



链滴

# fastJson,jackJson,Gson 序列化和反序列化实例

作者: [jsy](#)

原文链接: <https://ld246.com/article/1495590418814>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 直接看实例

```
import java.util.Map;
import org.codehaus.jackson.map.ObjectMapper;
import com.alibaba.fastjson.JSONObject;
import com.google.gson.Gson;

public class JsonTest {

    private static ObjectMapper objectMapper = new ObjectMapper();

    private static Gson gson=new Gson();

    private static JSONObject jsonObject=new JSONObject();

    private static long count=10000;
    /**
     * @param args
     * @throws Exception
     */
    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub
        gson();
        System.gc();
        jackJson();
        System.gc();
        fastJson();
        System.gc();
        System.out.println("-----");
        gsonJson();
        System.gc();
        jackJsonJson();
        System.gc();
        fastJsonJson();
    }

    private static long gson(){
        Demo demo=new Demo();
        demo.setCmd("1");
        demo.setContent("dd");
        long endTime1=System.currentTimeMillis();

        for(int i=0;i<count;i++){
            gson.toJson(demo);
        }
        long endTime2=System.currentTimeMillis();
        System.out.println("gson--obj--json:"+(count*1000)/(endTime2-endTime1));
        return endTime2-endTime1;
    }

    private static long jackJson() throws Exception{
```

```

Demo demo=new Demo();
demo.setCmd("1");
demo.setContent("dd");
long endTime1=System.currentTimeMillis();
for(int i=0;i<count;i++){
    objectMapper.writeValueAsString(demo);
}
long endTime2=System.currentTimeMillis();
System.out.println("jackJson--obj--json:"+(count*1000)/(endTime2-endTime1));
return endTime2-endTime1;
}

```

```

private static long fastJson(){
    Demo demo=new Demo();
    demo.setCmd("1");
    demo.setContent("dd");

    long endTime1=System.currentTimeMillis();
    for(int i=0;i<count;i++){
        jsonObject.toJSONString(demo);
    }
    long endTime2=System.currentTimeMillis();
    System.out.println("fastJson--obj--json:"+(count*1000)/(endTime2-endTime1));
    return endTime2-endTime1;
}

```

```

private static long gsonJson(){
    String json="{\"cmd\": \"1\", \"sss\": \"dd\"}";

    long endTime1=System.currentTimeMillis();

    for(int i=0;i<count;i++){
        gson.fromJson(json, Map.class);
    }
    long endTime2=System.currentTimeMillis();
    System.out.println("gson--json--obj:"+(count*1000)/(endTime2-endTime1));
    return endTime2-endTime1;
}

```

```

private static long jackJsonJson() throws Exception{
    String json="{\"cmd\": \"1\", \"sss\": \"dd\"}";

    long endTime1=System.currentTimeMillis();
    for(int i=0;i<count;i++){
        objectMapper.readValue(json, Map.class);
    }
    long endTime2=System.currentTimeMillis();
    System.out.println("jackJson--json--obj:"+(count*1000)/(endTime2-endTime1));
    return endTime2-endTime1;
}

```

```

private static long fastJsonJson(){

```

```

String json="{\"cmd\": \"1\", \"sss\": \"dd\"}";

long endTime1=System.currentTimeMillis();
for(int i=0;i<count;i++){
    jsonObject.parseObject(json, Map.class);
}
long endTime2=System.currentTimeMillis();
System.out.println("fastJson--json--obj:"+(count*1000)/(endTime2-endTime1));
return endTime2-endTime1;
}

}

class Demo {
    private String cmd;
    private String content;
    public String getCmd() {
        return cmd;
    }
    public void setCmd(String cmd) {
        this.cmd = cmd;
    }
    public String getContent() {
        return content;
    }
    public void setContent(String content) {
        this.content = content;
    }
}

```

## 总结

把Java对象JSON序列化，Jackson速度最快，在测试中比Gson快接近50%，FastJSON和Gson速度近。

把JSON反序列化成Java对象，FastJSON、Jackson速度接近，Gson速度稍慢，不过差距很小。

参考[知乎](#)