

1. [DOC接口操作简要说明.pdf](#)

2. **idea插件：** 由于工程中用到了lombok注解代码生成，需要在IDE中安装对应的lombok插件；

安装使用教程：

- i. [Java奇淫巧技之Lombok](#)
- ii. [Lombok 安装、入门 - 消除冗长的 java 代码](#)
- iii. [Lombok的使用和原理](#)

3. **Controller层：** 只能包含数据的封装转换逻辑，顺序调用执行，不包含业务逻辑，尽量保证Controller每个方法代码行数最多不超过10；

- i. Controller类中方法通过@MethodDoc来标注请求方法明细, 包括:所属模块, 入参DTO, 出参DTO;
- ii. 建议：Controller每个请求对应的方法入参/出参，须定义对应DTO封装类；
- iii. APP请求token授权：如果Controller中的接口方法，需要APP Token校验授权，必须使用@HttpAuth(isRequireAuth = true, isNeedReLoginWhenExpire = true)注解进行标示；

4. **Service层：** 直接调用Manager层, 顺序调用执行， 不包含任何的业务逻辑处理，不能包含任何HttpServlet API；

5. **Manager层：** 作为DAO层和BIZ层之间粘合层, 顺序调用执行， 只包含粘合逻辑, 不能包含任何HttpServlet API, 该层只做各层联通性测试；

6. **Biz层：** 作为业务逻辑处理层, 禁止出现访问DAO层等其他层的代码, 不能包含任何HttpServlet API, 该层不依赖任何上下文环境，可针对业务逻辑进行单元测试；

7. **异常处理：** Controller层, Service层, Manager层中的代码不包含任何异常处理语句，一律将异常向上层抛出；

- i. ServiceException：该异常会提示用户返回的信息，会回滚事务；
- ii. MessageException：该异常会提示用户返回的信息，不回滚事务；

8. **请求加密：** 已在请求的拦截层做了处理，开发者无需关心；

9. 请求/响应JSON数据统一：已在请求的拦截层做了处理，开发者无需关心；
10. 请求Version控制：Version控制只在Controller层方法级别，通过Spring MVC @RequestMapping注解的headers属性, 添加version来控制；

```
@RequestMapping(value = "test0", headers = {"version=1.0.0",  
"platform=PC", "device=PC"})
```

11. SVN提交日志规范：

- i. 项目提交：[项目名称]功能模块—简要描述；
- ii. Bug修改：[项目名称]Bug:bug号已修复；