Delmar\_core : 放置框架、平台级别的代码 我们认为这是最底层的框架

Delmar\_system : 放置用户、角色、权限、操作日志等 系统应用级别的

Delmar\_base : 基础数据，我们的一个想法是可以所有的基础数据都放在这里面，无论以后拆解哪一个模块单独使用，也可以全部包含delmar\_base 但可以屏蔽掉某些基础代码维护的菜单键。

Delmar\_common: 放置基本上大部分模块都要用到的功能，比如说附件上传/下载、邮件、

这里面方式的基本上都是每一个业务都能够用到的功能，但和某个具体的业务又没有关系。

Delmar-utils : 工具类的，纯粹工具类的，和业务一点关系都没有

Delmar\_doc： 文档

Delmar\_cargo :核心业务，但这里cargo的概念有些小了，但之前代码已经写了，所以也不去进行改动了，这里面会放置比如：CRM、报价、货代系统、仓储系统等

这里面主要为JAVA代码，不包含前端代码

Delmar\_web: 前端代码，这里需要注意的是web当中关于业务的划分一定要和delmar\_cargo 保持一致，即delmar\_cargo划分几个在delmar\_web当中就要划分几个，即数量delmar\_cargo>=Delmar\_web， 但这里需要注意的是delmar\_web当中也包含了delmar\_base以及delmar\_core里面的前端代码实现。

Delmar\_ws : 我们自身的web\_service接口，目前没有按照模块划分，这里也需要按照模块划分，后续需要把目前完成的web\_service进行整理

Delmar\_devs: Development 开发过程当中做的一些工具、提高开发效率的，以及部署的代码生成的工具等，

Delmar\_expimp ：相应数据库结构的初始化、数据表数据的初始化以及后续相应的备份操作，但我认为主要的还是数据结构和数据表数据的初始化，如此这里面便涉及到相应的安装问题。

Delmar\_mobile : 类似delmar\_web 但主要应用于移动端的访问

这里已经形成一个概念为：对于web端和手机端，他们后台的JAVA代码可以公用，但是前端以及control都是各自单独的。