**移动App—萌萌**

**后台系统设计说明书（v0.2）**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订时间** | **修订说明** | **修订者** |
| V0.1 | 2014/12/5 | Draft | 程圆建 |
| V0.2 | 2014/12/13 | 完善总体设计和部分详细设计 | 程圆建 |

目录

[1. 系统概述 4](#_Toc406232853)

[2. 系统架构 4](#_Toc406232854)

[3. 总体设计 4](#_Toc406232855)

[3.1 安全认证 5](#_Toc406232856)

[3.2 服务接口 5](#_Toc406232857)

[3.3 数据接口 6](#_Toc406232858)

[3.4 参数配置 6](#_Toc406232859)

[3.5 异常处理 6](#_Toc406232860)

[4. 详细设计 6](#_Toc406232861)

[4.1 接口设计 6](#_Toc406232862)

[4.2 数据库设计 7](#_Toc406232863)

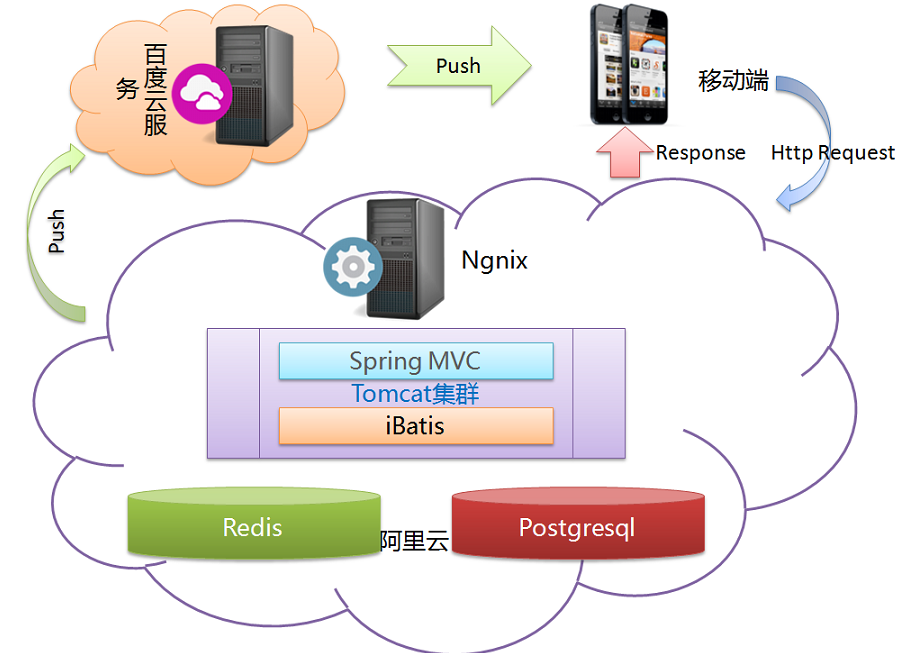
[4.3 全局返回码 7](#_Toc406232864)

## 系统概述

本系统主要为但不限于移动App（萌萌）提供后台服务和管理功能。其中服务包括主动推送和被动调用两种形式。主动推送采用百度云推送服务，把各种动态消息、订阅消息推送至iOS/Andorid客户端 ；被动调用以Restful接口服务的形式供App/Website通过Http请求调用。管理模块供后台技术人员和运维人员使用，包括常规的会员信息管理、宠物信息管理、黑名单/过滤管理、群组管理、营销服务管理、推送管理、日志监控、负载监控等功能。

## 系统架构

本系统逻辑控制层由Spring MVC实现，并暴露Restful接口；数据持久层采用iBatis；数据库用Postgresql做持久化，Redis做系统缓存，承载于Ngnix+Tomcat集群之上。



## 总体设计

注意：本文以下所有涉及的参数设置都是动态可配置。

### 安全认证

本系统针对不同的调用者提供常规服务、涉及数据私密的限制服务，其中限制服务在调用过程中必须经过身份验证，否则不予调用。

调用者在使用限制服务时首先要提供三个必要参数，分别是appid(一般为用户email)、secretid（时间戳）、signature（设备标识，iOS/Android/WebSite/MiniSite）。

然后调用 security/accesstoken的rest服务，来获取accesstoken。该token经过一系列组合后再采用base64加密，在调用限制服务时，需加上?access\_token=TOKEN参数方可调用成功，否则会返回：

{"retcode":-1,"retmsg":"Access Denied, Invalid Access Token"}

每次请求的TOKEN会内置有效时间，在到达有效期时失效。需要重新调用服务获取，为避免恶意调用，同时也限制了调用次数（200次/每天），建议每隔一段时间调用后，暂存入设备本地。

为减轻服务器压力，防止恶意攻击，限制服务也设置了调用次数（1000次/每天）。

其他重要的私密信息，如密码，会采用MD5加密存入数据库。

#### 3.1.1开发者模式

为开发阶段提供了开发者模式（正式上线视情况关闭），使用开发者特殊的token，不受有效期和调用次数的限制，以下为开发者token：

fF6JMWH0R9YVGm7F9vG0GA==

NUd+NwnV1ZYVJqxDLQrSVg==

### 服务接口

本系统目前仅提供两种视图：application/json和text/html，如：

<http://www.moepets.com/user/query/1.json>

或者

<http://www.moepets.com/user/query/1> （不用加任何后缀，默认为restful接口）会返回json格式数据报文，提供服务。

第二种主要供网站调用，需要加.html后缀用于识别，如：

<http://www.moepets.com/home.html>

会返回html页面，提供给官网使用。

更多详细接口说明，请参见后文。

### 数据接口

本系统提供两种数据接口，持久化和内存，分别基于Postgresql和Redis。对于需要频繁交互的数据会放入Redis，且后台会通过quartz启动一个定时任务，每天早上4点把redis里的数据持久化至Postgresql数据库。同时Redis也采用RDB机制，保证自身数据持久化和灾备，这样在服务器宕机后，不需要进行大量的数据同步，也可以很快恢复。

### 参数配置

本系统涉及的所有的参数均支持动态可配，如黑名单/ACL、敏感词配置、任务策略配置等。配置在两个地方，一个是程序内部的appconfig.properties，一个在数据库中，系统启动时会通过Filter初始化方法init()把数据库中的参数配置读入到redis。系统获取参数会优先取程序内置参数，然后再取redis，若都没有，则返回空。

### 异常处理

本系统自定义了两种异常：SystemException和BizLogicException

SystemException表示系统自身的异常，一般发生了这类异常，需要技术人员进行排查原因。

BizLogicException表示业务逻辑异常，一般为操作不合法造成。

所有异常信息，都会通过log4j记载下来，且每半天自动备份一次。

## 详细设计

### 接口设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 接口URI | 入参 | 出参 | 示例说明 |
| 用户类 | /user/query/{id} | 用户id |  | 获取id=1的用户信息 |
| 宠物类 | /pet/query/{id} | 宠物id |  |  |

### 数据库设计

#### 4.2.1审计体系

对于关键信息表，均会加audit trigger进行审计。

#### 4.2.2数据库连接池

采用DHCP连接池，由Spring framework统一管理。

#### 4.2.3 Tablespace/User

#### 4.2.4 Table/View

#### 4.2.5 Sequence

### 全局返回码

返回码仅针对json报文，不同于HTTP状态码。

|  |  |  |
| --- | --- | --- |
| 返回码 | 返回信息 | 说明 |
| 1 | OK | 调用成功 |
| -1 | Unknown Exception | 发生系统无法预料的异常信息 |
| -10001 | Access Denied. Invalid Access Token | Access token无效 |