一、为什么项目分层?

- 项目分层

  将项目工程结构标准化，对结构进行分层，提高项目可读性，降低维护成本

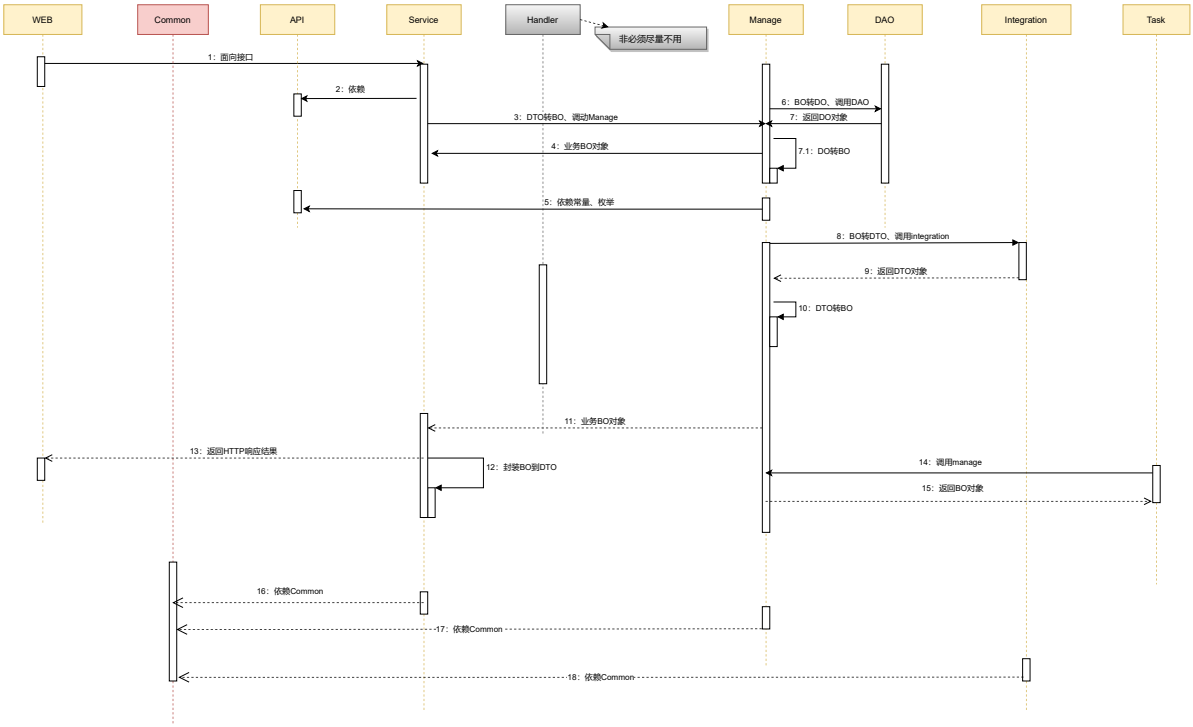
- 代码分层

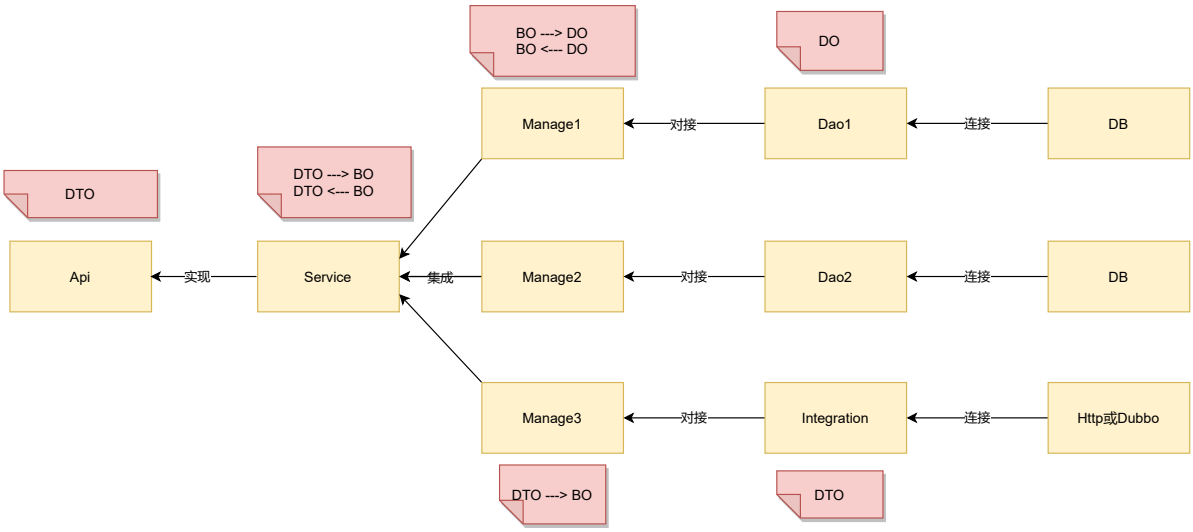
  从对象的维护来看项目分层

二、项目分层

项目分层名词	词解释
API层	提供标准的接口、常量和枚举。提供统一对外的接口。可以单独打jar包供外部系统使用。标准入参XXXRequestDTO，标准出参XXXResponseDTO。（一本书的目录）
Service层	API层接口的实现层；入参XXXRequestDTO转换为XXXRequestBO对象，透传到下一层作为入参；业务处理结果XXXResponseBO转换为XXXResponseDTO对象，透传到上层作为返回结果。（一章的目录）内部分为biz（在线业务处理层）和task（离线业务处理层）。
Manage层	外部系统的衔接层，获取外部接口的返回结果；1、对接DAO：一个managee对应一个DAO，包含对一个表的各种操作；2、对接integration：一个manage对应一个integration，获取外部接口得到结果。如果把DB，Redis，HTTP，RPC等都看成一个接口，Manage层就是为了这些外部系统的出入参做准备的。类似于“对外的一个Service层”。
DAO层	数据库交互层，与Redis、DB等数据库进行交互。DO对象
Integration层	外部系统集成层，HTTP、RPC等外部系统调用从这里开启，获取外围系统的数据。
Web层	对外暴露Http服务层。
Task层	暴露定时任务
Common层	封装常量、枚举供Service、Manage、Integration层调用

> 各个Module内部需要按照业务功能进行划分，划分成不同的package。





三、代码分层

各种对象名词解释

- DTO：Data Transfer Object，数据传输对象，用于系统之间数据交互的一种对象。系统对外提供的接口的出参和入参需要定义为DTO对象；调用外部系统的接口的出参和入参需要定义为DTO对象。
- BO：Business Object，业务对象。系统业务逻辑处理及内部层级调用所用的一种对象。转换方式：接口入参DTO对象--->业务逻辑BO对象；业务逻辑BO对象--->接口出参DTO对象。
- DO对象：Data Object，数据对象。把数据库当成一个外部系统，此系统与Java应用之间进行数据交互。DO是直接面向数据库交互的一种对象。转换方式：业务逻辑BO对象--->数据库DO入参对象；数据库DO出参对象--->业务逻辑BO对象。

四、目标及优势

结构清晰、层次分明、可读性强、易维护、易扩展、方便定位、边界清晰、职责清晰、可复用。

五、快速搭建脚手架

1、快速构建脚手架：https://www.jb51.net/article/249956.htm

2、如何在IDEA中添加maven的archetype：https://blog.csdn.net/qq_52827181/article/details/121479427