# 基础项目架构设计

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修改日期 | 修改人 | 修改内容 |
| 1.0 | 20160421 | 朱瑞泽 | 初始创建 |
|  |  |  |  |

**目录**

[基础项目架构设计 1](#_Toc449090523)

[开发环境 3](#_Toc449090524)

[项目分层 3](#_Toc449090525)

[说明 3](#_Toc449090526)

[结构图 5](#_Toc449090527)

[部署图 6](#_Toc449090528)

## 开发环境

|  |  |  |
| --- | --- | --- |
| JDK | 1.7 |  |
| WEB容器 | Tomcat 7.x |  |
| 版本控制 | SVN/Git | 推荐用Git，SVN只做代码仓库。 |
| 开发工具 | eclipse/myeclipse/IntelliJ IDEA | 推荐使用IntelliJ IDEA |
| 构建工具 | Maven 3.x |  |
| 数据库 | MySQL 5.5+ |  |
| 消息队列 | RabbitMQ |  |
| 缓存 | Redis 3.x |  |
| 文件服务 | FastDFS | 存储图片 |
| 文档数据库 | MongoDB 3.x | 记录日志 |

## 项目分层

### 说明

项目划分为2层，分别为服务层、应用层，依赖的公共服务单独开发并在公共组件里提供接口客户端。

#### *服务层*

* 服务层按模块竖向分隔，彼此之间互相独立，跨模块之间的调用在应用业务Service里完成；
* 模块服务使用Dubbo注册中心暴露服务接口；
* 持久层主要使用Hibernate，通过业务模型自动更新表结构，MyBatis只用于跨表的复杂查询；
* 服务实现类使用Spring的注解事务；

#### *应用层*

* Spring + SpringMVC作为主要开发框架，视图用Freemarker模版引擎生成HTML展示到前端；
* Spring Session过滤器通过Redis共享Session；
* 使用Dubbo注册中心订阅并注入服务供应用业务Service调用；

#### *流程*

* 应用平台接收前端请求或定时任务调用服务层的模块服务，模块服务通过来源切换对应的数据库（每个应用平台有独立的数据库和一个共享总库）来执行Service方法。
* 需要跨模块的异步任务只需要向消息队列发出消息，由执行方监听处理。

### 结构图



### 部署图

