

[首页](#) [新闻](#) [论坛](#) [问答](#) [专栏](#) [博客](#) [文摘](#) [圈子](#) [招聘](#) [服务](#) [搜索](#)[Java](#) [Web](#) [Ruby](#) [Python](#) [敏捷](#) [MySQL](#) [普元](#) [Dorado](#) [图书](#) [MSUP开放日](#)[专栏首页](#) → [Java编程](#) → [SOA](#) → [专栏: Apache CXF 专栏](#) → [章节:基础知识](#)

WebService开发笔记 1 -- 利用cxfr开发WebService竟然如此简单

原创作者: [jacally](#) 阅读:7301次 评论:7条 更新时间:2008-12-18

WebService开发笔记 1 -- 利用cxfr开发WebService竟然如此简单

现在的项目中需要用到SOA概念的地方越来越多,最近我接手的一个项目中就提出了这样的业务要求,需要在.net开发的客户端系统中访问java开发的web系统,这样的业务需求自然需要通过WebService进行信息数据的操作。下面就将我们在开发中摸索的一点经验教训总结以下,以供大家参考。

在[WebService开发笔记 2 -- VS 2005 访问WebServcie更简单](#)中作一个跨平台访问WebServcie服务的例子....

在[WebService开发笔记 3 -- 增强访问 WebService 的安全性](#)通过一个简单的用户口令验证机制来加强一下WebService的安全性....

我们项目的整个架构使用的比较流行的WSH MVC组合,即webwork2 + Spring + Hibernate;

1. 首先集成Apache CXF WebService 到 Spring 框架中;

apache cxfr 下载地址:<http://people.apache.org/dist/incubator/cxf/2.0.4-incubator/apache-cxf-2.0.4-incubator.zip>

在spring context配置文件中引入以下cxfr配置

Xml代码

```
1. <import resource="classpath*:META-INF/cxf/cxf.xml" />
2. <import resource="classpath*:META-INF/cxf/cxf-extension-soap.xml" />
3. <import resource="classpath*:META-INF/cxf/cxf-servlet.xml" />
```

在web.xml中添加过滤器:

Xml代码

```
1. <servlet>
2.     <servlet-name>CXFServlet</servlet-name>
```

文章信息

专栏: [Apache CXF 专栏](#)



- 由[YuLimin](#)在2008-12-18创建
- 由[YuLimin](#)在2008-12-18更新
- 标签: cxf webservises

相关新闻

- [CXF 2.0 发布了](#)
- [EJB 3.0和Spring 2.5 : 在Spring中使用EJB 3.0](#)
- [一个Java的Restful服务框架, 支持JPA、JAAS、分布式资源对象](#)

相关讨论

- [WebService开发笔记 1 -- 利用cxfr开发WebService竟然如此简单](#)
- [Apache CXF 学习笔记](#)
- [CXF 2.0 webservice 学习笔记系列_1](#)
- [何为CXF?](#)
- [cxf+spring+struts2的helloWorld](#)

```

3.         <servlet-class>
4.             org.apache.cxf.transport.servlet.CXFServlet
5.         </servlet-class>
6.     </servlet>
7.     <servlet-mapping>
8.         <servlet-name>CXFServlet</servlet-name>
9.         <url-pattern>/services/*</url-pattern>
10.    </servlet-mapping>

```

2. 开发服务端WebService接口:

Java代码

```

1.  /**
2.   * WebService接口定义类.
3.   *
4.   * 使用@WebService将接口中的所有方法输出为Web Service.
5.   * 可用annotation对设置方法、参数和返回值在WSDL中的定义.
6.   */
7.  @WebService
8.  public interface WebServiceSample {
9.
10.
11.      /**
12.       * 一个简单的方法, 返回一个字符串
13.       * @param hello
14.       * @return
15.       */
16.      String say(String hello);
17.
18.      /**
19.       * 稍微复杂一些的方法, 传递一个对象给服务端处理
20.       * @param user
21.       * @return
22.       */
23.      String sayUserName(
24.          @WebParam(name = "user")
25.          UserDTO user);
26.
27.      /**
28.       * 最复杂的方法, 返回一个List封装的对象集合
29.       * @return
30.       */

```

相关博客

- [利用cxf开发WebService](#)
- [利用cxf 开发 WebService](#)
- [WebService开发笔记 1 -- 利用cxf开发WebService竟然如此简单](#)
- [cxf应用](#)
- [WebService开发笔记 1 -- 利用cxf开发WebService竟然如此简单](#)

```
31.         public
32.         @WebResult(partName="o")
33.         ListObject findUsers();
34.
35.     }
```

由简单到复杂定义了三个接口,模拟业务需求;

3.实现接口

Java代码

```
1.  /**
2.   * WebService实现类.
3.   *
4.   * 使用@WebService指向Interface定义类即可.
5.   */
6.  @WebService(endpointInterface = "cn.org.coral.biz.examples.webservice.WebServiceSample")
7.  public class WebServiceSampleImpl implements WebServiceSample {
8.
9.      public String sayUserName(UserDTO user) {
10.         return "hello "+user.getName();
11.     }
12.
13.     public String say(String hello) {
14.         return "hello "+hello;
15.     }
16.
17.     public ListObject findUsers() {
18.         ArrayList<Object> list = new ArrayList<Object>();
19.
20.         list.add(instancUser(1,"lib"));
21.         list.add(instancUser(2,"mld"));
22.         list.add(instancUser(3,"lq"));
23.         list.add(instancUser(4,"gj"));
24.         ListObject o = new ListObject();
25.         o.setList(list);
26.         return o;
27.     }
28.
29.     private UserDTO instancUser(Integer id,String name){
30.         UserDTO user = new UserDTO();
31.         user.setId(id);
32.         user.setName(name);
```

```
33.         return user;
34.     }
35. }
```

4. 依赖的两个类:用户对象与List对象

Java代码

```
1.  /**
2.   * Web Service传输User信息的DTO.
3.   *
4.   * 分离entity类与web service接口间的耦合, 隔绝entity类的修改对接口的影响.
5.   * 使用JAXB 2.0的annotation标注JAVA-XML映射, 尽量使用默认约定.
6.   *
7.   */
8.  @XmlAccessorType(XmlAccessType.FIELD)
9.  @XmlType(name = "User")
10. public class UserDTO {
11.
12.     protected Integer id;
13.
14.     protected String name;
15.
16.     public Integer getId() {
17.         return id;
18.     }
19.
20.     public void setId(Integer value) {
21.         id = value;
22.     }
23.
24.     public String getName() {
25.         return name;
26.     }
27.
28.     public void setName(String value) {
29.         name = value;
30.     }
31. }
```

关于List对象,参照了[有关JWS的一个问题](#)中的描述:DK6.0 自带的WebService中 WebMethod的参数好像不能是ArrayList 或者其他List

传递List需要将List 包装在其他对象内部才行 (个人理解 如有不对请指出),我在实践中也遇到了此类问题.通过以下封装的对象即可以

传递List对象.

Java代码

```
1.  /**
2.   * <p>Java class for listObject complex type.
3.   *
4.   * <p>The following schema fragment specifies the expected content contained within this cla
5.   *
6.   * <pre>
7.   * <complexType name="listObject">
8.   *   <complexContent>
9.   *     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
10.  *       <sequence>
11.  *         <element name="list" type="{http://www.w3.org/2001/XMLSchema}anyType" maxOccurs="u
12.  *           nbounded" minOccurs="0"/>
13.  *       </sequence>
14.  *     </restriction>
15.  *   </complexContent>
16.  * </complexType>
17.  * </pre>
18.  *
19.  */
20. @XmlAccessorType(XmlAccessType.FIELD)
21. @XmlType(name = "listObject", propOrder = { "list" })
22. public class ListObject {
23.
24.     @XmlElement(nillable = true)
25.     protected List<Object> list;
26.
27.     /**
28.     * Gets the value of the list property.
29.     *
30.     * <p>
31.     * This accessor method returns a reference to the live list,
32.     * not a snapshot. Therefore any modification you make to the
33.     * returned list will be present inside the JAXB object.
34.     * This is why there is not a <CODE>set</CODE> method for the list property.
35.     *
36.     * <p>
37.     * For example, to add a new item, do as follows:
38.     * <pre>
39.     *     getList().add(newItem);
```

```

40.      * </pre>
41.      *
42.      *
43.      * <p>
44.      * Objects of the following type(s) are allowed in the list
45.      * {@link Object }
46.      *
47.      *
48.      */
49.      public List<Object> getList() {
50.          if (list == null) {
51.              list = new ArrayList<Object>();
52.          }
53.          return this.list;
54.      }
55.
56.      public void setList(ArrayList<Object> list) {
57.          this.list = list;
58.      }
59.
60.  }

```

5.WebService 服务端 spring 配置文件 ws-context.xml

Xml代码

```

1.  <beans xmlns="http://www.springframework.org/schema/beans"
2.      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.      xmlns:jaxws="http://cxf.apache.org/jaxws"
4.      xsi:schemaLocation="http://cxf.apache.org/jaxws http://cxf.apache.org/schemas/jaxws.x
sd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/sp
ring-beans.xsd"
5.      default-autowire="byName" default-lazy-init="true">
6.
7.      <jaxws:endpoint id="WebServiceSample"
8.          address="/WebServiceSample" implementor="cn.org.coral.biz.examples.webservice
.WebServiceSampleImpl"/>
9.
10. </beans>

```

WebService 客户端 spring 配置文件 wsclient-context.xml

Xml代码

```

1. <beans xmlns="http://www.springframework.org/schema/beans"
2.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3.     xmlns:jaxws="http://cxf.apache.org/jaxws"
4.     xsi:schemaLocation="http://cxf.apache.org/jaxws http://cxf.apache.org/schemas/jaxws.x
sd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/sp
ring-beans.xsd"
5.     default-autowire="byName" default-lazy-init="true">
6.
7.     <!-- ws client -->
8.     <bean id="identityValidateServiceClient" class="cn.org.coral.admin.service.IdentityVa
lidateService"
9.         factory-bean="identityValidateServiceClientFactory" factory-method="create"
10.     />
11.     <bean id="identityValidateServiceClientFactory"
12.         class="org.apache.cxf.jaxws.JaxWsProxyFactoryBean">
13.         <property name="serviceClass"
14.             value="cn.org.coral.admin.service.IdentityValidateService" />
15.         <property name="address"
16.             value="http://88.148.29.54:8080/coral/services/IdentityValidateService" />
17.     </bean>
18.
19. </beans>

```

6. 发布到tomcat服务器以后通过以下地址即可查看自定义的webservice接口生成的wsdl:

<http://88.148.29.54:8080/aio/services/WebServiceSample?wsdl>

7. 调用WebService接口的JUnit单元测试程序

Java代码

```

1. package test.coral.sample;
2.
3. import org.springframework.test.AbstractDependencyInjectionSpringContextTests;
4.
5. import cn.org.coral.biz.examples.webservice.WebServiceSample;
6. import cn.org.coral.biz.examples.webservice.dto.UserDTO;
7.
8. public class TestWebServiceSample extends
9.     AbstractDependencyInjectionSpringContextTests {
10.     WebServiceSample webServiceSampleClient;
11.
12.     public void setWebServiceSampleClient(WebServiceSample webServiceSampleClient) {

```

```

13.         this.webServiceSampleClient = webServiceSampleClient;
14.     }
15.
16.     @Override
17.     protected String[] getConfigLocations() {
18.         setAutowireMode(AUTOWIRE_BY_NAME);
19.         //spring 客户端配置文件保存位置
20.         return new String[] { "classpath:/cn/org/coral/biz/examples/webservice/wscli
nt-context.xml" };
21.     }
22.
23.     public void testWSClinet(){
24.         Assert.hasText(webServiceSampleClient.say(" world"));
25.     }
26. }

```

[学了Java，还找不到工作？](#)

为何学苹果手机开发工资高一倍？名企委托培养，本月报名学费75折...

[iphone.peixun.it](#)

Google 提供的广告

评论 共 7 条 [发表评论](#)

7 楼 [jayje](#) 2010-01-04 15:45 [引用](#)

能发个Demo给我吗？我学习中 非常需要高手的指点。万分感谢！liujie_1988@foxmail.com

6 楼 [TTLtry](#) 2009-12-05 13:47 [引用](#)

java的webservie还是第一次接触，学习java中如何调用webservice服务

5 楼 [ouly](#) 2009-11-03 08:15 [引用](#)

[u][u]参照**vsd**[b][b][color=red][color]



4 楼 [xiechunmei13](#) 2009-10-10 14:22 [引用](#)

记下一笔。学习中Ing.....

3 楼 [vions](#) 2009-08-07 12:42 [引用](#)

楼主引出了.net与java webservice的交互，要是能进一步谈谈两个平台的安全验证问题可能更有深度

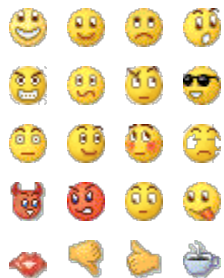
2 楼 [jiake0504](#) 2009-04-16 15:04 [引用](#)

挺爽，HO HO 正找这资料，还有这方面的东东，发我邮箱：jiake0504@hotmail.com,再次 谢谢。

非常不错! 谢谢了

发表评论

表情图标



字体颜色:

字体大小:

对齐:

提示: 选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

您还没有登录, 请[登录](#)后发表评论(快捷键 Alt+S / Ctrl+Enter)

[广告服务](#) | [JavaEye黑板报](#) | [关于我们](#) | [联系我们](#) | [友情链接](#)

© 2003-2010 JavaEye.com. 上海炯耐计算机软件有限公司版权所有 [[沪ICP备05023328号](#)]