江苏移动12580商盟平台

事件服务中心

设计初稿

北京宽连十方数字技术有限公司

2013年5月2日

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ]草稿 [V]正在发布 [ ]正在修改 | 文件标识： | V0.9 |
| 当前版本： | V0.9 |
| 作者： | 成凡 |
| 完成日期： | 2013-05-02 |

修改历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V0.9 | 成凡 |  | 2013-05-02 | 文档创建 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 概述 4](#_Toc355283318)

[1.1 平台现状 4](#_Toc355283319)

[1.2 预期结构 4](#_Toc355283320)

[1.2.1 模型 4](#_Toc355283321)

[1.2.2 角色 5](#_Toc355283322)

[1.2.3 典型业务1：新用户中心里的“用户成长体系” 5](#_Toc355283323)

[2 功能说明 5](#_Toc355283324)

[2.1 事件的定义 5](#_Toc355283325)

[2.2 事件生成 6](#_Toc355283326)

[2.3 事件传输协议 6](#_Toc355283327)

[2.4 事件存储 6](#_Toc355283328)

[2.5 WebUI 6](#_Toc355283329)

[2.6 事件发送 6](#_Toc355283330)

[2.7 HA 6](#_Toc355283331)

[2.8 关于JMS 7](#_Toc355283332)

[3 程序结构 7](#_Toc355283333)

[3.1 处理流程 7](#_Toc355283334)

[3.2 事件接收处理流程 8](#_Toc355283335)

# 概述

## 平台现状

目前平台有若干由事件驱动的业务，例如验证抽奖、满额赠、积分计算等。

此类业务的业务实现涉及支付中心、验证中心、Partners、积分计算中心和对应的业务处理程序等。

* **验证抽奖**

当用户身份验证时和验证码验证时，验证中心向抽奖程序发送送消息

* + 事件： 用户身份验证、用户验码
  + 事件发起： 验证码中心
  + 事件接收： 抽奖业务处理程序
* **满额赠业务**

支付完成时，Partners向满额赠业务发送用户完成支付的消息

* 事件： 用户支付成功
* 事件发起： Partners
* 事件接收： 满额赠业务处理程序
* **用户、商户积分计算**

用户身份验证、用户验码等情况下验证中心向积分服务程序发送消息

* 事件： 用户身份验证、用户验码、……
* 事件发起： 验证码中心、支付中心、……
* 事件接收： 用户积分

此类业务新增、修改都需要修改能力层程序。而且还有若干能遇见到的此类业务需要修改，例如在建的商户中心中的“商户积分规则”，在建的用户中心里的“用户成长体系”。此类业务皆需要修改各能力层程序。

所以需要统一建设一个分发事件的程序——事件服务中心

## 预期结构

### 模型

基于此类事件驱动的业务，将各环节分为：事件产生、事件收集、事件分发三个阶段

* 事件产生

各程序在事件产生环节生成统一的事件，并将消息发往事件中心。

例如：

用户身份验证时，就生成身份验证的事件。

验码时，生成验码事件。

支付完成时，生成支付完成事件。

* 事件收集

事件中心统一接收来自各程序的事件，并存储上。

* 事件分发

事件中心将事件主动分发给注册的各业务程序（事件消费者）



### 角色

在这个模型中涉及三个角色：事件产生者，事件服务，事件消费者，各角色的职责是：

* 事件生产者：产生事件并向事件服务提交事件
* 事件服务：接收、存储、分发事件
* 事件消费者：向事件服务注册事件监听并接收事件

### 典型业务1：新用户中心里的“用户成长体系”

需求1：用户各种行为可以获得不同的积分

事件生产者：支付中心、验证中心、……

产生事件：用户开始支付、用户完成支付、用户身份验证、用户验码、……

事件消费者：监听事件，并依据配置的积分规则给用户增加积分

# 功能说明

## 事件的定义

事件只包含如下信息：

* 事件的类型：用于区分消息类型
* 事件消息体：包含消息具体内容。所含信息由事件生产者决定。

例如，用户身份验证的事件：{类型：用户身份验证，消息体：{用户ID，商户ID}}

## 事件生成

在产生事件的环节，消息生产者创建事件对象，并通过事件中心提供的客户端将事件对象发往事件中心。

## 事件传输协议

通讯协议使用Http

事件消息体使用JSON协议封装

请求响应通过事务ID做唯一标识，用于过滤重复请求。

## 事件存储

事件使用com.cplatform.queue.SerializableObjectQueue的文件队列方式存储在本地磁盘中。

针对每个注册的消费者建立独立的文件队列，互不干扰。

## WebUI

事件中心提供简单的WebUI供维护和监控使用。提供功能包括：

* 注册事件类型
* 注册事件消费者
* 当前性能统计
* 当前各队列状态

## 事件发送

事件发送采用针对每个消费者建一个发送线程方式，互不干扰。或自动适配消费者状态，分不同的线程池发送。

## HA

对于高可用性的要求是：针对服务需要保证服务不中断，针对数据需要做到不丢失

外层通过负载均衡将来自事件生产者的事件随机发往各事件中心实例。可用的负载均衡有（HAProxy、Nginx等）

事件存储通过磁盘Raid来保证数据完整。

当某个事件中心实例失效后，则通过负载均衡可以保证对外服务继续有效。

如事件中心因硬件原因无法启动时，则可手工将磁盘上的数据复制到其他事件中心继续处理。



## 关于JMS

现阶段，尚没有合适的JMS系统满足使用简单、性能高、高可用性等条件。

RabbitMQ，安装使用不便。

IBM MQ性能不足，使用不便。

ActiveMQ高可用性方案不可靠。

# 程序结构

## 处理流程



1. 通过消费者注册接口注册“将哪些事件发往哪个地址”，数据存放在数据库中
2. 事件接收服务将收到的事件发给时间分派模块
3. 事件分派模块从数据库中获取消息的消费者配置
4. 事件分派模块将事件存储在对应消费者队列中
5. 事件发送线程从对应队列中获取需要发送的事件
6. 事件发送线程将事件发给消费者

## 事件接收处理流程

通过事务ID判断是否是重复发送的事件，用于防止响应接收异常产生的重发。

