Atitit xml转json总结

目录

[1.1. XML和JSON之间没有直接映射；元素类型问题 1](#_Toc11955)

[1.2. Xml与json的对应关系 2](#_Toc12560)

[1.3. 范例 2](#_Toc12345)

[2. Jsonlib的问题，，不支持type属性 2](#_Toc6834)

[3. org.json也可以 推荐 2](#_Toc29532)

[4. Other 3](#_Toc29004)

[4.1. 通过转换map中转也麻烦，因为要建立bean，或者特定格式的 3](#_Toc12067)

[4.2. 测试一下可以转换简单的不能转复杂的，嵌套的xml 5](#_Toc8277)

## XML和JSON之间没有直接映射；元素类型问题

Fastjson冒失是潜入一个@type指明类型

XML带有类型信息（每个元素都有一个名称）以及命名空间。因此，除非每个JSON对象都嵌入了类型信息，否则转换将是有损的。

但这并不一定要紧。重要的是JSON的使用者知道数据协定。例如，给定以下XML：

<books>

<book author="Jimbo Jones" title="Bar Baz">

<summary>Foo</summary>

</book>

<book title="Don't Care" author="Fake Person">

<summary>Dummy Data</summary>

</book></books>

您可以将其转换为：

{

"books": [

{ "author": "Jimbo Jones", "title": "Bar Baz", "summary": "Foo" },

{ "author": "Fake Person", "title": "Don't Care", "summary": "Dummy Data" },

]

}

消费者不需要知道books集合中的每个对象都是一个book对象。

## Xml与json的对应关系

String xml = "<hello><test to='toUser@longbourn.lit/study' >1.2</test><test2>123</test2></hello>";

{"test":{"@to":"toUser@longbourn.lit/study","#text":"1.2"},"test2":"123"}

Xml元素通过json子对象模式实现，，xml属性通过 sjon属性@prop实现，其实我觉得直接抓换为json属性也可呀。。Xml标签内容，通过json属性#text来实现。。

## 范例

<message from="romUser@xxx.com">

<body>bodytxt</body>

</message>

[{"@from":"romUser@xxx.com","body":"bodytxt"}](mailto:{\"@from\":\"romUser@xxx.com\",\"body\":\"bodytxt\"})

# Jsonlib的问题，，不支持type属性

丢失root跟标签xml标签

# org.json也可以 推荐

支持type属性比较完美，支持root总标签，不丢失标签

{"message":{"from":"romUser@xxx.com","to":"toUser@longbourn.lit/study","type":"groupchat","body":"bodytxt"}}

# Other

## 通过转换map中转也麻烦，因为要建立bean，或者特定格式的

2. xml字符串 转JSON

    xml转JON 需要借助 jackSon的 fastxml包来实现

<!-- fasterxml -->

<dependency>

<groupId>com.fasterxml</groupId>

<artifactId>jackson-xml-databind</artifactId>

<version>0.6.2</version>

</dependency>

/\*\*

\* xml 转json

\*/

public static JSONObject convertXmlToJson(String xml) throws IOException {

XmlMapper xmlMapper = new XmlMapper();

JSONObject param = xmlMapper.readValue(xml, JSONObject.class);

return param;

}

利用XmlMapper这个类， 简单两段代码就可以搞定 但是该方法有个缺点， 如果你的xml字符串转换的内容里 有数组的情况下， 直接这样转是不行的。 这种情况就要自己建立相应的实体类来做接收。 注：实体要有相应的getset方法， 并要保证和xml中的字段名一一对应。

另外一种方式是使用org.json来实现，这种方式更简单，只需要两个jar包即可，下载地址[http://mvnrepository.com/artifact/org.json/json](http://mvnrepository.com/artifact/org.json/json" \t "https://www.cnblogs.com/111testing/p/_blank)，随便下载一个使用比较多的jar包版本即可，具体实现代码见下 

[IMG_256](https://www.cnblogs.com/111testing/p/javascript:void(0);)

public class JsonUtils {

public static String xml2jsonString() throws JSONException, IOException {

InputStream in = JsonUtils.class.getResourceAsStream("student.xml");

String xml = IOUtils.toString(in);

JSONObject xmlJSONObj = XML.toJSONObject(xml);

return xmlJSONObj.toString();

}

public static void main(String[] args) throws JSONException, IOException {

String string = xml2jsonString();

System.out.println(string);

}

}

[IMG_257](https://www.cnblogs.com/111testing/p/javascript:void(0);)

　　简单对比一下使用json-lib的实现方式，前面的代码基本一致，区别是这里使用的是org.json.XML类，调用的是toJSONObject方法，接受的是一个xml格式的字符串，生成一个JSONObject对象，这里也是一样，调不调用jsonobject的toString方法输出效果都一样，xml文件内容一样，输出的格式见下 

{"student":{"sex":"man","name":"zhangsan"}}

最后总结一下：   
　　１ json-lib依赖的jar包很多，需要全部集齐，org.json仅仅需要两个jar包即可实现，一个org.json另一个是commons-io   
　　2 两者输出的xml格式不同，前者不带根标签需要手动添加，会区别标签的属性和子标签，后者带有根标签，标签的属性和子标签不会区分对待，因此根据自己的实际情况自行选择修改即可。   
　　PS:如果还有其他的更好的xml转json方式，希望各位大神能告诉一下，再次先谢过了，那么这篇到此结束先了

## 测试一下可以转换简单的不能转复杂的，嵌套的xml

Atitit.xml转换map 通用对象的实现

**public** **class** XMLParser

**public** **static** Map<String,Object> **getMapFromXML**(String **xmlString**) **throws** ParserConfigurationException, IOException, SAXException {

//这里用Dom的方式解析回包的最主要目的是防止API新增回包字段

DocumentBuilderFactory **factory** = DocumentBuilderFactory.***newInstance***();

DocumentBuilder **builder** = **factory**.***newDocumentBuilder***();

InputStream **is** = Util.***getStringStream***(**xmlString**);

Document **document** = **builder**.***parse***(**is**);

//获取到document里面的全部结点

NodeList **allNodes** = **document**.***getFirstChild***().***getChildNodes***();

Node **node**;

Map<String, Object> **map** = **new** ***HashMap***<String, Object>();

**int** **i**=**0**;

**while** (**i** < **allNodes**.***getLength***()) {

**node** = **allNodes**.***item***(**i**);

**if**(**node** **instanceof** Element){

**map**.***put***(**node**.***getNodeName***(),**node**.***getTextContent***());

}

**i**++;

}

**return** **map**;

}

还有个。。wechatx4pay 。。。

思路，先转换为潜逃的map。。但是貌似只能手工转，这些框架都不能像自动转换

Exception in thread "main" net.sf.json.JSONException: java.lang.NullPointerException: Cannot invoke "String.compareToIgnoreCase(String)" because "type" is null

at net.sf.json.xml.XMLSerializer.read(XMLSerializer.java:331)

at hsimprjMavenAid.Xml2Json.xmppmsg(Xml2Json.java:32)

at hsimprjMavenAid.Xml2Json.main(Xml2Json.java:16)

Caused by: java.lang.NullPointerException: Cannot invoke "String.compareToIgnoreCase(String)" because "type" is null

at net.sf.json.xml.XMLSerializer.setValue(XMLSerializer.java:1236)

at net.sf.json.xml.XMLSerializer.processObjectElement(XMLSerializer.java:1095)

at net.sf.json.xml.XMLSerializer.read(XMLSerializer.java:322)

... 2 more

原因是因为是有个type属性，可能关键词重名了。。。

在Java中将XML转换为JSON的最快方法 - Thinbug