

java认证:怎样区分ActionService和Dao功能Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/583/2021_2022_java_E8_AE_A4_E8_AF_81_c104_583998.htm 在不分层的系统里，我们可以

将所有的代码都写到一个地方，比如struts的Action类。在这里，我们不仅要处理页面逻辑，还要做业务逻辑，还要做数据访问。比如说：
`public String addUser() { if(user == null) {
return FAIL_NO_USER. } Result result = null.`

```
if(Role.ADMIN.equals(user.getRole())) { result =  
doSomethingForAdmin(user). } else { result =  
doSomethingForOthers(user). } Transaction trans =  
sess.beginTransaction(). Query query = sess.createQuery("0update  
Result set level = :level"). query.setParameter("level",  
result.getLevel()). query.executeUpdate(). trans.commit().
```

```
sess.close(). return SUCCESS. }
```

那么上面的代码，哪些部分是页面的部分，哪些是业务处理，哪些是数据访问呢?我认为，这个划分方法是：Action里只做和页面相关的事，不操作业务对象.Service不依赖于任何表现技术，不操纵任务用于表现的对象，对于业务对象，尤其是跨多个业务对象的操作，要放到Service里面来.最后，单纯的业务对象的存取，组装放到DAO里完成。上面所说的业务对象，就是像上例中role, result等和业务相关的对象，而SUCCESS, inputID等，则是页面相关的部分。因些，可以将上例改为：
`public String
addUser() { if(user == null) { return FAIL_NO_USER. } Result
result = service.process(user). dao.0update(result). return
SUCCESS. }` 在service里: `public Result process(User user) { Result`

```
result = null. if(Role.ADMIN.equals(user.getRole())) { result =  
doSomethingForAdmin(user) . } else { result =  
doSomethingForOthers(user). } return result. } 在dao里 : public  
void 0update(Result result) { Transaction trans =  
sess.beginTransaction(). Query query = sess.createQuery("0update  
Result set level = :level"). query.setParameter("level",  
result.getLevel()). query.executeUpdate(). trans.commit().  
sess.close(). } 这样分层，看起来会显得很麻烦，但事实上确实  
是大有好处，首先：代码更易读。每一层的每个方法的意义  
和目的更加明确，读起来受的干扰更少。拆开后的每一层  
都更容易测试。具体如何分层，还需要在开发中，多多体会  
，这没有绝对的界限，也许一开始放在action里的页面的控制  
后来会上升为业务规则，并被其它地方重用，然后被移  
入service.也许某一块对数据的存取也变得非常复杂，包含了  
业务逻辑，然后被移入service.也有可能发现以前写的service  
根本没有想像的那样的业务逻辑，只是帮助做了一些页面的  
流程控制，然后被重构成Action的一个方法，等等。更多优  
质资料尽在百考试题论坛 百考试题在线题库 java认证更多详  
细资料 100Test 下载频道开通，各类考试题目直接下载。详细  
请访问 www.100test.com
```