



黑客派

# 0622 公司项目 CodeReview 问题及总结

作者: [seanlee](#)

原文链接: <https://hacpai.com/article/1498612377589>

来源网站: 黑客派

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

```
<div>
  &nbsp;
</div>
<div>
  <span>后台代码部分</span>
</div>
<div>
  1. 枚举类的使用：当定义的某个字段有多种状态值或表示值时，可以创建一个枚举类将所有状态表
  出来，并供其他使用到该字段的方法统一调用。枚举类定义可参考 '/banking-dbbp-product-comp/
  rc/main/java/com/vivebest/banking/dbbp/product/comp/Enums/ProductTypeEnum.java'。
</div>
<div>
  2. 异常处理，在调用其它项目接口时一定要进行异常处理，在抛出异常时需要使用logger.info()对
  常进行打印，异常处理格式如下，
</div>
<div>
  &nbsp;
  <a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Foqb6m3iar.bkt.clouddn.co
  %2Ff2f186b300e2466486cb36f545acf4a2.jpg" class="fancybox" target="_blank" rel="nofollow
  ugc"></a>
</div>
<div>
  3. 日志打印不用转换成json对象再输出，如果是打印实体类对象在编写时直接构建toString()方法即
  ，方式是 右键实体类--&gt; Source --&gt; Generate toString()。
</div>
<div>
  4. 后台日志打印主要有info和debugger两种形式，在进入后台Action时使用logger.info()打印信息
  在抛出异常前使用logger.info()打印信息，调用其它方法是使用logger.debugger()来打印信息。inf
  主要是用来判断参数信息是否正确，debugger主要用来查看程序走到了哪一步。
</div>
<div>
  5. 打印日志时要使用占位符 '{'，不建议使用字符串拼接 '+' 的形式，使用 '+' 是字符串拼接
  会浪费系统资源，后期调整为error级别时，仍会进行字符串的运算，在进行压力测试的时候或者是高
  发的情况下，影响系统的性能。使用 ',' 符进行占位，效率比较高，方便日后日志筛选与查看。
</div>
<div>
  6. 自定义异常处理时，异常的定义要规范化，同时要包含一定的业务含义，使代码具有可读性。
</div>
<div>
  7. 有返回值的实体类或者List和Map都不能使用New()操作，直接在定义时直接初始化=null即可，
  ： List resultList = null或者直接 List<PublishAuditor> resultList = publishAuditorService.q
  ery(publishAuditor);使用New()操作会消耗额外资源。
</div>
<div>
  8. 删除操作、插入操作和更新操作都不需要有返回值接收，查询操作时选择适当的容器来进行接收
  询数据。
</div>
<div>
  9. 日期类型的转换有公共的Utils类中有方法可以直接调用，可以避免手写转换，在第三方jar包 org.
  pache.http.client.util.DateUtils中可直接时候，使用方法为 Date date = DateUtils.parseDate()。
</div>
<div>
```

10. 在Action执行完数据库操作后向前台返回数据时，不需要手动自定义标识符，系统框架会自动回前台可识别的标识符。不需要特意putResponse出执行成功或失败的标识，平台级的报文中会有res ltType为N的，同时前端会对错误进行处理的。

</div>

<div>

11. 当从前台接收到的某个字段需要经常做转换或者处理时，可以在定义接收该字段的实体类中的Set )方法中进行处理，避免了在其他层频繁处理。

</div>

<div>

12. Action层主要负责业务和模块的组装，用来写业务的流程。Service层用来进行具体的业务操作数据处理。当Action接收到参数时，在Action进行数据的组装。Action层处理大的业务逻辑，即主要务逻辑流程的串接，Service层做具体的小业务的流程，以及数据的处理等

</div>

<div>

13. 在使用MyBatis在XML里面编写SQL语句时，需要联表查询并编写多个Join语句时，应该把每个Jo n语句单独成行，方便一眼看出关联了哪些表。如下：

</div>

<div>

&nbsp;

<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Foqb6m3iar.bkt.clouddn.co %2F93042d6b8cd342febab83539f3411fdd.png" class="fancybox" target="\_ blank" rel="nofoll w ugc"></a>

</div>

<div>

14. 对于数据的组装和赋值，应多使用工具类，如dozzer 或者beanUtils中的些对象之间的赋值，尽 减少set方法。使代码具有简洁、可读性。

</div>

<div>

&nbsp;

</div>

<div>

<span>前端代码</span>

</div>

<div>

1. 前端的组件如tab id或者name进行命名的时候要有明确含义，尽量避免tab-1、tab-2这种无意义命名。

</div>

<div>

2. 编写JS文件时要在对应页面下创建JS文件夹，并且JS文件也要有适当的注释，关键字段和方法也 加上注释。

</div>

<div>

3. 在一个页面有多个Tab展示不同的加载数据时，可以给tab添加定时器，在点击一个tab后对后面的 ab进行预加载，不必一次性加载全部数据，也不必只有点击之后才加载数据。

</div>

<div>

4. 当前端页面多次用到同一个组件的某个属性时，可以添加一个class id，并在CSS或通过Jquery来 行统一控制。

</div>

<div>

&nbsp;

</div>

```
<div>
  &nbsp;
</div>
<div>
  &nbsp;
</div>
```