由于SOAP中值的类型就是XML Schema中的基本类型，默认只支持简单类型和数组。所以在进行对象传递的过程中，需要进行序列化和反序列化。

一个类的传递必须包括两部分第一部分为调用端的声明序列化，第二部分是服务器端的反序列化。所以现在分为两步走，查清axis中如何定义一个序列化的类。

第一个人需要在服务器端加一个xml控制文件就好了，在调用端需要加上

1. QName qn = **new** QName("urn:SessionData", "SessionData");
2. call.registerTypeMapping(SessionData.**class**, qn,
3. **new** BeanSerializerFactory(SessionData.**class**, qn),
4. **new** BeanDeserializerFactory(SessionData.**class**, qn));

但是这里的urn后边的SessionData是不一定的，第二个SessionData是类名，第二函数中调用的SessionData都是类名。

1. call.addParameter("arg0", qn, ParameterMode.IN);//定义一个参数类型，如果是String类型的参数可以不需要这句话
2. Address result = (Address) call.invoke(new Object[]{address});//这里传递给service的是一个对象

有的人会说声明类的时候需要这么声明

1. **public class** User implements Serializable {
2. }

需要说明的是，这里传递的User和服务器端的User是同一个对象，在实际开发中应该不是同一个对象的。如java调用.net的WebService，传递的对象应该不是同一个。需要我们自己根据wsdl文件中的描述创建JavaBean。注意的是对象一个要序列化。

new QName("urn:User", "User")这里的QName的urn:User和wsdd文件中的对应，参数User也是和wsdd文件中的qname=*"myNSD:User"*对应的

这里是将对象序列化和反序列化的配置

call.registerTypeMapping(User.class, qn, new BeanSerializerFactory(User.class, qn), new BeanDeserializerFactory(User.class, qn));

User是Mapping的对象，也就是更加wsdl中参数描述创建的Java对象

call.addParameter("i", XMLType.XSD\_INT, ParameterMode.IN);设置参数信息，ParameterMode.IN代表传递的参数，XMLType.XSD\_INT,代表参数的类型，前面的i是参数名称

call.setReturnClass(User[].class);设置返回值的类型

特别说明：

A、 如果你调用的方法有返回值，一定要设置返回值的类型。call.setReturnClass

B、 如果你调用的方法有参数，一定要设置参数的类型call.addParameter

C、 记得添加wsdl4j.jar，序列化转换的时候需要用到，否则会出现找不到类型异常

Wsdd文件怀疑是使用axis框架开发webservice所用到的文件。