

开发工具 idea

[\\lichun\soft\java\idea](#)

需要配置 maven,tomcat

引入的 java 包，统一在 pom.xml 中管理

例:

```
<dependency>
                                <groupId>mysql</groupId>
                                <artifactId>mysql-connector-java</artifactId>
                                <version>5.1.25</version>
</dependency>
```

在 pom.xml 的 plugin 标签中注册新增的模块以及 sql.xml 文件

```
<configuration>
                                <sources>
                                <source>${basedir}/SystemManagement/</source>
                                <source>${basedir}/workflowEngine/</source>
                                <source>${basedir}/SatelliteReport/</source>
                                <source>${basedir}/SatelliteGraph/</source>
                                <source>${basedir}/common/</source>
                                </sources>
</configuration>
```

```
<resource>                                <directory>${
{basedir}/SatelliteGraph/sql/mapper</directory>
                                <targetPath>sql/mapper</targetPath>
</resource>
```

配置文件在 phalaenopsis/common/conf 下面

1 applicationContext.xml 配置上下文

其中

<bean

```
id="exceptionHandler" class="phalaenopsis.common.method.MyExceptionHandler" />
```

配置统一的异常拦截器

程序中不要自己捕获异常，不然事务无法回滚

2 applicationContext_dao.xml 配置 mybatis, 实体类映射

3 applicationContext_datasource.xml 配置数据库连接及事务

4 config.properties 配置固定参数

例

上传文件存放路径

UploadFolder=/UploadFiles/

phalaenopsis.common.entity.AppSettings 中获取

5 dispatcher-servlet.xml 配置消息转换器 以及拦截器

```
<mvc:interceptors>
```

```

<mvc:interceptor>
    <mvc:mapping    path="/**/*"    />
    <bean
class="phalaenopsis.common.method.CrossInterceptor">
        <property    name="ignoreUrls"
            value="/api;/Sys/Auth/Login;/Sys/Auth/loginForLC;/Sys/A
uth/Test;/Foundation/DataDictionary/DownloadApp;/Foundati
on/DataDictionary/GetLatestVersion;/SatelliteGraph/Mobile/Ch
eckConnection;/LawCase/LawCaseService/GetSignLink;/Satellit
eGraph/Service/test"></property>
    </bean>
</mvc:interceptor>

```

```

        <!--      liulang      add      start      -->
                                <mvc:interceptor>
                                <mvc:mapping      path="/**"      />
                                <bean
class="phalaenopsis.common.interceptor.LogInterceptor">
                                </bean>
                                </mvc:interceptor>
        <!--      liulang      add      end      -->
</mvc:interceptors>

```

所有前端发送的请求,会被拦截进入前置方法, 如果身份票据验证不通过并且 url 不在 ignoreUrls 中, 则无法访问服务

6 ehcache.xml 设置用户缓存, 系统缓存

7 jdbc.properties 数据库连接

```

jdbc.kotei.url                      =
jdbc.kotei.username =
jdbc.kotei.password =

```

8 workflow-config.xml 工作流配置

基本结构

Controller 层 公布前端访问的接口

类名上加入标签

@Controller

@RequestMapping("/LawCase/LawCaseService")

使用@Autowired 实例化 service

@Autowired

```
private LawCaseCoordinate coordinateService;
```

例

```
/**
 *      获  取  案  件  坐  标  数  据  .
 *      @param    id      案  件  id  号
 *      @return    返  回  坐  标  信  息
 */
@RequestMapping(value = "/GetCaseShape", method =
RequestMethod.GET)
@ResponseBody
public Polygon getCaseShape(@PathParam("id") final String id)
{
    return    coordinateService.getCaseShape(id);
}

/**
 *      删  除  实  测  坐  标  信  息 .
 *      @param    info    案  件  信  息
 *      @return    返  回  是  否  成  功
 */
@RequestMapping(value = "/DeleteRealCoordinate", method =
RequestMethod.POST)
@ResponseBody
public boolean deleteRealCoordinate(@RequestBody final
CaseBaseInfo info) {
    return    coordinateService.deleteCoordinate(info);
}
```

Service 层 实现业务逻辑

类名上加入标签@Service("lawCaseService")

使用@Autowired 实例化接口

@Autowired

```
private IWorkflowDao workflowDao;
```

在方法上加入@Transactional 开启事务

Dao 层 接口

类名加入标签@Repository("iLawCaseDao")

Entity 实体

Sql/mapper xml 编写 sql

```
<mapper
```

```
namespace="phalaenopsis.lawcase.dao.ILawCaseServiceDao">
```

Xml namespace 指向接口

在 common/method 下面有一些工具类

例 ExportExcel 导出 excel

common/method/Tools

DateUtils GuidHelper StrUtil

Common/temple/Generator 自动生成实体，增删改查