



链滴

json 文件读写 (java 对象和 json 文件互相转换)

作者: [JellyfishMIX](#)

原文链接: <https://ld246.com/article/1605146212088>

来源网站: [链滴](#)

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

前言

最近遇到了业务需求，java对象转换为json文件，json文件转换为java对象。这个需求可以拆分为：

1. json 序列化反序列化
2. java IO

json 序列化反序列化我们使用 alibaba 的 fastjson，很好用。

直接看demo代码吧。

依赖

```
<!-- fastjson https://mvnrepository.com/artifact/com.alibaba/fastjson -->
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.56</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.12</version>
</dependency>
```

代码

实体类 Person (使用了lombok依赖)

三个 lombok 注解必须加，如果未使用 lombok，请在此实体类加 setter & getter，全参构造方法无参构造方法。

```
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Person {
  private Integer id;
  private String name;
  private Integer age;
}
```

object2JsonFile

```
/**
 * Object 转换为 json 文件
 *
 * @param finalPath finalPath 是绝对路径 + 文件名，请确保欲生成的文件所在目录已创建好
 * @param object 需要被转换的 Object
```

```

*/
public static void object2JsonFile(String finalPath, Object object) {
    JSONObject jsonObject = (JSONObject) JSON.toJSON(object);

    try {
        OutputStreamWriter osw = new OutputStreamWriter(new FileOutputStream(finalPath), StandardCharsets.UTF_8);
        osw.write(jsonObject.toJSONString());
        osw.flush();
        osw.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    System.out.println(jsonObject.toJSONString());
}

```

jsonFile2Object

```

/**
 * json 文件转换为 Object
 *
 * @param finalPath finalPath 是绝对路径 + 文件名，请确保欲生成的文件所在目录已创建好
 * @param targetClass 需要被转换的 json 对应的目标类
 * @param <T> 需要被转换的 json 对应的目标类
 * @return 解析后的 Object
 */
public static <T> T jsonFile2Object(String finalPath, Class<T> targetClass) {
    String jsonString;
    File file = new File(finalPath);
    try {
        FileInputStream inputStream = new FileInputStream(file);
        int size = inputStream.available();
        byte[] buffer = new byte[size];
        inputStream.read(buffer);
        inputStream.close();
        jsonString = new String(buffer, StandardCharsets.UTF_8);
        T object = JSON.parseObject(jsonString, targetClass);
        return object;
    } catch (IOException e) {
        e.printStackTrace();
        throw new RuntimeException("IO exception");
    }
}

```

测试类的方法（maven项目结构中的测试类）

```

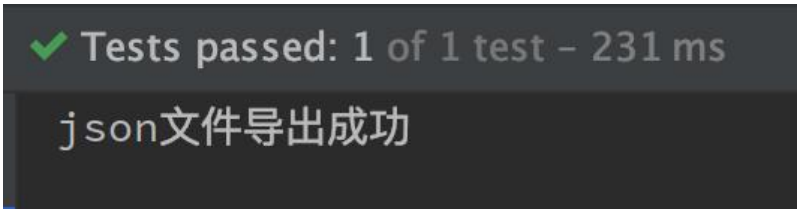
@Test
void object2JsonFile() {
    Person person = new Person(22, "王多鱼", 19);
    String finalPath = "/Users/qianshijie/Temporary/skycomm/devsyn/test.json";
    JsonUtil.object2JsonFile(finalPath, person);
}

```

```
@Test
void jsonFile2Object() {
    String finalPath = "/Users/qianshijie/Temporary/skycomm/devsyn/test.json";
    Person person = JsonUtil.jsonFile2Object(finalPath, Person.class);
    System.out.println(person.toString());
}
```

运行结果

根据 java 对象生成 json 文件成功:



生成的json文件 (可使用vim查看) :

```
{"name": "王多鱼", "id": 22, "age": 19}
~
~
```

读取 json 文件转换为 java对象 成功:

