XMLBuilder开源代码：

https://github.com/cmz2013/xmlbuilder/

Java开发者会经常遇到创建不同的XML，一般都是逐个节点的去拼接，这种做法看似简单，但不同的XML需要不同的拼接算法，而且XML的结构改变时，必须修改算法。

XMLBuilder帮助Java开发者创建标准的XML，提高开发效率。使用XMLBuilder，程序员只需要设计XML模板，然后调用XMLBuilder提供的接口，即可创建想要的XML。以下是一个简单的XML模板：

<object\_0 attribute\_1=*""*>

<node1 attribute\_2=*""*>

<node2 attribute\_3=*""* text\_value=*""*/>

<node3>

<memobjects\_0 object\_data=*""*>

<memobject\_1 attribute\_4=*""* otherattr=*"normal attribute"*>

</memobject\_1>

</memobjects\_0>

<memobject\_2 object\_data=*""* attribute\_5=*""*>

</memobject\_2>

</node3>

</node1>

</object\_0>

1.节点object\_\*，\*代表节点名称，如果该节点为根节点则在模板中只有一个，对应一个数据实例（即Java对象）；如果该节点不为根节点则在模板中可以有多个，分别对应一个数据实例集。

2. object\_\*节点及子节点的属性：

1）attribute\_\*，\*代表属性名称，其值为object\_\*对应数据实例的属性名。按此方式定义的属性，程序在生成XML时其值将从对应的数据实例中获取。

2）text\_value，节点文本值，只有在叶子节点中才有效，其值为数据实例的属性名。如果数据实例该属性返回的是数据集，程序在生成XML时创建多个节点，依次设值。

3）其他普通属性。程序在生成XML时，普通属性的值直接从模板中获取。

3. 对象节点object\_\*的子元素：

1）memobjects\_\*，\*代表节点名称，对应一个数据实例或一个数据实例集，属性object\_data、attribute\_\*的值为object\_\*对应数据实例的属性名。属性object\_data返回值为memobjects\_\*对应的数据实例。如果memobjects\_\*对应的是一个数据实例集，程序在生成XML时会生成多个节点，每个节点分别对应一个数据实例。

2）memobject\_\*，\*代表节点名称，对应一个数据实例，属性object\_data、attribute\_\*的值为object\_\*对应数据实例的属性名。属性object\_data的返回值为memobject\_\*对应的数据实例。

3）其他普通元素。

4）memobjects\_\*节点和memobject\_\*节点也可以包含子元素：memobjects\_\*或memobject\_\*或其他普通元素，这些子元素及其对象属性对应的数据实例是父节点object\_data属性返回的数据实例。

4. 所有节点名称和属性名称，在实际应用中自定义。