**狮桥研发DEMO文档**

**2016年3月(编写日期)**

**V1.0(版本号)**

关于本文档

对于本文档的版权与保密信息的描述。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 主 题 | | 项目结构及技术使用规范 | | | | | | |
| 说　　明 | |  | | | | | | |
| 适用对象 | | 系统研发人员 | | | | | | |
| 修订历史 | | | | | | | | |
| 版本 | 章节 | | 类型 | 修订日期 | 修订者 | 复核日期 | 复核人 | 修订说明 |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |

**说明：类型－创建（C）、修改（U）、删除（D）、增加（A）；**

目录

[关于本文档 2](#_Toc495499825)

[1. 前言 4](#_Toc495499826)

[1.1 编写目的 4](#_Toc495499827)

[1.2 适用范围 4](#_Toc495499828)

[1.3 术语定义 5](#_Toc495499829)

[1.4 参考文档 5](#_Toc495499830)

[2. 项目结构 5](#_Toc495499831)

[2.1 CMM-BASE 5](#_Toc495499832)

[2.2 FL-DEMO-FACADE 6](#_Toc495499833)

[2.3 FL-DEMO-SERVICE 7](#_Toc495499834)

[2.4 FL-DEMO-WEB 9](#_Toc495499835)

[2.5 结构图 10](#_Toc495499836)

[3. 项目配置 11](#_Toc495499837)

[3.1 MAVEN 11](#_Toc495499838)

[3.2 DUBBO 11](#_Toc495499839)

[3.2.1 说明 11](#_Toc495499840)

[3.3 项目依赖 11](#_Toc495499841)

[4. 命名规范 12](#_Toc495499842)

[4.1 类方法命名规范说明 12](#_Toc495499843)

[4.2 配置文件命名规范说明 12](#_Toc495499844)

[4.2.1 DUBBO 12](#_Toc495499845)

[4.2.2 MABATIS 12](#_Toc495499846)

[4.2.3 PROPERTIES 13](#_Toc495499847)

[4.2.4 XML 13](#_Toc495499848)

[4.2.5 FILE 13](#_Toc495499849)

[4.3 包命名规范说明 13](#_Toc495499850)

[4.4 JSP命名规范 13](#_Toc495499851)

[4.5 其它规范 14](#_Toc495499852)

[4.5.1 暴露DUBBO API接口 14](#_Toc495499853)

[4.5.2 内部接口实现类 14](#_Toc495499854)

[4.5.3 接口和方法注释 15](#_Toc495499855)

[4.5.4 实体类注释 15](#_Toc495499856)

[5. 常用工具 15](#_Toc495499857)

[5.1 JAVA使用说明 15](#_Toc495499858)

[5.1.1 CollectionUtils 15](#_Toc495499859)

[5.1.2 DateUtils 16](#_Toc495499860)

[5.1.3 DecimalUtils 16](#_Toc495499861)

[5.1.4 FileUtils 16](#_Toc495499862)

[5.1.5 HttpClientUtils 16](#_Toc495499863)

[5.1.6 StringUtils 16](#_Toc495499864)

[5.2 JS使用说明 17](#_Toc495499865)

[5.2.1 CollectionUtils 17](#_Toc495499866)

[5.2.2 DateUtils 17](#_Toc495499867)

[5.2.3 TipUtils 17](#_Toc495499868)

[5.2.4 NumberUtils 17](#_Toc495499869)

[5.2.5 StringUtils 17](#_Toc495499870)

[5.2.6 UIUtils 17](#_Toc495499871)

[6. 研发代码步骤(以Demo为例子) 17](#_Toc495499872)

# 前言

## 编写目的

有效管理项目结构及技术规范,防止研发过程中造成结构混乱，技术混乱现象以及影响项目稳定性、扩展性等。

由以下项目组成：

1. CMM-BASE为公共类及公共工具的定义。
2. FL-\*-FACADE为公共暴露接口及常量类的定义。
3. FL-\*-SERVICE 为服务支撑项目，其中定义持久层相关文件及对外接口实现等。
4. FL-\*-WEB 为客户端项目，其中定义数据转换实现和所需文件。

## 适用范围

本文档适用读者包括以下人员：

* 项目PMO
* 系统研发组成员

## 术语定义

PO : 持久层对象

DTO: 服务端与客户端传递数据的对象

VO: 客户端展示数据使用对象

DAO: 持久层

SERVICE: 服务层 也是服务提供者

CONTROLLER: REST控制层

BASE: 基类

TRANSFORM: 数据转换类

## 参考文档

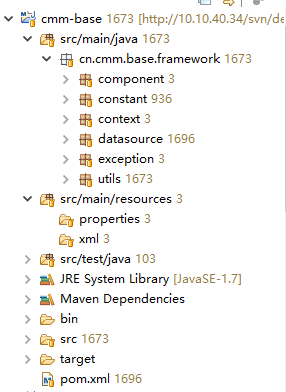
无.

# 项目结构

## CMM-BASE

CMM-BASE项目中中定义公共工具项目。

结构如下：



## FL-DEMO-FACADE

FL-DEMO-FACADE 项目是客户端和服务端所拥有共同类及接口定义。

由各个业务模块的常量类、DTO类、查询对象、暴露接口、分页类,如果需要建立暴露接口、DTO、常量类 ,按照demo模块自定义即可。

1、自定义类名称规范注意事项：

暴露服务接口：I + 模块名称 + ApiService 例：IDemoApiService

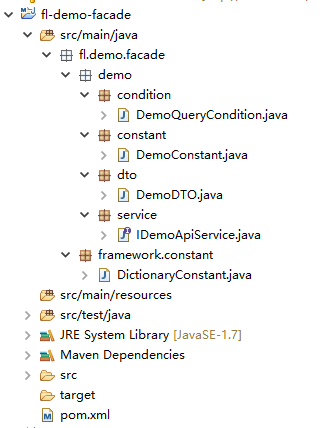
DTO : 实体类名称 + DTO 继承 BaseDTO 例：DemoDTO

常量类： 模块名称+Constant 或 实体名称+Constant 例：DemoConstant

查询对象: 模块名称 + QueryCondition 例：DemoQueryCondition 或者

模块名称 + QC 例：DemoQC

结构如下：



## FL-DEMO-SERVICE

FL-DEMO-SERVICE项目为客户端提供业务支撑(同时暴露Dubbo接口)。

其中由 DAO 、 PO 、 SERVICE、TRANSFORM 、FRAMEWORK(研发勿动)定义以及配置文件 DUBBO、MYBATIS、PROPERTIES、SPRING 等组成，研发人员可按照demo模块进行配置开发 。

1、自定义类名称规范注意事项：

DAO: 实体类名称 + Mapper 继承IBaseDao 例：DemoMapper

PO: 实体类名称 例：Demo

SERVICE: I + 实体类名称 + Service 例：IDemoService

SERVICE.IMPL：实体类名称 + ServiceImpl 例：DemoServiceImpl

暴露接口实现：模块名称 + ApiServiceImpl 例：DemoApiServiceImpl

TRANSFORM: 实体类名称 + Transform 例：DemoTransform

参照Demo模块类继承模式。

2、自定义配置文件名称规范注意事项(名称小写)：

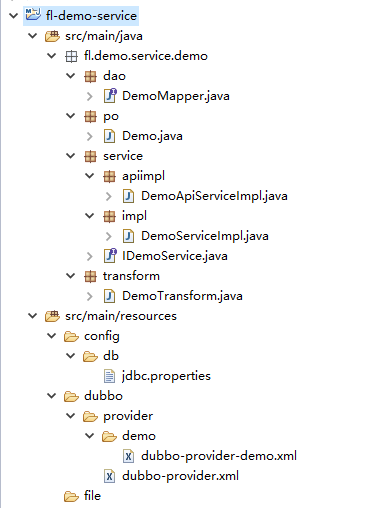
DUBBO: dubbo-provider- + 实体类名称(多名词要用中划线分割) + .xml 例：dubbo-provider-demo.xml

MYBATIS: 实体类名称 + -result-mapper.xml

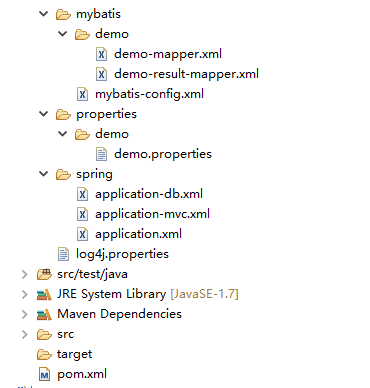
例：demo--result-mapper.xml

实体类名称 + -mapper.xml

例：demo-mapper.xml

PROPERTIES：模块名 + .properties 例：demo.properties

结构如下：



## FL-DEMO-WEB

FL-DEMO-WEB为客户端(DUBBO消费方) , 需要请求FL-SERVICE服务端(DUBBO 服务提供方)请求业务数据。

其中由CONTROLLER 、SERVICE、TRANSFORM、VO、FRAMEWORK及配置文件组成。研发人员可按照demo模块进行开发 。

1、自定义类名称规范注意事项：

CONTROLLER：实体类名称 + Controller 或 模块名 + Controller

例：DemoController

SERVICE: I + 实体类名称 + Service 例： IDemoService

SERVICE.IMPL: 实体类名称 + ServiceImpl 例：DemoServiceImpl

TRANSFER: 实体类名称 + VOTransform 例：DemoVOTransform

VO: 实体类名称 + VO 例：DemoVO

参照Demo模块类继承模式。

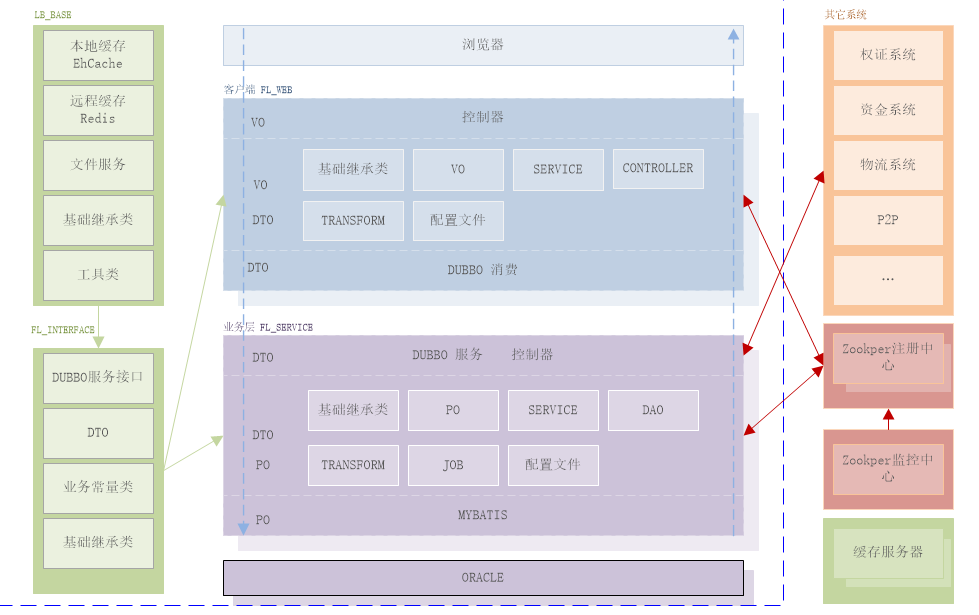
2、自定义配置文件名称规范注意事项(名称小写)：

DUBBO: dubbo-consume-实体类名称.xml

例：dubbo-consume-demo.xml

PROPERTIES：模块名 + .properties 例：demo.properties

## 结构图





# 项目配置

## MAVEN

项目统一由Maven管理, 结构为：

cmm-root  
    　　 |----pom.xml  
    　　 |----fl-demo-web  
       |----pom.xml  
    　　 |----fl-demo-service  
        　　 |----pom.xml  
    　 |----fl-demo-facade   
       |----pom.xml  
    　　 |----cmm-base  
         |----pom.xml

## DUBBO

### 说明

1. DUBBO 配置分为服务提供方(FL\_SERVICE)和消费方(FL\_WEB)
2. 服务方主配置在resources/dubbo/provider/dubbo-provider.xml中。
3. 消费方主配置在resources/dubbo/consume/ dubbo-consume.xml中。
4. 各业务模块需在同级目录下建立模块文件夹后建立属于自己的服务xml,参照demo模块.

## 项目依赖

FL-DEMO-SERVICE 依赖 FL-DEMO-FAÇADE

FL-DEMO-WEB 依赖 FL-DEMO-FACADE

# 命名规范

## 类方法命名规范说明

1、所有方法名称均采用驼峰命名法。

2、Mapper 方法命名：新增(insert\*) 、删除(delete\*)、修改(update\*)、查询(select\* find\* get\*)、分页查询(select\*Page)。

3、Service 方法命名：新增(add\*) 、删除(remove\*)、修改(modify\*)、查询(select\* get\*)、分页查询(search\*Page)。

4、类名以驼峰命名法且首字母必须大写。

5、所有接口方法要抛出BaseException运行时异常, 根据业务处理逻辑对错误使用异常模式抛出,比如参数为空, 则 throw new BaseException(BaseExceptionConstant. BASE\_EXCEPTION\_CODE\_600,

具体描述信息);

## 配置文件命名规范说明

### DUBBO

1、消费方配置文件都配置在resources/dubbo/consume下以模块名称(小写)区分, 自定义文件(均小写)格式：dubbo-consume-模块名称.xml。

2、服务方配置文件都配置在resources/dubbo/provider下以模块名称(小写)区分,自定义文件(均小写)格式: dubbo-provider-模块名称.xml

### MABATIS

1、所有SQL配置文件都在resources/mybatis目录下定义自己的模块(小写), 每个实体对应两个文件 \*-mapper 与 \*-result-mapper 其中\*为实体名称以名词分割后用中划线连接的字符串(小写),如：demo-demo-mapper.xml demo-demo-result-mapper

### PROPERTIES

1、所有属性配置文件都在resources/properties目录下定义自己的模块(小写), 属性文件名称定义为小写, 其中内部命名要严格按照：模块名.文件名.properties.\* 格式, 如果命名不规范就会在全局缓存中不存在数据的时候直接查询文件失败。如：demo.demo.properties.local.is.open

### XML

1、所有属性配置文件都在resources/xml目录下定义自己的模块(小写), 文件名称定义为小写, 格式：application-模块-自定义.xml。

### FILE

1、所有文件存放位置都在resources/file目录下定义自己的模块(小写), 将文件放入即可。

## 包命名规范说明

1. 所有包在fl.demo.façade 或者 fl.demo.service或者 fl.demo.web目录下,如果各业务需要分别建立属于自己的模块包,包名必须全部为小写, 模块包下建立po 、vo、dto等包即可。

## JSP命名规范

**大模块名+小模块名+关键词(如下) ，命名均已驼峰命名法, 如果名称过长可以对单独词语进行缩写(如system 可以缩写为sys, customer可以缩写为cust ),但是名称一定要有明确的含义。**

**1、增加 add**

**2、删除 remove delete del**

**3、修改 update modify**

**4、查询列表 list**

**5、管理 manage**

**6、设置 setting**

**如客户:**

**新增： custCstBscInfoAdd**

**列表页：custCstBscInfoList**

**修改：custCstBacInfoModify**

**管理页面：custCstBacInfoManage**

## 其它规范

### 暴露DUBBO API接口

1、所有接口必须抛出throws BaseException 运行时异常。

2、实现类中只能调用其它服务层接口获取数据。

3、类中不允许出现非DTO对象进行数据传输, 如 PO VO。

4、类中不允许出现复杂业务逻辑,只能做基本的验证操作、异常处理、操作日志。所有的异常均已throw new BaseException(BaseExceptionConstant. BASE\_EXCEPTION\_CODE\_600,

具体错误信息描述)模式向上抛出。

### 内部接口实现类

1、所有接口必须抛出throws BaseException 运行时异常。

2、BaseBizImpl 中已经实现了基本方法如下：

public Long add(DTO entity) throws BaseException;

/\*\* public Long add(DTO entity) throws BaseException;

public boolean removeById(Long id) throws BaseException;

public boolean modify(DTO entity) throws BaseException;

public DTO selectById(Long id) throws BaseException;

public Pager searchListPage(Pager page) throws BaseException;

public List<DTO> searchList(C condition) throws BaseException;

如果有需要可以重写本方法,或者自定方法即可,

1. 内部接口绝对不可以直接调用其它服务层的Mappr服务,只能调用Service接口获取数据, 返回的对象不能包含PO对象。PO对象只能在自己的服务中使用。

### 接口和方法注释

1、接口类要明确写上是什么接口及用途说明。

2、方法体内要处理什么业务要在接口中写明。

3、方法体中复杂逻辑一定要标明处理思路及业务逻辑说明。

4、修改接口和方法时要同时更新注释。

### 实体类注释

1、注明实体类所属对象描述。

2、所有属性和get set方法必须写明字段说明,这样有助于调用者在调用的时候明确看到属性是干什么的, 也同时避免在很多字段的情况下自己还得去和模型或者数据库对比含义。

7、查询对象命名为：模块名 + QueryCondition 或模块名 + QC

# 常用工具

## JAVA使用说明

项目中如果用到与工具类相关的验证等务必使用工具类.

### CollectionUtils

对集合空验证和非空的验证。

### DateUtils

对时间的转换以及时间的获取等。

### DecimalUtils

对计算的加减乘除的处理，程序中所有Double类型的加减乘除运算必须要使用本工具处理。

### FileUtils

对文件的简单处理。

### HttpClientUtils

对发送POST请求工具类,根据参数配置好参数后使用。

### StringUtils

字符串工具类，对字符串的判空、 邮箱格式匹配、是否是整数、是否是手机号。

## JS使用说明

### CollectionUtils

### DateUtils

### TipUtils

### NumberUtils

### StringUtils

### UIUtils

定义统一公共使用的JS方法,如：

成功提示

错误提示

警告提示

# 研发代码步骤(以Demo为例子)

参考FL-DEMO-SERVICE DEMO模块

FL-DEMO-WEB DEMO 模块