**DPOP组基础技术框架-开发规范**

|  |  |
| --- | --- |
| **组内审核时间** |  |
| **重要性** |  |
| **紧迫性** |  |
| **技术评审拟制人** | 胡海亮 |
| **技术评审提交日期** | 2014/7/15 |
| **需求变更控制时间点** |  |

百度联盟产品组

（版权所有翻版必究）

**修改记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **更新时间** | **内容** | **修改人** | **版本** |
| 2014/7/15 | DPOP组基础技术框架-开发规范 | 胡海亮 | V1.0.0 |
|  |  |  |  |

目录

[**DPOP组基础技术框架-开发规范** 1](#_Toc395289930)

[**修改记录** 1](#_Toc395289931)

[目录结构 3](#_Toc395289932)

[工程目录结构 3](#_Toc395289933)

[包目录结构 3](#_Toc395289934)

[配置文件目录结构 5](#_Toc395289935)

[web静态资源目录结构 6](#_Toc395289936)

[开发规范 7](#_Toc395289937)

[DPOP 基础架构总览 7](#_Toc395289938)

[BO开发规范 9](#_Toc395289939)

[Mapper开发规范 9](#_Toc395289940)

[Dao开发规范 10](#_Toc395289941)

[Service开发规范 11](#_Toc395289942)

[Controller开发规范 12](#_Toc395289943)

# 目录结构

目录结构属于工程的概要设计领域，目录结构的层次关系清晰度能体现一个项目是否清晰，也决定了项目后期的维护成本。也决定着新研发成员学习的成本大小。下面详细介绍下各个层次的目录结构规范。

## 工程目录结构

DPOP组项目管理是油maven工具来管理的,结构如下：

resources目录下存放的是工程资源配置，本人把这些配置细分层sql、config。如上图

1、conf目录下面rd- properties存放的是RD开发环境相关的配置信息，以此类推。

2、test包中resources下面的mockDB是存放单测mock 数据库的脚步

3、conf目录下面无特别情况下不存放properties配置，只存放XML程序配置

4、\*properties目录下存放properties配置信息

## 包目录结构

java package :

**平台/框架 包结构：**

com.baidu.dpop.frame.core.base

com.baidu.dpop.frame.core.context

com.baidu.dpop.frame.core.mybatis

com.baidu.dpop.frame.core.util

com.baidu.dpop.frame.remote.client

**组织 公司 小组 平台 层次 组件**

**模块代码包结构：**

com.baidu.dpop.rmp.common . constant

com.baidu.dpop.rmp.common . utils

com.baidu.dpop.rmp.material.bo

com.baidu.dpop.rmp.material.dao

com.baidu.dpop.rmp.material.dao.mapper

com.baidu.dpop.rmp.material.service

com.baidu.dpop.rmp.material.web.controller

com.baidu.dpop.rmp.material.web.validator

**组织 公司 小组 项目 模块 层次 功能**

## 配置文件目录结构

**环境资源文件：**

普通的环境通过工程启动的时候指定参数：-Dspring.profiles.active=rd来指定，如果没指定默认使用线上online环境。

**校验信息配置文件：**

系统级别提示信息：

模块级别提示信息：

每个模块都会有自己的校验规则和提示信息，故本人将不同的模块的提示信息放置不同的模块下。

**工程中spring配置文件:**

spring配置文件命名：

“applicationContext -功能模块.xml”

## web静态资源目录结构

webapp/static下面存放项目的静态资源，规范如下图：

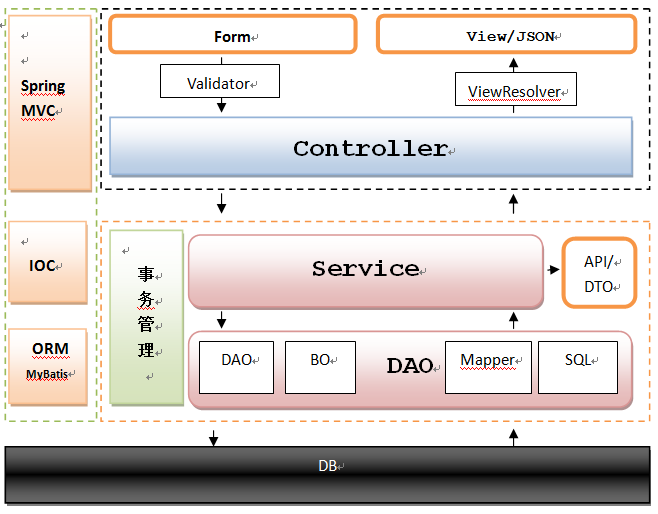
Webapp/ WEB-INF/ templates目录下面存放项目的页面view:

# 开发规范

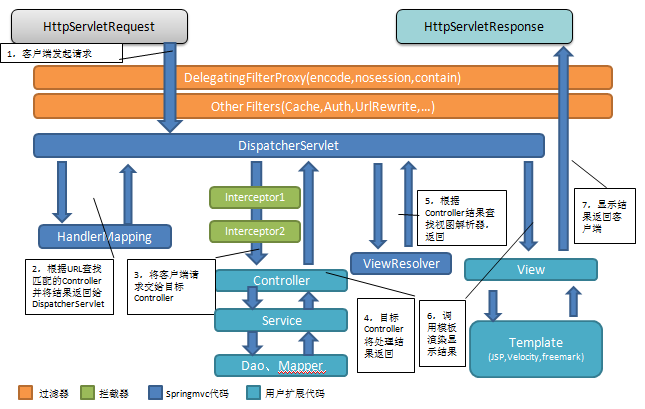
Dpop组基础框架的是由SpringMVC+Mybatis组成，故一些基础规范和设计在这两框架之上做的封装。

## DPOP 基础架构总览

**框架的层次关系图：**



**请求流转图：**



**业务层的继承图：**



在图中，上面是抽象和接口，下面是具体的模块实现。

***注：Module代表模型，是模块的名称。***

## BO开发规范

首先看看BO层的继承关系：



***注：Module代表模型，是模块的名称。***

ModuleBase是代码生成工具生成出来的BO基类，不允许手工修改此基类。用户扩展的属性需Module中扩展。如果DB模型变化了，比如新增了一个字段或者删除了一个字段，只需再次生成ModuleBase即可。

## Mapper开发规范

Mapper是Mybatis框架的SQL映射层，DPOP代码生成工具会生成：ModuleMapper.java、ModuleBaseMapper.xml、ModuleMapper.xml三个文件。如上图：ModuleMapper继承与泛型Mapper类获取一些公共的操作方法。而SQL存放在两个配置文件中：ModuleBaseMapper.xml、ModuleMapper.xml，其中和BO同理ModuleBaseMapper.xml存放一些基础映射和基础方法，不能手工修改，用户自定义扩展的方法请存放在ModuleMapper.xml中。

## Dao开发规范

Dao层是每个存在DB的项目必备的层，故此层规范大致和其他一样。保证无状态、只负责数据层交互。但DPOP组内的规范，必须按照如下基础关系来实现：



继承BaseDao获取一些框架授予的基础方法，其他用户自定义扩展的操作需在此架构下扩展。

## Service开发规范

服务层是一个完整的业务逻辑划分，需保证事务一致性。一个服务执行完成之后事务提交，发生异常事务回滚。首先看看继承体系：



事务控制通过在方法标注：

@Transactional(propagation = Propagation.*REQUIRED*, rollbackFor = RuntimeException.**class**)

## Controller开发规范

Comtroller是SpringMVC的控制层，继承规范如下：



JsonBaseController规范了JsonResult相关的的操作，如下：

**public** JsonResult markSuccessResult(Object data,String resultInfo)

**public** JsonResult markErrorResult(String errorCoder)

**public** JsonResult markErrorResult(Errors errors)

**private** JsonResult markJSONResult(String isSuccess,Object data,String resultInfo)

业务Controller可直接调用此些方法返回结果。