- **SOAP(Simple Object Access Protocal)简单对象访问协议**

SOAP技术把基于HTTP的Web技术与XML的可扩展性相结合，实现异构程序和平台之间的互操作性，使应用能够被不同的用户所访问。

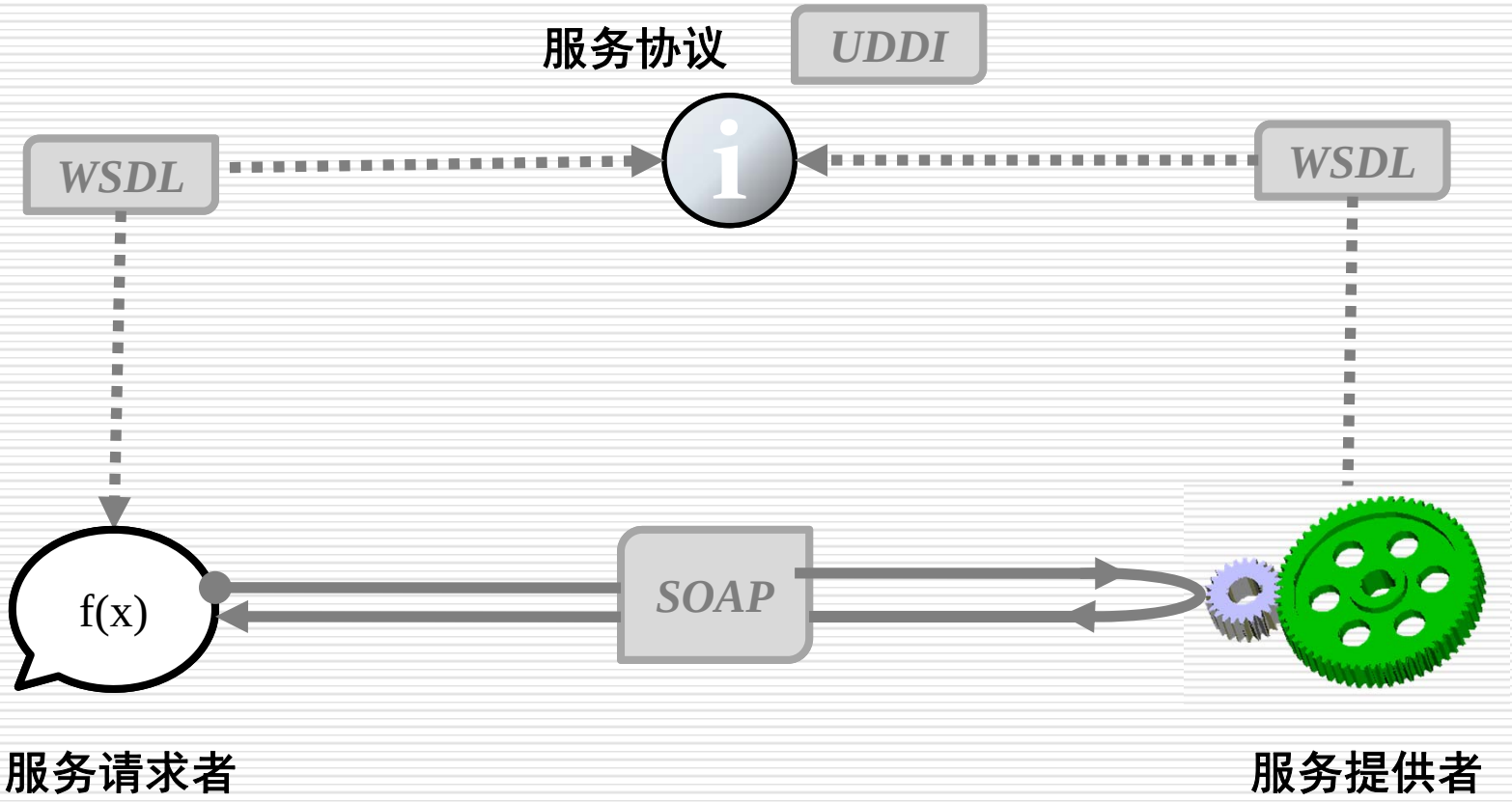
- **WSDL(Web Services Description Language) Web服务描述语言**

WSDL是一种用于描述Web服务的XML格式。WSDL提供服务的详细操作信息。

- **UDDI(Universal Description, Discovery and Integration) 通用描述、发现与集成协议**

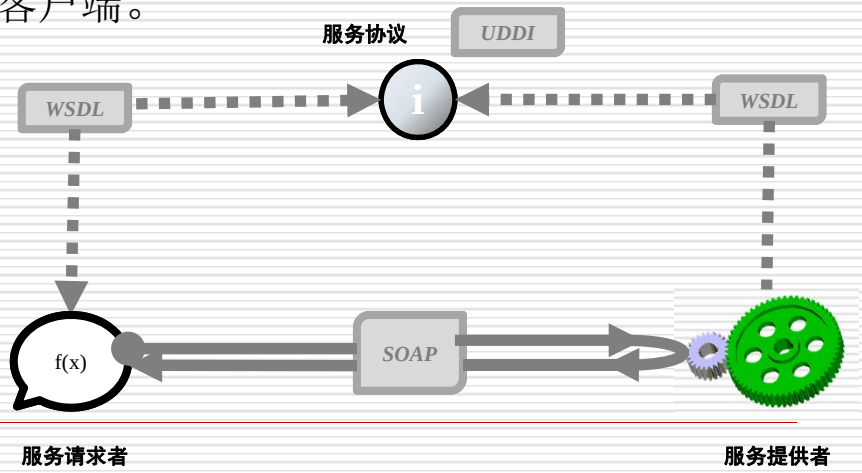
UDDI是一个独立平台的，基于XML语言的注册表和机制。注册表记录了互联网上的商务应用。它也提供了等级和查找Web服务应用程序的机制。

Web服务的架构



Web服务调用原理

- 服务提供者首先建立Web服务，然后把服务发布给所有用户。
- 任何平台上的用户可以通过阅读其WSDL文档生成一个SOAP请求消息。该SOAP消息嵌入到HTTP POST请求中发送到Web服务所在的Web服务器。
- Web服务器把请求转发给Web服务请求处理器，请求处理器解析SOAP请求，然后调用Web服务生成相应的SOAP应答。
- Web服务器得到SOAP应答后通过HTTP送回客户端。



高层接口

- 使用高层接口，不需要知道SOAP和XML的任何信息，就可以生成和使用一个Web服务。
- Soap Tookit 2.0 通过提供SoapClient和SoapServer两个COM对象来完成这些功能。

低层接口

使用低层接口必须对SOAP和XML有所了解。这种接口可以对SOAP的处理过程进行控制，特别是要做特殊处理的时候。

1. 创建一个HttpConnector对象负责HTTP连接。
2. 创建SoapSerializer对象，用于生成SOAP消息。
3. SOAP消息作为Playload通过HttpConnector被发送到服务端。
4. 生成呢个一个SoapReader对象，负责读取服务端返回的SOAP消息。

Web服务的应用

SUN YAT-SEN UNIVERSITY

企业之间的应用

- Web服务用于电子商务应用的标准和开发工具。
- 企业间的电子商务（B2B）：Web服务应用于以企业采购、物流和分销内容的供应链
- 企业与消费者之间的电子商务（B2C）：Web服务应用于涉及到零售以及中间业务的支付系统

企业内部的应用

- 采用中间件应用服务器软件作为工具将企业各项应用都进行改造和开发
- 企业内部应用软件的网络化，包括现在流行的ERP以及CRM等

访问WebService

SUN YAT-SEN UNIVERSITY

- 在Android SDK中没有提供调用WebService的库，因此，需要使用第三方类库来调用WebService，适合手机的WebService客户端类库也有一些，比较常用的为Ksoap2。
- Ksoap2是一个SOAP Webservice客户端包。主要用于资源受限制的Java环境如Applets或J2ME应用程序（CLDC/CDC/MIDP）。

使用Ksoap2访问WebService

SUN YAT-SEN UNIVERSITY

1. 下载Ksoap2包http://code.google.com/p/ksoap2-android/

2. 将下载后的jar文件复制到Eclipse工程的lib目录中（若无该目录，可以新建一个），并在Eclipse工程中引用这个jar包。

使用Ksoap2访问WebService

SUN YAT-SEN UNIVERSITY

- 指定WebService的URL，命名空间和调用的方法名

```
String url =
"http://webservice.webxml.com.cn/WebServices/WeatherWebService.asm
x";
String nameSpace = "http://WebXml.com.cn/";
String methodName = "getSupportCity";
```

- 命名空间和调用方法可参考网站

http://webservice.webxml.com.cn/WebServices/WeatherWebService.asmx

http://webservice.webxml.com.cn/WebServices/WeatherWebService.asmx?WSDL

使用Ksoap2访问WebService

从WSDL查看WebService的Namespace

- 进入 http://webservice.webxml.com.cn/WebServices/WeatherWebService.asmx?WSDL

```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://WebXml.com.cn/"
    xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://WebXml.com.cn/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"><a href="http://www.webxml.com.cn/"
```

nameSpace

使用Ksoap2访问WebService

SUN YAT-SEN UNIVERSITY

进入 http://webservice.webxml.com.cn/WebServices/WeatherWebService.asmx 查看**WebService**提供的方法

使用Ksoap2访问WebService

设置调用方法的参数值。

此步是可选的，如果方法没有参数，可以省略这一步。设置方法的参数值的代码如下

```
SoapObject request = new SoapObject(nameSpace, methodName);
request.addProperty("param1 ",?"value1");
request.addProperty("param2",?"value2");
```

使用Ksoap2访问WebService

查看Method的输入输出参数

- 进入

  http://webservice.webxml.com.cn/WebServices/WeatherWebService.asmx?op=getWeatherbyCityName

  查看

getWeatherbyCityName

WebService读入参数格式定义

WebService输出参数格式定义

SOAP 1.1

以下是 SOAP 1.2 请求和响应示例。所显示的占位符需替换为实际值。

```
POST /WebServices/WeatherWebService.asmx HTTP/1.1
Host: webservice.webxml.com.cn
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://WebXml.com.cn/getWeatherbyCityName"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:x
  <soap:Body>
    <getWeatherbyCityName xmlns="http://WebXml.com.cn/">
      <theCityName>string</theCityName>
    </getWeatherbyCityName>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:x
  <soap:Body>
    <getWeatherbyCityNameResponse xmlns="http://WebXml.com.cn/">
      <getWeatherbyCityNameResult>
        <string>string</string>
        <string>string</string>
      </getWeatherbyCityNameResult>
    </getWeatherbyCityNameResponse>
  </soap:Body>
</soap:Envelope>
```

使用Ksoap2访问WebService


```
POST /WebServices/WeatherWebService.asmx HTTP/1.1
Host: webservice.webxml.com.cn
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://WebXml.com.cn/getWeatherbyCityName"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w
  <soap:Body>
    <getWeatherbyCityName xmlns="http://WebXml.com.cn/">
      <theCityName>string</theCityName>
    </getWeatherbyCityName>
  </soap:Body>
</soap:Envelope>
```

把读入的**String**变量**theCityName**

```
SoapObject request = new SoapObject(nameSpace, methodName);
request.addProperty("theCityName", "广州");
```

使用Ksoap2访问WebService

- 生成调用WebService方法的SOAP请求信息。该信息由SoapSerializationEnvelope对象描述，代码如下：

```
SoapSerializationEnvelope envelope =
    new SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.bodyOut = request;
```

- 创建SoapSerializationEnvelope对象时需要通过SoapSerializationEnvelope类的构造方法设置SOAP协议的版本号。该版本号需要根据服务端WebService的版本号设置。在创建SoapSerializationEnvelope对象后，还需要设置SoapSerializationEnvelope类的bodyOut属性，该属性的值就是在第1步创建的SoapObject对象。

使用Ksoap2访问WebService

- 创建HttpTransportSE对象。通过HttpTransportSE类的构造方法可以指定WebService的WSDL文档的URL，代码如下：

```
HttpTransportSE ht = new HttpTransportSE (url );
```

- 使用call方法调用WebService方法，代码如下：

```
ht.call(null, envelope);
```

call方法的第1个参数一般为null，第2个参数就是在之前创建的SoapSerializationEnvelope对象。

使用Ksoap2访问WebService

- 获得WebService方法的返回结果，代码如下：

```
SoapObject result = (SoapObject) envelope.bodyIn;
SoapObject detail = (SoapObject)
result.getProperty("getWeatherbyCityNameResult");
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="htt
  <soap:Body>
    <getWeatherbyCityNameResponse xmlns="http://WebXml.com.cn/">
      <getWeatherbyCityNameResult>
        <string>string</string>
        <string>string</string>
      </getWeatherbyCityNameResult>
    </getWeatherbyCityNameResponse>
  </soap:Body>
</soap:Envelope>
```

结果由两个**string**组成，返回**getWeatherCityNameResult**

使用Ksoap2访问WebService

- 若读取的类型是int时，则是SoapPrimitive，不是SoapObject

```
SoapObject result = (SoapObject) envelope.bodyIn;
SoapPrimitive detail = (SoapPrimitive)
result.getProperty("addResult");
```

Ksoap2的常用接口

SUN YAT-SEN UNIVERSITY

接口
org.kSOAP2.SoapEnvelope org.kSOAP2.SoapSerializationEnvelope org.kSOAP2.SoapObject org.KSOAP2.transport.HttpTransport

- SoapEnvelope与SOAP规范中的SOAP Envelope相对应，封装了head和body。
- SoapSerializationEnvelope对SoapEnvelope进行了扩展来支持SOAP序列化规范，能够把简单对象自动进行序列化。

Ksoap2的常用接口

SUN YAT-SEN UNIVERSITY

接口
org.kSOAP2.SoapEnvelope org.kSOAP2.SoapSerializationEnvelope org.kSOAP2.SoapObject org.KSOAP2.transport.HttpTransport

- SoapObject能够构造SOAP调用。
- HttpTransport屏蔽了网络请求或访问以及获取服务器SOAP的具体细节。

Ksoap2和Web服务

SUN YAT-SEN UNIVERSITY

- 利用Web服务传递String给MIDP(Mobile Information Device Profile，移动信息设备配置文件)
- 首先在服务器端编写主服务类

服务器端

```
Public class kSOAPWS {

      public kSOAPWS() {}

      public String WSMethod (String user, String pwd) {
            return  “WSResponse”;
      }
}
```

Ksoap2和Web服务

KSOAP调用服务器的Web服务有6步：

1. 指定Web服务的命名空间和调用方法的名称；
2. 设置调用方法的参数（可选）；
3. 生成调用Web服务的SOAP请求信息；
4. 指定Web服务的WSDL文档的URL；
5. 利用call调用Web服务；
6. 利用getResponse方法获得Web服务的返回结果。

指定Web服务的命名空间和调用方法的名称

1. 利用SoapObject类完成调用

```
SoapObject request =new SoapObject(ServiceNamespace, MethodName);
```

- *ServiceNamespace* – Web服务的命名空间，可从WSDL文档中找到；
- *MethodName* – 所调用方法的名字。

设置调用方法的参数（可选），如果方法没有方法，则这一步可以省略。

```
request.addProperty("User","Password");
```

- addProperty方法设置的参数需要与Web服务类中的方法参数顺序保持一致。

利用SoapSerializationEnvelope对象生成调用Web服务的SOAP请求信息

```
SoapSerializationEnvelope en =
        new SoapSerializationEnvelope (SoapEnvelope.VER11);
en.bodyOut = request;
```

1. SoapEnvelope.VER11是SOAP协议的版本号，该版本号要与服务器端Web服务的版本号一致；
2. 在创建SOAP序列化封装对象后，需要设置属性bodyOut为第一步的SoapObject对象。

KSOAP和Web服务

- 创建HttpTransportSE对象，通过这个对象的构造方法指定Web服务的WSDL文档的URL

```
HttpTransportSE HT = new HttpTransportSE(WSDL_URL);
```

- 使用call方法调用Web服务

```
HT.call(null, en);
```

  - 第二个参数是第3步创建的SOAP序列化封装对象。

- 使用getResponse方法获得Web服务的返回结果

```
SoapObject SO = (SoapObject) en.getResponse();
```

- 利用第3步创建的SOAP序列化封装对象获得的Web服务的返回结果，并强制类型转换为SoapObject类。

KSOAP和Web服务

SUN YAT-SEN UNIVERSITY

- KSOAP调用Web服务需要运用HttpTransport类，实际上是调用了HttpConnection作为网络连接。
- 在KSOAP调用Web服务的时候，如果由于某种原因，Web服务不能立即返回，Android界面上的组件仍然需要处于活动状态供用户使用，不能造成阻塞。
- 为了防止UI组件的阻塞，KSOAP调用Web服务的时候，必须另起一个线程。

KSOAP的类型映射

SUN YAT-SEN UNIVERSITY

- KSOAP能够把四种SOAP类型映射为Java类型：

SOAP Type	Java Type
xsd:int	java.lang.Integer
xsd:long	java.lang.Long
xsd:string	java.lang.String
xsd:boolean	java.lang.Boolean

- 其余类型需要进行类型映射，把成员变量序列化为byte[]，通过网络传送后再放序列化。

KSOAP的类型映射

SUN YAT-SEN UNIVERSITY

- 在KSOAP中，利用Base64把二进制流编码为ASCII字符串，使二进制数据能够通过XML/SOAP传输；
- Org.kSOAP2.serialization中的MarshalBase64的目的就是把SOAP XML中的xsd:base64Binary元素序列化为Java字节数组类型。
- KSOAP2提供Marshaldate和MarshalHashtable类来把相应的元素序列化为Java的Data和Hashtable类型。

KSOAP应用实例

SUN YAT-SEN UNIVERSITY

KSOAP2是第三方开发的专门用于在移动设备调用WebService的类库。

- 使用KSOAP2调用WebService可分为6步来完成，主要使用了SoapObject对象来指定了要调用的方法
- 通过HttpTransportSE对象的call方法来调用WebService的方法
- 通过getResponse方法返回结果。

通过Web服务查询亚马逊网上书店书目

- 提交包含关键字的书目查询，如果查询成功，将会返回一系列书名节点，每一本书都提供作者、出版社、出版日期、价格等信息
- 书名节点在一个”Details”节点下，查询结果的总数放在TotalResult节点
- 每页10个结果，可以通过查看TotalPages节点来确定需要多少页
- 要测试工程，必须到亚马逊http://www.amazon.com/webservice注册获取Access Key ID

KSOAP应用实例

SUN YAT-SEN UNIVERSITY

亚马逊的书目查询**Web**服务：

http://soap.amazon.com/schemas3/AmazonWebServices.wsdl

- 关键词查询请求方法：

```
▼<operation name="KeywordSearchRequest">
   <soap:operation soapAction="http://soap.amazon.com"/>
  ▼<input>
     <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://soap.amazon.com"/>
   </input>
  ▼<output>
     <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://soap.amazon.com"/>
   </output>
 </operation>
```

KSOAP应用实例

SUN YAT-SEN UNIVERSITY

KSOAP2可以简单地通过SoapObject的getProperty方法来得到书详细信息的节点，存储到Vector对象中：

```
HttpTransport ht= new HttpTransport("http://soap.amazon.com/onca/soap3");

ht.call(null, envelope);

SoapObject result = (SoapObject) envelope.getResult();

Vector resultVector = (Vector) result.getProperty("Details");
```

KSOAP应用实例

SUN YAT-SEN UNIVERSITY

在Vector对象中存储了一组SoapObject对象，每个SoapObject对象对应一本书的DOM(Document Object Model)对象，要得到每一本书的书名和价格：

```
for(int i = 0; i < resultVector.size(); i++){
      SoapObject detail = (SoapObject) resultVector.elementAt(i);
       System.out.println("书名>>"+(String) detail.getProperty("ProductName"));
      System.out.println("日期>>"+(String) detail.getProperty("ReleaseDate"));
      System.out.println("价格>>"+(String) detail.getProperty("ListPrice"));
}
```

KSOAP应用实例

SUN YAT-SEN UNIVERSITY

```
//WSDL文档的URL
private static final String NAMESPACE
= "http://webservices.amazon.com/AWSECommerceService/2006-05-17";
//从亚马逊网站获取的Access Key ID
private static final String AMAZON_WEBSERVICE_KEY = "";

public AmazonSearchClient() {
        if (AMAZON_WEBSERVICE_KEY.length() == 0) {
            System.out.println("Please substitute your own amazon webservice key
before running this code.");
        }
```

具体程序：

```
else {
        Request requestObject = new Request();
        requestObject.author = "Whyte";
        requestObject.searchIndex = "Books";
          //第1步：创建SoapObject对象，并制定Web服务的命名空间
         SoapObject request
                  = new SoapObject(NAMESPACE, "ItemSearch");
        //第2步：设置Web服务方法的参数
         request.addProperty("SubscriptionId",AMAZON_WEBSERVICE_KEY);
         request.addProperty("Request", requestObject);
```

具体程序：

```
//第3步：创建SoapSerializationEnvelope对象，并制定Web服务的版 本
    SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
    envelope.setOutputSoapObject(request);
    requestObject.register(envelope);
    registerObjects(envelope);
//第4步：创建HttpTransportSE对象，并指定WSDL文档的URL
      HttpTransportSE httpTransportSE= new
HttpTransportSE("http://soap.amazon.com/onca/soap?Service=AWSECommerceService");
    httpTransportSE.setXmlVersionTag("<?xml version=\"1.0\" ， encoding=\"UTF-8\"?>");
```

KSOAP应用实例

```
try {
            //第5步：调用Web服务
            httpTransportSE.call("http://soap.amazon.com", envelope);
          //第6步：使用getResponse方法获得Web服务方法的返回结果
            BookItems response = (BookItems)  envelope.getResponse();
          System.out.println(response);
        } catch (Exception e) {
          e.printStackTrace();
          }
    }
}
```

Web服务

- 创建可互操作的分布式应用程序的新平台
- 为了达到跨平台操作，Web服务是完全基于XML、XSD等独立于平台、独立于软件供应商的标准的。
- Web服务适用于应用程序集成、B2B集成、代码和数据重用，以及通过Web进行客户端和服务器的通信。

Questions?