



链滴

按业务拆分模块的疑惑

作者: [xinyue](#)

原文链接: <https://ld246.com/article/1581133517099>

来源网站: 链滴

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

前情概要

打算做一个 Java admin 快速开发框架；将常用的库、权限管理、用户管理...啥的集成，方便平做点小项目。

项目结构

-

- xinyue-platform 是 parent 项目，类型是 pom

- xinyue-admin 是 Web 项目，用来最后打包运行的后台管理服务

- xinyue-business 是所有的业务逻辑模块，模块内按照业务分包(业务包内分层)，如图：

-

- xinyue-core 是项目核心模块，包含一些 base 基类等等

- xinyue-common 作为公共模块，放一些常量、工具类、枚举啥的

- xinyue-generator/xinyue-task 暂时可以忽略，还没设计 joy:

依赖关系

-

- xinyue-admin 依赖 xinyue-business

- xinyue-business 依赖 xinyue-core

- xinyue-core 依赖 xinyue-common

设计初衷

因为未来比如用这个 platform 做电商项目，可能还需要对外提供 xinyue-api-h5/xinyue-api-a plet 一类的服务；所以希望可以将业务逻辑下沉到单独的模块，可以供 admin(后台管理系统)和各终端 API 调用

这也是为什么 xinyue-business 模块内部，分层的时候只有 entity、repository、service 层的因了，因为具体这些服务对外提供的接口；可能路径不同，也可能需要在不同的 Web 层在进行一些务逻辑

所以 controller 层被单独拿到了 xinyue-admin 模块(也有可能是未来的 xinyue-api-xxx 模块)如图：

疑问

目前的设计存在哪些问题？

-

- 由于自己还在上学，没接触过大型项目，都是在 Google 和开源项目中学习了一些思路

- 自己根据自己的业务特点设计了如上的结构，不知道是否标准，或者说存在着哪些问题呢

- 之前参考的项目多是按业务模块划分，模块内 entity、repository、service、controller 分层，些疑惑：

-

- 比如分为 user 模块和 order 模块，不采用微服务，那么最终还是要打包到某个 Web 服务中发运行的

- 可是如果 order 模块某些逻辑需要依赖 user 模块，直接在 order.pom 中添加 user 的依赖会有点怪怪的

- 并且，这样岂不是 maven 会有重复依赖的问题，比如 order 和 user 都有 springBoot-starer-w b/data--jpa 的依赖，order 再依赖 user，那么 springBoot-starer-web/data--jpa 不就重复依赖了？

<h3 id="business-按业务分包是否可以直接拆分为-xxx-模块">business 按业务分包是否可以直接分为 xxx 模块</h3>

有的时候在想，目前的 business 内按照业务分包，会不会导致未来各个业务包之间边界越来越糊的问题(同在一个模块，可以直接互相调用其他包)

比如同上的情况；order 的某些逻辑依赖 user，那么可以直接在 orderService 中调用 user 提的 service/repository，没有任何壁垒

那么各个业务包是否丧失了高内聚、低耦合的核心

若拆分为各个业务模块，又是否能解决这些问题

<h3 id="项目中的依赖关系如何管理">项目中的依赖关系如何管理</h3>

<p>参考、分析了 4、5 个 GitHub 上高分的类似快速开发平台，发现这些项目被 Idea 的 Maven 提 omitted for duplicate，其实也是我第一个问题里面提到的重复以来的问题；提示 maven 应该会自动处理，主要就是心里面有点想不通，这样会不会有隐患</p>

<p></p>

<p></p>

<h3 id="结语">结语</h3>

<p>希望各位可以给一些参考意见，小弟不胜感激</p>