|  |
| --- |
| 上海新致软件有限公司 |
| 微信公众号概要设计 |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 版本 | 作者 | 修订历史 |
| V0.3 | 甯尤刚 |  |

|  |
| --- |
|  |

# 引言

## 编写目的

此概要设计说明书是为了说明整个系统的体系架构，以及需求用例的各个功能点在架构中的体现，为系统的详细设计人员进行详细设计时的输入参考文档。

## 背景

本系统微信公众账号是公司一个基础组件。通过该系统，用户可以通过微信客户端与系统通信。

## 定义

## 参考资料

# 总体设计

## 简述

## 架构设计

### 系统逻辑架构图

### 系统物理架构图

#### 总物理架构图

#### XXX服务器

### 逻辑到物理映射

### 系统边界

### 功能架构

### 工程架构

# 通用模块设计

## 消息解析、消息拦截、分发模块

### 功能描述

* 消息解析：对微信后台服务器通过http协议发送过来的消息格式进行解析，如当接收消息时，将xml转化成javaBean，响应消息时，将javaBean转化为xml。
* 消息拦截：对消息进行拦截，如用户没有进行账号映射则给予提示、如对拦截到的消息以日志的形式进行保存。其中拦截规则可以配置。
* 消息分发：对文本消息、事件消息进行分发处理。如对文本消息，当用户回复 “1”，后台分发给“1”对应的程序处理，回复“2”，后台分发给“2”对应的程序处理。对事件消息，根据不同的事件类型交给不同的程序处理。

### 用例



|  |  |
| --- | --- |
| 用例名称： | 消息处理 |
| 用例编号： | message-001 |
| 参与者： | 微信客户端 |
| 用例说明： | 描述微信客户端发送文本、事件等消息的处理的过程 |
| 前置条件： | N/A |
| 基本事件流： | 1. 微信用户通过http发送xml文本消息(或其他消息格式) 2. 系统进行文本消息解析(或其他消息格式解析)到javaBean 3. 对拦截到的消息进行拦截处理,如是否做账号绑定，关键字过滤。 4. 系统根据文本消息内容分发到相应的文本处理器(或事件处理器)； 5. 系统将文本处理器(或事件处理器)得到的结果转化为xml，向用户输出 |
| 其他事件流： | N/A |
| 异常事件流： | N/A |
| 后置条件： | N/A |

### 流程图

### 类图设计

#### 消息关系类图设计



#### 消息处理类图设计



### 类/接口设计

#### 微信Servlet类

|  |  |
| --- | --- |
| 类功能描述 | 处理“从微信后台服务器发送到公众账号号http请求”的微信servet |
| 类全路径 | com.newtouch.weixin.WeixinServlet |
| 初始化 | 方法名:public void init(ServletConfig servletConfig) throws ServletException |
| 方法参数: |
| 处理流程:实例化私有属性weixinUtil |
| 处理get请求 | 方法名: public void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException |
| 输入参数: |
| 处理流程:将请求交给私有属性weixinUtil |
| 处理post请求 | 方法名: public void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException |
| 输入参数: |
| 处理流程:将请求交给私有属性weixinUtil |
| 销毁servlet | 方法名: public void destroy() |
| 输入参数: |
| 处理流程:销毁weixinUtil对象 |

#### 微信工具类

|  |  |
| --- | --- |
| 类功能描述 | 微信http请求处理的工具类 |
| 类全路径 | com.newtouch.weixin.WeixinUtil |
| 处理http请求(如文本、事件、图片等xml消息) | 方法名: public void dealMessage(HttpServletRequest request, HttpServletResponse response) throws IOException |
| 方法参数: |
| 返回值: |
| 处理流程:  先将私有属性request、response初始化   1. 请求如果来自微信后台服务器   先进行“验证消息真实性”，校验成功后，根据不同的消息类型调用相应的“微信消息解析和处理器接口”进行消息解析和消息分发处理，最后调用微信结果工具类将结果bean转化为xml消息，返回给客户端  二、请求如果来自非微信后台服务器  跳转到首页(暂时这样处理) |
| 验证消息真实性 | 方法名: private boolean checkSignature(String signature, String timestamp, String nonce) |
| 方法参数: signature:微信加密签名，结合了开发者填写的token参数和请求中的timestamp参数、nonce参数  timestamp:时间戳  nonce:随机参数 |
| 返回值:boolean，通过校验为true，否则为false |
| 处理流程: N/A |
| 注意事项: 防止其他非法程序恶意请求,所以需要校验，可以参考官方提供的php代码，转化为相应的java代码 |
| 将request中的二进制流解析到属性message | 方法名: private Map parseInputStream (HttpServletRequest request) throws Exception |
| 方法参数: request |
| 返回值:Map |
| 处理流程:  一、根据不同的消息类型，构造不同的IMessageHandler，如果为文本，则为MessageTextHandler，如果为事件，则为MessageEventHandler  二、调用相应的“微信消息解析和处理器接口”将request中的二进制流根据不同的消息类型(如文本事件)进行解析到baseMessage  三、最后将一、二的结果放入Map |

#### 微信消息解析和处理器接口

|  |  |
| --- | --- |
| 接口功能描述 | 微信消息解析和处理的接口 |
| 接口全路径 | com.newtouch.weixin.handler.MessageHandler |
| 解析消息，将element中的dom元素，解析到message | 方法名: public BaseMessage parseMessage(Element element) |
| 方法参数: element:从微信服务器传过来的element |
| 返回值: BaseMessage |
| 消息处理(文本消息对应文本处理器，事件消息对应事件处理器) | 方法名: public BaseResultMessage handleMessage(BaseMessage message); |
| 方法参数:message，解析后的原始消息 |
| 返回值:BaseResultMessage，处理后的消息结果Bean |

|  |  |
| --- | --- |
| 抽象类功能描述 | 微信消息解析和处理的抽象类 |
| 抽象类全路径 | com.newtouch.weixin.handler. AbstractMessageHandlerHelper |
| 解析特殊节点消息 | 方法名: public abstract void parseSpecialMessage(BaseMessage message, Element root); |
| 方法参数:  BaseMessage:将element解析到baseMessage  element:从微信服务器传过来的element |
| 返回值:void |
| 处理特殊类型的消息 | 方法名: public abstract BaseResultMessage handleSpecialMessage(BaseMessage message); |
| 方法参数: BaseMessage:baseMessage |
| 返回值: BaseResultMessage,消息返回bean |
| 解析消息 | 方法名: public BaseMessage parseMessage(Element root) |
| 方法参数: Element:element |
| 返回值: BaseMessage,消息到的message |
| 处理流程:   1. 先解析各种消息的公共部分   二、在调用子类的解析规则，对消息的特殊节点进行处理。 |
| 处理消息 | 方法名: public BaseResultMessage handleMessage(BaseMessage message) |
| 方法参数: BaseMessage:待处理的消息 |
| 返回值: BaseResultMessage,处理后的消息 |
| 处理流程:   1. 【拦截处理】先调用拦截器链进行拦截处理，校验通过，进入后续流程。校验不通过，则直接返回，不进行后续处理。(拦截规则与拦截的业务代码必须分离)   二、消息处理：调用子类的处理规则进行处理。 |

|  |  |
| --- | --- |
| 类功能描述 | 拦截器链处理 |
| 类路径 | com.newtouch.weixin.handler.InterceptChain |
| 构造方法 | 方法名: private InterceptChain() |
| 处理流程:读取拦截器配置文件 |
| 获取单例 | 方法名: public static InterceptChain getInstance() |
| 返回值: InterceptChain |
| 拦截链对拦截消息进行处理 | 方法名: public BaseResultMessage doChain(BaseMessage baseMessage) |
| 参数值: BaseMessage：被拦截的消息 |
| 返回值: BaseResultMessage:拦截之后处理的消息 |

|  |  |
| --- | --- |
| 接口功能描述 | 拦截器接口 |
| 接口路径 | com.newtouch.weixin.handler. MesssageInterceptor |
| 单个的拦截器进行的消息拦截处理 | 方法名: public BaseResultMessage intercept(BaseMessage baseMessage) |
| 参数值: BaseMessage：被拦截的消息 |
| 返回值: BaseResultMessage:拦截之后处理的消息 |

|  |  |
| --- | --- |
| 类功能描述 | 文本消息解析和处理器 |
| 类全路径 | com.newtouch.weixin.handler.MessageTextHandler |
| 构造器 | 方法名: private MessageTextHandler() |
| 处理流程:读取文本消息分发的xml配置文件 |
| 获取单例 | 方法名: public static MessageTextHandler getInstance() |
| 返回值: MessageTextHandler |
| 处理流程:利用单例设计模式，返回MessageTextHandler |
| 注意事项:其他方式，如懒汉式，过于复杂，建议饥饿式 |
| 解析文本消息类型的特殊节点(Content节点)消息到message | 方法名:public void parseSpecialMessage(BaseMessage baseMessage,Element root) |
| 方法参数:  element:从微信服务器传过来的element  BaseMessage:将element解析到BaseMessage |
| 处理流程:将dom中的Content节点值解析到message |
| 对文本消息类型的消息进行处理 | 方法:  public BaseResultMessage handleSpecialMessage(BaseMessage message) |
| 方法参数: message，原始消息bean |
| 返回值:Message，经过处理后的消息结果bean |
| 处理流程:对文本消息类型的消息进行处理.  参考方法:  当回复”a”时，由”a”对应的程序进行处理  当回复”a”时，由”b”对应的程序进行处理…..  这种回复文本与程序的关系，可以写在配置文件XXX.xml中，这样解析xml，得到对应回复文本与程序的关系，利用反射机制构造“程序实例”，进行处理。如xml参考格式如下:  <xml>  <text>  <processor>  <replyContent>a</replyContent>  <className>com.newtouch.xxx.xxxx</className>  </processor>  <processor>  <replyContent>b</replyContent>  <className>com.newtouch.xxx.xxxx</className>  </processor>  </text>  </xml> |

|  |  |
| --- | --- |
| 类功能描述 | 事件消息解析和处理器 |
| 类全路径 | com.newtouch.weixin.handler.MessageEventHandler |
| 构造方法 | 方法名: private MessageEventHandler() |
| 处理流程:读取事件消息分发的xml配置文件 |
| 获取单例 | 方法名: public static MessageEventHandler getInstance() |
| 返回值: MessageEventHandler |
| 处理流程:利用单例设计模式,返回MessageEventHandler |
| 解析事件消息类型的特殊节点(Event、EventKey等节点)消息到message | 方法:public void parseSpecialMessage(BaseMessage,Element root) |
| 方法参数: element:从微信服务器传过来的element  BaseMessage:从elelment解析后得到的事件Message |
| 处理流程:将dom中的Event、EventKey节点值解析到message |
| 对事件消息类型的消息进行处理 | 方法名:public abstract BaseResultMessage handleSpecialMessage(BaseMessage message) |
| 方法参数: message，原始消息bean |
| 返回值:Message，经过处理后的消息结果bean |
| 处理流程:对事件消息类型的消息进行处理.  参考方法:  当为关注事件时，由关注事件对应的程序进行处理。  当为取消关注事件时，由取消关注事件对应的程序进行处理。  当为菜单点击事件时，有eventKey对应的程序进行处理。…….  这种事件类型与程序的关系，可以写在配置文件XXX.xml中，这样解析xml，得到不同事件了性与程序的关系，利用反射机制构造“程序实例”，进行处理。如xml参考格式如下:  <xml>  <event>  <processor>  <eventType>subscribe</eventType>  <className>com.newtouch.xxx.xxxx</className>  </processor>  <processor>  <eventType>unsubscribe</eventType>  <className>com.newtouch.xxx.xxxx</className>  </processor>  <processor>  <eventType>CLICK</eventType>  <subProcessor>  <eventKey>a</eventKey>  <subClassName>com.newtouch.xxx.xxxx</subClassName>  </subProcessor>  <subProcessor>  <eventKey>b</eventKey>  <subClassName>com.newtouch.xxx.xxxx</subClassName>  <subProcessor>  </processor>  </event>  </xml> |

## 微信菜单管理

### 功能描述

通过微信菜单功能界面，对微信菜单进行维护，包括查看、新增、删除、修改。

### 用例



### 类图设计



### 类/接口设计

|  |  |
| --- | --- |
| 接口类功能描述 | 微信菜单的服务接口类 |
| 接口类全路径 | com.newtouch.weixin.menu.MenuService |
| 读取微信菜单 | 方法名: public String getMenu(); |
| 方法参数: |
| 返回值:微信菜单JSON数据 |
| 处理流程:  直接从本地数据库的微信菜单表读取即可。 |
| 读取微信菜单到java对象 | 方法名: public Menu getMenu(); |
| 方法参数: |
| 返回值:微信菜单JSON数据对应的Menu对象 |
| 处理流程:   1. 先从本地数据库的微信菜单表读取JSON串   二、将读取的Json转化为Menu对象 |
| 创建微信菜单 | 方法名: public boolean createMenu(String json); |
| 方法参数: menuJson:从前端返回的菜单JSON字符串 |
| 返回值:true:成功、false:失败 |
| 处理流程:   1. 先调用获取token的http接口   二、利用token，调用腾讯创建菜单的http接口，保存到微信服务器上。  三、保存到本地的数据库 |
| 备注:之所以需要保存到本地，是因为jackjson在反序列化多态数据类型时，必须指定”@class”属性，以方便jackjson反序列化时能够通过@class属性构造子类对象。 |
| 删除微信菜单 | 方法名: public boolean deleteMenu(); |
| 方法参数: |
| 返回值:true:成功，false失败 |
| 处理流程:   1. 先调用获取token的http接口   二、利用token，调用删除菜单的http接口。  三、删除本地服务器的微信菜单数据。 |

## 微信日志保存

### 功能描述

* 公众账号访问微信菜单功和回复文本信息时，将操作流水记录进行保存。

### 用例



### 类图设计



### 类/接口设计

|  |  |
| --- | --- |
| 实现类功能描述 | 微信菜单日志拦截类 |
| 实现类全路径 | com.newtouch.weixin.menu.OperLogRecordInterceptor |
| 拦截微信的日志 | 方法名: public BaseResultMessage intercept(BaseMessage baseMessage) |
| 方法参数: BaseMessage:需要处理的原始消息 |
| 返回值:保存菜单后直接返回null |
| 处理流程:  一、读取本地数据库的微信菜单菜单表，将“根据菜单key或url匹配菜单名称”匹配规则加入缓存。以后匹配规则获取走缓存。  二、根据匹配规则，获取菜单名称，构造微信菜单日志OperLog对象。  三、调用“微信菜单日志记录服务类”将微信菜单日志进行保存入库。 |
| 接口类功能描述 | 微信菜单日志记录服务类 |
| 接口类全路径 | com.newtouch.weixin.menu.OperLogRecordService |
| 保存微信菜单访问日志 | 方法名: public void save(OperLog operLog) |
| 方法参数: OperLog,微信日志bean |
| 返回值: |
| 处理流程(建议选A,通杀)  方案A(并发量大对数据库的压力较大)   1. 将OperLog先放入本机web应用服务器的缓存(java.util.ArrayList<OperLog>)。   以下两个步骤异步处理:  二、当缓存中的数目达到N条进行批量入库（N条可配置）。  三、即使没有达到N条，达到超时时间也批量入库（超时时间可配置）。  方案B(并发量小对数据库的压力很小)  一、 直接异步将operLog保存入库。 |

# 业务模块设计

## 用户管理模块设计

### 功能描述

* 提供用户查询功能，可以查看用户的关注情况。
* 管理员手工触发，对已离职员工的解绑操作。
* 后台一个自动调度，对离职的员工进行解绑。

### 用例

|  |  |
| --- | --- |
| 用例名称： |  |
| 用例编号： | employee-001 |
| 参与者： | NewtouchOne 框架 |
| 用例说明： | 描述用户的展示和查询。 |
| 前置条件： | N/A |
| 基本事件流： | 1通过newtouchone框架发送对应的报文给后台action处理。  2后台action处理完之后将结果返回给前台。 |
| 其他事件流： | N/A |
| 异常事件流： | N/A |
| 后置条件： | N/A |

### 类图设计



### 类/接口设计

|  |  |
| --- | --- |
| 类功能描述 | 处理“用户展示查询功能” |
| 类全路径 | com.newtouch.wplatform.system.service.WxUserService |
| 查询功能 | 方法名:public page queryEmployee(String employeeid,String activeFlag,boolean isWxUser,int pageIndex,int pageSize) throws Exception |
| 方法参数:  employeeid:员工编号、aciveFlag:员工状态(离职、在职)、isWxUser:是否微信用户 |
| 处理流程: |
| 手工解除关联(单个或多个) | 方法名称：public boolean deleteWxUser(String[] employeeids) throws Exception |
| 方法参数：employeeid数组 |
| 处理流程:管理员手动解除关联 |
| 自动解除关联(凌晨调度) | 方法名称：public void deleteWxUser() throws Exception |
| 方法参数：N/A |
| 处理流程:晚上凌晨2:00，通过quartz调度，可以直接同spring结合起来使用。 |

## 新致精彩模块设计

### 功能描述

* 提供后台管理界面，支持协会活动的增加，删除，修改和查询。在手机端可以查看录入的协会活动
* 提供后台管理功能，支持旅游条目的增加，删除，修改和查询。在手机端可以查看录入的旅游活动。

### 用例

|  |  |
| --- | --- |
| 用例名称： |  |
| 用例编号： | union-001 |
| 参与者： | NewtouchOne 框架 |
| 用例说明： | 描述协会活动、旅游活动的增删改查。 |
| 前置条件： | 对手机通过公众平台访问，需要关注 |
| 基本事件流： | 1.通过newtouchone框架发送对应的报文给后台action处理。  2.后台action处理完之后将结果返回给前台。 |
| 其他事件流： | N/A |
| 异常事件流： | N/A |
| 后置条件： | N/A |

### 类图设计



### 类/接口设计

|  |  |
| --- | --- |
| 接口类功能描述 | 处理“协会活动的增，删，改，查功能” |
| 接口类全路径 | com.newtouch.wplatform.wonderful.service.UnionService |
| 协会活动增加功能 | 方法名:public void addUnion (Union union) throws Exception |
| 方法参数: Union |
| 处理流程:新增工会活动 |
| 协会活动删除功能 | 方法名:public void deleleUnion l(Long[]ids) throws Exception |
| 方法参数: ids,工会活动id数组 |
| 处理流程:删除 |
| 协会活动修改功能 | 方法名:public void updateUnion (Union union) throws Exception |
| 方法参数: Union |
| 处理流程:修改 |
| 协会活动查询功能 | 方法名:public Page queryUnion(String unionName,int pageIndex,int pageSize) throws Exception |
| 方法参数: unionName:工会活动名称 |
| 处理流程:查询 |

|  |  |
| --- | --- |
| 接口类功能描述 | 处理“旅游项目增，删，改，查功能” |
| 接口类全路径 | com.newtouch.wplatform.wonderful.service.TravelService |
| 旅游项目增加功能 | 方法名:public void addTravel (Travel travel) throws Exception |
| 方法参数: Travel |
| 处理流程:新增 |
| 旅游项目删除功能 | 方法名:public void deleteTravel (Long[] ids) throws Exception |
| 方法参数: ids:旅游项目id数组 |
| 处理流程:删除 |
| 旅游项目修改功能 | 方法名:public void updateTravel (Travel travel) throws Exception |
| 方法参数: TravelActivity |
| 处理流程:修改 |
| 旅游项目查询功能 | 方法名:public Page queryTravel (String travelName,int pageIndex,int pageSize) throws Exception |
| 方法参数: travelName:旅游名称 |
| 处理流程:查询 |

## 微信日志分析

### 功能描述

* 系统管理员对微信的日志进行统计分析。做客户偏好、菜单点击统计等分析。

### 用例



### 类/接口设计

|  |  |
| --- | --- |
| 接口类功能描述 | 微信菜单日志分析服务接口 |
| 接口类全路径 | com.newtouch.weixin.menu.OperLogAnalysisService |
| 菜单分析-趋势 | 方法名: public List<TrendBean> getTrend(String employeeid,String menuName,String beginDate,String endDate,int cycleType); |
| 方法参数:  employeeid:员工编号、menuName:菜单名称、beginDate:开始日期、endDate:结束日期、cycleType:分析周期 |
| 返回值:前端flash展示需要的xml |
| 处理路程(建议方案A,比较java的正宗策略)  A方案:   1. 取出趋势分析展示的数据，如普通的java.util.List。   二、将list转化为json  三、由前台对将json转化为xml(需要有js的封装)  B方案:  一、构造趋势分析的sql, 将数据放入String[][].  二、调用”fusionchat通用模块”，将二维数组转化为xml. |
| 备注:实现时，需要考虑由于日志流水的统计力度到小时，数据很细，如果按照日、月的统计，势必在sql中要做group by、sum，这样sql统计效率会随着数据量的增加而出现瓶颈，建议:后台跑个程序，对数据做日、月的汇总，对日的统计放到日表，对月的统计放到月表，这样sql的查询效率将有很大的提升。 |
| 菜单分析-排名 | 方法名: public String getOrder(String employeeid,String menuName,String beginDate,int orderType,int num); |
| 方法参数:  employeeid:员工编号、menuName:菜单名称、beginDate:开始日期、endDate:结束日期、orderType:排名类型(1：前N名，2：后N名)，num:取多少名 |
| 返回值:前端flash展示需要的xml |
| 处理路程:  雷同趋势分析的思路 |
| 菜单分析-对比 | 方法名: public String getCompare(String employeeid,String menuName, String compareMenuName,String beginDate,int cycleType); |
| 方法参数:  employeeid:员工编号、menuName:菜单名称、compareMenuName:对比菜单名称、beginDate:开始日期、endDate:结束日期、cycleType:分析周期 |
| 返回值:前端flash展示需要的xml |
| 处理路程:  与趋势雷同 |
| 菜单分析-明细 | 方法名: public Page<OperLog> getDetail(String employeeid,String menuName,String beginDate,String endDate,int pageIndex,int pageSize); |
| 方法参数:  employeeid:员工编号、menuName:菜单名称、beginDate:开始日期、endDate:结束日期 |
| 返回值: Page <OperLog> |
| 明细导出 | 方法名: public void exportOperLog(HttpServletResponse response, String employeeid,String menuName,String beginDate,String endDate) |
| 方法参数:  Response:http的response,employeeid:员工编号、menuName:菜单名称、beginDate:开始日期、endDate:结束日期 |
| 返回值:void |
| 处理流程:   1. 构造导出全量查询的sql   二、调用“大数据量通用导出”模块，传入sql，文件名、导出列等参数，将数据全量导出 |

# 数据库设计

## [eds库]用户基本表tabEmployee

略，通过链接服务器访问

## [eds库]部门表tabDepartment

略

## [eds库]用户休假表tabxiujia

略，在休假应用中，还用到了EDS库的存储过程。

## [eds库]休假明细表tabxiujiaSec

略

## [eds库]休假设置表xiujiaSet

略

## [eds库]用户考勤表tab\_pcd\_checkdetail

略

## [微信库] 微信工会活动表wx\_union

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **编码** | **注释** | **默认值** | **数据类型** | **是否为空** |
| Id | id | 主键id |  | BIGINT | 否 |
| 所属部门 | dept\_id | 从部门表中读取tabdepartment中读取 |  | VARCHAR(64) | 否 |
| 工会活动名称 | union\_name |  |  | VARCHAR(64) | 否 |
| 组织者 | organizer\_id | 从用户基本表tabEmployee中读取 |  | VARCHAR(64) | 否 |
| 活动内容 | content |  |  | VARCHAR(64) | 否 |
| 协会描述 | description |  |  | VARCHAR(64) | 是 |

## [微信库]旅游活动表 wx\_travel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **编码** | **注释** | **默认值** | **数据类型** | **是否为空** |
| Id | id | 主键id |  | BIGINT | 否 |
| 旅游活动名称 | travel\_name | 旅游活动名称 |  | VARCHAR(64) | 否 |
| 旅游地点 | travel\_address | 旅游地点 |  | VARCHAR(128) | 否 |
| 出发时间 | start\_date | 出发时间 |  | Date | 否 |
| 目标人数 | target\_person\_count | 目标人数 |  | Integer | 否 |
| 组织者 | organizer\_name | 组织者姓名 |  | VARCHAR(16) | 否 |
| 组织者电话 | organizer\_phone | 组织者电话 |  | VARCHAR(16) | 否 |
| 人均消费 | ave\_fee | 人均消费 |  | int | 否 |
| 行程安排 | travel\_arrange | 行程安排 |  | VARCHAR(128) | 否 |

## [微信库]微信用户表wx\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **编码** | **注释** | **默认值** | **数据类型** | **是否为空** |
| 用户映射id | id | 主键id |  | BIGINT | 否 |
| 员工id | employeeid | 员工id |  | VARCHAR(16) | 否 |
| 微信用户在公众平台的用户id | openid | 微信用户在公众平台的用户id |  | VARCHAR(64) | 否 |

## [微信库]微信日志表wx\_menu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **编码** | **注释** | **默认值** | **数据类型** | **是否为空** |
| 主键id | id | 主键id |  | BIGINT | 否 |
| 微信菜单JSON串 | menu | 含@class属性的json菜单数据 |  | text | 否 |

还有中设计方式，将json串的各个字段展开，json字段与表字段一致。此处为了方便直接存JSON串。

## [微信库]微信操作日志表wx\_oper\_log

不需要微信菜单表，因为接口调用频率可以满足要求。

自定义菜单创建:1000。

自定义菜单查询:10000。

自定义菜单删除:10000。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **编码** | **注释** | **默认值** | **数据类型** | **是否为空** |
| 主键id | id | 主键id |  | BIGINT | 否 |
| 访问用户 | openid | 访问用户,提升入库以及查询效率 |  | VARCHAR(64) | 否 |
| 访问菜单名称 | menu\_name | 访问菜单名称(针对click、view) |  | VARCHAR(16) | 是 |
| 回复文本 | reply\_text | 回复文本 |  | VARCHAR(512) | 是 |
| 操作时间 | operate\_time | 操作时间 |  | TIMESTAMP | 否 |

暂不分表