姓名 张栋

学院 信息学院

学号 2017104146

简介

Webservice:跨语言跨平台的远程调用技术。Web service 即web服务，它是一种跨编程语言和跨操作系统平台的远程调用技术即跨平台远程调用技术。

JAVA 中共有三种WebService 规范，分别是JAX-WS（JAX-RPC）、JAXM&SAAJ、JAX-RS。

webService三要素：soap、wsdl、uddi

JAX-WS 的全称为 Java API for XML-Based Webservices ，早期的基于SOAP 的JAVA 的Web 服务规范JAX-RPC（Java API For XML-Remote Procedure Call）.

JAXM（JAVA API For XML Message）主要定义了包含了发送和接收消息所需的API，SAAJ（SOAP With Attachment API For Java，JSR 67）是与JAXM 搭配使用的API，为构建SOAP 包和解析SOAP 包提供了重要的支持，支持附件传输等.

JAX-RS 是JAVA 针对REST(Representation State Transfer)风格制定的一套Web 服务规范.

Soap

SOAP 是一种网络通信协议

SOAP即Simple Object Access Protocol简易对象访问协议

SOAP 用于跨平台应用程序之间的通信

SOAP 被设计用来通过因特网(http)进行通信

SOAP ＝ HTTP+XML，其实就是通过HTTP发xml数据

SOAP 很简单并可扩展支持面向对象

SOAP 允许您跨越防火墙

Socket是所有通信的基础也是语言个无关平台无关。

Socket使用的是tcp协议，传输效率高。适合传递大数据高并发场景，高并发的情况需要实现多线程并且使用到线程池，编码复杂。Sockt的高并发框架mina。

Socket只是流的传输，传输的格式需要程序员自己定义。

Webservice使用的是soap协议，soap协议基于http协议的应用层协议，本质就是http+xml。Soap协议是w3c标准，传输效率低。使用传输数据不是太大的场合，也是支持高并发的，受限于web容器。支持soap协议和wsdl两者都是国际通用标准，不需要自定义数据格式，只需要面向对象开发。

WSDL

WSDL是一个用于精确描述Web服务的文档，WSDL文档是一个遵循WSDL-XML模式的XML文档。WSDL 文档将Web服务定义为服务访问点或端口的集合。在 WSDL 中，由于服务访问点和消息的抽象定义已从具体的服务部署或数据格式绑定中分离出来，因此可以对抽象定义进行再次使用。消息，指对交换数据的抽象描述；而端口类型，指操作的抽象集合。用于特定端口类型的具体协议和数据格式规范构成了可以再次使用的绑定。将Web访问地址与可再次使用的绑定相关联，可以定义一个端口，而端口的集合则定义为服务。

一个WSDL文档通常包含8个重要的元素，即definitions、types、import、message、portType、operation、binding、service元素。这些元素嵌套在definitions元素中，definitions是WSDL文档的根元素。

UDDI

UDDI是一种用于描述、发现、集成Web Service的技术，它是Web Service协议栈的一个重要部分。通过UDDI，企业可以根据自己的需要动态查找并使用Web服务，也可以将自己的Web服务动态地发布到UDDI注册中心，供其他用户使用

这一个项目实现了一个查询天气的web service接口

1、编写一个SEI,也就是一个接口

public interface WeatherInterface {

String queryWeather(String cityName);

}

2、编写一个SEI实现类，需要实现SEI接口，而且还需要在这个实现类上面添加一个@Webservice注解

@Webservice

public class WeatherInterfaceImpl implements WeatherInterface {

public String queryWeather(String cityName) {

System.out.println("接收到客户端发送的城市名称:"+cityName);

String result="晴,高温预警";

return result;

}

}

3、发布服务。使用Endpoint的静态方法publish。

public class WeatherServer {

public static void main(String[] args) {

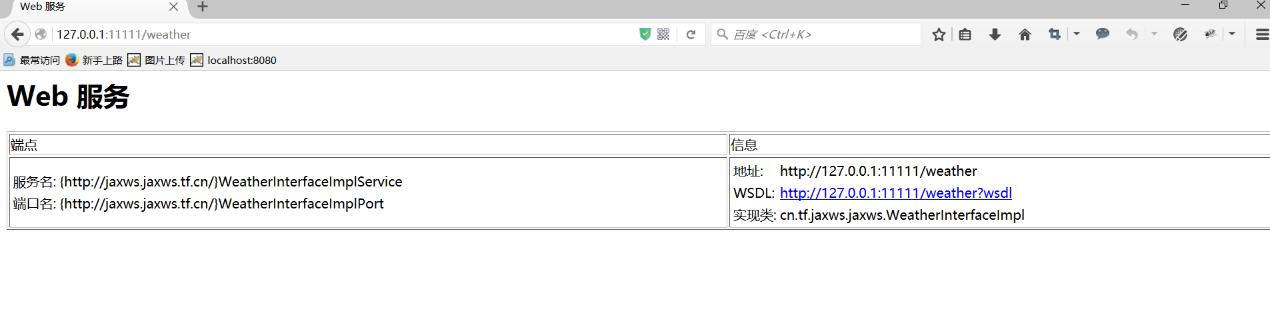
//发布服务

Endpoint.publish("http://127.0.0.1:11111/weather",newWeatherInterfaceImpl());

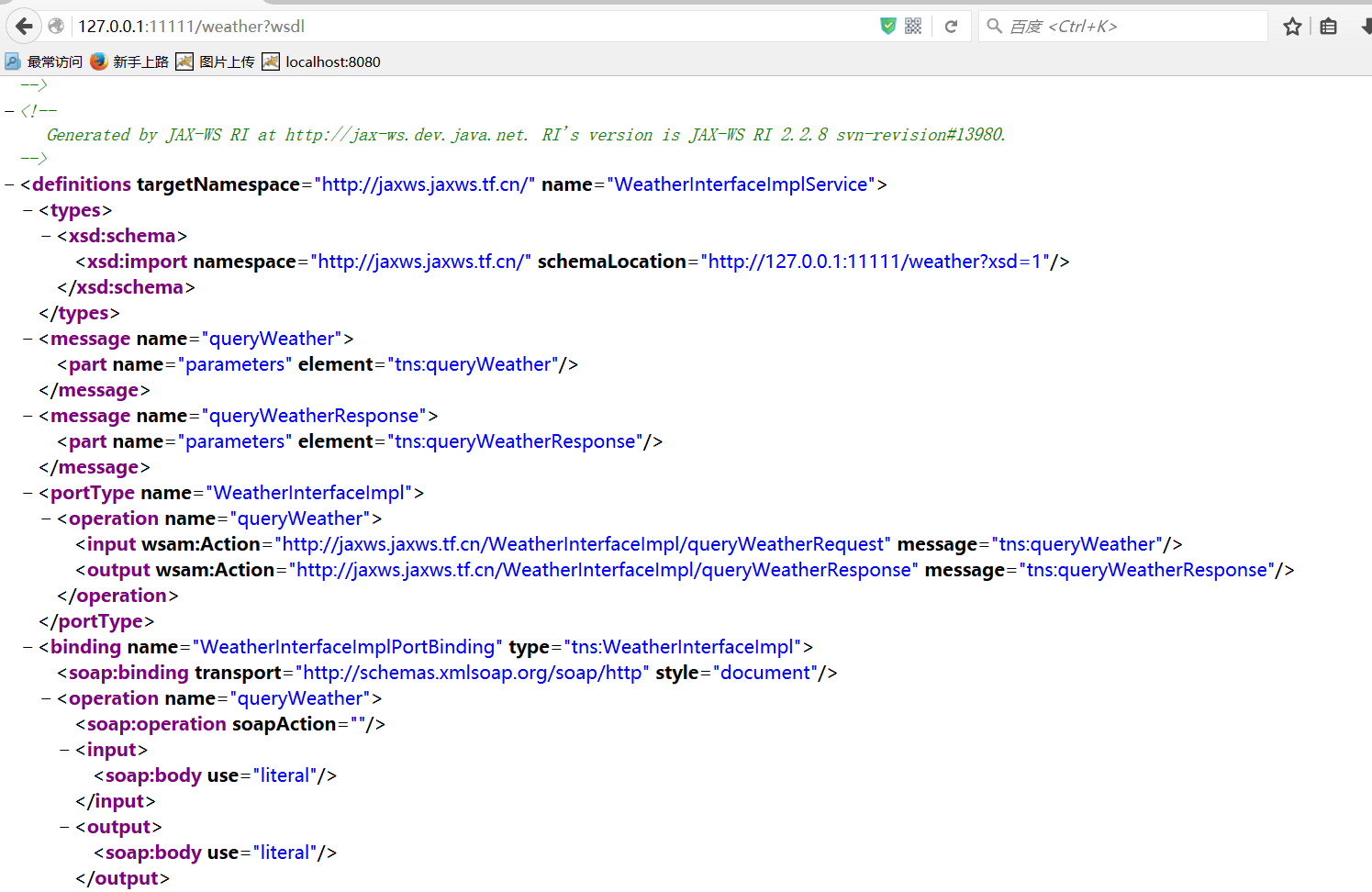
}

}

访问地址：<http://127.0.0.1:11111/weather>



<http://127.0.0.1:11111/weather?wsdl> 中看到ImplPort



看到效果则说明启动成功了。

##客户端

使用Wsimport生成客户端调用代码，

在jdk的安装目录的bin目录中，有一个wsimport命令。

可以根据wsdl文档生成客户端调用代码。

生成WebServiceWSGEN

在src目录下通过cmd运行以下命令：

wsimport -s . http://127.0.0.1:11111/weather?wsdl

###实现步骤

使用过程生成的代码来创建客户端

1、创建一个服务视图对象

2、从服务试图获得porttype（SEI）对象

3、调用服务端方法

4、打印结果

public class WeatherClient {

public static void main(String[] args) {

WeatherInterfaceImplService service=new WeatherInterfaceImplService();

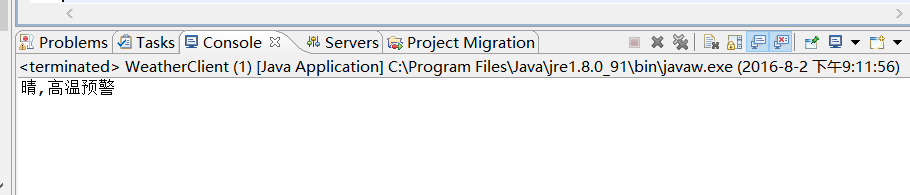
WeatherInterfaceImpl portType=service.getWeatherInterfaceImplPort();

String result=portType.queryWeather("衡阳");

System.out.println(result);

}

}



项目地址：https://github.com/sdksdk0/WebWeather-BaseDemo-