**用户成长体系服务设计文档**

版本：V0.1

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **时间** | **修订人** | **版本** | **修订内容** |
| 2015-8-31 | 顺炽国 | V0.1 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 概要 5](#_Toc428895588)

[1.1. 背景 5](#_Toc428895589)

[1.2. 目标 5](#_Toc428895590)

[1.3. 名词约定 5](#_Toc428895591)

[2. 系统框架 6](#_Toc428895592)

[2.1. 系统关系图 6](#_Toc428895593)

[2.2. 成长体系关系 7](#_Toc428895594)

[2.3. 系统使用人员 8](#_Toc428895595)

[2.4. 传输协议及编码格式 8](#_Toc428895596)

[2.5. 安全、权限 9](#_Toc428895597)

[2.6. 单服务器情况 10](#_Toc428895598)

[2.7. 多服务器情况 11](#_Toc428895599)

[2.7.1. 风险及规避方法 11](#_Toc428895600)

[3. 系统方案 11](#_Toc428895601)

[3.1. 业务接口 12](#_Toc428895602)

[3.1.1. 更新用户成长信息 13](#_Toc428895603)

[3.1.2. 查询用户积分信息 13](#_Toc428895604)

[3.1.3. 查询积分变更信息 14](#_Toc428895605)

[3.1.4. 查询等级信息 16](#_Toc428895606)

[3.1.5. 注册用户信息推送 17](#_Toc428895607)

[3.1.6. 解除用户信息推送 18](#_Toc428895608)

[3.1.7. 推送用户信息 19](#_Toc428895609)

[3.2. 扩展接口 19](#_Toc428895610)

[3.2.1. 文件访问接口 19](#_Toc428895611)

[3.2.2. 数据访问接口 19](#_Toc428895612)

[3.2.3. 缓存接口 19](#_Toc428895613)

[3.2.4. 集群服务接口 20](#_Toc428895614)

[3.3. 日志记录 20](#_Toc428895615)

[3.4. 系统初始化 20](#_Toc428895616)

[3.5. 成长规则处理 21](#_Toc428895617)

[3.5.1. 创建规则 21](#_Toc428895618)

[3.5.2. 成长规则调用流程 22](#_Toc428895619)

[3.5.3. 成长规则调用信息流 23](#_Toc428895620)

[2.1. 用户信息查询 23](#_Toc428895621)

[2.2. 数据结构 24](#_Toc428895622)

[2.3. 相关开源项目 25](#_Toc428895623)

[3. 实施计划 25](#_Toc428895624)

[4. 风险评估 26](#_Toc428895625)

[5. 性能指标 26](#_Toc428895626)

# 概要

## 背景

目前用户成长体系（用户积分、经验信息管理）是由myusd服务提供，随着需求的增加，用户成长体系规则变更以及涉及的程序改动，在原有服务上实现新的规则修改成本日益增加，所以在B端的用户成长体系需求出来时，实现一个通用的用户成长体系服务。

## 目标

实现用户成长体系服务的基本功能，支持多个成长体系的管理，信息相互隔离。各个成长体系支持成长规则维护，积分、经验等可设置参数的动态配置，成长规则的权限管理（提交、审批、规则测试等），支持规则的后续动态扩展（基于插件模式）。

服务需要能满足高并发访问的需求，支持分布式部署，可方便的实现横向扩展。

## 名词约定

UIAC 统一认证服务中心

UGS User Growing System用户成长体系服务

# 系统框架

## 系统关系图

说明：

1. 接口模块中与UIAC通讯主要是验证用户有效性，暂时不对用户的有效性（是否禁用）作处理（需要识产品需求而定）
2. 缓存模块主要提供快速信息查询使用
3. 数据存储适配层用以切换不同的数据库，而不需要修改代码

## 成长体系关系



说明：

1. 各个成长体系自成一体，维护各自的积分类别、用户等级、成长规则、用户积分信息等规则
2. 成长规则插件仅负责积分写入业务逻辑处理，配置成长规则时在各个配置点写入积分变化情况。

## 系统使用人员



说明：

1. 系统仅考虑向日葵用户的成长体系，对于临时用户不在考虑范围内。
2. 后台使用人员的权限结构独立。

## 传输协议及编码格式

用户成长体系服务服务运行于内网，由业务应用服务调用，建议采用JSON+HTTP的方式。

## 安全、权限



用户成长体系服务运行在内网，因此暂时不考虑接口调用授权机制。



管理后台的权限需要实现RBAC权限架构，维护各个体系内的用户级别、积分规则等功能，产品人员可以进入后台配置、测试体系的详细规则，交由管理人员审核，发布，管理人员可以禁用指定的规则。

## 单服务器情况



说明：

1. 因成长体系服务无须双向通讯，因此仅提供查询和更新接口即可。
2. 公共会话信息存放在缓存服务中，UGS服务实现无状态服务，方便横向扩展。

## 多服务器情况



在单服务器访问量支撑不住的时候，需要横向扩展到多服务器以缓解服务器压力，用户成长体系服务将会话信息存储到第三方缓存系统，服务本身提供无状态服务机制，因此可以视业务负载情况平滑扩展到多台服务。

### 风险及规避方法

1. **单点问题***负载均衡服务*属于单点服务，如果其中一台出现问题，会导致各自负责的业务无法使用，对于此类问题，最简单的解决办法是每个服务增加一个热备，在某台服务出现问题时主动切换到备份机器提供服务。

# 系统方案

系统用例图



## 业务接口

### 更新用户成长信息

#### 功能说明

执行用户成长信息更新操作。

#### 约定

**Resource** growing/{GrowingTypeCode}/{uid}

**METHOD**  POST

#### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| GrowingTypeCode | String | Yes | 成长体系类别代码 |
| Uid | String | Yes | 要更新的用户ID |
| RuleCode | String | Yes | 要更新的规则代码名称 |
| Description | String | No | 规则描述信息，不输入则由默认填写规则名称 |

#### 错误代码

#### 返回值

{

result: Boolean, -- true-成功，false-失败

}

### 查询用户积分信息

#### 功能说明

新查询用户的积分信息。

#### 约定

**Resource** growing/{GrowingTypeCode}/{uid}

**METHOD**  GET

#### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| GrowingTypeCode | String | Yes | 成长体系类别代码 |
| Uid | String | Yes | 要更新的用户ID |

#### 错误代码

#### 返回值

{

Uid:string, --用户ID

TypeCode:string, --成长体系代码

TypeName:string --成长体系名称

Points: --积分详细信息，数组格式，包含不同积分类型的详细信息

[{

PointCode:String, --积分代码

PointName:String, --积分名称

Point: int, --积分项

Grade:int --对应等级（如果有配置的话，则计算，否则默认为0）

},

……

]

}

### 查询积分变更信息

#### 功能说明

查询用户积分的历史变更信息。

#### 约定

**Resource** history/{GrowingTypeCode}/{uid}/{pageNum}/{pageSize}

**METHOD**  GET

#### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| GrowingTypeCode | String | Yes | 成长体系类别代码 |
| Uid | String | Yes | 要更新的用户ID |
| pageSize | Int | No | 分页大小，不输入默认为10，最大限制为50 |
| pageNum | Int | No | 第几页，不输入默认为第1页 |
| queryDateBegin | String | No | 可选，查询开始时间，格式为:yyyy-MM-dd |
| queryDateEnd | String | No | 可选，查询结束时间 |
| ruleCodes | String | No | 可选，默认为查询所有动作代码，多个动作用逗号（，）分隔 |

#### 错误代码

#### 返回值

{

Total: int --记录总数

PageSize:int --分页大小

PageNum: int --当前页数

Result:[obj1,obj2,……] --查询结果，数组格式

}

每个Obj的格式如下：

{

RuleCode:String --积分变更的规则代码

RuleName:String --积分变更的规则名称

AddDate：Date --积分增加时间

Point:int --积分数量

Description:String --描述信息

}

### 查询等级信息

#### 功能说明

查询系统中定义的用户等级定义信息。

#### 约定

**Resource** grade/{GrowingTypeCode}

**METHOD**  GET

#### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| GrowingTypeCode | String | Yes | 成长体系类别代码 |

#### 错误代码

#### 返回值

[{

GradeName: String --等级名称

PointCode:String --积分类别编码

PointName: String --积分类别名称

Details:[obj1,obj2,……] --积分详细信息

}]

每个Obj的格式如下：

{

Level:int --等级

Point:int --等级积分数量

}

### 注册用户信息推送

#### 功能说明

在用户的积分、等级等信息发生变化时，可以主动通知到已经注册的观察者。

#### 约定

**Resource**  observer/{GrowingTypeCode}

**METHOD**  POST

#### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| GrowingTypeCode | String | Yes | 成长体系类别代码 |
| Callback | String | Yes | 回调Url |
| subjectType | String | Yes | 订阅类别，可设置多个，如：信息变更，等级变更等，待约定 |

#### 错误代码

#### 返回值

{

true|false --成功返回true，否则返回false

}

注：

1. 推送的回调接口必须为有效的Http地址，且能正确接收POST请求
2. 推送的信息内容详见3.1.9章节定义的推送内容

### 解除用户信息推送

#### 功能说明

解除已经注册的用户推送注册信息。

#### 约定

**Resource**  observer/{GrowingTypeCode}

**METHOD**  DELETE

#### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| GrowingTypeCode | String | Yes | 成长体系类别代码 |
| Callback | String | Yes | 回调Url |
| subjectType | String | Yes | 订阅类别，可设置多个，如：信息变更，等级变更等，待约定 |

#### 错误代码

#### 返回值

{

true|false --成功返回true，否则返回false

}

### 推送用户信息

#### 功能说明

因考虑到性能问题，推送不会采用实时推送的方式，而会在缓冲区满或到指定推送时间后主动进行推送。

#### 推送内容

以数组的格式推送过来

[authObj1, authObj2,……]

其中每个对象格式如下：

{

uid: string, --对应用户ID

growingTypeCode: String, --成长体系类别

messageType: string, --消息类别

addTime:datetime --添加时间

message:string --消息内容

}

注：

1. 消息内容格式识消息类别而定，具体再约定

## 扩展接口

### 文件访问接口

提供本地文件的访问接口，如：日志、配置文件读写。

### 数据访问接口

提供对数据库访问接口的抽象，业务层仅使用接口，交由不同的数据库处理层实现数据库的驱动和存储。以便于后期数据迁移。因此功能较常用，此处不作特殊说明。

### 缓存接口

缓存接口抽象出来主要目的是为了后续缓存迁移工作，在用户量小的情况，可以使用本地缓存来实现，用户量大的时候，则需要使用外部缓存，如Memcached、Redis等。

本地缓存的实现可以考虑基于LRU或相关的扩展算法来实现。

在用户成长体系服务中，缓存主要是缓存用户的积分信息。



### 集群服务接口

保留，暂时不考虑。

## 日志记录

目前日志统一记录在文件中，格式统一。在任何主要操作，如：查询、修改、登录、注销等都需要记录下用户的操作信息，记录的信息包括：调用的方法、操作人、操作参数、返回值等。

## 系统初始化

系统初始化时，需要初始化部份功能，其中对于用户的积分信息，建议延时加载到缓存中。

## 成长规则处理

### 创建规则



说明：

1. 成长规则插件需要提炼出可配置项，如：加分数量，可能的中转条件等，配置界面需要约定一套动态生成的规则。
2. 每套分组流程可以设置多个规则名称，业务端应用使用规则名称触发规则动作

### 成长规则调用流程



### 成长规则调用信息流



## 成长插件框架



说明：

1. 插件与成长体系服务之间使用IGrowingService对象通讯，此对象在插件初始化时，由PlguinManager传入
2. 插件需要提供初始化（init）方法供PlguinManager调用，用以初始化插件相关的代码
3. DefaultGrowingPlugin是一个默认的抽象类，用以提供一些公共的插件方法，用户编写的成长规则插件继承DefaultGrowingPlguin，实现Invoke方法即可。
4. 所有加载的业务插件实例，都会存放在缓存中，当有匹配的调用请求发生时，查找到对应的实例，触发插件执行操作





## 信息推送服务



## 数据结构



说明：

1. 可以建立多个成长体系，互相隔离
2. 每个成长体系下可以定制多个规则方案分组，不过同一时间仅能使用一个方案分组
3. 每个成长体系可以制定自己的积分项目，创建一个成长体系后，默认会有一个用户积分项，可以增加多个积分项目
4. 成长规则统一用插件处理，系统初始化时加载
5. 每个方案分组中可以有多个成长规则，用以累加用户的成长积分等操作

## 相关开源项目

1. 模版引擎grovvy
2. Java动态编译库:javassit、cglib
3. JSON序列化：Gson、fastJSON
4. 日志记录：log4j、slf4j
5. 数据库:mysql、PostgreSQL、Mongodb
6. 缓存：memcache、redis、Ehcache
7. IOC框架：Spring
8. ORM框架：Hibernate

# 实施计划

考虑到时间及产品需求因素，目前整个项目计划分为三次迭代来实施，计划如下：

**第一迭代：**

1. 优先满足产品业务需求
2. 实现产品的基本框架（不包含管理界面、成长规则插件动态扩展等）
3. 实现业务接口功能
4. 实现业务插件的基本接口（以满足目前产品需求为主）
5. 业务插件仅实现业务需求需要的一次性积分累加和按天限制的插件
6. 数据库实现基本的数据库表操作

**第二次迭代：**

1. 完善成长规则插件接口功能，支持扩展和业务服务互通调用
2. 支持成长规则插件自动加载、更新
3. 实现基本的成长规则插件逻辑

**第三次迭代：**

1. 支持业务规则管理界面，业务规则分组管理，插件配置逻辑等
2. 支持成长体系后台用户、权限管理功能
3. 支持成长体系审核、发布流程
4. 支持性能监控、插件规则管理
5. 支持用户信息查询

**附：第一次迭代开发时间表**



# 任务列表

|  |  |  |  |
| --- | --- | --- | --- |
| **阶段** | **任务内容** | **负责人** | **预计完成时间** |
| 系统设计 | 设计文档编写 | 顺炽国 | 2015-9-6 |
| 设计文档评审 | 顺炽国 | 2015-9-6 |
| 数据库设计 | 顺炽国 | 2015-9-7 |
| 开发 | 程序框架搭建 | 顺炽国 | 2015-9-10 |
| 成长插件基本框架 | 顺炽国 | 2015-9-14 |
| 数据库模块 | 待定 | 2015-9-14 |
| 一次赠送积分业务插件 | 待定 | 2015-9-15 |
| 多次赠送积分业务插件 | 待定 | 2015-9-16 |
| 积分变更业务模块 | 待定 | 2015-9-16 |
| 积分查询业务模块 | 待定 | 2015-9-18 |
| 业务接口模块 | 待定 | 2015-9-18 |
| 测试 | 内部集成测试 | 所有人 | 2015-9-24 |
| 整合测试 | 所有人 | 2015-9-29 |

# 风险评估

# 性能指标