|  |  |  |  |
| --- | --- | --- | --- |
| 深圳市快传技术有限公司 | **文档编号** | **产品版本** | **密级** |
| flypaas-admin | 1.0 |  |
| **产品名称：**Flypaas管理后台 | | **共页** |

**Flypaas管理后台详细设计说明书**

**(仅供内部使用)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 拟制： |  |  | 日期： | 2014-07-16 |
| 审核： |  |  | 日期： |  |
| 核准： |  |  | 日期： |  |
| 签发： |  |  | 日期： |  |
| **文档版本：1.0** | | | | |

**版权所有侵权必究**文件修改记录

| 修改日期 | 版本 | 修改页码、章节、条款 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- |
| 2014-07-16 | 1.0 | 初始编写 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 综述 5](#_Toc393815483)

[2. 代码框架说明 6](#_Toc393815484)

[2.1 目录结构 6](#_Toc393815485)

[2.2 源文件说明 7](#_Toc393815486)

[2.3 架构及配置说明 9](#_Toc393815487)

[2.3.1 架构说明 9](#_Toc393815488)

[2.3.2 框架简介 9](#_Toc393815489)

[2.3.3 配置及使用 10](#_Toc393815490)

[2.3.4 jar包统计 25](#_Toc393815491)

[2.4 编码规范 26](#_Toc393815492)

[2.5 运行配置 29](#_Toc393815493)

[2.5.1 开发环境 29](#_Toc393815494)

[2.5.2 测试环境 31](#_Toc393815495)

[2.5.3 线上环境 33](#_Toc393815496)

[3. 系统的分析与设计 34](#_Toc393815497)

[3.1 系统的实现目标 34](#_Toc393815498)

[3.2 系统分析 35](#_Toc393815499)

[3.2.1 系统功能分析 35](#_Toc393815500)

[3.2.2 系统流程分析 36](#_Toc393815501)

[3.3 系统数据库设计 36](#_Toc393815502)

[3.3.1 系统E-R图 36](#_Toc393815503)

[3.3.2 数据库表设计 37](#_Toc393815504)

[4. 模块设计和实现 39](#_Toc393815505)

[4.1 公共模板 39](#_Toc393815506)

[4.1.1 登录 39](#_Toc393815507)

[4.1.2 权限过滤 40](#_Toc393815508)

[4.1.3 自定义标签 41](#_Toc393815509)

[4.2 信息管理 42](#_Toc393815510)

[4.2.1 管理员中心 42](#_Toc393815511)

[4.2.2 开发者管理 50](#_Toc393815512)

[4.2.3 应用管理 52](#_Toc393815513)

[4.2.4 client管理 53](#_Toc393815514)

[4.3 财务管理 55](#_Toc393815515)

[4.3.1 开发者账务 55](#_Toc393815516)

[4.3.2 client账务 57](#_Toc393815517)

[4.3.3 资费套餐 58](#_Toc393815518)

[4.3.4 账单信息 60](#_Toc393815519)

[4.4 运营统计 60](#_Toc393815520)

[4.4.1 开发者注册充值统计 60](#_Toc393815521)

[4.4.2 Client注册充值统计 60](#_Toc393815522)

[4.4.3 充值日志 60](#_Toc393815523)

[4.5 审核管理 60](#_Toc393815524)

[4.5.1 资质审核 60](#_Toc393815525)

[4.5.2 应用审核 62](#_Toc393815526)

[4.5.3 短信模板审核 63](#_Toc393815527)

[4.6 业务管理 65](#_Toc393815528)

[4.6.1 测试DEMO 65](#_Toc393815529)

[4.6.2 语音 66](#_Toc393815530)

[5. 总结 67](#_Toc393815531)

[6. 参考文献 68](#_Toc393815532)

# 综述

关键词：Flypaas、管理后台、详细设计

摘 要：本文描述Flypaas管理后台的详细设计

缩略语说明：

|  |  |  |
| --- | --- | --- |
| **缩略语** | **英文全名** | **中文解释** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

本文描述Flypaas管理后台的设计、开发过程，方便归档、查阅。

本文的读者对象是具有java初级开发能力的技术人员，需掌握以下基础知识：

1. J2EE技术
2. HTML、CSS、JS技术
3. SQL技术

开发环境如下：

1. 开发工具：JDK 1.7、Eclipse、Navicat Premium
2. 版本控制工具：Git
3. 数据库：MySQL 5.1
4. 服务器：Tomcat 7.0

线上环境运行配置：

在tomcat的bin下catalina.sh里，位置cygwin=false前，添加以下内容，将环境配置为production（线上）

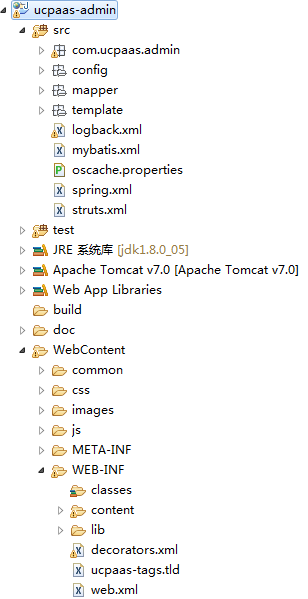
JAVA\_OPTS="-Dspring.profiles.active=production"

项目的SVN路径：svn://172.16.12.16/yunzixun/YXun\_Operate/tags/flypaas-admin

# 代码框架说明

## 目录结构

下图2.1-1是对项目总体的目录结构进行说明：



src源码（图2.2-1有详细说明）

JAVA类

系统配置文件

MyBatis的SQL映射文件

FreeMarker模板文件

Logback日志配置文件

MyBatis的XML配置文件

OSCache缓存配置文件

Spring配置文件

Struts 2配置文件

测试类

相关文档

web页面相关配置（图2.2-2有详细说明）

公共jsp页面：如400、500、footer

CSS样式文件

图片

JS脚本文件

功能JSP页面

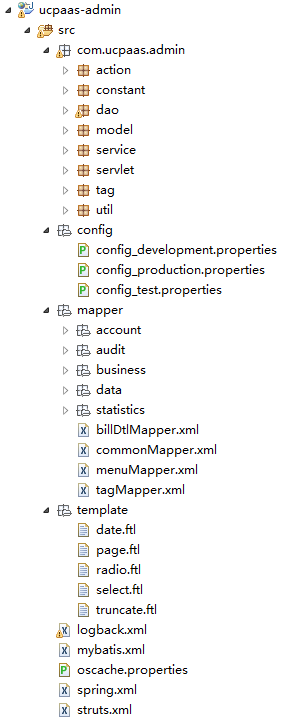
SiteMesh页面布局框架配置文件

自定义标签配置文件

图2.1-1总体的目录结构

## 源文件说明

下图2.2-1是对项目src源码中的目录结构进行说明：



Strut 2的action类，按不同的业务大类再分子包

常数类

对MyBatis封装的持久层操作类

POJO类

业务层实现类，按不同的业务大类再分子包

Servlet类

自定义标签实现类

工具类：rest接口、缓存、加密、文件操作、web、字符串操作等

开发环境的配置文件

线上环境的配置文件

测试环境的配置文件

账务管理的SQL映射文件

审核管理的SQL映射文件

业务管理……

信息管理……

运营统计……

公共业务……

菜单业务……

自定义标签业务……

日期标签的模板文件

分页标签的模板文件

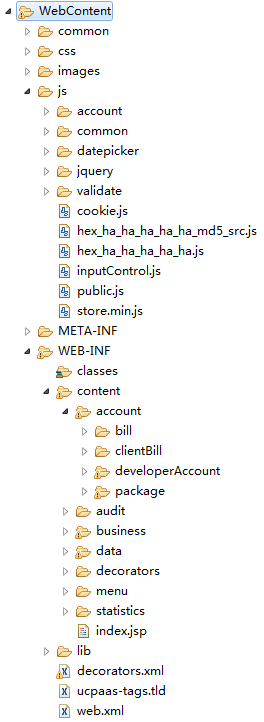
单选框标签……

下拉框标签……

字符串截短标签……

图2.2-1src源码的目录结构

下图2.2-2是对项目WebContent源码中的目录结构进行说明：



财务管理相关JS脚本

页面样式操作相关JS脚本

日期时间选择器（My97 DatePicker）……

Jquery、jquery.form……

JS校验（jquery.validate）……

Cookie操作……

MD5加密源码……

MD5加密min版……

文本框输入控制：如只能输入数字……

公共……

实现浏览器的本地存储……

财务管理的JSP页面

财务管理-账单信息的JSP页面

财务管理-client账务……

财务管理-开发者账务……

财务管理-资费套餐……

审核管理……

业务管理……

信息管理……

SiteMesh框架使用的装饰页面……

菜单子页面……

运营统计……

登录页面……

图2.2-2WebContent源码的目录结构

## 架构及配置说明

### 架构说明

本系统基于B/S的开发模式，前台使用JSP技术开发实现，结合JS脚本、CSS样式控制页面，控制层使用Strut 2框架，业务层使用Spring框架，持久层使用MyBatis框架对MySQL数据库进行封装。

同时使用了其他辅助框架提高系统性能、稳定性、可维护性和用户体验，如使用OSCache对页面进行缓存以提高页面响应速度、使用SiteMesh对页面进行布局控制最终得到布局和外观一致的页面以减小代码复杂度提高开发效率……

### 框架简介

|  |  |  |
| --- | --- | --- |
| **序号** | **框架名称** | **简介** |
|  | DBCP | DBCP(DataBase connection pool),数据库连接池。是 apache 上的一个 java 连接池项目，也是 tomcat 使用的连接池组件。 |
|  | FastDFS | FastDFS是一个开源的轻量级分布式文件系统，它对文件进行管理，功能包括：文件存储、文件同步、文件访问（文件上传、文件下载）等，解决了大容量存储和负载均衡的问题。特别适合以文件为载体的在线服务，如相册网站、视频网站等等。 |
|  | Gson | Gson（又称Google Gson）是Google公司发布的一个开放源代码的Java库，主要用途为串行化Java对象为JSON字符串，或反串行化JSON字符串成Java对象。 |
|  | HttpClient | HttpClient 是 Apache Jakarta Common 下的子项目，可以用来提供高效的、最新的、功能丰富的支持 HTTP 协议的客户端编程工具包，并且它支持 HTTP 协议最新的版本和建议。 |
|  | joda-time | Joda-Time提供了一组Java类包用于处理包括ISO8601标准在内的date和time。可以利用它把JDK Date和Calendar类完全替换掉，而且仍然能够提供很好的集成。 |
|  | Logback | Logback是由log4j创始人设计的又一个开源日志组件。logback当前分成三个模块：logback-core,logback- classic和logback-access。logback-core是其它两个模块的基础模块。logback-classic是log4j的一个改良版本。此外logback-classic完整实现SLF4J API使你可以很方便地更换成其它日志系统如log4j或JDK14 Logging。logback-access访问模块与Servlet容器集成提供通过Http来访问日志的功能。 |
|  | MyBatis | MyBatis 是支持普通 SQL 查询,存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射,将接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java 对象)映射成数据库中的记录。 |
|  | MySQL | MySQL[1] 是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司。Mysql是最流行的关系型数据库管理系统，在WEB应用方面MySQL是最好的RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。 |
|  | OSCache | OSCache由OpenSymphony设计，它是一种开创性的JSP定制标记应用，提供了在现有JSP页面之内实现快速内存缓冲的功能。  OSCache是个一个广泛采用的高性能的J2EE缓存框架，OSCache能用于任何Java应用程序的普通的缓存解决方案。 |
|  | SiteMesh | SiteMesh 是一个网页布局和修饰的框架，利用它可以将网页的内容和页面结构分离，以达到页面结构共享的目的。  OS(OpenSymphony)的SiteMesh是一个用来在JSP中实现页面布局和装饰（layout and decoration）的框架组件，能够帮助网站开发人员较容易实现页面中动态内容和静态装饰外观的分离。 |
|  | Spring | Spring是一个开源框架，Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson 在其著作Expert One-On-One J2EE Development and Design中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。 |
|  | Struts 2 | Struts 2是Struts的下一代产品，是在 struts 1和WebWork的技术基础上进行了合并的全新的Struts 2框架。其全新的Struts 2的体系结构与Struts 1的体系结构差别巨大。Struts 2以WebWork为核心，采用拦截器的机制来处理用户的请求，这样的设计也使得业务逻辑控制器能够与ServletAPI完全脱离开，所以Struts 2可以理解为WebWork的更新产品。虽然从Struts 1到Struts 2有着太大的变化，但是相对于WebWork，Struts 2的变化很小。 |

表2.3.2-1 框架简介

### 配置及使用

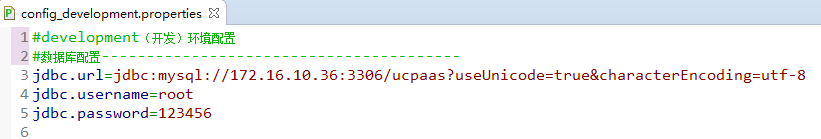
注：在配置及使用前均需引入相应的jar包，详情请参见表2.3.4-1。

#### DBCP

BDCP配置及使用步骤：

1. 根据不同的运行环境（开发、测试、线上），在相应的配置文件

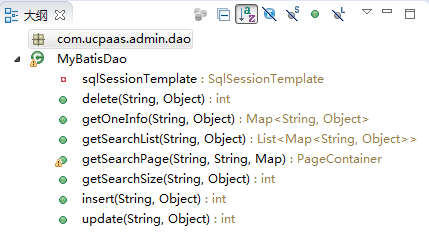
（/flypaas-admin/src/config/config\_development.properties、config\_test.properties、config\_production.properties）中配置连接数据库的参数，如url、用户名、密码等



1. 在Spring 的配置文件（/flypaas-admin/src/spring.xml）中对数据源进行封装（sqlSessionTemplate、transactionManager），以方便注入及做事务处理



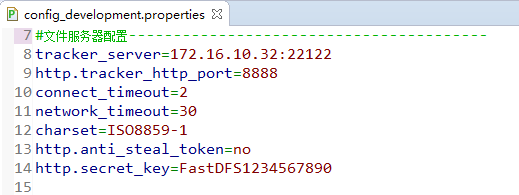
1. 在持久层类（MyBatisDao.java）中注入SqlSessionTemplate，然后就可以对数据库进行操作了



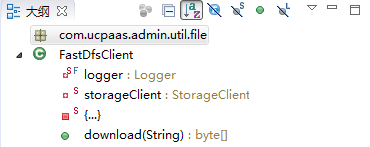
#### FastDFS

1. 在文件服务器上安装FastDFS的服务器端，并做好相关配置，启动服务
2. 根据不同的运行环境（开发、测试、线上），在相应的配置文件

（/flypaas-admin/src/config/config\_development.properties、config\_test.properties、config\_production.properties）中配置文件服务器的参数，如IP、端口、秘钥等



1. 在工具类（FastDfsClient.java）中调用FastDFS的jar包提供的方法连接上文件服务器，然后进行上传、下载、删除、获取文件信息等操作。



#### Gson

1. 使用Gson的jar包提供的方法就可以将java对象和json字符进行相互的转换
2. 详见工具类JsonUtils.java

/\*\*

\* json字符串工具类

\*

\* **@author**xiejiaan

\*/

**publicclass** JsonUtils {

**privatestatic** Gson *gson* = **new** Gson();

/\*\*

\* 将对象转换成为json字符串

\*

\* **@param** obj

\* **@return**

\*/

**publicstatic** String toJson(Object obj) {

**return***gson*.toJson(obj);

}

/\*\*

\* 将json字符串转换成为对象

\*

\* **@param** json

\* **@param** classOfT

\* **@return**

\*/

**publicstatic**<T> T toObject(String json, Class<T>classOfT) {

**return***gson*.fromJson(json, classOfT);

}

}

#### HttpClient

1. 使用HttpClient的jar包提供的方法就可以在java程序中方便的动态发送http请求和处理响应了
2. 详见工具类HttpClientUtils.java

/\*\*

\* 发送get请求

\*

\* **@param** url

\* **@return**

\*/

**publicstatic** String doGet(String url) {

String msg = **null**;

**try** {

msg = Request.*Get*(url).execute().returnContent().asString();

} **catch** (Exception e) {

***LOGGER***.error("发送get请求失败, url=" + url, e);

}

***LOGGER***.debug("发送get请求, url=" + url + ", msg=" + msg);

**return**msg;

}

#### joda-time

1. 使用joda-time的jar包提供的方法就可以方便的处理日期、时间，举例如下

//根据类型（日、周、月）获取下一个周期的日期

if (type.equals("day")) {

DateTime dt = new DateTime(parseDate(date, "yyyyMMdd"));

table = table + dt.plusDays(1).toString("yyyyMMdd");

} elseif (type.equals("week")) {

DateTime dt = new DateTime(parseDate(date, "yyyyMMdd"));

table = table + dt.plusWeeks(1).toString("yyyyMMdd");

} else {

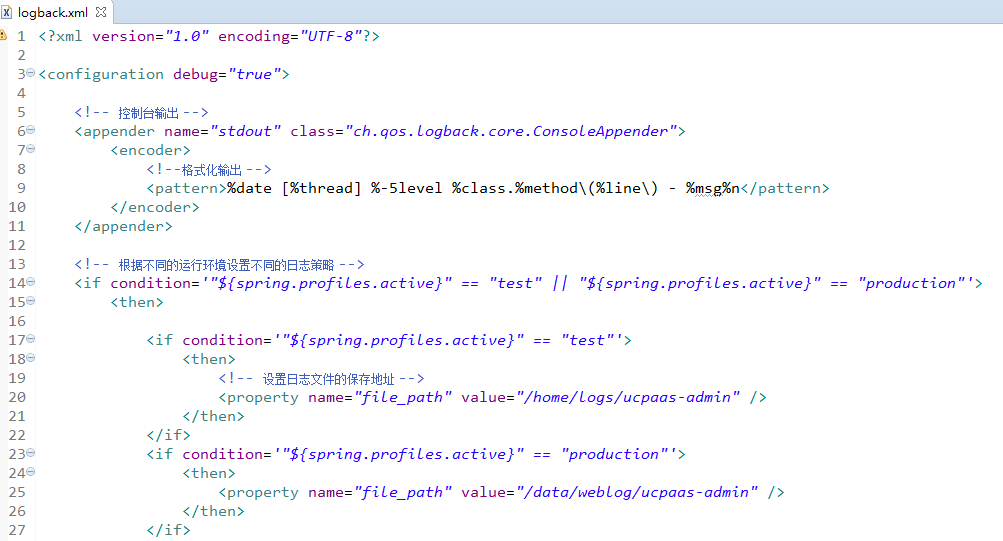
DateTime dt = new DateTime(parseDate(date, "yyyyMM"));

table = table + dt.plusMonths(1).toString("yyyyMMdd");

}

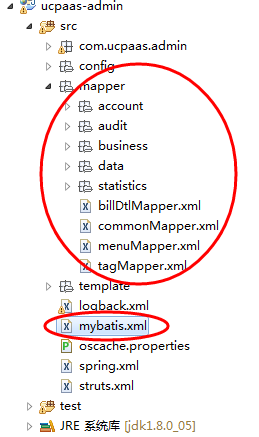
#### Logback

1. 在Logback的配置文件（/flypaas-admin/src/logback.xml）中定义日志输出格式、目的地、输出级别等，可以根据不同的运行环境设置不同的日志策略：控制台输出、debug级别的文件输出、error级别的文件输出、error级别日志发送邮件等



#### MyBatis

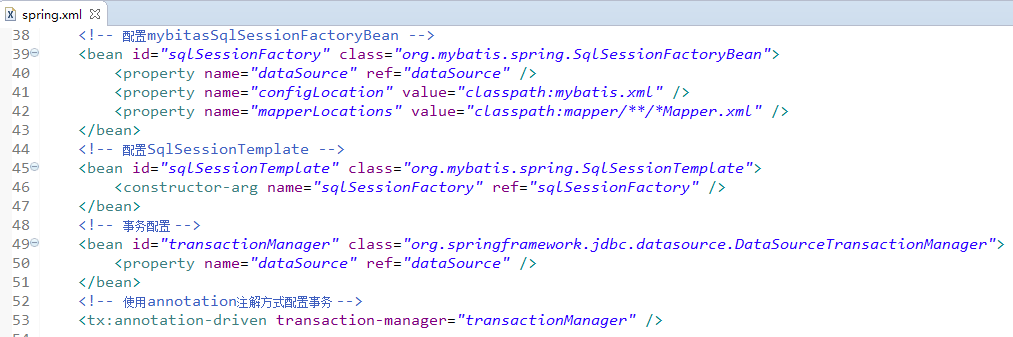
1. 定义MyBatis的XML配置文件（/flypaas-admin/src/mybatis.xml）、MyBatis的SQL映射文件（/flypaas-admin/src/mapper/\*\*/\*Mapper.xml）



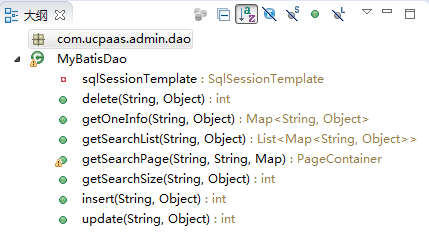




1. 在Spring的配置文件（/flypaas-admin/src/spring.xml）中加载定义好的配置文件，并将数据源注入sqlSessionFactory，然后获取到sqlSessionTemplate



1. 在持久层类（MyBatisDao.java）中注入SqlSessionTemplate，然后就可以对数据库进行操作了



1. 在业务类中根据SQL映射文件中定义好的唯一标示符执行相应的sql语句

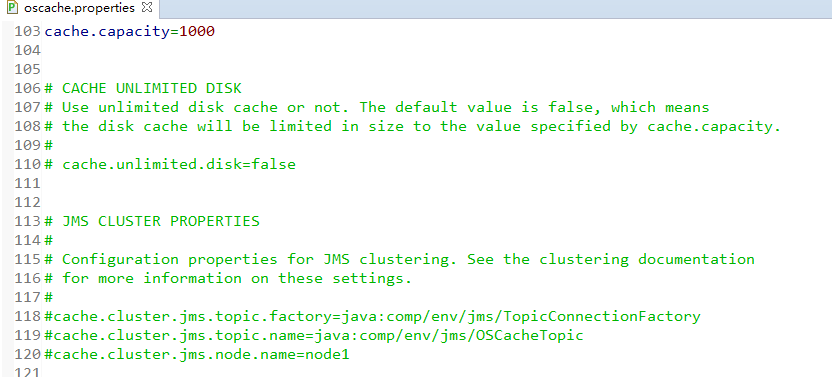


#### MySQL

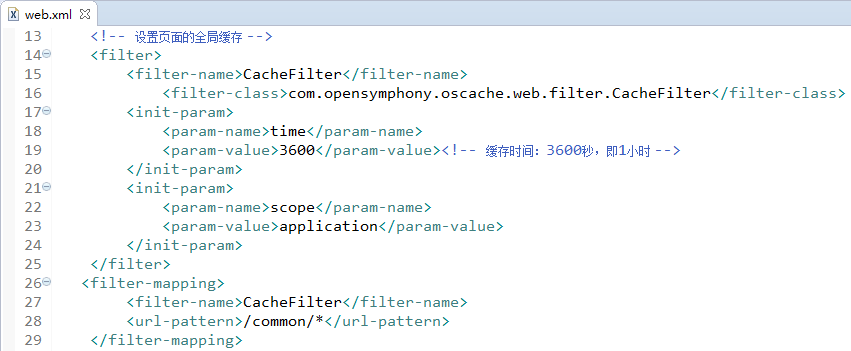
1. 使用MySQL的驱动类，根据url、用户名、密码连接上MySQL数据库，然后操作数据库中的数据
2. 使用DBCP对其封装为数据库连接池，方便维护连接

#### OSCache

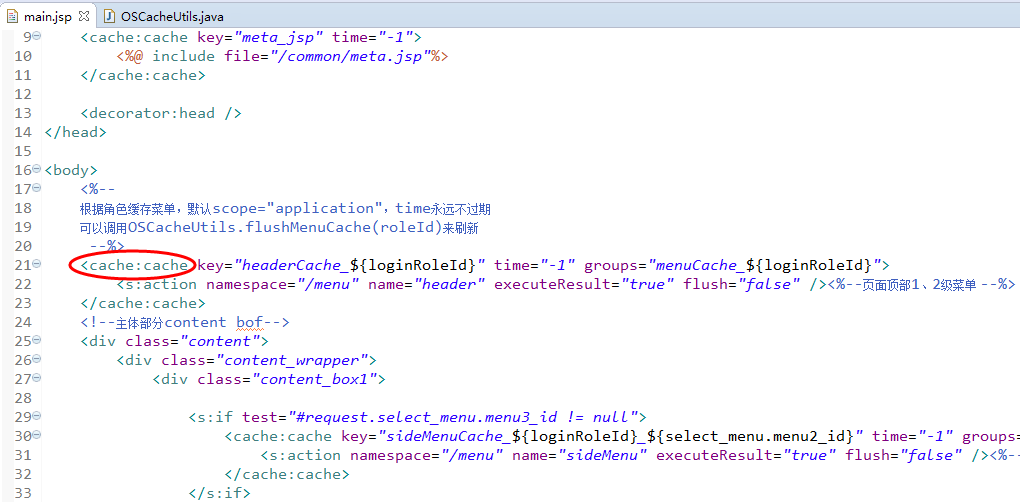
1. 定义OSCache缓存配置文件（/flypaas-admin/src/oscache.properties）

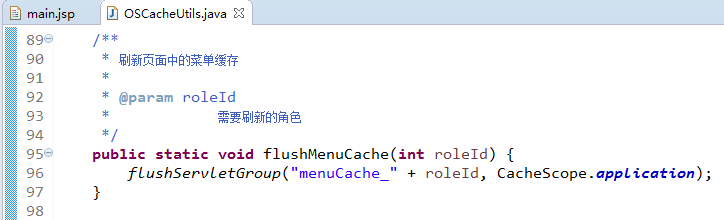


1. 有以下3种使用方式：
   1. 使用其提供的CacheFilter过滤特定的url，并对该url的响应内容进行缓存，参见web.xml

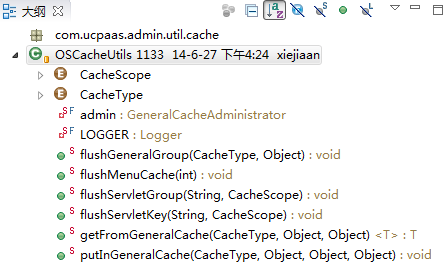


* 1. 使用其提供的标签（<cache:cache）将页面局部内容进行缓存，同时可以设置缓存时间、缓存的key等，根据key可以在程序中动态刷新，参见/flypaas-admin/WebContent/WEB-INF/content/decorators/main.jsp、OSCacheUtils.flushMenuCache(roleId)方法





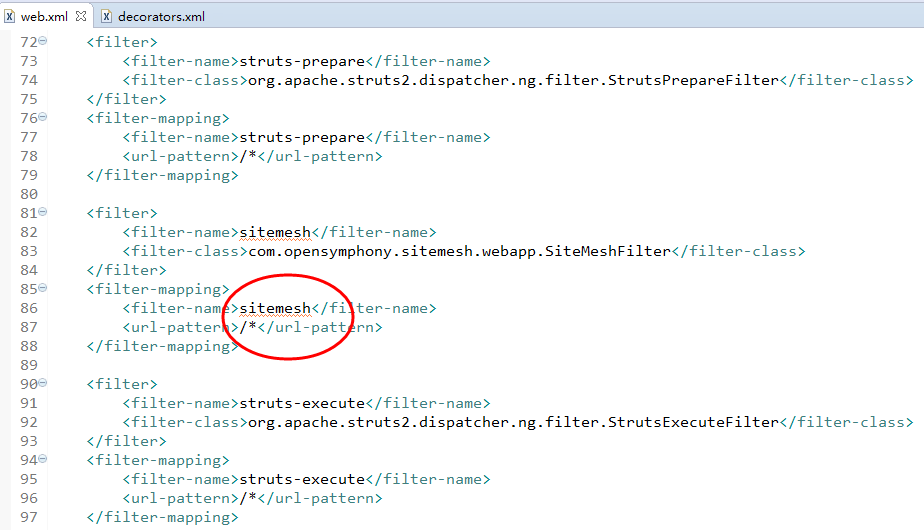
* 1. 使用其提供的方法可以将对象添加进缓存、刷新、删除等，参见OSCacheUtils.java



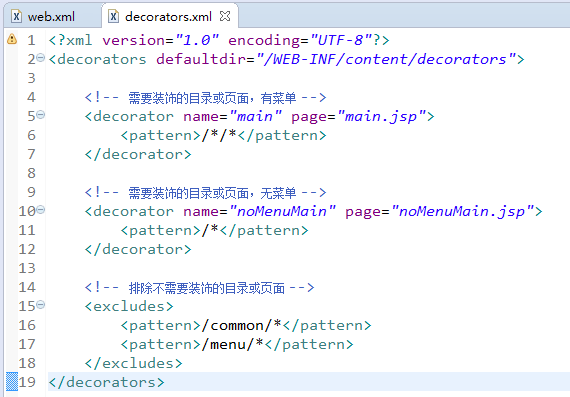
#### SiteMesh

1. 在web.xml中配置SiteMeshFilter，并过滤特定的url

注意：SiteMeshFilter的过滤顺序必须在StrutsPrepareFilter之后、StrutsExecuteFilter之前



1. 在SiteMesh配置文件（/flypaas-admin/WebContent/WEB-INF/decorators.xml）中定义装饰规则和策略

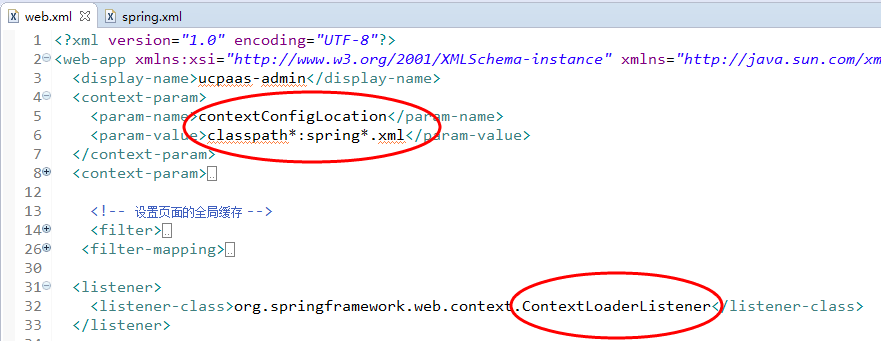


1. 在装饰页面使用其提供的标签（<decorator:）在相应的地方添加进被装饰页面html的title、head、body内容

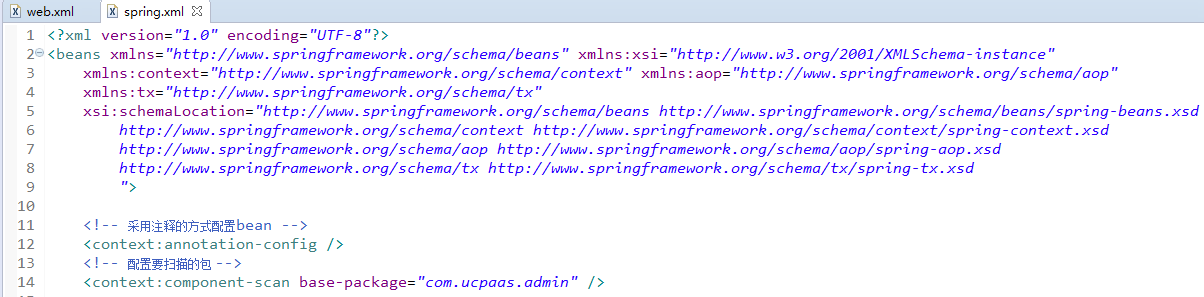


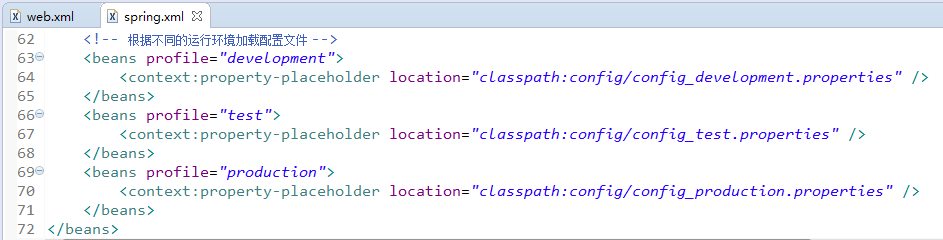
#### Spring

1. 在web.xml中配置Spring配置文件的路径和ContextLoaderListener

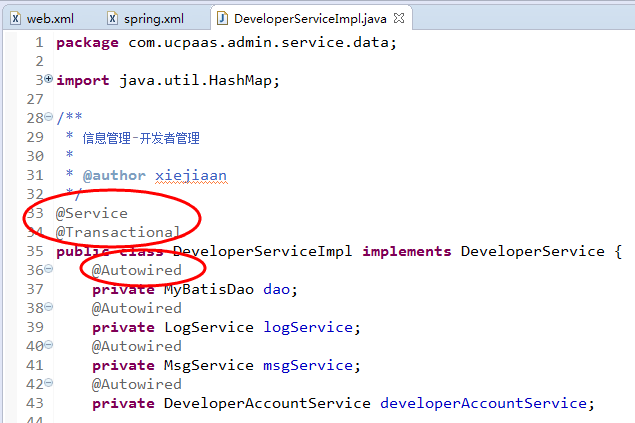


1. 在Spring的配置文件中定义配置内容，如采用注解的方式配置bean、事务管理、根据不同的运行环境加载配置文件等





1. 在java类中使用注解方式注入bean

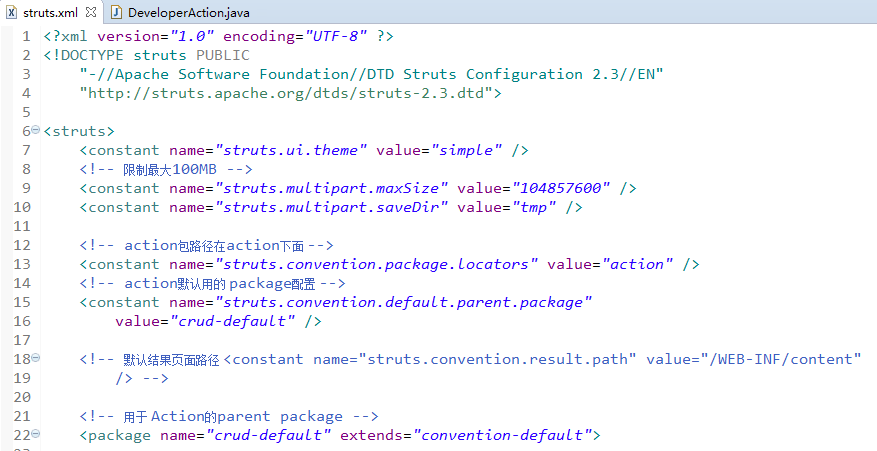


#### Struts 2

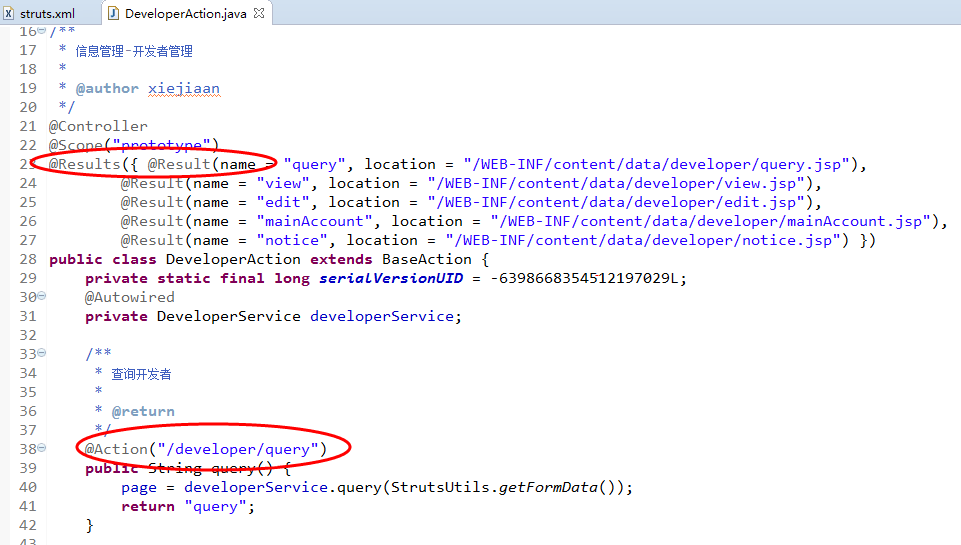
1. 在web.xml中配置Struts 2的Filter，由于使用了SiteMesh，不能直接使用StrutsPrepareAndExecuteFilter，而应使用StrutsPrepareFilter、StrutsExecuteFilter



1. 定义Struts 2的配置文件（/flypaas-admin/src/struts.xml），使用注解方式配置action和result，需要引入相应插件包（struts2-convention-plugin-2.3.16.3.jar）



1. 在action类的方法上使用@Action声明为一个映射url，在类名上使用@Results声明多个返回结果



### jar包统计

下表2.3.4-1是项目的jar包统计（参见【/flypaas-admin/doc/框架说明.txt】文件），可以对代码环境有一个总体的认识

|  |  |  |
| --- | --- | --- |
| **序号** | **框架名称** | **包含的jar包** |
|  | DBCP | commons-dbcp2-2.0.1.jar  commons-pool2-2.2.jar |
|  | FastDFS | fastdfs\_client\_v1.22.jar |
|  | Gson | gson-2.2.4.jar |
|  | HttpClient | fluent-hc-4.3.4.jar  httpclient-4.3.4.jar  httpcore-4.3.2.jar |
|  | joda-time | joda-time-2.3.jar |
|  | Logback | logback-access-1.1.2.jar  logback-classic-1.1.2.jar  logback-core-1.1.2.jar  slf4j-api-1.7.7.jar  javax.mail.jar  commons-compiler.jar  janino.jar |
|  | MyBatis | mybatis-3.2.7.jar  mybatis-spring-1.2.2.jar |
|  | MySQL | mysql-connector-java-5.1.30-bin.jar |
|  | OSCache | oscache-2.4.1.jar |
|  | SiteMesh | sitemesh-2.4.2.jar |
|  | Spring | aopalliance-1.0.jar  asm-3.3.jar  asm-commons-3.3.jar  javassist-3.11.0.GA.jar  spring-aop-4.0.4.RELEASE.jar  spring-beans-4.0.4.RELEASE.jar  spring-context-4.0.4.RELEASE.jar  spring-context-support-4.0.4.RELEASE.jar  spring-core-4.0.4.RELEASE.jar  spring-expression-4.0.4.RELEASE.jar  spring-jdbc-4.0.4.RELEASE.jar  spring-tx-4.0.4.RELEASE.jar  spring-web-4.0.4.RELEASE.jar |
|  | Struts 2 | commons-fileupload-1.3.1.jar  commons-io-2.2.jar  commons-lang3-3.3.2.jar  commons-logging-1.1.3.jar  freemarker-2.3.19.jar  ognl-3.0.6.jar  struts2-convention-plugin-2.3.16.3.jar  struts2-core-2.3.16.3.jar  struts2-spring-plugin-2.3.16.3.jar  xwork-core-2.3.16.3.jar |

表2.3.4-1jar包统计

## 编码规范

1. Package命名：以com.flypaas.admin作为前缀，之后是类的分类名称：action、constant、dao、model、service、servlet、tag、util，其中action、service的子package为各个业务功能的名称：account、audit、business、data、statistics，有对应关系

如：com.flypaas.admin.action.data包中的类表示“信息管理”功能的action类

com.flypaas.admin.service.data包中的类表示“信息管理”功能的service类

1. 所有功能jsp放在/flypaas-admin/WebContent/WEB-INF/content文件夹中，按各个业务功能建立子文件夹
2. 类名、变量名、文件夹名、文件名：使用驼峰式命名法(CamelCase)，其中类名的首字母大写
3. Java类格式化：默认所有Java采用Eclipse自带的格式进行格式化，其中每行最多120个字符，以保持风格统一，便于SVN版本比较
4. 类注释：类的注释需要有创建者的信息，以便维护

/\*\*

\* 信息管理-开发者管理

\*

\* **@author**xiejiaan

\*/

**publicclass** DeveloperAction **extends** BaseAction {

……

1. 方法注释：尽量将各个参数和返回值都进行注释，以便调用时参考

/\*\*

\* 消息业务

\*

\* **@author**xiejiaan

\*/

**publicinterface** MsgService {

/\*\*

\* 直接发送消息

\*

\* **@param** sid

\* 开发者sid

\* **@param** msgType

\* 消息类型

\* **@param** msgTitle

\* 消息标题

\* **@param** msgDesc

\* 消息内容

\* **@return**是否发送成功

\*/

**boolean** sendMsg(String sid, MsgType msgType, String msgTitle, String msgDesc);

……

1. 常数：多次使用的常数需要在常数类中定义，并声明为：public static final

/\*\*

\* 系统常量

\*

\* **@author**xiejiaan

\*/

**publicclass** SysConstant {

/\*\*

\* 项目中的金额费率，1元=1000000

\*/

**publicstaticfinal** String ***money\_rate*** = "1000000";

**publicstaticfinalint*money\_rate\_int*** = 1000000;

/\*\*

\* tb\_bill\_event\_type表中“保底消费”项的id

\*/

**publicstaticfinal** String ***min\_consumption\_event\_id*** = "1017";

……

1. 常数作为形参时最好定义为枚举类型（enum），避免传入的实参有误，可以在enum中定义getValue方法获取真正的值

/\*\*

\* 消息常量

\*

\* **@author**xiejiaan

\*/

**publicclass** MsgConstant {

/\*\* 消息类型 \*/

**publicenum**MsgType {

/\*\* 公告 \*/

***type\_1***(1),

/\*\* 资费变更 \*/

***type\_2***(2),

/\*\* 充值确认 \*/

***type\_3***(3),

/\*\* 余额提醒 \*/

***type\_4***(4),

/\*\* 系统消息 \*/

***system\_msg***(5),

/\*\* 管理员指定消息 \*/

***admin\_msg***(6);

**privateint**value;

MsgType(**int**value) {

**this**.value = value;

}

**publicint** getValue() {

**return**value;

}

}

……

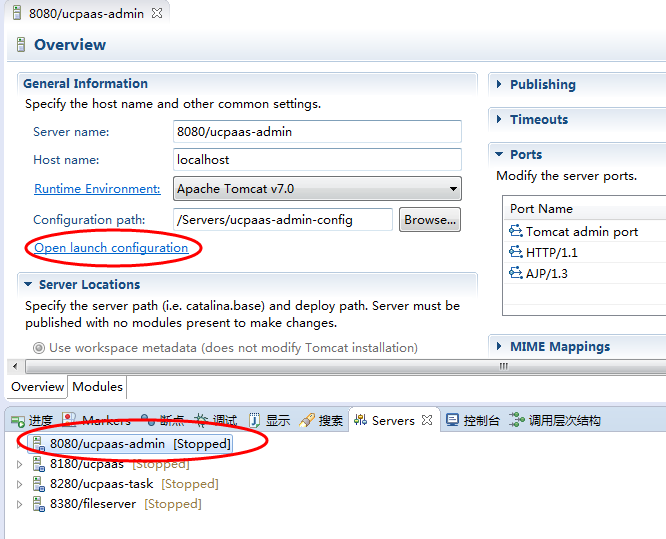
1. JSP、html、CSS文件：尽量符合html 5的规范，如标签和属性名小写（<div></div>）、标签需要有关闭（br/>）、不使用过时的属性和样式等
2. 其他编码规范请参见具体的代码

## 运行配置

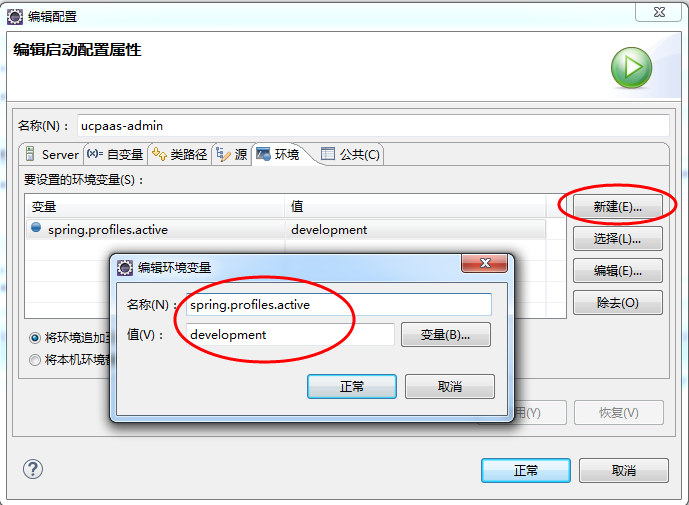
本系统运行时会根据不同的运行环境加载配置文件（参见Logback、Spring的配置），所以只需要定义好环境变量（spring.profiles.active）的值（development、test、production）即可，具体方法如下：

### 开发环境

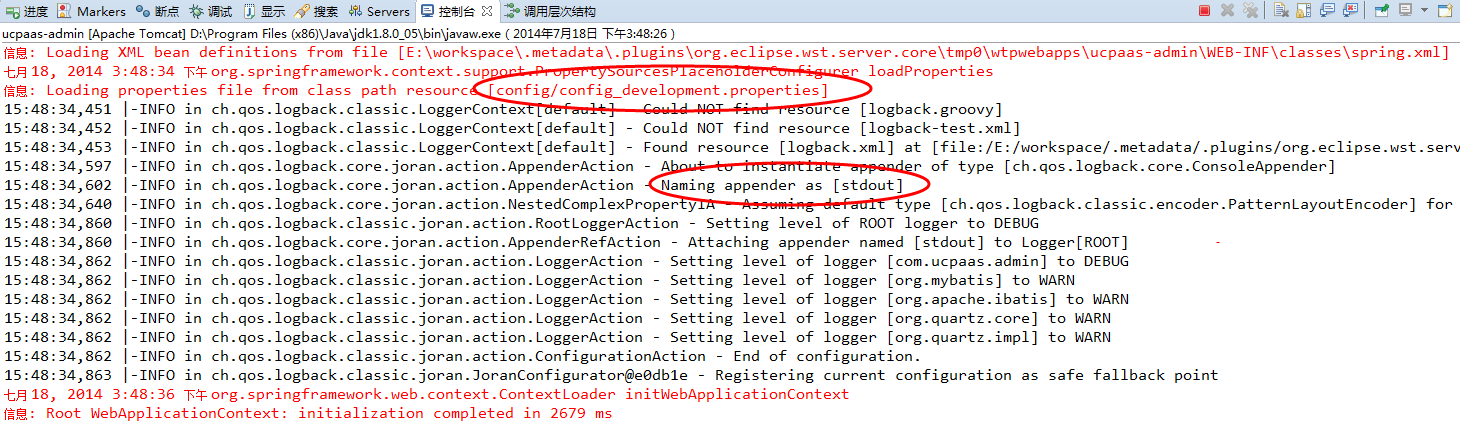
1. 在Eclipse中将项目添加到Tomcat服务器中
2. 打开Tomcat服务器的配置窗口，点击“Open launch configuration”



1. 在弹出窗口中选择“环境”选项卡，点击“新建”，输入环境变量的名称（spring.profiles.active）和值（development）



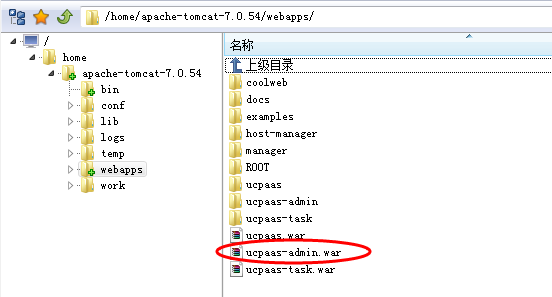
1. 启动该Tomcat服务器，在控制台会输出Spring、Logback在开发环境中的配置信息



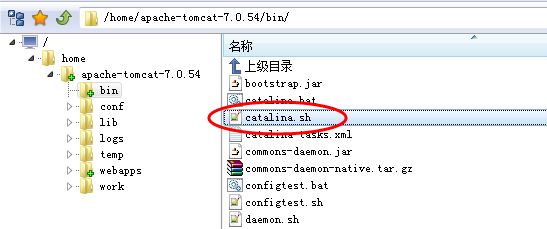
1. Perfect！开发环境配置成功

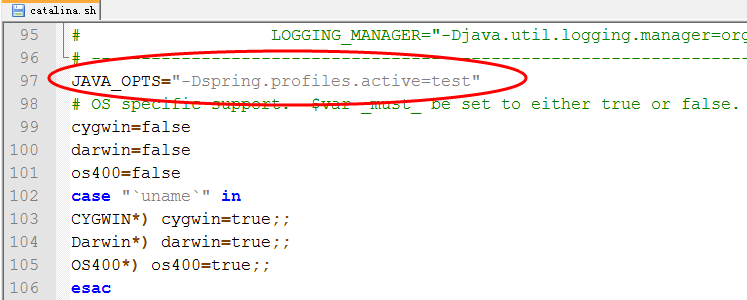
### 测试环境

1. 将项目导出为war包，上传到测试机（172.16.10.32）上的Tomcat目录中（/home/apache-tomcat-7.0.54/webapps/）

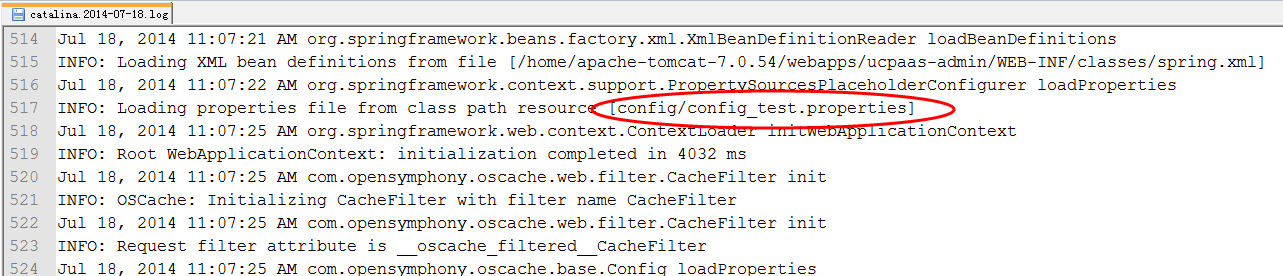


1. 编辑<Tomcat>/bin/catalina.sh文件，在cygwin=false前，添加以下内容将环境配置为test（测试）：JAVA\_OPTS="-Dspring.profiles.active=test"

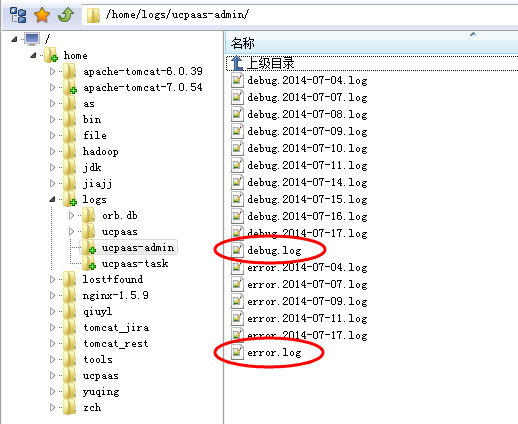




1. 使用命令行（./startup.sh）启动Tomcat，在控制台看到Spring的配置信息



1. 同时测试机上会生成debug级别的日志文件、error级别的日志文件（配置在/flypaas-admin/src/logback.xml中），并且一天一个文件



1. Perfect！测试环境配置成功

### 线上环境

1. 可以参考测试环境的配置，只是将环境变量的值设置为production，即在<Tomcat>/bin/catalina.sh文件，在cygwin=false前，添加以下内容将环境配置为production（线上）：JAVA\_OPTS="-Dspring.profiles.active=production"
2. 或者在Tomcat启动参数中添加参数：-Dspring.profiles.active=production
3. Perfect！线上环境配置成功

# 系统的分析与设计

## 系统的实现目标

帮助运营人员对“flypaas融合通讯开放平台”进行全方位的数据管理，主要包括资源方信息管理、权限管理、开发者财务管理、运营统计、审核管理、业务管理等。

## 系统分析

### 系统功能分析

图3.2.1-1 系统功能结构图

系统功能特征如下：

1. 资源方注册

2. 信息管理（最高权限的人员可以操作）

①管理员管理

查看管理员列表、修改密码、权限管理、操作日志查看、任务日志查看。

②开发者管理

查询开发者列表、查看详情、冻结/解冻，发送通知、查看主账号。

③资源方管理

资源方列表信息、修改资源方信息，资源方消息列表。

④应用管理

查看所有应用、查看应用详情、上线/下线，修改开发 者套餐类型。

⑤Client管理

查看客户列表、查看客户详情。

3. 账务管理

①开发者账务管理

查看开发者列表、操作（查看详情，赠送，充值，扣费）。

②Client账务管理

查看用户列表账户余额，查看详情。

③资源方账务管理

资源方账单列表，资源方的余额以及充值、提款记录。

④资费套餐管理

套餐列表，套餐添加、启用/关闭、修改、删除。查看套餐使用详情。

⑤账单统计

查询账单，导出账单。

4. 运营统计

①开发者注册充值统计

②客户注册充值信息统计

③充值日志

5. 审核管理

①开发者审核

查询开发者信息，并进行审核。

②应用审核

查询应用信息，并进行审核。

③资源方审核

查询资源方信息，并进行审核。

6. 业务管理

查看业务数据：测试DEMO、语音、短信、即时消息等

### 系统流程分析

## 系统数据库设计

### 系统E-R图

使用PowerDesigner导出的系统E-R图如下，可以放大查看：

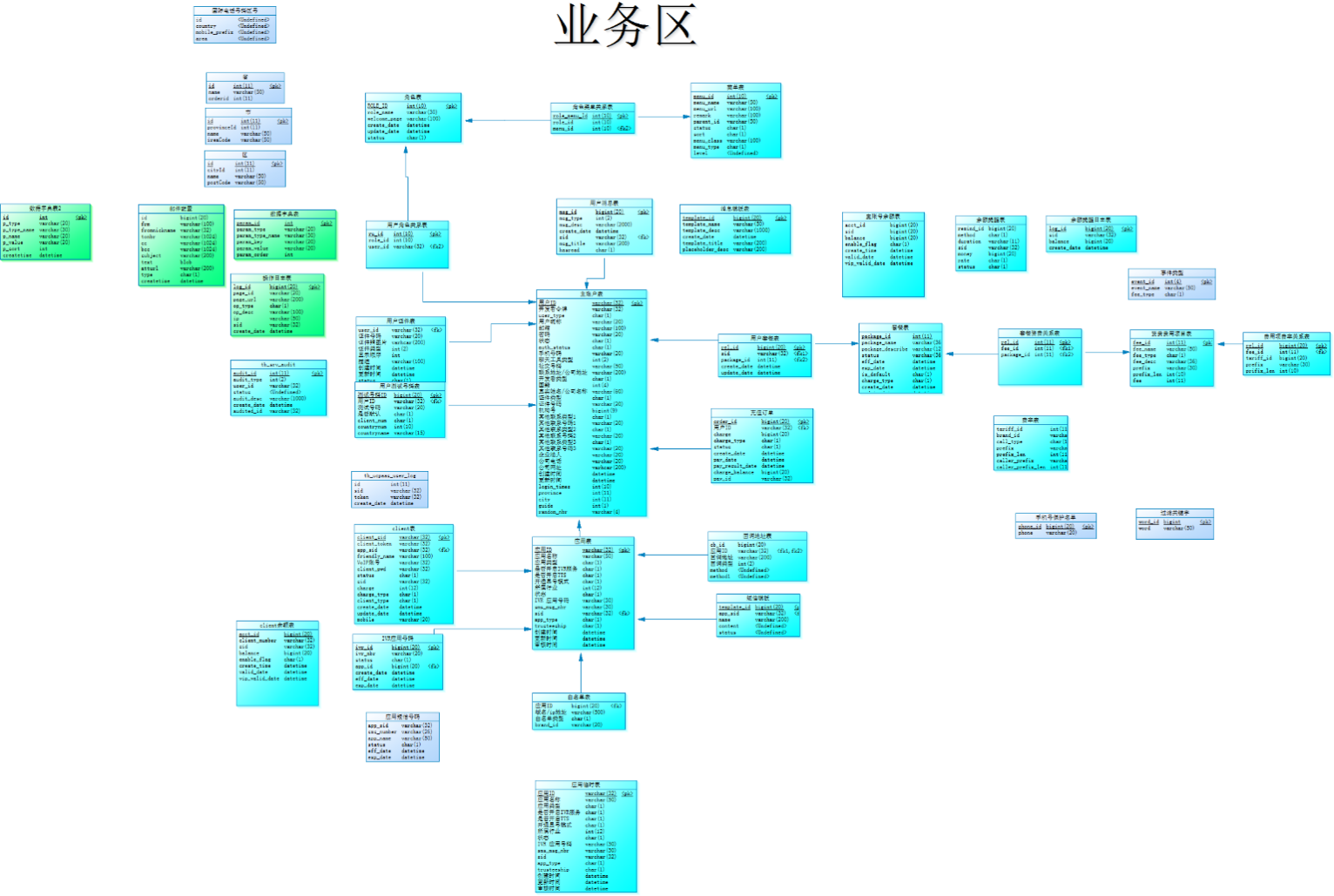


图3.3.1-1系统E-R图

主要是围绕开发者、应用为中心进行数据的存储

### 数据库表设计

|  |  |  |
| --- | --- | --- |
| **序号** | **表名** | **注释** |
|  | tb\_ams\_params | 系统参数表 |
|  | tb\_bill\_acct\_balance | 余额表 |
|  | tb\_bill\_acct\_package\_rel | 用户套餐表 |
|  | tb\_bill\_call\_log\_20140619 |  |
|  | tb\_bill\_call\_log\_20140620 |  |
|  | tb\_bill\_client\_balance | client余额表 |
|  | tb\_bill\_event\_type |  |
|  | tb\_bill\_fee\_item | 资费费用项目表 |
|  | tb\_bill\_fee\_item\_rel | 套餐资费关系表 |
|  | tb\_bill\_package | 套餐表 |
|  | tb\_bill\_servvaliduser |  |
|  | tb\_bill\_tariff |  |
|  | tb\_bill\_time\_acct |  |
|  | tb\_bill\_time\_acct\_internation |  |
|  | tb\_bill\_time\_acct\_interval |  |
|  | tb\_bill\_time\_acct\_world |  |
|  | tb\_bill\_white\_list | 白名单表 |
|  | tb\_srv\_audit | 审核记录表 |
|  | tb\_srv\_callback | 回调地址表 |
|  | tb\_srv\_ivr | IVR应用号码 |
|  | tb\_srv\_payment\_order | 充值订单 |
|  | tb\_srv\_remind | 余额提醒表 |
|  | tb\_srv\_remind\_log |  |
|  | tb\_srv\_sms\_template | 短信模板表 |
|  | tb\_srv\_sql | sql配置表 |
|  | tb\_srv\_white\_list | 白名单表 |
|  | tb\_flypaas\_application | 应用表 |
|  | tb\_flypaas\_city | 市 |
|  | tb\_flypaas\_client | 子帐号表 |
|  | tb\_flypaas\_client\_copy | 子帐号表 |
|  | tb\_flypaas\_country\_mobile\_prefix | 国际电话号码区号 |
|  | tb\_flypaas\_district | 区 |
|  | tb\_flypaas\_filter\_keyword | 过滤关键字 |
|  | tb\_flypaas\_mailprop | 邮件配置表 |
|  | tb\_flypaas\_menu | 菜单表 |
|  | tb\_flypaas\_oprate\_log | 操作日志表 |
|  | tb\_flypaas\_params | 数据字典表 |
|  | tb\_flypaas\_province | 省 |
|  | tb\_flypaas\_role | 角色表 |
|  | tb\_flypaas\_role\_menu | 角色菜单关系表 |
|  | tb\_flypaas\_safe\_phone\_list |  |
|  | tb\_flypaas\_temp\_application | 应用表 |
|  | tb\_flypaas\_test\_number | 用户测试号码表 |
|  | tb\_flypaas\_user | 用户表 |
|  | tb\_flypaas\_user\_log | 变更日志 |
|  | tb\_flypaas\_user\_msg | 用户消息表 |
|  | tb\_flypaas\_user\_msg\_template | 用户消息模板表 |
|  | tb\_flypaas\_user\_pic | 用户证件表 |
|  | tb\_flypaas\_user\_role | 用户角色关系表 |

表3.3.2 数据库表信息统计

每张表的详细设计请到MySQL数据库中查看

# 模块接口设计和实现

## 公共模板

### 登录

|  |  |
| --- | --- |
| **功能描述** | 根据用户名、密码登录系统，并保存登录状态 |
| **页面url** | http://localhost:8080/flypaas-admin/ |
| **页面效果** |  |
| **请求参数** | 用户名、密码 |
| **格式校验** | 1. 用户名不能为空 2. 密码不能为空 |
| **响应结果** | 1. 登录结果需要提示相应信息 2. 若登录成功则跳转到后台页面 |
| **实现的action** | LoginAction.login() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/index.jsp |
| **实现步骤** | 1. 判断是否已登录（session中是否有loginSid），若已登录则提示“当前已登录，请勿重复登录”并执行第6步 2. 判断用户名、密码是否为空，若为空则提示“用户名或密码不能为空” 3. 根据用户名、密码查询用户表（tb\_flypaas\_user），根据查询结果做以下判断：    1. 若结果为null，则提示“用户名或密码错误，请重新输入”    2. 若角色状态已删除，则提示“管理身份（" + user.get("roleName") + "）已删除，登录失败”    3. 若用户状态锁定，则提示“用户已锁定，登录失败”    4. 若用户状态已删除，则提示“用户已删除，登录失败” 4. 更新登录信息：登录次数 5. 保存登录状态：将当前登录用户sid、roleId保存在session中 6. 页面跳转到”管理员资料”页面 |

### 权限过滤

|  |  |
| --- | --- |
| **功能描述** | 根据当前登录用户的管理员身份判断请求的url是否有访问权限 |
| **页面url** | http://localhost:8080/flypaas-admin/admin/view |
| **页面效果** |  |
| **请求参数** |  |
| **格式校验** |  |
| **响应结果** | 1. 若有权限则正常显示页面 2. 若无权限则跳转到403页面 |
| **实现的filter** | AuthorityFilter |
| **JSP页面** |  |
| **实现步骤** | 1. 判断请求的url是否符合如下条件，若符合则直接跳过不做控制，执行第5步    1. url包含.且不是.action，即请求静态文件，如.js文件、.css文件、.jpg文件等    2. url以某些特定的路径开始，如：/menu/（菜单页面）    3. url等于某些特定的路径，如：/,/index,/login,/logout 2. 判断是否为登录状态（即session中是否有roleId），若未登录则跳转到403页面 3. 根据url查询菜单表（tb\_flypaas\_menu），若无记录（即没有配置菜单）则执行第5步，若有记录则取出menuId 4. 根据menuId、roleId查询角色权限表（tb\_flypaas\_role\_menu）    1. 若查询无记录则表示无权限，跳转到403页面    2. 若查询有记录则表示有权限，执行第5步 5. 正常显示页面 |

### 自定义标签

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **案例** |
|  | 分页标签 | <u:page page="${page}" formId="mainForm" /> |
|  | 下拉框标签 | <u:select id="user\_status" value="${param.user\_status}" placeholder="开发者状态" dictionaryType="user\_status" /> |
|  | 日期标签 | <u:date id="start\_date" value="${param.start\_date}" placeholder="开始时间" maxId="end\_date,-1" maxToday="-1" /> |
|  | 单选框标签 | <u:radio name="charge\_type" value="${data.charge\_type}" dictionaryType="pay\_type" excludeValue="" /> |
|  | 权限控制标签 | <u:authority menuId="38"><s:set var="menuId\_38" value="true"/></u:authority> |
|  | 字符串截短标签 | <u:truncate length="10" value="${page\_url}" /> |
|  | 参数格式化标签 | <u:ucparams key="${op\_type}" type="op\_type" /> |
|  | des3加密标签 | <u:des3 value="${img\_url}" /> |

表4.1.3-1 自定义标签列表

|  |  |
| --- | --- |
| **功能描述** | 分页标签：将查询结果的分页信息显示在页面上 |
| **页面url** | http://localhost:8080/flypaas-admin/developer/query |
| **页面效果** |  |
| **请求参数** | 分页查询容器（PageContainer）、表单id（formId） |
| **格式校验** |  |
| **响应结果** | 1. 若总页数等于1，则显示“共1条纪录每页显示30条当前1/1页” 2. 若总页数大于1，则显示“共38条纪录每页显示30条当前2/2页” |
| **实现的类** | PageTag |
| **模板页面** | /flypaas-admin/src/template/page.ftl |
| **实现步骤** | 1. 将请求参数（PageContainer、formId）传入指定的模板文件（page.ftl） 2. 在page.ftl中实现显示逻辑，执行以下步骤：    1. 显示“共1条纪录每页显示30条当前1/1页”    2. 若总页数等于1，则停止执行；若大于1则继续执行    3. 若当前页号大于1，则显示“第1页、上一页”按钮    4. 若当前页号小于总页数，则显示“下一页、最后一页”按钮 |

## 信息管理

### 管理员中心

#### 管理员资料

|  |  |
| --- | --- |
| **功能描述** | 显示当前登录用户的个人信息 |
| **页面url** | http://localhost:8080/flypaas-admin/admin/view |
| **页面效果** |  |
| **请求参数** |  |
| **格式校验** |  |
| **响应结果** | 1. 显示“管理员资料”页面 |
| **实现的action** | AdminAction.view() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/admin/view.jsp |
| **实现步骤** | 1. 从session中获取当前登录用户的sid 2. 根据sid执行以下sql语句获取个人信息   SELECT  a.sid,  a.email,  a.mobile,  a.realname,  c.role\_id,  c.role\_name  FROM  tb\_flypaas\_user a,  tb\_flypaas\_user\_role b,  tb\_flypaas\_role c  WHERE  a.sid = b.sid  AND b.role\_id = c.role\_id  AND a.user\_type = 3  AND a.sid = #{sid}   1. 在JSP页面使用jstl标签显示信息：${data.role\_name} |

#### 管理员列表

|  |  |
| --- | --- |
| **功能描述** | 查询显示所有管理员的信息 |
| **页面url** | http://localhost:8080/flypaas-admin/admin/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、管理员身份 |
| **格式校验** |  |
| **响应结果** | 显示“管理员列表”页面 |
| **实现的action** | AdminAction.query () |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/admin/query.jsp |
| **实现步骤** | 1. 获取request中的请求参数 2. 根据请求参数执行以下sql语句获取所有管理员信息   SELECT  u.sid,  u.email,  u.mobile,  r.role\_name,  u.realname,  date\_format(  u.create\_date,  '%Y-%m-%d %H:%i:%S'  ) AS create\_date,  u.`status`  FROM  tb\_flypaas\_user u,  tb\_flypaas\_user\_role ur,  tb\_flypaas\_role r  WHERE  u.sid = ur.sid  AND ur.role\_id = r.role\_id  AND u.user\_type = 3  AND u.`status` = 1  <if test="text != null">  AND (  u.sid LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  OR u.mobile LIKE concat('%', #{text}, '%')  OR u.realname LIKE concat('%', #{text}, '%')  )  </if>  <if test="role\_id != null">  AND r.role\_id = #{role\_id}  </if>  ORDER BY  u.create\_date DESC   1. 在JSP页面使用Struts 2的循环控制标签（<s:iterator value="page.list">）依次输出每个管理员的信息 |

#### 修改密码

|  |  |
| --- | --- |
| **功能描述** | 修改密码 |
| **页面url** | http://localhost:8080/flypaas-admin/admin/editPassword |
| **页面效果** |  |
| **请求参数** | 当前密码、新密码 |
| **格式校验** | 1. 当前密码、新密码、再次输入新密码不能为空 2. 当前密码、新密码长度为6~20位 3. 新密码、再次输入新密码必须一致 |
| **响应结果** | 显示修改结果 |
| **实现的action** | AdminAction.savePassword() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/admin/editPassword.jsp |
| **实现步骤** | 1. 获取request中的请求参数 2. 对当前密码、新密码进行MD5加密 3. 执行以下sql语句保存新密码    1. 若执行成功，则提示“修改成功”    2. 若执行失败，则提示“当前密码错误，修改失败”   UPDATE tb\_flypaas\_user a  SET a.`password` = #{newPassword},  a.update\_date = now()  WHERE  a.user\_type = 3  AND a.sid = #{sid}  AND a.`password` = #{password} |

#### 权限配置

|  |  |
| --- | --- |
| **功能描述** | 编辑管理身份 |
| **页面url** | http://localhost:8080/flypaas-admin/authority/edit?role\_id=18 |
| **页面效果** |  |
| **请求参数** | 管理身份、权限配置 |
| **格式校验** | 1. 管理身份不能为空、以下拉列表动态显示管理员身份 |
| **响应结果** | 显示添加结果 |
| **实现的action** | AuthorityAction.save() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/admin/authority/edit.jsp |
| **实现步骤** | 1. 管理身份查重，若已重复则提示“管理身份已被使用，请重新输入”，若不重复则继续执行 2. 修改管理身份名称   UPDATE tb\_flypaas\_role r  SET r.role\_name = #{role\_name},  r.update\_date = now()  WHERE  r.role\_id = #{role\_id}   1. 若修改失败，则提示“管理身份不存在，修改失败”，否则继续 2. 若权限配置未做修改，则执行第6步，否则继续 3. 删除原来的权限配置   DELETE  FROM  tb\_flypaas\_role\_menu  WHERE  role\_id = #{role\_id}   1. 添加新的权限配置   INSERT INTO tb\_flypaas\_role\_menu (role\_id, menu\_id)  VALUES  <foreach collection="list" item="item" separator="," >  (#{item.role\_id},#{item.menu\_id})  </foreach>   1. 刷新该管理身份的菜单缓存   OSCacheUtils.flushMenuCache(roleId);   1. 刷新该管理身份的权限缓存   OSCacheUtils.flushGeneralGroup(CacheType.authority\_url, roleId);  OSCacheUtils.flushGeneralGroup(CacheType.authority\_menuId, roleId);   1. 提示“修改成功” |

#### 操作日志

|  |  |
| --- | --- |
| **功能描述** | 查询系统操作日志 |
| **页面url** | http://localhost:8080/flypaas-admin/oprateLog/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、操作类型 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | OprateLogAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/oprateLog/query.jsp |
| **查询sql** | SELECT  l.log\_id,  u.sid,  u.email,  u.realname,  l.page\_url,  l.op\_type,  l.op\_desc,  l.ip,  date\_format(  l.create\_date,  '%Y-%m-%d %H:%i:%S'  ) AS create\_date  FROM  tb\_flypaas\_oprate\_log l,  tb\_flypaas\_user u  WHERE  l.sid = u.sid    <if test="text != null">  AND (  u.email LIKE concat('%', #{text}, '%')  OR l.page\_url LIKE concat('%', #{text}, '%')  OR l.op\_desc LIKE concat('%', #{text}, '%')  OR l.ip LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND l.create\_date >= #{start\_date}  </if>  <if test="end\_date !=null">  <![CDATA[ AND l.create\_date <= #{end\_date} ]]>  </if>  <if test="op\_type != null">  AND l.op\_type = #{op\_type}  </if>  ORDER BY  l.create\_date DESC  ${limit} |

#### 任务日志

|  |  |
| --- | --- |
| **功能描述** | 查询定时任务日志 |
| **页面url** | http://localhost:8080/flypaas-admin/taskLog/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、任务、任务类型、持续时间 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | TaskLogAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/taskLog/query.jsp |
| **查询sql** | SELECT  l.log\_id,  t.task\_id,  t.task\_name,  t.task\_type,  l.data\_date,  date\_format(  l.start\_date,  '%Y-%m-%d %H:%i:%S'  ) AS start\_date,  date\_format(  l.end\_date,  '%Y-%m-%d %H:%i:%S'  ) AS end\_date,  l.duration,  l.remark  FROM  flypaas\_statistics.tb\_flypaas\_task\_log l,  flypaas\_statistics.tb\_flypaas\_task t  WHERE  l.task\_id = t.task\_id    <if test="text != null">  AND (  l.log\_id LIKE concat('%', #{text}, '%')  OR l.data\_date LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND l.start\_date >= #{start\_date}  </if>  <if test="end\_date != null">  <![CDATA[ AND l.start\_date <= #{end\_date} ]]>  </if>  <if test="task\_id != null">  AND t.task\_id = #{task\_id}  </if>  <if test="task\_type != null">  AND t.task\_type = #{task\_type}  </if>  <choose>  <when test='duration == "0"'>  AND l.duration IS NULL  </when>  <when test='duration == "1"'>  AND l.duration = '00:00:00'  </when>  <when test="duration != null">  AND l.duration >= #{duration}  </when>  </choose>  ORDER BY  l.start\_date DESC  ${limit} |

### 开发者管理

|  |  |
| --- | --- |
| **功能描述** | 查询开发者，并进行管理(查看详情，冻结，发送通知，查看主账号) |
| **页面url** | http://localhost:8080/flypaas-admin/developer/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、开发者状态、认证状态 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | DeveloperAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/developer/query.jsp |
| **查询sql** | SELECT  u.sid,  u.email,  u.mobile,  u.username,  date\_format(  u.create\_date,  '%Y-%m-%d %H:%i:%S'  ) AS create\_date,  u.`status`,  u.oauth\_status,  b.enable\_flag  FROM  tb\_flypaas\_user u  LEFT JOIN tb\_bill\_acct\_balance b ON b.sid = u.sid  WHERE  u.user\_type IN (1, 2)    <if test="text != null">  AND (  u.sid LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  OR u.mobile LIKE concat('%', #{text}, '%')  OR u.username LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND u.create\_date >= #{start\_date}  </if>  <if test="end\_date !=null">  <![CDATA[ AND u.create\_date <= #{end\_date} ]]>  </if>  <if test="user\_status != null">  AND u.`status` = #{user\_status}  </if>  <if test="oauth\_status != null">  AND u.oauth\_status = #{oauth\_status}  </if>  <if test="wallet\_status != null">  AND b.enable\_flag = #{wallet\_status}  </if>  ORDER BY  u.create\_date DESC  ${limit} |

### 应用管理

|  |  |
| --- | --- |
| **功能描述** | 查询应用，并进行管理（查看应用详情，强制下线） |
| **页面url** | http://localhost:8080/flypaas-admin/app/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、审核状态、归属行业 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | AppAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/app/query.jsp |
| **查询sql** | SELECT  app.app\_sid,  app.app\_name,  u.email,  date\_format(  app.create\_date,  '%Y-%m-%d %H:%i:%S'  ) AS create\_date,  app.`status`,  app.app\_kind,  app.industry  FROM  tb\_flypaas\_application app Join tb\_flypaas\_user u on u.sid = app.sid  WHERE app.industry !=0    <if test="text != null">  AND (  app.app\_sid LIKE concat('%', #{text}, '%')  OR app.app\_name LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND app.create\_date >= #{start\_date}  </if>  <if test="end\_date != null">  <![CDATA[ AND app.create\_date <= #{end\_date} ]]>  </if>  <if test="status != null">  AND app.`status` = #{status}  </if>  <if test="industry != null">  AND app.industry = #{industry}  </if>    ORDER BY  app.create\_date DESC  ${limit} |

### client管理

|  |  |
| --- | --- |
| **功能描述** | 查询client，并进行管理（查看客户详情） |
| **页面url** | http://localhost:8080/flypaas-admin/client/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、归属行业、状态 |
| **格式校验** | 模糊查询的值不能为空 |
| **响应结果** | 显示查询结果 |
| **实现的action** | ClientAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/data/client/query.jsp |
| **查询sql** | SELECT  c.client\_number,  app.app\_name,  u.email AS developer\_email,  app.industry,  c.`status`,  c.mobile,  date\_format(  c.create\_date,  '%Y-%m-%d %H:%i:%S'  ) AS create\_date  FROM  tb\_flypaas\_user u,tb\_flypaas\_application app,tb\_flypaas\_client c  WHERE  u.sid = app.sid and app.app\_sid = c.app\_sid  AND c.create\_date BETWEEN #{start\_date} AND #{end\_date}  <if test="client\_number != null">  AND c.client\_number = #{client\_number}  </if>  <if test="app\_name != null">  AND app.app\_name = #{app\_name}  </if>  <if test="mobile != null">  AND c.mobile = #{mobile}  </if>    <if test="email !=null">  and c.sid = (select sid from tb\_flypaas\_user where email = #{email} limit 1)  </if>  <if test="status != null">  AND IF (c.`status` = 2, 1, c.`status`) = #{status}  </if>  <if test="industry != null">  and app.industry = #{industry}  </if>  ORDER BY  c.create\_date DESC  ${limit} |

## 财务管理

### 开发者账务

|  |  |
| --- | --- |
| **功能描述** | 查询开发者的账务信息，并进行管理 |
| **页面url** | http://localhost:8080/flypaas-admin/developerAccount/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、钱包金额、开始时间、结束时间、钱包状态 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | DeveloperAccountAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/account/developerAccount/query.jsp |
| **查询sql** | SELECT  t.\*  FROM  (  SELECT  b.acct\_id,  b.sid,  u.email,  b.balance / ${money\_rate} AS balance,  p.package\_name,  b.enable\_flag AS wallet\_status,  date\_format(  (  SELECT  p.pay\_date  FROM  tb\_srv\_payment\_order p  WHERE  p.sid = b.sid  AND p.`status` = 2  ORDER BY  p.pay\_date DESC  LIMIT 0,  1  ),  '%Y-%m-%d %H:%i:%S'  ) AS last\_recharge\_date  FROM  tb\_bill\_acct\_balance b  LEFT JOIN tb\_flypaas\_user u ON u.sid = b.sid  LEFT JOIN tb\_bill\_acct\_package\_rel pr ON pr.sid = b.sid  LEFT JOIN tb\_bill\_package p ON p.package\_id = pr.package\_id  WHERE  1 = 1    <if test="text != null">  AND (  b.acct\_id LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_balance != null">  AND b.balance >= #{start\_balance} \* ${money\_rate}  </if>  <if test="end\_balance != null">  <![CDATA[ AND b.balance <= #{end\_balance} \* ${money\_rate} ]]>  </if>  <if test="wallet\_status != null">  AND b.enable\_flag = #{wallet\_status}  </if>  ) t  WHERE  1 = 1  <if test="start\_date != null">  AND t.last\_recharge\_date >= #{start\_date}  </if>  <if test="end\_date != null">  <![CDATA[ AND t.last\_recharge\_date <= #{end\_date} ]]>  </if>  ORDER BY  t.last\_recharge\_date DESC  ${limit} |

### client账务

|  |  |
| --- | --- |
| **功能描述** | 查询client的账务信息，并进行管理 |
| **页面url** | http://localhost:8080/flypaas-admin/clientBill/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、钱包状态、归属行业 |
| **格式校验** | 模糊查询的值不能为空 |
| **响应结果** | 显示查询结果 |
| **实现的action** | ClientBillAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/account/clientBill/query.jsp |
| **查询sql** | SELECT  t.acct\_id,  t.client\_number,  b.app\_sid,  c.app\_name,  a.email,  t.balance / ${money\_rate} AS balance,  e.package\_name,  date\_format(  t.create\_time,  '%Y-%m-%d %H:%i:%S'  ) AS create\_time,  c.industry,  t.enable\_flag  FROM  tb\_flypaas\_user a, tb\_flypaas\_application c ,  tb\_bill\_acct\_package\_rel d ,tb\_bill\_package e ,  tb\_flypaas\_client b ,tb\_bill\_client\_balance t  WHERE  c.sid = a.sid  and d.package\_id = e.package\_id  and d.sid = a.sid  and b.app\_sid = c.app\_sid  and t.client\_number = b.client\_number  and t.create\_time BETWEEN #{start\_date} AND #{end\_date}  <if test="mobile != null">  AND b.mobile = #{mobile}  </if>  <if test="email !=null">  and t.sid = (select sid from tb\_flypaas\_user where email = #{email} limit 1)  </if>  <if test="app\_name != null">  AND c.app\_name = #{app\_name}  </if>  <if test="wallet\_status != null">  AND t.enable\_flag = #{wallet\_status}  </if>  <if test="industry != null">  AND c.industry = #{industry}  </if>  ORDER BY  t.create\_time DESC  ${limit} |

### 资费套餐

|  |  |
| --- | --- |
| **功能描述** | 查询资费套餐信息，并进行管理 |
| **页面url** | http://localhost:8080/flypaas-admin/package/query |
| **页面效果** |  |
| **请求参数** | 套餐名称 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | PackageAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/account/package/query.jsp |
| **查询sql** | SELECT  p.package\_id,  p.package\_name,  p1.param\_value AS pay\_type\_name,  p.`status`,  p2.param\_value AS status\_name,  (  SELECT  i.fee  FROM  tb\_bill\_fee\_item\_rel ir,  tb\_bill\_fee\_item i  WHERE  ir.package\_id = p.package\_id  AND i.fee\_id = ir.fee\_id  AND i.fee\_id LIKE '${min\_consumption\_event\_id}%'  ) / ${money\_rate} AS min\_consumption,  (  SELECT  count(0)  FROM  tb\_bill\_acct\_package\_rel pr  WHERE  pr.package\_id = p.package\_id  ) AS developer\_number,  date\_format(  p.create\_date,  '%Y-%m-%d %H:%i:%S'  ) AS create\_date  FROM  tb\_bill\_package p  LEFT JOIN tb\_flypaas\_params p1 ON p1.param\_key = p.charge\_type  AND p1.param\_type = 'pay\_type'  LEFT JOIN tb\_flypaas\_params p2 ON p2.param\_key = p.`status`  AND p2.param\_type = 'package\_status'  WHERE  1 = 1    <if test="text != null">  AND p.package\_name LIKE concat('%', #{text}, '%')  </if>  ORDER BY  p.create\_date DESC  ${limit} |

### 账单信息

## 运营统计

### 开发者注册充值统计

### Client注册充值统计

### 充值日志

## 审核管理

### 资质审核

|  |  |
| --- | --- |
| **功能描述** | 查询开发者的资质审核信息，并进行审核 |
| **页面url** | http://localhost:8080/flypaas-admin/qualification/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、认证状态 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | QualificationAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/audit/qualification/query.jsp |
| **查询sql** | SELECT  u.sid,  u.email,  u.mobile,  u.realname,  date\_format(  u.update\_date,  '%Y-%m-%d %H:%i:%S'  ) AS update\_date,  u.oauth\_status,  a.audit\_desc,  u.user\_type  FROM  tb\_flypaas\_user u  LEFT JOIN tb\_srv\_audit a ON a.audited\_id = u.sid AND a.audit\_type = 1  WHERE  u.user\_type IN (1, 2)  AND u.oauth\_status IN (2, 3, 4)    <if test="text != null">  AND (  u.sid LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  OR u.mobile LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND u.update\_date >= #{start\_date}  </if>  <if test="end\_date != null">  <![CDATA[ AND u.update\_date <= #{end\_date} ]]>  </if>  <if test="oauth\_status != null">  AND u.oauth\_status = #{oauth\_status}  </if>  ORDER BY  u.update\_date DESC  ${limit} |

### 应用审核

|  |  |
| --- | --- |
| **功能描述** | 查询应用审核信息，并进行审核 |
| **页面url** | http://localhost:8080/flypaas-admin/appAudit/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、审核状态 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | AppAuditAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/audit/appAudit/query.jsp |
| **查询sql** | SELECT  app.app\_sid,  app.app\_name,  u.email,  date\_format(  app.update\_date,  '%Y-%m-%d %H:%i:%S'  ) AS update\_date,  app.`status`,  a.audit\_desc,  app.app\_kind,  app.industry  FROM  tb\_flypaas\_temp\_application app  JOIN tb\_flypaas\_user u ON u.sid = app.sid  LEFT JOIN tb\_srv\_audit a ON a.audited\_id = app.app\_sid  AND a.audit\_type = 2  WHERE  app.`status` IN (1, 2, 4, 6)  and app.industry !=0    <if test="text != null">  AND (  app.app\_sid LIKE concat('%', #{text}, '%')  OR app.app\_name LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND app.update\_date >= #{start\_date}  </if>  <if test="end\_date != null">  <![CDATA[ AND app.update\_date <= #{end\_date} ]]>  </if>  <if test="status != null">  AND app.`status` = #{status}  </if>  ORDER BY  app.update\_date DESC  ${limit} |

### 短信模板审核

|  |  |
| --- | --- |
| **功能描述** | 查询短信模板审核信息，并进行审核 |
| **页面url** | http://localhost:8080/flypaas-admin/smsAudit/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间、审核状态 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | SmsAuditAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/audit/smsAudit/query.jsp |
| **查询sql** | SELECT  t.template\_id,  t.`name`,  t.content,  app.app\_name,  u.email,  date\_format(  t.update\_date,  '%Y-%m-%d %H:%i:%S'  ) AS update\_date,  t.`status`,  a.audit\_desc  FROM  tb\_srv\_sms\_template t  JOIN tb\_flypaas\_application app ON app.app\_sid = t.app\_sid  JOIN tb\_flypaas\_user u ON u.sid = app.sid  LEFT JOIN tb\_srv\_audit a ON a.audited\_id = t.template\_id  AND a.audit\_type = 3  WHERE  t.`status` IN (1, 2, 3)    <if test="text != null">  AND (  t.template\_id LIKE concat('%', #{text}, '%')  OR t.`name` LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND t.update\_date >= #{start\_date}  </if>  <if test="end\_date != null">  <![CDATA[ AND t.update\_date <= #{end\_date} ]]>  </if>  <if test="status != null">  AND t.`status` = #{status}  </if>  ORDER BY  t.update\_date DESC  ${limit} |

## 业务管理

### 测试DEMO

|  |  |
| --- | --- |
| **功能描述** | 查询测试DEMO的信息 |
| **页面url** | http://localhost:8080/flypaas-admin/testDemo/query |
| **页面效果** |  |
| **请求参数** | 模糊查询的值、开始时间、结束时间 |
| **格式校验** |  |
| **响应结果** | 显示查询结果 |
| **实现的action** | TestDemoAction.query() |
| **JSP页面** | /flypaas-admin/WebContent/WEB-INF/content/business/testDemo/query.jsp |
| **查询sql** | SELECT  u.sid,  u.email,  app.app\_sid,  app.`status`,  date\_format(  app.create\_date,  '%Y-%m-%d %H:%i:%S'  ) AS create\_date,  (  SELECT  sum(y.fee) / ${money\_rate}  FROM  flypaas\_statistics.v\_bill\_client\_year y  WHERE  y.app\_sid = app.app\_sid  ) AS sum\_fee  FROM  tb\_flypaas\_application app  JOIN tb\_flypaas\_user u ON u.sid = app.sid  WHERE  app.app\_type = 0    <if test="text != null">  AND (  u.sid LIKE concat('%', #{text}, '%')  OR u.email LIKE concat('%', #{text}, '%')  OR app.app\_sid LIKE concat('%', #{text}, '%')  )  </if>  <if test="start\_date != null">  AND app.create\_date >= #{start\_date}  </if>  <if test="end\_date != null">  <![CDATA[ AND app.create\_date <= #{end\_date} ]]>  </if>  ORDER BY  app.create\_date  ${limit} |

### 语音

# 总结

本项目主要功能的开发历时2个半月，各种功能、配置和优化系逐步添加，若有错漏之处，烦请指正，谢谢！

# 参考文献

1. DBCP连接池原理分析

<http://elf8848.iteye.com/blog/1931778>

1. FastDFS介绍

<http://baike.baidu.com/view/2466025.htm>

1. 使用Gson实现Json和Java对象的转换

<http://forestqqqq.iteye.com/blog/1942328>

1. HttpClient官网

<http://hc.apache.org/httpclient-3.x/>

1. 替代JDK日期的开源项目：joda-time

<http://www.cnblogs.com/cm4j/archive/2013/05/16/cm4j.html>

1. Logback介绍及入门

<http://blog.csdn.net/wangjunjun2008/article/details/18732019>

1. Logback Appenders文档

<http://logback.qos.ch/manual/appenders.html>

1. struts2配合log4j打印异常栈信息

<http://tdcq.iteye.com/blog/706459>

1. MyBatisMapper XML 文件

<http://mybatis.github.io/mybatis-3/zh/sqlmap-xml.html>

1. Ibatis中#和$的区别

<http://hjxia.iteye.com/blog/186286>

1. Mybatis if test中字符串比较

<http://www.cnblogs.com/andysd/p/3738197.html>

1. OSCache简述

<http://www.cnblogs.com/Irving/archive/2013/04/05/3000722.html>

1. OsCache报错error while trying to flush writer

<http://www.52itstyle.com/thread-765-1-1.html>

1. SiteMesh的使用

<http://www.cnblogs.com/china-li/archive/2013/05/15/3080154.html>

1. spring profile 配置不同的环境

<http://haiziwoainixx.iteye.com/blog/2066427>

1. Srping 静态属性注入

<http://sunlzx.iteye.com/blog/1671521>

1. Struts使用convention-plugin实现零配置

<http://topmanopensource.iteye.com/blog/398263>