1. 项目代码分层：

业务代码放在dkd.okay.business下，每个大模块建一个子包，下分bo（实体）、dao、service、controller四个子包，其中bo包内建立map子包用于存放mybatis配置文件。

1. 各模块引用：

每个模块的controller只能引用自己或其他模块的service，不能直接引用dao；每个模块service只能引用自己模块的dao和其他模块的service，不能直接引用其他模块的dao；注入使用@Autowired，service层要在类上加上@Transactional

1. 命名

系统中不管是包名、类名、方法名、变量名，也不管是前台或者后台，起名时都要使用有实际意义的英文来命名，不要出现类似a,b,c或者name1,name2之类的情况。

1. 前后台传值
2. 查询：不要再创建类似博迈科的form文件，若查询条件少，可以直接在后台controller中定义（也可以使用实体），如：query(String name,String code)；若查询条件较多，可以使用实体进行封装，如：query(User user)；若查询条件字段实体中没有，可以在实体中增加临时字段使用（一定要写注释）。
3. 表单提交：单个单据可以直接提交；若是类似订单头体结构的单据，可以在表单提交时调用common.js中的changeDetails方法来进行修改表体输入框的name，在提交时由框架自动封装。
4. 细节
5. 写完一个功能后首先要自审代码，看是否有冗余代码，若无特殊情况，不能出现黄色叹号警告和冗余代码。
6. 确认无用的代码不要注释，直接删掉。
7. 开发过程中公用功能的代码放在common或者util包中（后台）或者common.js（前台），同一功能不要写两份代码。
8. 注意代码格式，控制代码层级缩进，要使用换档键，不能使用空格键。
9. 比较复杂的方法或代码块要写上注释。
10. Xml文件的sql语句里不要写复杂的判断，若是数据层级的判断，写在dao文件中；业务层级的判断，写在service文件中。
11. 可以一个sql语句实现的功能，不要写成两个；例如查询头体结构单据。
12. 前台做dom操作时要使用jquery实现。
13. 提交SVN时一定要写注释。
14. 业务单据修改保存时要校验状态是否发生变化，避免并发的问题。
15. 公用选择页面

在本模块要引用其他模块单据信息的时候，查询其他单据信息页面只返回数据（接收返回数据的方法在本模块页面定义），查询页面不对本模块单据编辑页面进行操作。

例如选择物料，在选择物料的弹出页面要定义父级页面接收数据的方法：

**Var material;//假设此对象为物料对象；实际需要选择页面进行组装成js对象。**

**Var handleMaterial = window.top.mainFrame.handleMaterial;// handleMaterial方法是在需要引用物料的页面进行定义的处理函数，进行实际的dom操作**

**if(handleMaterial && typeof(handleMaterial) == "function"){**

**handleMaterial(material);**

**}**