(收藏)映射集合（Mapping collections）

XFire开发时，在返回数据类型时遇到了一些麻烦，查到这样一篇文单，非常不错，故收藏之。

翻译： zilong3927 原文地址： http://docs.codehaus.org/display/XFIRE/Mapping+collections
调用 Web Services 时，经常需要返回集合（ collection ）作为结果，或者接受 collection 型的参数。 SOAP 本身就支持这一点。
但是这一机制的问题在于， java 语言的 collections 是无类型的（ untyped ）. 因此，如果要在 Java 1.4 当中支持 collections ， 就需要做一些额外的工作。

Java 5 & 范型（Generics）

首先而且是推荐的做法是在 JDK5 当中使用范型（ generics ）。范型能够使你在代码当中为你的 collections 指定类型信息， 从而允许 xfire 自动地推导出 collection 类型，生成正确的 wsdl 等等。
下面示例了如何写这样的一个方法：

```
public Collection< String > getValuesForIds(Collection< Integer >);
```

Java 1.4 & 集合（Collections）

有些情况下并不总能够使用范型（ generics ）. 例如，如果你的部署环境使用 JDK 1.4 ， 或者你想暴露一些遗留的服务，而同时又不打算修改任何代码也不打算进行移植。
对于这样的一些情况而言， 你需要生成一个 xml 映射文件，来指定方法和它们对应的集合类型（ collection types ） .
这个 xml 文件的名字必须是 <className>.aegis.xml ， 其中 className 是你的服务（ service ）的接口类

< 2007年4月 >

日	一	二	三	四	五	六
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

常用链接

我的随笔
我的评论
我的参与
最新评论

留言簿(3)

给我留言
查看公开留言
查看私人留言

（unqualified class）的名字。
下面最好通过一个例子来展示这个 xml 文件的格式。 我们想要展现的服务有这样的一个接口：

```
public interface MyService1{
  String getFoo();
  Collection getCollection();
  void setList( int id ， java.util.List);
}
```

既然代码中的 collections 没有指定类型， 我们剧需要生成一个 xml 文件来指定所需要的类型。 这个文件的路径应该和 MyService1.class 在同一个包（package）当中， 并且它的名字应该是 MyService1.aegis.xml
对于这个接口来说，一个最简单的映射文件如下：

```
<mappings>
  <mapping>
    <method name= "getCollection" >
      <return-type componentType= "java.lang.String" />
    </method>
    <method name= "setList" >
      <parameter index= "1" componentType= "java.lang.String" />
    </method>
  </mapping>
</mappings>
```

注意这个映射文件确切地指定了所需要的信息，不包含任何冗余。 例如， getFoo 方法没有被指定，这是由于它没有包含任何 collections ，因此能够在没有任何映射信息的情况下暴露给使用者。
其次， setCollection 方法没有指定索引为 0 的参数。 这是由于该参数类型为 int ，因此不需要任何映射
如果我们有多个方法，都匹配指定的映射又该怎么办？这种情况下， 映射就对所有匹配的方法均有效。
所以，如果在我们的接口中增加以下的方法：

```
void setList(int id ，java.util.List，boolean persist);
```

那么现在我们的映射定义对于两个 setList 方法都有作用。这种情况下， 我们不必为额外的参数（译者注：此处指 boolean persist ）指定两次映射 . 映射文件就指定了所有那些第二个参数为 List 的方法，并假定 List 中包含的都是 strings 。
如果我们想让那个具有 3 个参数的方法，其中的 list 不包含 Strings ， 而是实际上包含 Dates? 这种情况下， 就需要一个更确切的映射来覆盖（override）原先那个更一般的， 所以我们的映射文件需要添加下面这个定义：

```
    <method name= "setList" >
      <parameter index= "1" componentType= "java.lang.String" />
```


随笔分类

Hibernate(2) XML
java 开发 (28) XML
Linux(13) XML
Spring(4) XML
web开发(19) XML
即时通讯(2) XML
大杂烩(10) XML
数据库(7) XML
软件设计(2) XML
问题记录(1) XML
项目管理(1) XML

随笔档案

2009年11月 (1)
2009年9月 (3)
2009年8月 (1)
2009年7月 (1)
2009年6月 (2)
2009年5月 (1)
2009年3月 (4)
2009年2月 (1)
2009年1月 (1)
2008年12月 (2)
2008年11月 (1)
2008年10月 (1)
2008年9月 (2)
2008年8月 (1)
2008年7月 (3)
2008年6月 (2)
2008年5月 (2)
2008年3月 (2)
2008年1月 (15)
2007年12月 (8)
2007年11月 (15)
2007年7月 (2)

```
    <parameter index= "2" class= "boolean" />
  </method>
```

注意一下类型属性。 现在这个映射将对所有那些第二个参数为 List ， 第三个参数为 boolean 型的方法适用。 在我们的接口当中，这个映射唯一地确定了一个特定的方法，使用这个映射就能够解释方法当中的 List 参数。
在优先顺序方面， 更确切的映射总是优先于更一般的。
让我们考虑下面这个复杂一些的例子：

```
public  interface  MyService2
{
  Collection getCollection(); //method 1
  Collection getCollection( int id); //method 2
  Collection getCollection( String id); //method 3
  Collection getCollectionForValues( int value ， Collection c); //method 4
  Collection getCollectionForValues( String id ， Collection c); //method 5
}
```

映射文件的内容为：

```
<mappings>
  <mapping>
    <!-- mapping 1 -->
    <method name= "getCollection" >
      <return-type componentType= "java.lang.Double" />
    </method>
    <!-- mapping 2 -->
    <method name= "getCollection" >
      <return-type componentType= "java.lang.Float" />
      <parameter index= "0" class= "int" />
    </method>
    <!-- mapping 3 -->
    <method name= "getCollectionForValues" >
      <return-type componentType= "java.math.BigDecimal" />
    </method>
    <!-- mapping 4 -->
    <method name= "getCollectionForValues" >
      <parameter index= "0" class= "java.lang.String" />
      <parameter index= "1" componentType= "java.util.Date" />
    </method>
```

2007年6月 (3)
2007年5月 (7)
2007年4月 (7)

收藏夹

EXT(5) XML
java 开发(9) XML
Linux(3) XML
开源(1) XML
搜索技术(1) XML

搜索

最新评论 XML

1. re: eclipse 生成javadoc乱码问题解决
用到，谢谢
--ADB
2. re: Linux启动级别&Samba服务自启动设置[未登录]
rwerqwrewqrwq
--1
3. re: Linux启动级别&Samba服务自启动设置[未登录]
asdas
--1
4. re: 利用Openssl 建立自己的证书。[未登录]
谢谢
--zhang
5. re: Tomcat JVM设置
谢谢了
--zgw

```
    <!-- mapping 5 -->
    <method name= "getCollectionForValues" >
      <return-type componentType= "java.util.Calendar" />
      <parameter index= "0" class= "int" />
      <parameter index= "1" componentType= "java.lang.Bit" />
    </method>
  </mapping>
</mappings>
```

这个文件的格式是不需要做过多解释的。但有几点还是需要加以说明。
先来看一下第一个映射 (mapping 1) 。 这个映射指定了所有 getCollection 方法所返回的 collections contain 均包含 java.lang.Doubles 。 如果没有指定其他的 getCollection 映射， 那么这个映射将对方法 1 ， 2 ， 3 都适用。
但是，第二个映射更加明确地指定了它所适用的方法。即如果 getCollection 方法的第一个参数是 int 型，那么该方法所返回的 collection 包含的是 Float 型。 由于这条规则更加明确，它将为方法 2 覆盖掉第一个映射，这是满足映射约束标准的。
使用以上的规则，不难推导出方法 4 和方法 5 返回的 collections 结果的组件类型（ component types ）。

Collections on Javabeans

对于使用 collections 的 java beans 来说，语法也是类似的。 例如，比方说我们有一个 Company bean ， 包含了一个 List ， 其中的对象是 employees:

```
public class Company
{
  private Collection employees;
  Collection getEmployees() { return employees; }
  public void setEmployees(Collection employees) { this .employees = employees };
}
```

除了可以使用 <method> & <parameter> 元素外， 也可以使用 <property> 元素 :

```
<mappings>
 <mapping>
  <property componentType= "org.codehaus.xfire.Employee" />
 </mapping>
</mappings>
```

Handling Maps

Java Maps 并不能很好地映射到 XML Schema (no pun intended)，因为 XML Schema 中没有 Map 的概念，客户端也是这样， Maps 被转换成 {key ， value} 元组的集合。 除了要提供 value 的类型以外，你还必须为 Aegis 提供 key 的类型 :

```
public  class GiftService {
   Map getGiftList() { /* returns a map of NiceChild => Present */ }
}
```

映射文件应该像下面这样 :

```
<mappings>
 <mapping>
  <method name= "getGiftList" >
   <return-type keyType= "org.codehaus.xfire.NiceChild" componentType= "org.codehaus.xfire.Prese
nt" >
  </method>
 </mapping>
</mappings>
```

这将生成下面的类型 :

```
<xsd:complexType name=  "NiceChild2PresentMap"  >
 <xsd:sequence>
  <xsd:element name= "entry" minOccurs= "0" maxOccurs= "unbounded" >
   <xsd:complexType>
    <xsd:sequence>
     <xsd:element name= "key" type= "ns1:NiceChild" minOccurs= "0" maxOccurs= "1" />
     <xsd:element name= "value" type= "ns1:Present" minOccurs= "0" maxOccurs= "1" />
    </xsd:sequence>
   </xsd:complexType>
  </xsd:element>
 </xsd:sequence>
</xsd:complexType>
```

Collections of Collections of Collections of....

在某些情况下，你可能想要传递 Collections of Collections 。比方说你有一个返回 List of a List of Doubles 的服务 (不要问为什么你要做这样一件事情 ...):

```
public class ListService
{
```

```
 public List getListOfListOfDoubles
 {
  List l = new ArrayList();
  List doubles = new ArrayList();
  doubles.add( new Double (1.0));
  l.add(doubles);
  return l;
 }
}
```

要处理这种情况，我们需要引进一个新的 <component> 元素。 下面是一个很好的例子 :

```
<mappings>
 <mapping>
  <method name= "getListofListofDoubles" >
   <return-type componentType= "#someDoubles" />
  </method>
  <component name= "someDoubles" class= "java.util.List" componentType= "java.lang.Double" />
 </mapping>
</mappings>
```

正像你在这里所看到的，返回类型的 componentType 是一个指向 <component> 的引用，而不是一个类。 组件类型 "#someDoubles" 引用到名字为 "someDoubles" 的 <component> 。
Aegis 将会自动给这些 collections 命名为 ArrayOfDouble 和 ArrayOfArrayOfDouble 。 你也可以改变这些名字。 要设置你自己的名字， 提供一个 "typeName" 属性即可 :

```
<mappings>
 <mapping>
  <method name= "getListofListofDoubles" >
   <return-type componentType= "#someDoubles" typeName= "LotsOfDoubles" />
  </method>
  <component name= "someDoubles" class= "java.util.List" typeName= "SomeDoubles" componentType= "java.lang.Double" />
 </mapping>
</mappings>
```

IT新闻:

· Gmail实验室关闭5项功能 6项功能正式启用

· 360回应阻拦QQ医生装补丁：不是所有补丁都必要

· 连线杂志：Google算法统治互联网的秘诀

· Google进入社交领域 庞大用户基数显示侵略性

· Firefox 3.7 Alpha 2将于本周发布

专题：Android iPad jQuery Chrome OS

标题 re: (收藏)映射集合（Mapping collections）

姓名

主页

验证码 * 3195

内容(请不要发表任何与政治相关的内容)

☑ Remember Me?

提交 登录

[使用Ctrl+Enter键可以直接提交]

每天10分钟，轻松学英语

博客园首页随笔:

· 推荐：做人的基本原则 — 温家宝
· 桌面程序界面设计分享
· [VSTS] Visual Studio 2010 自定义代码段新特性
· 在Asp.net MVC使用jqGrid--代码少点再少点
· 记住在php编写时容易范的错误！

知识库:

· 连线杂志：Google算法统治互联网的秘诀
· 妄想or未来？界面的虚拟现实化
· 从.NET说开去（我的.NET 4.0系列课程开讲）
· 有了HTML5，Flash还能走多远？
· 关于URL编码

相关文章:

通用业务引用查询服务实现
通过Findbugs 找出程序中隐藏的bug
java反编译工具
Eclipse 常用插件.
ibatis 自动生成的主键
项目构建工具之Maven
AXIS快速生成客户端调用文件
JSON 是什么东西？

LDAP/Directory 与数据库比较摘要
XML/JSON的类库，Xstream!