



链滴

# JDK14 中 record 实践尝鲜

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1586169476630>

来源网站: [链滴](#)

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



## 他来了他来了! 小刀终于更新了

他来了他来了,不好意思, 他来晚了. 大家好, 我是小刀,前几天看到朋友圈被jdk14的GA版本刷屏了,我直忍到现在才写,,,唉,实在是惭愧惭愧.

按道理讲啊,其实也不算晚吧, 因为之前写过一篇jdk14中 `jpackage`命令的使用. 原文如下:

<https://www.lixiang.red/articles/2020/03/01/1583033948949.html>

好,下面我们就进入今天的主题-jdk14中的record的使用

## 使用形式

其实要是看代码编写的话,就一行

```
// 定义
record Image(@SerializedName("media_id") String mediaId){}
// 使用
Image image = new Image("传mediaId");
// 取里面的值
image.mediaId();
```

如上我定义了一个record,Image,里面有一个String mediaId 那按以前的写法呢,大致是这样的

```
static class ImageClass{
    // 定义一个属性
    private String mediaId;
    // 定义一个构造函数
    public ImageClass(String mediaId) {
        this.mediaId = mediaId;
    }
}
```

```
// 定义一个get方法
public String getMediaId() {
    return mediaId;
}
}
```

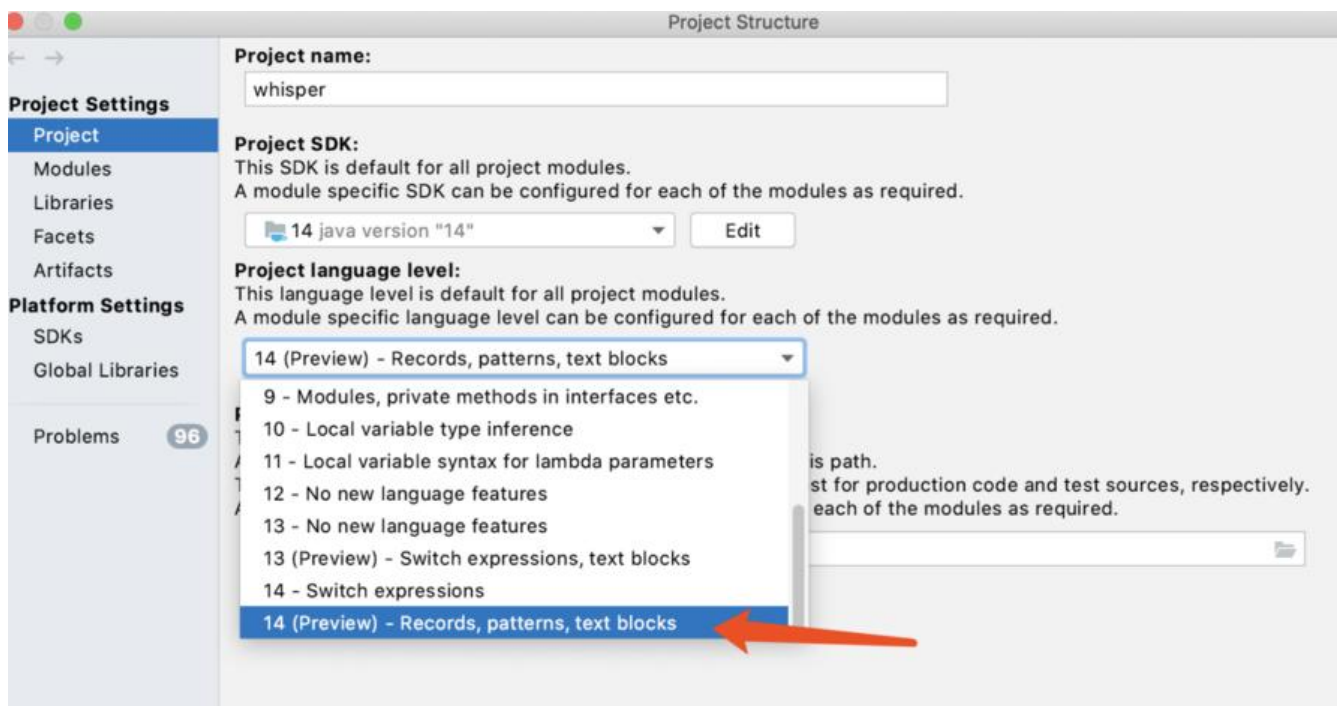
但是现在!!! 现在只要一行就可以搞定了!!! 真得是相见恨晚的感觉啊!!! 在做微信公众号开发的小伙伴该有这种体验,那么多的嵌套实体类. 用map写吧, 不雅观, 用 static class 写吧, 马上一个文件就要好几行了. 这时候如果领导同意的话,可以试试 jdk14 中的 record哦~~ (话说,,我们公司现在连jdk11还没上,你们呢?)

## 运行配置

因为这个功能还在预览当中,所以需要一些额外的配置才能跑起来

## idea

idea 需下载2020.1 RC 版本,然后在项目设置中,选择语言级别为14-preview,如下图所示



## gradle

gradle需要升级到 6.3 及以后才支持jdk14, maven未验(已经很久没用maven了,gradle真香)

可使用如下命令进行升级:

```
./gradlew wrapper --gradle-version=6.3
```

## build.gradle

在gradle的脚本文件中需要添加如下配置

```

sourceCompatibility = '14'

tasks.withType(JavaCompile) {
    options.compilerArgs += "--enable-preview"
}

tasks.withType(Test) {
    jvmArgs += "--enable-preview"
}

tasks.withType(JavaExec) {
    jvmArgs += '--enable-preview'
}

```

## 最后一步把fastJson换成gson

下面我们用一個main函数演示一下

```

public static void main(String[] args) {
    record OutRecord(String name){
    }
    OutRecord outRecord = new OutRecord("xiaodao");
    // 直接对record序列化,都没有结果
    Gson gson = new Gson();
    System.out.println("gson:" + gson.toJson(outRecord));
    System.out.println("fastJson:" + JSON.toJSONString(outRecord));

    // 外面用个类包装一下
    Holder holder = Holder.init("xiaodao");
    System.out.println("gson:" + gson.toJson(holder));
    System.out.println("fastJson:" + JSON.toJSONString(holder));
}

// 来于包装的类
static class Holder{
    // 声明一个record
    InnerRecord record;
    // 定义一个record
    record InnerRecord(String name){}

    /**
     * 推荐这种实例过种封状在类里面的写法
     * @param name
     * @return
     */
    public static Holder init(String name){
        Holder holder = new Holder();
        InnerRecord innerRecord = new InnerRecord(name);
        holder.record = innerRecord;
        return holder;
    }
}

```

输出结果如下:

```

/Users/lixiang/soft/jdk-14.jdk/Contents/Home/bin/java
gson:null
fastJson:{}
gson:{"record":{"name":"xiaodao"}}
fastJson:{}

```

可以看到,不管是直接record还是外面有包装类的情况下, fastJson都不能序列化,但gson可以在外面有装类的情况下进行序列化

## 撒花撒花! 本篇完结

关于record, 小刀觉得还是挺好用的, 只作值对象的话,几乎可以取代静态内部类了,代码简洁.嗯,,,可以自己实践下哦

今年的大环境确实不好, 不仅仅是应届的不好找工作, 我们在职的也天天提心吊胆. 但是呢, 这又不能和人说,只能自己背负着前行, 别都压在自己心里了,来这里说一说吧. 全匿名的世界, 不用担心别人怎么看你 你就是你, 心友们都在这等你🍓heart

感谢黑客派的各位兄弟姐妹们的支持~~!!

