

翼龙贷统一架构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **版本** | **编写/修订说明** | **修订人** | **修订日期** | **备注** |
| 1 | v1.0.0 | 初版 | 秦雪飞 | 2017/11/10 | v1.0.0初版 |
| 2 | v1.0.1 | 1.增加在线生成项目使用说明 | 秦雪飞 | 2017/11/16 | v1.0.1版本 |
|  |  |  |  |  |  |

[翼龙贷统一架构 1](#_Toc29481)

[0. 概述 3](#_Toc22034)

[0.1. 背景 3](#_Toc17822)

[0.2. 目标 3](#_Toc31376)

[1. 统一开发环境 3](#_Toc11173)

[1.1. 优势 3](#_Toc22677)

[1.2. 工具 4](#_Toc16746)

[1.2.1. IDEA 4](#_Toc26476)

[1.2.2. Jdk1.7 4](#_Toc1428)

[1.2.3. Tomcat 4](#_Toc9601)

[1.2.4. Maven 5](#_Toc5561)

[1.2.5. SVN 5](#_Toc22409)

[1.2.6. JUnit 5](#_Toc4965)

[2. 定期会议 6](#_Toc29146)

[2.1. 会议需求 6](#_Toc7823)

[2.2. 会议章程 6](#_Toc1446)

[3. Code Review 6](#_Toc30825)

[4. 统一架构 7](#_Toc6395)

[4.1. 统一项目结构 8](#_Toc14824)

[4.1.1. 主项目结构 9](#_Toc18343)

[4.1.2. Api项目结构 10](#_Toc9094)

[4.2. 统一Jar包版本 11](#_Toc15487)

[4.3. 项目生成工具 12](#_Toc30724)

[4.3.1. 在线生成工具 12](#_Toc19862)

[4.3.2. Windows的Bat执行工具 13](#_Toc8751)

[4.3.3. Maven插件方式 14](#_Toc7010)

# 概述

## 背景

随着公司规模的扩展，公司开发人员和项目都在不断扩增。相应的也带来了开发沟通，项目管理，代码管理等各式的问题。为了解决此问题，特推广翼龙贷统一架构！由翼龙贷研发中心架构组研发定制。

## 目标

架构组为了解决随着公司规模扩展带来的问题，提出以下几个目标：

* 统一开发环境，用管理代码的方式来管理环境，缩短各个项目组开发人员沟通成本
* 定期开展需求会议，收集各个项目组对公司各种组件的需求
* 定期Code Review代码，共同学习，共同进步。
* 提供架构模板，新项目和老项目重构需要以此为基础（内置中间件）

# 统一开发环境

## 优势

相信每一位coder都有自己熟悉的一套开发环境，但是遇到其他使用另外一套开发环境的同事，双方在沟通，看代码的时候， 就跟看天书一样了。We are a team！ 有多少人真正理解这句话的意思。我们是一个团队，就应该你看我写的代码就跟看自己写的一样，这样遇到问题解决的效率才会更上一个档次。

优势1：新员工加入能够更快的融入团队

优势2：员工之间沟通成本变低

优势3：集成代码变方便

优势4：coding变得更有效率

## 工具

**SVN**

**Maven**

**Tomcatt**

**IDEA**

**Jdk1.7**

**Junit**

### IDEA

IDEA 全称IntelliJ IDEA，是[java语言](http://baike.baidu.com/view/229611.htm" \t "http://baike.baidu.com/_blank)开发的集成环境，IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、[J2EE](http://baike.baidu.com/view/1507.htm" \t "http://baike.baidu.com/_blank)支持、[Ant](http://baike.baidu.com/subview/42824/15924876.htm" \t "http://baike.baidu.com/_blank)、[JUnit](http://baike.baidu.com/view/66926.htm" \t "http://baike.baidu.com/_blank)、[CVS](http://baike.baidu.com/view/20982.htm" \t "http://baike.baidu.com/_blank)整合、代码审查、 创新的[GUI设计](http://baike.baidu.com/view/66012.htm" \t "http://baike.baidu.com/_blank)等方面的功能可以说是超常的。

|  |  |
| --- | --- |
| **软件名称** | IntelliJ IDEA |
| **版本号** | 15.0.2 |
| **下载地址** | <ftp://192.168.1.55/ftp/software/%D2%ED%C1%FA%B4%FB%BC%DC%B9%B9/>  用户名：eloancn 密码：eloancn |
| **备注** | 【key.txt】：为idea的激活码  【Idea-config.dotx】：为idea的设置文档 |

### Jdk1.7

JDK是整个java开发的核心，它包含了JAVA的运行环境，JAVA工具和JAVA基础的类库。

|  |  |
| --- | --- |
| **软件名称** | JDK |
| **版本号** | 1.7.0\_75 |
| **下载地址** | <ftp://192.168.1.55/ftp/software/%D2%ED%C1%FA%B4%FB%BC%DC%B9%B9/>  用户名：eloancn 密码：eloancn |
| **备注** |  |

### Tomcat

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器。

|  |  |
| --- | --- |
| **软件名称** | Tomcat |
| **版本号** | 7.0.55 |
| **下载地址** | <ftp://192.168.1.55/ftp/software/%D2%ED%C1%FA%B4%FB%BC%DC%B9%B9/>  用户名：eloancn 密码：eloancn |
| **备注** |  |

### Maven

[Maven](http://baike.baidu.com/view/336103.htm" \t "http://baike.baidu.com/view/_blank)项目对象模型(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的软件[项目管理工具](http://baike.baidu.com/view/3169783.htm" \t "http://baike.baidu.com/view/_blank)。

|  |  |
| --- | --- |
| **软件名称** | [Maven](http://baike.baidu.com/view/336103.htm" \t "http://baike.baidu.com/view/_blank) |
| **版本号** | 3.2.1 |
| **下载地址** | <ftp://192.168.1.55/ftp/software/%D2%ED%C1%FA%B4%FB%BC%DC%B9%B9/>  用户名：eloancn 密码：eloancn |
| **备注** | setting.xml已经配置好 |

### SVN

SVN是Subversion的简称，是一个开放源代码的版本控制系统。

|  |  |
| --- | --- |
| **软件名称** | SVN |
| **版本号** | 1.8.13 |
| **下载地址** | <ftp://192.168.1.55/ftp/software/%D2%ED%C1%FA%B4%FB%BC%DC%B9%B9/>  用户名：eloancn 密码：eloancn |
| **备注** |  |

### JUnit

JUnit是一个Java语言的单元测试框架。多数Java的开发环境都已经集成了JUnit作为单元测试的工具。

|  |  |
| --- | --- |
| **软件名称** | JUnit |
| **版本号** | Idea中有内置的junit非常好用 |
| **下载地址** |  |
| **备注** | 在开发过程中，每个方法必须要有对用的单元测试。  单元测试的难易程度，直接反应了代码的好坏。代码写的越好对应的单元测试越容易编写覆盖。 |

# 定期会议

## 会议需求

为了架构组能够更好的服务于公司各个业务部门。决定每个月定期开展会议，收集各个业务部门对公司各组件的需求，并加以修改升级使其更完善健壮。

## 会议章程

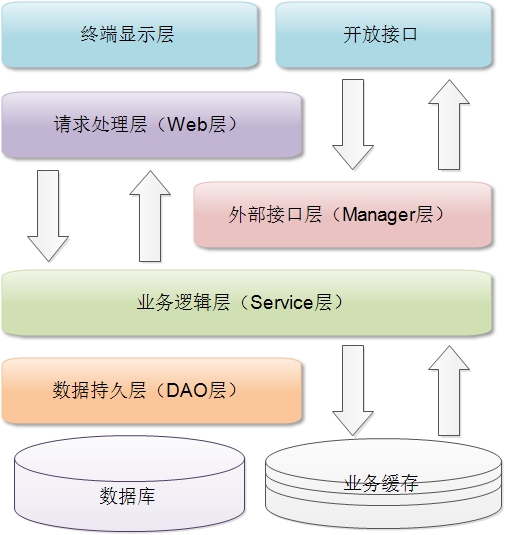
|  |  |
| --- | --- |
| **时间** | 每月偶数周的周三 |
| **主持人** | 刘罗家 |
| **参与人** | 架构组，各部门leader |
| **地址** | 魏公村某会议室 |
| **会议记录** | 每次会议落成文档提交wiki |
| **备注** |  |

# Code Review

* 代码在上线之前，一定要组织组内Code Review
* 可以定期组织组员分享写的比较好的代码，进行Code Review

|  |  |
| --- | --- |
| **会议** | Code Review |
| **时间** | 定期 |
| **主持人** | 部门leader |
| **参与人** | 组员 |
| **地址** | 某会议室 |
| **备注** |  |

# 统一架构



**数据库：** 使用mysql数据库，阿里巴巴[com.alibaba.druid.pool.DruidDataSource]数据源链接池，mysql链接驱动[mysql-connector-java.5.1.37]版本。

\*\*\*注释：数据库具体版本以DB运维为准。

**业务缓存：**使用Redis[2.9.0]版本。主要用于静态业务的缓存，Redis锁等。

**数据持久层（DAO层）：**使用mybatis[3.3.0]版本，直接对应数据库表的原子操作层。

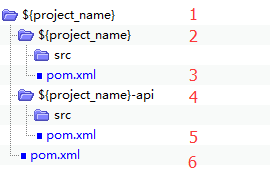
**业务逻辑层（Service层）：**可以调用DAO层的操作，封装业务处理。

**外部接口层（Manager层）：**目前内部调用使用Dubbo[2.8.4]版本，Manager层可以调用业务逻辑Service层通过Dubbo对外提供服务。

**请求处理层（Web层）：**调用业务逻辑Service层，对用户在界面发起的请求响应。

**终端显示层：**使用easyui，jquery，将后台响应的数据展示给用户。

## 统一项目结构



①：项目的总目录，相当于根文件夹。

②：主项目，包含项目的主体代码。

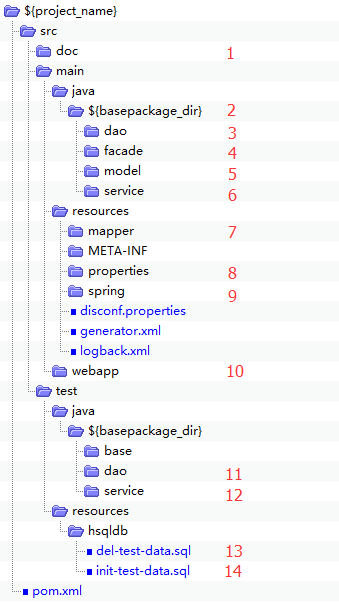
③：主项目的pom配置文件。（引用父pom【<groupId>com.el.parent</groupId><artifactId>el-pom</artifactId><version>1.0.1</version>】）

④：Api项目，项目需要对外提供接口时，提供给调用者引用。包含接口方法和返回实体Dto。

⑤：Api项目的pom配置文件。（引用父pom【<groupId>com.el.parent</groupId><artifactId>el-pom</artifactId><version>1.0.1</version>】）

⑥：项目的父pom，帮助打包。（辅助打包）

### 主项目结构



①：主项目的doc文件夹，存放项目的设计，开发文档等。

②：主项目的包根路径。

③：主项目的数据持久层（DAO层）

④：主项目的外部接口层（Manager层）-实现类

⑤：主项目的数据持久层（DAO层）-实体类Model

⑥：主项目的业务逻辑层（Service层）

⑦：主项目的数据持久层（DAO层）-Mapper配置文件

⑧：主项目的属性文件存放目录

⑨：主项目的spring配置文件（集成disconf，cas，redis，codis，rabbitmq，dubbo，cat）

⑩：主项目的终端展示层，存放页面，JS，CSS等

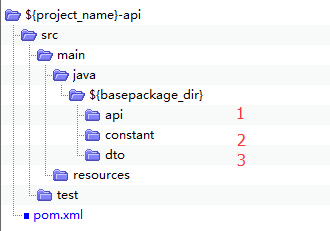
⒒：Junit单元测试DAO层

⒓：Junit单元测试Service层

⒔：Junit单元测试数据驱动，在执行单元测试结束的时候，会自动执行此文件中的delete语句

⒕：Junit单元测试数据驱动，在执行单元测试开始的时候，会自动执行此文件中的insert语句

### Api项目结构



①：Api项目对外提供的接口类

②：Constants常量帮助类

③：Api项目对外提供接口类的返回对象Dto

## 统一Jar包版本

\*\*\*注释：以下Jar包引入使用此版本，避免引起Jar冲入。（实时维护）

|  |  |  |
| --- | --- | --- |
| Name | Version | Remark |
| spring | 4.3.5.RELEASE |  |
| mybatis | 3.3.0 |  |
| mybatis-spring | 1.2.0 | Mybatis集成spring |
| redis.clients | 2.9.0 |  |
| spring-data-redis | 1.8.8.RELEASE | Redis集成spring |
| Rabbitmq | 1.7.4.RELEASE |  |
| aspectjrt | 1.7.3 | AOP语法 |
| aspectjweaver | 1.7.3 | AOP动态代理 |
| javassist | 3.15.0-GA | AOP动态代理 |
| shiro | 1.2.1 |  |
| servlet-api | 3.1.0 |  |
| commons-codec | 1.10 | commons |
| commons-lang3 | 3.6 | commons |
| slf4j-api | 1.7.25 | Log |
| logback-classic | 1.2.3 | Log |
| log4j-over-slf4j | 1.7.25 | Log |
| jcl-over-slf4j | 1.7.25 | Log |
| jul-to-slf4j | 1.7.25 | Log |
| log4jdbc-log4j2-jdbc4 | 1.15 | Log |
| jasypt | 1.9.2 | jasypt加密 |
| jasypt-spring3 | 1.9.2 | jasypt加密 |
| jackson | 2.7.4 | Json |
| junit | 4.12 |  |
| dubbo | 2.8.4 |  |
| cat-client | 1.3.6-SNAPSHOT | Cat监控 |
| disconf | 2.6.30 |  |
| mysql-connector-java | 5.1.37 |  |
| alibaba.druid | 1.0.12 |  |

## 项目生成工具

Maven插件方式

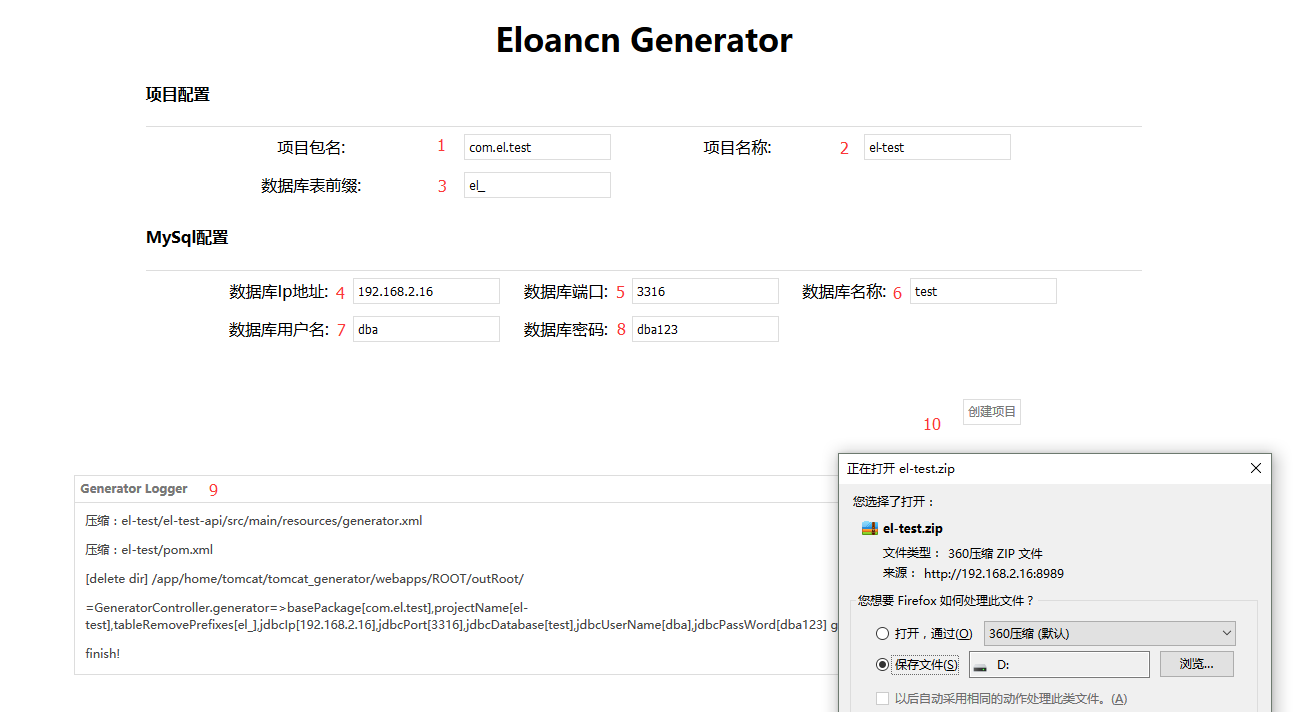
在线生成方式

项目生成提供了在线生成方式。生成项目之后，在开发过程中修改数据库的表结构，需要重新生成DAO层代码的时候，可以使用Maven插件的方式，重新生成底层代码。

### 在线生成工具

|  |
| --- |
| 在线地址（暂定）：http://192.168.2.16:8989/ |

直接访问该地址，填写参数，点击提交即可自动生成项目，并下载。具体如下：



①：项目的包根目录

②：项目的名称

③：数据库表的前缀，可以过滤掉，多个使用逗号分隔

④：数据库的IP地址

⑤：数据库的端口号

⑥：数据库的名称

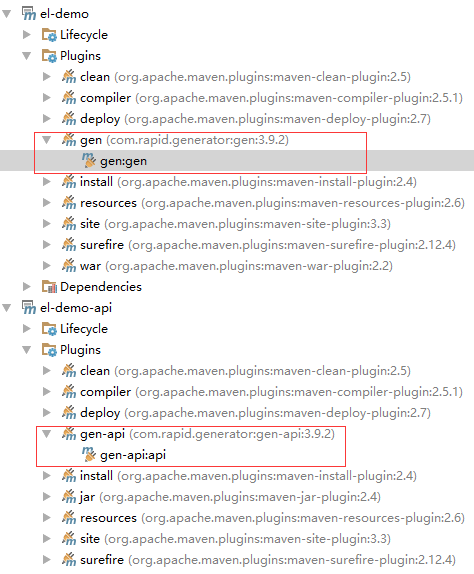
⑦：数据库的用户名

⑧：数据库的密码

⑨：点击Generator按钮后，后台操作日志输出台

⑩：点击Generator按钮后，即可下载项目

### Maven插件IDEA配置



生成的项目中自带【Maven方式的插件】，因修改表结构，需要重新生成底层文件的时候， 直接删除原来文件， 单击此插件即可生成。

### Maven插件Eclipse配置

直接配置eclipse的Maven Build即可。

