[Jackson、JSON-lib、Gson性能对比](http://blog.csdn.net/jackpk/article/details/17141061)

2013-12-05 14:36 15133人阅读 [评论](http://blog.csdn.net/jackpk/article/details/17141061#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/jackpk/article/details/17141061#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

xml\_json（5） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

[**两款JSON类库Jackson与JSON-lib的性能对比(新增第三款测试)**](http://blog.csdn.net/blog/738933)

url：<http://wangym.iteye.com/blog/738933>

**最后更新：2010-08-18，增加对GSON的性能**[**测试**](http://lib.csdn.net/base/softwaretest)

近日做一些性能优化工作，在挑选JSON类库时，发现除了一般常用的JSON-lib外，还有一款号称性能最快的JSON处理器Jackson，于是用上了刚学会的JMeter，对这两个类库进行了简单的性能对比。

Jackson：<http://jackson.codehaus.org/>

JSON-lib：<http://json-lib.sourceforge.net/>

Gson：<http://code.google.com/p/google-gson/>

**测试环境：**

1、工作电脑：Intel双核E8400 共6GHz，内存4GB，WinXP

2、JSON-lib用最新的JDK15，GSON版本是最新的v1.4，Jackson也是最新的v1.5.5，JDK-v1.6.0\_20，JMeter-v2.4

3、测试时不开启任何无关进程，每完成一项测试后关闭JMeter整理内存后，再进行下一项测试，每项测试运行3次，取平均值

4、JSON转[**Java**](http://lib.csdn.net/base/java)Bean意为将JSON格式转换成[**Java**](http://lib.csdn.net/base/javase)类，这个类内包括Map、List、Date、Integer/Long/Double、String等类型的属性，Java Bean转Json则同理。另外，两者互转，每次转换的数据都是随机生成

**测试结果：**

\* 吞吐量的值越大越好，总耗时的值越小越好

**JSON转Bean，5个线程并发，约200字节对象，1千万次转换：**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Jackson | JSON-lib | Gson |
| TPS | 64113.7 | 8067.4 | 13952.8 |
| 总耗时(秒) | 155 | 1238 | 700 |

**Bean转JSON，5个线程并发，约200字节对象，1千万次转换：**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Jackson | JSON-lib | Gson |
| TPS | 54802 | 15093.2 | 17308.2 |
| 总耗时(秒) | 181 | 661 | 560 |

**JSON转Bean，5个线程并发，约2K对象，1千万次转换：**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Jackson | JSON-lib | Gson |
| TPS | 37314 | 2406.9 | 3657.50 |
| 总耗时(秒) | 267 | 4120 | 2720 |

**Bean转JSON，5个线程并发，约2K对象，1千万次转换：**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Jackson | JSON-lib | Gson |
| TPS | 30922.2 | 4274.8 | 4977.00 |
| 总耗时(秒) | 322 | 2320 | 2000 |

**测试总结：**

1、**显而易见，无论是哪种形式的转换，Jackson > Gson > Json-lib。**

**Jackson的处理能力甚至高出Json-lib有10倍左右**

2、JSON-lib似乎已经停止更新，最新的版本也是基于JDK15，而Jackson的社区则较为活跃；

3、**在测试性能的同时，又以人肉方式对这三个类库转换的正确性 进行了检查 ，三者均达100%正确**；

4、JSON-lib在转换诸如Date类型时较为累赘，如以下是两者的转换结果：

***JSON-lib：***

{"brithday":{"date":17,"day":2,"hours":9,"minutes":24,"month":7,"seconds":26,"time":1282008266398,"timezoneOffset":-480,"year":110}}

***Jackson：***

{"brithday":1282008123101}

5、JSON-lib依赖commons系列的包及ezmorph包共5个，而Jackson除自身的以外只依赖于commons-logging  
6、Jackson提供完整基于节点的Tree Model，以及完整的OJM数据绑定功能。

**Jackson使用示例：**

**JacksonMapper：**

创建为**饿汉式单例模式** ，Jackson用于转换的核心类ObjectMapper无需每次都new一个object，官网上的一句话：can reuse, share globally

Java代码

1. /\*\*
2. \* @author xuanyin
3. \*
4. \*/
5. **public** **class** JacksonMapper {
7. /\*\*
8. \*
9. \*/
10. **private** **static** **final** ObjectMapper mapper = **new** ObjectMapper();
12. /\*\*
13. \*
14. \*/
15. **private** JacksonMapper() {
17. }
19. /\*\*
20. \*
21. \* @return
22. \*/
23. **public** **static** ObjectMapper getInstance() {
25. **return** mapper;
26. }
28. }

**JSON转Bean：**

Java代码

1. ......
2. String json = "...";
3. ObjectMapper mapper = JacksonMapper.getInstance();
4. YourBean bean = mapper.readValue(json, **new** YourBean().getClass());
5. ......

**Bean转JSON：**

Java代码

1. ......
2. YourBean bean = **new** YourBean();
3. ......
4. ObjectMapper mapper = JacksonMapper.getInstance();
5. StringWriter sw = **new** StringWriter();
6. JsonGenerator gen = **new** JsonFactory().createJsonGenerator(sw);
7. mapper.writeValue(gen, bean);
8. gen.close();
9. String json = sw.toString();
10. ......

\* 上面两段代码中的YourBean当然也可以是Java的基本类型

文章不足之处欢迎大家留言指正：)

11 楼 [wangym](http://wangym.iteye.com/) 2010-08-18   [引用](http://blog.csdn.net/jackpk/article/details/17141061)

wangliang\_5290 写道

wangym 写道

wangliang\_5290 写道

我碰到一个问题，要将字符串  
String json = "{a:1, b:2}";  
转化为 Map, 如果用Jackson的话，对字符串格式有严格要求，必须为  
String json = "{\"a\":1, \"b\":2}";  
否则报错，觉得不爽。  
  
不知道是对Json了解不够呢， 还是Jackson就是这么要求的？

在JAVA里定义这样一个String自然是需要"\"给后面的引号进行转义，无论是用哪个JSON类库都一样，但在实际应用中，没有这个问题。

实际应用中json串不一定是在java中定义的，有可能是从页面传过来的，java只是负责解析。我以前用的是json-lib没有问题， 但是换成jackson后出现这个问题。不知道是不是对jackson的API研究不够？  
1.json-lib 以下写法没有问题

Java代码

1. Map map = JSONObject.fromObject("{a:1, b:2}");

2.jackson 以下写法报异常

Java代码

1. Map map = **new** ObjectMapper().readValue("{a:1, b:2}", HashMap.**class**);

必须写成

Java代码

1. Map map = **new** ObjectMapper().readValue("{\"a\":1, \"b\":2}", HashMap.**class**);

不好意思一开始没明白你的意思  
请开启支持字段名称不带引号的转换模式,默认是关闭的.  
即将上面报异常的那句改成这样：

Java代码

1. Map map = **new** ObjectMapper().configure(Feature.ALLOW\_UNQUOTED\_FIELD\_NAMES, **true**).readValue("{a:1, b:2}", HashMap.**class**);

24 楼 [wangym](http://wangym.iteye.com/) 2011-01-18   [引用](http://blog.csdn.net/jackpk/article/details/17141061)

JSON转POJO时，若JSON中的某个字段在POJO中未定义，在默认情况下会抛异常转换失败，只要增加这个配置：

Java代码

1. Feature.FAIL\_ON\_UNKNOWN\_PROPERTIES, **false**    
   就可以转换正常，即只转换定义的字段，未定义的字段忽略。