# **controller和service层的一些见解**

基于controller和modal层并没有很好的实现模块化，因此，我们将modal层去掉，改为更加原子化的dao层；同时，将controller层的业务逻辑，划分成多个服务。每个服务可以组合使用dao层数据，组装成一个服务，比如用户的注册服务；而controller层，调用多个service服务完成url请求。   
  
简单来说，增加service层，替换modal层，第一是细化了数据模型，使得我们在改动某张表时，只需要改动dao层实现即可，最大化的减少了代码的改动成本；当然，更多的情况是service服务和controller可能都需要更改； service层将controller的逻辑分类，保证了controller的逻辑更加清晰。   
  
举个生活中的例子，用户预约某个酒店的客房，这是酒店首先会调用验证服务对用户提供的信息进行验证，之后调用预约服务进行预约，如果预约失败，酒店可能会把客户的预约信息提交给另外一家酒店请求它们的预约服务，然后将结果返回给客户。 对于服务层来说，需要判断酒店是否有空余客房，之后修改客房信息，同时将客房和用户信息存入临时表。这里至少需要两种不同的dao层服务实现service。   
  
所以整体上来看，controllrt->service->dao至少是一对一，更多的情况下是一对多。这也就是service层存在的意义了。

dao是[数据访问层](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E8%AE%BF%E9%97%AE%E5%B1%82&from=1012015a&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uhDdnHf4PHbvPWDznjwb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPH6LPWnsnjRk" \t "https://zhidao.baidu.com/question/_blank)，service是[业务逻辑层](https://www.baidu.com/s?wd=%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91%E5%B1%82&from=1012015a&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uhDdnHf4PHbvPWDznjwb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPH6LPWnsnjRk" \t "https://zhidao.baidu.com/question/_blank)，必须分开以减少藕合度，便于维护。DAO负责数据库的增删改查，针对单表操作  
Service调用DAO来实现业务逻辑操作，会涉及多表操作，有时候一个业务逻辑操作会调用多个DAO的方法，因此是在Service层中控制事务

|  |  |  |
| --- | --- | --- |
| UserDao { // 根据部门ID删除用户 deleteUserByDeptId(String deptid) } | DeptDao { // 根据部门ID来删除部门 deleteDeptById(String deptId) } | DeptService { // 删除部门和部门以下的人 deteteDeptAndUser(deptId) {  UserDao .deleteUserByDeptId(deptId);  DeptDao .deleteDeptById(deptId);  } } |

dao层一般只增删改查,操作数据库,无其他代码,Service层才是[业务逻辑层](https://www.baidu.com/s?wd=%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91%E5%B1%82&from=1012015a&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uhDdnHf4PHbvPWDznjwb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPH6LPWnsnjRk" \t "https://zhidao.baidu.com/question/_blank),这样层次分开了就,比较容易维护,dao层的方法也容易重用,一个查询,可能多个地方用到,假如dao也有业务逻辑了,那么就不能重用了.