# 前提

什么是WebService

Web service是一个平台独立的，低耦合的，自包含的、基于可编程的web的应用程序，可使用开放的XML（标准通用标记语言下的一个子集）标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的互操作的应用程序。

Web Service技术， 能使得运行在不同机器上的不同应用无须借助附加的、专门的第三方软件或硬件， 就可相互交换数据或集成。依据Web Service规范实施的应用之间， 无论它们所使用的语言、 平台或内部协议是什么， 都可以相互交换数据。Web Service是自描述、 自包含的可用网络模块， 可以执行具体的业务功能。Web Service也很容易部署， 因为它们基于一些常规的产业标准以及已有的一些技术，诸如标准通用标记语言下的子集XML、HTTP。Web Service减少了应用接口的花费。Web Service为整个企业甚至多个组织之间的业务流程的集成提供了一个通用机制。

调用网络上的WebService服务网址：<http://webxml.com.cn/>

# SOAP和WSDL概念

SOAP作为一个基于XML语言的协议用于在网上传输数据，是基于HTTP协议的

简单对象访问协议

SOAP的xml规范(约束)的组成如下：

Envelope – 必须的部分。以XML的根元素出现

Headers – 可选的

Body – 必须的。在body部分，包含要执行的服务器的方法。和发送到服务器的数据。

基于http协议

请求行

POST /WebServices/IpAddressSearchWebService.asmx HTTP/1.1

请求头

Host: ws.webxml.com.cn

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://WebXml.com.cn/getCountryCityByIp"

以下是请求体

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<getCountryCityByIp xmlns="http://WebXml.com.cn/">

<theIpAddress>string</theIpAddress>

</getCountryCityByIp>

</soap:Body>

</soap:Envelope>

WSDL 描述文档告诉你此请求的一些信息：由xml格式组成

WSDL(WebService Description Language):web 服务描述语言

就是一个xml文档，用于描述当前服务的一些信息（服务名称、服务的发布地址、服务提供的方法、方法的参数类型、方法的返回值类型等）

wsdl又可以叫：webService的使用说明书

# 基于jdk1.7发布一个WebService服务

## 服务端发布

步骤

第一步：创建一个Java项目

注意：这里不使用web项目，不用导第三方jar包

第二步：创建一个类，加入Webservice注解 ，添加方法

注意 要普通的方法：不能是静态方法

@javax.jws.WebService //类中使用此注解

public class A{

public String str(String name,int i) {

System.out.println("服务端被调用了"+i+"次");

return "您好！"+name+"欢迎您使用此weservice";

}

//main方法

public static void main(String[] args) {

// 参数1：必须带ip和端口、访问路径(可以随便定义一般都写类名)三样不可缺少

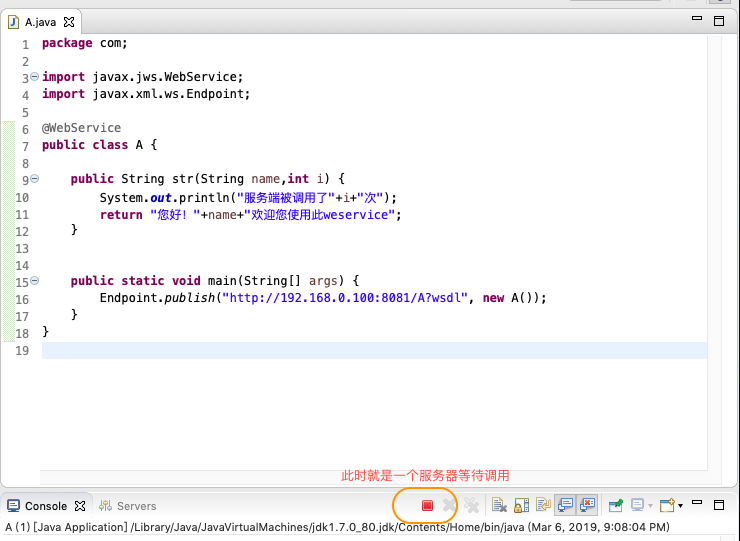
//参数2：类的对象

javax.xml.ws.Endpoint.publish("http://192.168.0.101:8081/A", new A()); //地址可以用来访问

}

}

第三步 直接运行主方法即可发布：此时主方法就在运行中(就是一个服务)等待调用

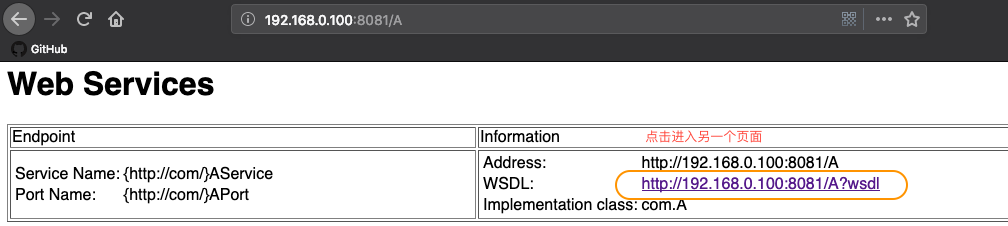


注意：不是调用一次就会停止的，永久服务

第四步 测试

发布服务后JDK就已经给你设置了WSDL(使用描述说明书)的访问了

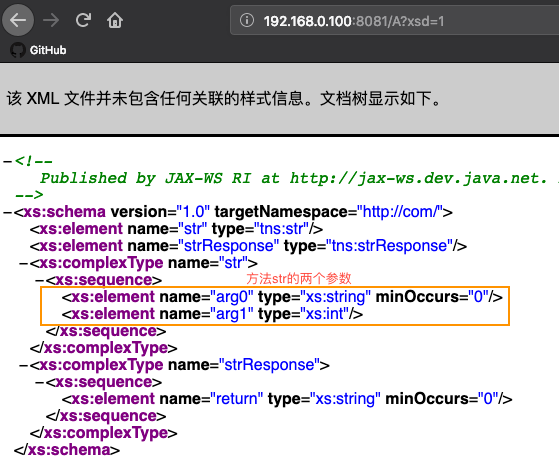
此时就可以通过浏览器url就可以进行访问了：<http://192.168.0.101:8081/A>



进入<http://192.168.0.101:8081/A>？wsdl



进入方法的详细页面：这里有点不同，貌似JDK发布的才有这个页面的



## 客户端调用

第一步 使用jdk7中bin目录下的命令wsimport解析生成代码

常用命令参数：

-s d:/wwl/test/ -s<指定目录生成java文件目录>

-p wwl.service -p<生成类的包名>

-d d:/wwl/test/ -d<指定目录生成class文件目录> 默认参数一般都不用写

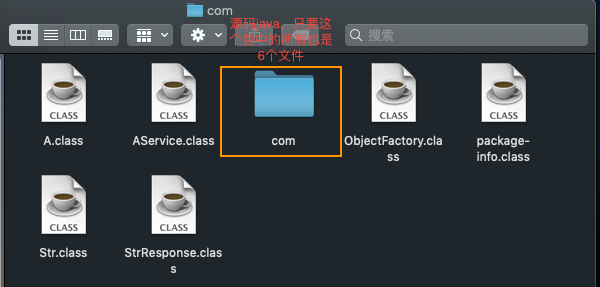
注意 使用命令解析的时候服务端一定要开启

Wsimport –s . –p wwl.service http://192.168.0.2:8081/A?wsdl 命令意思是：在本目录生成包名为wwl.service的类

注意：http://192.168.0.2:8081/A?wsdl 这个网址是在服务端自己定义的：服务端的ip和端口 + 工程名和默认参数wsdl

mac系统命令直接参数-s即可： wsimport -s com http://192.168.0.100:8081/A?wsdl

注意：需要自己先创建文件夹，不然报错



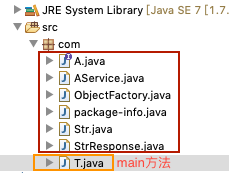
第二步 把生成的java类导入工程中的

大概有以下些类（了解）

A接口类 和服务端要调用的类名一致

AService ：调用需要用到类，继承Service

ObjectFactory，package-info ，Str（服务端方法名，这里变成类）



第三步 类中的使用

public static void main(String[] args) {

//第一 使用远程的类名+Service（默认会自动加一个Service）创建对象

AService a= new AService();

//第二 创建客户端代理对象，用于远程调用

A porxy = a.getAPort();

//第三 调用远程不是静态类的方法（静态类方法不能调用）

String str = porxy.str("你好", 1);

System.out.println(str);

}

注意 服务端不会就调用一次就停止的，永久服务