XStream两分钟教程(译)

Posted on 2007-05-08 18:14 qiyadeng 阅读(5093) 评论(7) 编辑 收藏

两分钟教程

这是一个关于XStream快速的介绍。快速浏览一下你会马上知道把一个对象转换到XML或是转换回来是多么的简单。你肯定会碰到下面的问题。

创建能够序列化的类

这是一组简单的类。XStream能把这些类的实例转换到XML或是转换回来。

public class Person {

private String firstName;

private String lastName;

private PhoneNumber phonex;

private PhoneNumber fax;

//构造函数或是其他方法

}

public class PhoneNumber {

private int code;

private int number;

//构造函数或是其他方法

}

注：注意到这些都是私有变量。XStream不关心变量的作用域。不需要getter或是setter方法。并且，XStream不限定需要默认的构造函数。

实例化XStream

使用XStream，简单实例化XStream类：

XStream xStream = new XStream();

你需要xstream-[version].jar and xpp3-[version].jar在classpath中。XPP3是一个非常快的XML拉式转换器工具。如果你不想包含这个依赖，你可以使用标准的JAXP DOM转换器来代替。

XStream xStream = new XStream(new DomDriver());//不需要XPP3库

注：这个类的简单设计是为了实现通用操作的。为了实现更复杂的操作你可以选择自己创建出不同方式。

现在，为了使用XStream来更精简的输出XML，你可以为自定义的类创建别名到XML的元素名的映射。这是使用XStream唯一需要的映射的，甚至这个都是可选的。

xStream.alians("person",Person.class);

注：这是可选的一步。没有这步XStream也可以很好的起作用，但是XML元素的名字就会包含每个类的全称(包括包名)，这将会使生成XML稍大。

序列号一个对象到XML

让我们创建一个Person的实例并且填充它的变量域：

Person joe = new Person("Joe","Walnes");

joe.setPhone(new PhoneNumber(123,"1234-456"));

joe.setFax(new PhoneNumber(123,"9999-999"));

现在转换到XML，你要做的是简单的调用XStream：

String xml = xstream.toXML(joe);

生成的XML看上去像这样：

<person>

<firstname>Joe</firstname>

<lastname>Walnes</lastname>

<phone>

<code>123</code>

<number>1234-456</number>

</phone>

<fax>

<code>123</code>

<number>9999-999</number>

</fax>

</person>

非常简单，像创建XML一样。

从XML反序列化一个对象

从XML重新构造一个对象：

Person newJoe = (Person)xStream.fromXML(xml);

XStream是多么的简单啊!

总结：

使用xStream.alias(String elementName, Class cls)为任何一个自定义类创建到类到元素的别名；

使用xStream.toXML(Object obj)转换对象到XML；

使用xStream.fromXML(String xml)转换XML到对象；