|  |  |  |
| --- | --- | --- |
| 文档名称 | 文档编码 | 编写时间 |
| SMS模块详细设计 | T-sms2 | 2017-03-01 |
|  |  |  |

# 模块功能介绍

是其他应用系统通过IM系统与客户端信息交互的通道。

# 模块协议

基于http协议，UTF-8编码格式的数据交互，包体采用xml或者json格式

# 模块参数说明

## IM->WCF(IM向WCF发送的消息)

### 发短信到wcf

**请求：**

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String |  | 应用编号 |
| MsgType | String | IM.SMS | 发送消息类型 |
| Sender | String | 发送者帐号 | 发送短信的用户帐号 |
| Targets | String | 手机号 | 多个帐号 |
| MSTitle | String | 标题 | 信息标题，可为空 |
| Flag | String | 0无需回执，1需要回执 |  |
| MsgId | String | 发送系统编号年月日时分秒4位序列号  示例：  系统编号1001  则某时消息ID  1001201408152020200001 | 唯一标识该消息 |
| MSContent | String | 最长140字节 | 消息内容 |
| TargetTime | YYYY-MM-DD HH:MM:SS |  |  |
| Priority |  |  | 优先级 |

**Json示例**：

**单条数据包：**

{"FromSys":"IM","MsgType":" IM.SMS "," Sender":"1001",SendName”:”testuser”,

“Targets":"user1","MSTitle":”模块化系统测试”,

"Flag":”0”,”MsgId”:”1001201408162020200001”,

"MSContent":"系统今天7点钟接受测试数据成功，谢谢！",

"TargetTime":"2012-08-07 07:20:30","Piority":"1"}

**批量发送：**

[{"FromSys":"IM","MsgType":"IM.SMS"," Sender":"1001",”SendName”:”testuser”,

"Targets":[“13266666666”,”13355555555”],"MSTitle":”模块化系统测试”,

“Flag”:”0”,”MsgId”:”1001201408162020200001”,

"MSContent":"系统今天7点钟接受测试数据成功，谢谢！","TargetTime":"2012-08-07 07:20:30","Piority":"1"}]

或：

{"FromSys":"IM","MsgType":"IM.SMS",

" Sender":"1001",”SendName”:”testuser”,

"Targets":”13222222222”,"MSTitle":”模块化系统测试”,

”Flag”:”0”,”MsgId”:”1001201408162020200001”,

"MSContent":"系统今天7点钟接受测试数据成功，谢谢！",

"TargetTime":"2012-08-07 07:20:30","Piority":"1"}，

{"FromSys":"IM","MsgType":"IM.SMS",

" Sender":"1001",”SendName”:”testuser”,

"Targets":"13222222222”, "MSTitle":”模块化系统测试”,

”Flag”:”0”,”MsgId”:”1001201408162020200001”,

"MSContent":"系统今天7点钟接受测试数据成功，谢谢！",

"TargetTime":"2012-08-07 07:20:30","Piority":"1"}，

{"FromSys":"IM","MsgType":"IM.SMS",

" Sender":"1001",”SendName”:”testuser”,

"Targets":“13222222222”, "MSTitle":”模块化系统测试”,

”Flag”:”0”,”MsgId”:”1001201408162020200001”,

"MSContent":"系统今天7点钟接受测试数据成功，谢谢！",

"TargetTime":"2012-08-07 07:20:30","Piority":"1"}

]

应答：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 系统标识 |
| Result | String | Ok/false | 该请求处理结果 |

示例：

{"FromSys":"IM","Result":"ok”}]

{"FromSys":"IM","Result":"false: 尚未设置可访问系统的IP”}

返回值：

信息接收成功：OK

信息接收失败：

格式 [false]:[错误信息]

1)false:IP地址：\*\*\*已被统一信息平台系统排除，请与管理员联系！

注：\*\*\*代表访问的IP地址

2) false:系统已经停止使用！

--传入是参数fromSys(来源系统)在统一信息平台中已经标志为停用状态

3) false:尚未设置可访问系统的IP! --传入是参数fromSys(来源系统)，在统一信息平台系统中尚未为来源系统（fromSys）配置可访问的IP地址

4) false:IP地址不在系统许可访问范围内！--统一信息平台系统已经为来源系统（fromSys）配置了可访问的IP地址，但来源系统的访问IP，不在已经配置的范围内

5) false:数据库保存时出错！

6) false:来源系统不能为空！--传入是参数fromSys为空

7) false:无法确认来源系统！--在系统信息表中找不到fromSys相应的代码，即系统尚未注册到统一信息管理平台中

8) false:目标接受者不能为空！--传入参数target不能为空

9) false:无法确认目标接受者！--传入参数target在人员信息表中找不到相应信息

10) false:发送内容不能为空！--传入参数msContent不能为空

11) false:im内容长度不能超过780个字节！

--传入参数msContent内容不能超过780个字节

12) false:IM 服务配置尚未完成，请联系RTX信息管理人员！--传入是参数fromSys相应的系统还没有配置IM服务信息

### 消息回执

表示所发消息成功或失败，

请求：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 系统标识 |
| msgtype | String | IM.SMS.Report | 发送消息类型 |
| msgid | String | 原始消息的msgid |  |
| Target | String | 目标用户帐号 |  |
| Result | String | 状态 | 成功： ok  失败：  false:发送失败  false:无此帐号 |
| Piority | String | 优先级 |  |

示例：

Json示例：

{"FromSys ":"IM", "MsgType":" IM.SMS.Report ",

”MsgId”:”1001201408162020200001”,“

"Target":"user1",

“Result”:” false:发送失败”,

"Piority":"1"}

应答：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 系统标识 |
| Result | String | Ok/false | 该请求处理结果 |

示例：

{"FromSys":"IM","Result":"ok”}

{"FromSys":" IM","Result":"false: 尚未设置可访问系统的IP”}

返回值：

信息接收成功：OK

信息接收失败：

格式 [false]:[错误信息]

1)false:IP地址：\*\*\*已被统一信息平台系统排除，请与管理员联系！

注：\*\*\*代表访问的IP地址

2) false:系统已经停止使用！

--传入是参数fromSys(来源系统)在统一信息平台中已经标志为停用状态

3) false:尚未设置可访问系统的IP! --传入是参数fromSys(来源系统)，在统一信息平台系统中尚未为来源系统（fromSys）配置可访问的IP地址

4) false:IP地址不在系统许可访问范围内！--统一信息平台系统已经为来源系统（fromSys）配置了可访问的IP地址，但来源系统的访问IP，不在已经配置的范围内

5) false:数据库保存时出错！

6) false:来源系统不能为空！--传入是参数fromSys为空

7) false:无法确认来源系统！--在系统信息表中找不到fromSys相应的代码，即系统尚未注册到统一信息管理平台中

8) false:目标接受者不能为空！--传入参数target不能为空

9) false:无法确认目标接受者！--传入参数target在人员信息表中找不到相应信息

10) false:发送内容不能为空！--传入参数msContent不能为空

11) false:im内容长度不能超过780字节

--传入参数msContent内容不能超过780字节

12) false:IM 服务配置尚未完成，请联系RTX信息管理人员！--传入是参数fromSys相应的系统还没有配置IM服务信息

## WCF->IM (wcf平台向IM发送消息)

### 发消息给IM（包括上行短信）

请求:

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 应用编号 |
| MsgType | String | IM.MSG | 发送消息类型 |
| Sender | String | 发送者帐号 | 发送短信的用户帐号 |
| Target | String | 接受者用户帐号 | 接受者用户 |
| Flag | String | 1需要；0 不需要 | 是否需要回执 |
| MsgId | String | 发送系统编号年月日时分秒4位序列号  示例：  系统编号1001  则某时消息ID  1001201408152020200001 | 唯一标识该消息 |
| MSContent | String | 最长2000字节 | 消息内容 |
| TargetTime | YYYY-MM-DD HH:MM:SS |  |  |
| Priority |  |  | 优先级 |
|  | String | 最长140字节 | 消息内容 |

Json示例：

**单条数据包：**

{"FromSys ":"WCF", "MsgType":" IM.MSG ",

" Sender":"1001",SendName”:”testuser”,

"Target":"user1","Flag":”0”,

”MsgId”:”1001201408162020200001”,“

"MSContent":"系统今天7点钟接受测试数据成功，谢谢！",

"TargetTime":"2012-08-07 07:20:30",

"Piority":"1"}

应答：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 系统标识 |
| Result | String | Ok/false | 该请求处理结果 |

示例：

{"FromSys":"WCF","Result":"ok”}

{"FromSys":" WCF","Result":"false: 尚未设置可访问系统的IP”}

返回值：

信息接收成功：OK

信息接收失败：

格式 [false]:[错误信息]

1)false:IP地址：\*\*\*已被IM系统排除，请与管理员联系！

注：\*\*\*代表访问的IP地址

2) false:系统已经停止使用！

--传入是参数fromSys(来源系统)在统一信息平台中已经标志为停用状态

3) false:尚未设置可访问系统的IP! --传入是参数fromSys(来源系统)，在统一信息平台系统中尚未为来源系统（fromSys）配置可访问的IP地址

4) false:IP地址不在系统许可访问范围内！IM平台系统已经为来源系统（fromSys）配置了可访问的IP地址，但来源系统的访问IP，不在已经配置的范围内

5) false:数据库保存时出错！

6) false:来源系统不能为空！--传入是参数fromSys为空

7) false:无法确认来源系统！--在IM系统信息表中找不到fromSys相应的代码，即系统尚未注册到IM平台中

8) false:目标接受者不能为空！--传入参数target不能为空

9) false:无法确认目标接受者！--传入参数target在人员信息表中找不到相应信息

10) false:发送内容不能为空！--传入参数msContent不能为空

11) false:im内容长度不能超过512个字！

--传入参数msContent内容不能超过512个字

12) false:IM 服务配置尚未完成，请联系IM信息管理人员！--传入是参数fromSys相应的系统还没有配置IM服务信息

### 短信状态报告

请求：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 系统标识 |
| msgtype | String | IM.SMS.Report | 发送消息类型 |
| msgid | String | 发送短信时的msgid |  |
| Target | String | 目标用户帐号 |  |
| Result | String | 状态 | 成功： ok  失败：  false:发送失败  false:无此帐号 |
| Piority | String | 优先级 |  |

示例：

Json示例：

{"FromSys ":"WCF", "MsgType":" IM.SMS.Report ",

”MsgId”:”1001201408162020200001”,“

"Target":"user1",

“Result”:” false:发送失败”,

"Piority":"1"}

应答：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 系统标识 |
| Result | String | Ok/false | 该请求处理结果 |

示例：

{"FromSys":"WCF","Result":"ok”}

{"FromSys":" WCF","Result":"false: 尚未设置可访问系统的IP”}

返回值：

信息接收成功：OK

信息接收失败：

格式 [false]:[错误信息]

1)false:IP地址：\*\*\*已被IM系统排除，请与管理员联系！

注：\*\*\*代表访问的IP地址

2) false:系统已经停止使用！

--传入是参数fromSys(来源系统)在统一信息平台中已经标志为停用状态

3) false:尚未设置可访问系统的IP! --传入是参数fromSys(来源系统)，在统一信息平台系统中尚未为来源系统（fromSys）配置可访问的IP地址

4) false:IP地址不在系统许可访问范围内！IM平台系统已经为来源系统（fromSys）配置了可访问的IP地址，但来源系统的访问IP，不在已经配置的范围内

5) false:数据库保存时出错！

6) false:来源系统不能为空！--传入是参数fromSys为空

7) false:无法确认来源系统！--在IM系统信息表中找不到fromSys相应的代码，即系统尚未注册到IM平台中

8) false:目标接受者不能为空！--传入参数target不能为空

9) false:无法确认目标接受者！--传入参数target在人员信息表中找不到相应信息

10) false:发送内容不能为空！--传入参数msContent不能为空

11) false:im内容长度不能超过512个字！

--传入参数msContent内容不能超过512个字

12) false:IM 服务配置尚未完成，请联系IM信息管理人员！--传入是参数fromSys相应的系统还没有配置IM服务信息

### 消息通知

请求:

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 应用编号 |
| MsgType | String | IM.Alert | 发送消息类型 |
| Sender | String | 发送者帐号 | 发送短信的用户帐号 |
| Targets | String | 接受者用户帐号 | 接受者用户 |
| MSTitle | String | 标题 | 信息标题 |
| Flag | String | 1需要；0 不需要 | 是否需要回执 |
| MsgId | String | 发送系统编号年月日时分秒4位序列号  示例：  系统编号1001  则某时消息ID  1001201408152020200001 | 唯一标识该消息 |
| MSContent | String | 最长700字节 | 消息内容 |
| SendTime | YYYY-MM-DD HH:MM:SS |  |  |
| Priority |  |  | 优先级 |

Json示例：

**单条数据包：**

{"FromSys ":"WCF", "MsgType":" IM.Alert ",

" Sender":"1001",SendName”:”testuser”,

"Target":"user1","Flag":”0”,

”MsgId”:”1001201408162020200001”,

"MSContent":"系统今天7点钟接受测试数据成功，谢谢！",

"TargetTime":"2012-08-07 07:20:30",

"Piority":"1"}

应答：

|  |  |  |  |
| --- | --- | --- | --- |
| 元素名称 | 类型 | 约束 | 描述 |
| FromSys | String | 1 | 系统标识 |
| Result | String | Ok/false | 该请求处理结果 |

示例：

{"FromSys":"WCF","Result":"ok”}

{"FromSys":" WCF","Result":"false: 尚未设置可访问系统的IP”}

返回值：

信息接收成功：OK

信息接收失败：

格式 [false]:[错误信息]

1)false:IP地址：\*\*\*已被IM系统排除，请与管理员联系！

注：\*\*\*代表访问的IP地址

2) false:系统已经停止使用！

--传入是参数fromSys(来源系统)在统一信息平台中已经标志为停用状态

3) false:尚未设置可访问系统的IP! --传入是参数fromSys(来源系统)，在统一信息平台系统中尚未为来源系统（fromSys）配置可访问的IP地址

4) false:IP地址不在系统许可访问范围内！IM平台系统已经为来源系统（fromSys）配置了可访问的IP地址，但来源系统的访问IP，不在已经配置 的范围内

5) false:数据库保存时出错！

6) false:来源系统不能为空！--传入是参数fromSys为空

7) false:无法确认来源系统！--在IM系统信息表中找不到fromSys相应的代码，即系统尚未注册到IM平台中

8) false:目标接受者不能为空！--传入参数target不能为空

9) false:无法确认目标接受者！--传入参数target在人员信息表中找不到相应信息

10) false:发送内容不能为空！--传入参数msContent不能为空

11) false:im内容长度不能超过512个字！

--传入参数msContent内容不能超过512个字

12) false:IM 服务配置尚未完成，请联系IM信息管理人员！--传入是参数fromSys相应的系统还没有配置IM服务信息

## 配置文件说明

打开/QY/cfg/ 目录下的qy\_sms.conf

[MODULE]

NetID = 0x00160001 //本网元ID

LogPath = /QY/log/SMS\_16001 //日志输出路径

LogName = PlatServer

LogFlag = 1

LogLevel = 5

LogExpiredDay = 15

DumpPath = /QY/log/SMS\_16001/dump

#接入服务模块IP和端口

SERVICEIP = 10.112.67.3 //本服务的 ip

UdpServerPort = 8004 //本服务监听udp端口

#统一消息平台

WDMsgIP = 115.159.153.235

WDMsgPort = 9003

#统一身份平台

WDIdentityIP = 10.199.201.189

WDIdentityPort = 80

#监听消息端口

WDPort = 7007

#是否启用短信MD5重复消息检测 0：不检测 1：启用检测

WdSmsCheckOffOn = 1

#短信MD5值在内存中保存时长 单位：分钟

WdSmsCheckTime = 5

#ip地址鉴权

WdSmsLegalIP = 10.128.46.45$10.112.67.13

#数据库连接信息

DBHost = 10.0.104.43

DBPort = 9017

DBUser = sxit

DBPwd = sxitprog

DBName = wangxin\_test

DBConnectNum = 5

#日志调试级别

DebugMode = 4

DebugSync = 2

QueueMaxNumPrint= 1000

#轻应用通知标志，0为不处理轻通知

LightAppNotice = 1

[SYNC\_CLIENT]

NetID = 0x00160001

PacketLen = 2500

SocketNum = 8

LogPath = /QY/log/SMS\_16001

LogName = SyncPlatClient

ServerNetID = 0x00160002

ServerPort = 7007

ServerIPAddr = 10.112.67.4

ServerSocketNum = 2

# 模块编译说明

1.cd /QY/src/SMS //切换到源文件目录

2.make //makefile 编译源文件

3.cp QY\_SMS –f /QY/bin //拷贝执行文件到执行目录

4.cd /QY/bin //切换的执行目录

5./QY/bin/QY\_SMS –f /QY/cfg/qy\_sms.conf //启动服务

# 模块流程设计

手机用户

消息平台

短信服务

核心服务

客户端

接入服务

## Wcf短信平台--->IM客户端

Wcf短信平台

SMS短信服务

客户端

Service接入服务

Core核心服务

发

送消息请求 IP认证，消息验证

NO，拒绝连接 YES,消息发送给核心服务 消息转发给接入服务 消息下发给客户端 返回接收应答给核心服务 返回应答给短信服务

返回发送成功应答

## IM客户端-->wcf短信平台

SMS短信服务

Service接入服务

Wcf短信平台

客户端

Core核心服务

发

送短信消息请求 发送短信请求转发刚给核心服务 转发短信请求给sms服务 组织json和http报文

请求，发送短信请求

消息验证 发送成功或者失败应答 转发应答 转发应答给接入服务

回复应答给客户端

# 相关数据结构，数据流转设计

## 消息统一接口包

typedef struct \_WD\_MSG

{

UINT32 dwMsgType; //消息类型 1:IM下行消息 2:IM提醒消息 3:短信回复消息

//4:短信发送回执 5:万信短信发送 6:wcf消息应答 7:万信客户端

修改个人信息 8:统一身份平台过来的应答消息

UINT32 dwBatch; //批次号

UINT32 dwTotal; //批次总数

UINT32 dwTryTime; //尝试次数 尝试两次发送，失败则返回应答给客户端

UINT32 dwSocket; //请求套接字ID

UINT32 dwReqTime; //请求时间

UINT8 cStep; //消息流程 0：收到客户端请求 1：建立连接，等待发送请求 2：

已发请求，等待应答 3：已收应答等待短信回执 4：已收短信回执 5:等待超时应答 6:等待回执超时 7:未知错误

UINT64 dwMsgID; //IM消息ID

UINT8 aszWDMsgID[WD\_MSGID]; //万达平台消息ID（IM这边不用填，可为空）

UINT8 aszContent[WD\_PACKETMAXLEN]; //消息内容

}WDMSG;

## 对外接口请求包

typedef struct \_outside\_interface\_req

{

UINT32 nPacketLen; //数据包总长度

UINT32 nOperID; //命令字

char strSerial[MAX\_SERIAL\_LEN]; //序列号

UINT32 nSocket; //socket号(固定为0)

char strPacket[MAX\_INTERFACE\_PACKET\_LEN];

}OutsideInterfaceReq;

## 对外接口应答包

typedef struct \_outside\_interface\_ack

{

UINT32 nPacketLen; //数据包总长度

UINT32 nOperID; //命令字

char strSerial[MAX\_SERIAL\_LEN]; //序列号

UINT32 nReturn; //返回结果标识位 0成功

char strPacket[MAX\_INTERFACE\_PACKET\_LEN];

}OutsideInterfaceAck;

## 用户登录内存表

struct r\_userlogin

{

UINT32 lUserId; //用户ID

UINT32 dwCompID; //企业ID

INT8 cLoginType; //客户端类型 1:android 2:IOS 3:PC 4:other

UINT32 lLoginIp; //用户登录IP

UINT32 lNetId; //用户登录时连接的网元ID

UINT32 lLoginTime; //本次登录时间

UINT8 cStatus; //0:离线 1:上线 2: 离开 3: 退出

INT8 aszUserName[USERNAME\_LEN]; //用户姓名

INT8 IsBlend; //是否为融合客户端

};

## 网元信息

struct terminal

{

UINT32 lNetid; //消息接收方

UINT16 iTerminalType; //终端类型1 CORE 2SERVICE 3MANAGER

UINT16 iMajorFlag; //主备标识

INT8 strIP[IP\_STR\_LENGTH]; //IP地址

UINT32 lPort; //端口

UINT32 dwUdpPort; //udp监听端口

UINT8 cIsAlive; //活跃时间

UINT32 dwAliveTime; //活跃时间

};

## 客户端请求数据缓存

typedef struct terminal\_head

{

UINT32 nPacketLen;

UINT32 nOperID;

char strMsgid[MSG\_ID\_LENGTH];

INT16 nReturn;

UINT32 nSocket;

char strIP[IP\_STR\_LENGTH];

UINT16 nNetFlag; //内网，外网标识 0:外网 WEB 1:内网

char strPacket[MAX\_PACKET\_BODY\_LENGTH];

}terminal\_head;

## IOS 通知

typedef struct iosnoticemsg

{

INT16 wMsgLen;

INT16 wCmdID;

INT32 dwSeq;

char token[DEVICETOKEN\_LEN];

UINT16 nPushNum;

INT16 dwContentLen;

char aszContent[200];

}iosnoticemsg;

## 计数器

typedef struct statcounter

{

UINT32 nMsgCount; //消息总量

UINT32 nMsgNotice; //消息通知总量

UINT32 nInsertErrocount; //单人消息插入错误计数

UINT32 nInsertSucced; //单人消息插入成功计数

UINT32 nGroupErr; //群消息插入错误计数

UINT32 nGroupSucced; //群消息插入成功计数

} statcounter;

## IM下行消息（万达->万信客户端）

typedef struct \_IM\_down

{

UINT8 aszMsgID[WD\_MSGID]; //消息ID

UINT8 aszFromSys[WD\_SYSNUM]; //系统编号

UINT8 aszMsgType[WD\_MSGTYPE]; //消息类型

UINT32 dwSender; //发送者

UINT32 dwNumber; //接收者数量

UINT32 aTarget[WD\_MAXRECVNUM]; //接收者ID

UINT8 cFlag; //该消息是否需要回执 0:不需要 1:需要

UINT8 aszMSContent[WD\_MSGMAXLEN]; //消息内容 最大长度为2000字节

UINT8 aszSendTime[WD\_SENDTIME]; //发送时间 格式：YYYY-MM-DD HH:MM:SS

UINT8 cPriority; //优先级 1:高优先级 0:一般

}IM\_down;

## IM提醒通知（万达->万信客户端）

typedef struct \_IM\_noitce

{

UINT8 aszMsgID[WD\_MSGID]; //消息ID

UINT8 aszFromSys[WD\_SYSNUM]; //系统编号

UINT8 aszMsgType[WD\_MSGTYPE]; //消息类型

UINT32 dwSender; //发送者

UINT32 dwNumber; //接收者数量

UINT32 aTarget[WD\_MAXRECVNUM]; //接收者ID

UINT8 cFlag; //该消息是否需要回执 0:不需要 1:需要

UINT8 aszMsTitle[WD\_HEADLINELEN]; //标题

UINT8 aszMSContent[WD\_MSGNOTICE]; //消息内容

UINT8 aszSendTime[WD\_SENDTIME]; //发送时间

UINT8 cPriority; //优先级 1:高优先级 0:一般

}IM\_notice;

## 回复的短信接口（万达->万信客户端）

typedef struct \_SMS\_up

{

UINT8 aszMsgID[WD\_MSGID]; //消息ID

UINT8 aszFromSys[WD\_SYSNUM]; //系统编号

UINT8 aszMsgType[WD\_MSGTYPE]; //消息类型

UINT32 dwSender; //发送者

UINT64 dwSenderMobile; //发送者手机号码

UINT32 dwTarget; //接收者ID

UINT8 cFlag; //该消息是否需要回执 0:不需要 1:需要

UINT8 aszMSContent[WD\_MSGMAXLEN]; //消息内容

UINT8 aszSendTime[WD\_SENDTIME]; //发送时间

UINT8 cPriority; //优先级 1:高优先级 0:一般

}SMS\_up;

## 回复短信下发给客户端

typedef struct \_SMS\_down\_to\_client

{

UINT32 dwSender; //发送者

UINT64 dwSenderMobile; //发送者手机号码

UINT8 cFlag; //该消息是否需要回执 0:不需要 1:需要

UINT32 dwTime; //短信回复时间

UINT8 aszMSgContent[WD\_MSGMAXLEN]; //消息内容

}SMS\_client;

## 短信下行（万信客户端->万达）

typedef struct \_SMS\_down

{

UINT8 aszMsgID[WD\_MSGID]; //消息ID

UINT8 aszFromSys[WD\_SYSNUM]; //系统编号

UINT8 aszMsgType[WD\_MSGTYPE]; //消息类型

UINT32 dwSender; //发送者

UINT32 dwNumber; //接收者人数

UINT64 aTarget[WD\_MAXRECVNUM]; //接收者（手机号）

UINT8 aszMSTitle[WD\_HEADLINELEN]; //标题(可为空)

UINT8 aszMSContent[WD\_CLIENTMAXLEN]; //消息内容

UINT8 aszSendTime[WD\_SENDTIME]; //发送时间

UINT8 cPriority; //优先级 1:高优先级 0:一般

}SMS\_down;

## 回执消息（万达->万信客户端）

typedef struct \_SMS\_receipt

{

UINT8 aszMsgID[WD\_MSGID]; //本消息ID

UINT8 aszFromSys[WD\_SYSNUM]; //系统编号

UINT8 aszMsgType[WD\_MSGTYPE]; //消息类型

UINT8 aszSourceMsgID[WD\_MSGID]; //原始消息ID

UINT32 dwTarget; //消息接收者ID

UINT8 aszResult[WD\_MSGSTATUS]; //短信下发状态

}SMS\_receipt;

## 短信回执给客户端

typedef struct WD\_SMS\_to\_client

{

UINT64 dwPhoneNum; //手机号码

UINT8 cResultType; //回执类型 0：成功 1：失败 当成功的时候回执描述为空

UINT8 aszDescribe[DESCRIBE\_LEN]; //发送失败的结果描述

}SMStoclient;

## IM/短信应答消息

typedef struct WD\_resp\_msg

{

UINT8 aszFromSys[WD\_SYSNUM]; //系统标识

UINT8 aszMsgID[WD\_MSGID]; //源消息ID

UINT8 aszSendTime[WD\_SENDTIME]; //发送时间

UINT8 aszResult[WD\_MSGRESULT]; //结果

UINT8 aszResultDesc[WD\_MSGSTATUS]; //结果描述

}WD\_respmsg;

## 万信消息发送请求

typedef struct WD\_SMS\_req

{

UINT32 dwBatch; //批次号

UINT32 dwTotal; //批次总数

UINT64 dwPhoneNum; //手机号码

UINT8 aszContent[WD\_CLIENTMAXLEN]; //短信内容

}SMSreq;

## 通讯数据包

struct data\_packet

{

UINT32 lSrcNetId; //源网元ID

UINT32 lDestNetId; //目的网元ID

UINT32 lPeerIp; //对端IP[用于业务层取IP地址]

UINT8 cPacketType; //数据类型[业务层不操作]

UINT16 iPacketLen; //包长

UINT32 lRouteInfo; //路由信息 主动发送为0 应答包需与请求包一致

UINT16 iPacketId; //包ID,应答包需与请求包一致

UINT16 iSerialId; //包辅助ID,用于一次请求有多个回应包，需要按序接收。

由服务端按序填入，从1开始，客户端按序取出。

INT32 lTimeMark; //时间戳

UINT16 iOperationCode; //操作码

INT16 iReturnCode; //返回码，错误码

UINT32 lFunctionCode; //数据包功能码URD中指明最终目的其他操作中辅助业务

UINT8 strBuffer[PACKET\_LENGTH]; //数据包内容

};

## 企业基本信息

struct base\_compinfo

{

UINT32 dwCompID; //企业ID

INT8 aszCompCode[COMPCODE\_LEN]; //企业代码

INT8 aszCompName[COMPNAME\_LEN]; //企业名称

INT8 aszLogo[COMPLOGO\_LEN]; //企业LOGO路径

UINT32 dwEstablish\_time; //成立时间

UINT32 dwUpdate\_time; //更新时间

UINT16 wUpdate\_type; //更新类型

};

## 部门基本信息

struct base\_deptinfo

{

UINT32 dwDeptID; //部门ID

UINT32 dwCompID; //企业ID

char aszDeptName[DEPTNAME\_MAXLEN+1]; //部门名称

INT32 dwPID; //上级部门ID

UINT32 dwUpdate\_time; //更新时间

UINT16 wUpdate\_type; //更新类型

UINT16 wSort; //排序序号

char aszDeptTel[20]; //部门联系方式

char aszDeptEnName[ENNAME\_LENGTH]; //部门英文名

};

## 员工基本信息

typedef struct base\_userinfo

{

UINT32 dwUserID; //用户ID

UINT32 dwCompID; //企业ID

INT8 aszEmail[EMAIL\_STR\_LEN]; //邮箱

INT8 aszPwd[PWD\_STR\_LEN]; //密码

INT8 aszUserName[USERNAME\_LEN]; //姓名

INT8 aszUserCode[USERCODE\_LEN]; //工号

INT8 cSex; //性别

UINT32 dwBirth; //出生日期

INT8 aszLocal[POST\_LEN]; //职务

INT8 aszTel[64]; //办公电话

INT8 aszPhone[NUMBER\_LENGTH]; //手机号码

INT8 aszHomeTel[20]; //宅电

INT8 aszEmergencyPhone[NUMBER\_LENGTH]; //紧急联系手机

UINT32 dwEntryDate; //入职时间

INT8 aszLogo[LOGID\_STRING\_LENTH]; //用户头像路径

INT8 aszSign[SIGN\_LEN]; //个性签名

UINT16 wPurview; //用户权限

UINT32 dwUpdate\_time; //更新时间

UINT16 wUpdate\_type; //更新类型

UINT8 cMsgSynType; // 0 PC全收 PC在线，移动端在线不收； 1 两端都收；

2移动端不收，PC全收

UINT8 cForbidden; // 0: 账号启用 1: 账号禁用

UINT16 wAccouttype; //账号类型 0:私人账号 1：值班号 （值班号不需要sso验证）

UINT32 dwLogoUpdateTime;

INT8 aszUserAddress[ADDRESS\_LENGTH]; //联系地址

INT8 aszFax[20]; //传真

INT8 aszPostcode[10]; //邮编

INT8 aszEnName[ENNAME\_LENGTH]; //英文名

}base\_UserInfo;

## 员工部门对应关系

typedef struct DEPTUSER

{

UINT32 dwCompID;

UINT32 dwUserID;

UINT32 dwDeptID;

INT8 aszUserCode[USERCODE\_LEN]; //工号

INT8 aszUserName[USERNAME\_LEN]; //姓名

INT8 aszLogo[LOGID\_STRING\_LENTH]; //用户头像路径

INT8 cSex; //性别

INT8 cRankID; //本人级别

INT8 cProfessionalID; //本人业务

UINT16 dwAreaID; //本人地域

UINT32 dwUpdate\_time; //更新时间

UINT16 wUpdate\_type; //更新类型

UINT16 wUserSort; //人员排序

}deptuser;

## 信息发送请求

struct send\_message\_req

{

UINT8 sMsgid[MSG\_ID\_LENGTH]; //系统MSGID

UINT32 dwSrcUserID; //源号码

UINT32 dwDstUserID; //目的号码

INT8 aszGroupID[GROUPID\_LEN]; //群组ID

INT8 cIsGroup;//是否群组聊天0: 不是 1:是 2:固定群组 3: 点对点消息同步自己

UINT32 dwGroupTime; //本地群组时间

UINT64 dwMsgID; //消息ID

UINT8 cMsgType;//0：文字 1：图片 2：语言片段 3：视频 4：附件 5：点对点握手请求

8：PC自动回复 7：长消息 6：文件接收回复 10：IOS推送/消息来自公

众平台 20：广播消息 21:短信发送 22：短信发送结果回执

23：万达平台的短信回复 24：万达平台的IM消息 25：万达平台的IM提醒

UINT8 cRead; //0:不发已读 1:发送已读回执

UINT8 cAllReply; //是否是一呼百应消息 0:不是 1:一呼百应 2:一呼万应消息回复

UINT64 dwSrcMsgID; //一呼万应源消息ID

UINT8 cOfflineMsg; //0:非离线消息 1:离线消息

UINT8 cBatchTotal; //同批次消息总数，8.5号开始用作 通知标识，

1告诉客户端不通知用户

UINT8 cBatchSerial; //消息下发次数，反复离线消息

UINT16 nOfflineBatchTotal; //离线消息同批次消息总数，不超过256

UINT16 nOfflineBatchSerial; //离线消息同批次消息序列

UINT8 cTerminalType; //信息接收方类型 1:PC端与移动端均发

2:只发送移动端//3:只发送给PC端

UINT16 dwMsgLen; //消息长度

UINT32 dwSendTime; //发送时间

INT8 aszMsgContent[MSG\_CONTENT\_LEN]; //网信文字内容

};

## 发送信息应答

struct send\_message\_resp

{

UINT8 sMsgid[MSG\_ID\_LENGTH]; //系统MSGID

UINT32 dwDstUserID; //目的号码

UINT64 dwMsgID; //消息ID

UINT16 iRetcode; //返回码

};

## 消息通知

struct message\_notice\_req

{

UINT8 sMsgid[MSG\_ID\_LENGTH]; //系统MSGID

UINT32 dwSrcUserID; //源号码

UINT32 dwDstUserID; //目的号码

INT8 aszGroupID[GROUPID\_LEN]; //群组ID

INT8 cIsGroup; //是否群组聊天0: 不是 1:是 2:固定群组

3:点对点消息同步给自己 4.群消息同步给自己

UINT32 dwGroupTime; //服务端本地群组时间

UINT64 dwMsgID; //消息ID 消息类型：0：文字 1：图片 2：语言片段

3：视频 4：附件 5：点对点握手请求 6：PC自动回复

7：长消息 8：文件接收回复 10：IOS推送/消息来自公众平台 20：广播消息

UINT8 cMsgType;

UINT8 cRead; //0:不发已读回执 1:发送已读回执

UINT8 cAllReply; //是否是一呼百应消息 0:不是 1:一呼百应

2:一呼万应消息回复

UINT64 dwSrcMsgID; //一呼万应源消息ID

UINT8 cOfflineMsg; //0:非离线消息

//1:离线消息

UINT8 cBatchTotal; //同批次消息总数，8.5号开始用作 通知标识，1告诉客户

端不通知用户 2.下发离线消息标记为已读

UINT8 cBatchSerial; //同批次消息序列，从1开始，不超过256

UINT16 nOfflineBatchTotal; //离线消息同批次消息总数，不超过256

UINT16 nOfflineBatchSerial; //离线消息同批次消息序列

UINT8 cTerminalType; //信息接收方类型 1:PC端与移动端均发

2:只发送移动端 3:只发送给PC端

UINT16 dwMsgLen; //消息长度

UINT32 dwSendTime; //发送时间

INT8 aszMsgContent[MSG\_CONTENT\_LEN]; //网信文字内容

};

## 消息通知应答

struct message\_notice\_resp

{

UINT16 iRetcode; //返回码

message\_notice\_req MsgNotice; //消息通知

};

## 广播信息发送请求

struct send\_broad\_req

{

UINT8 sMsgid[MSG\_ID\_LENGTH]; //系统MSGID

UINT32 dwSrcUserID; //源号码

UINT32 dwDestUserID; //目的号码

UINT8 cTerminalType;

UINT64 dwMsgID; //消息ID

UINT32 dwSendTime; //发送时间

UINT8 cMsgType; //0:文字 1:图片 2:语音

UINT8 cAllReply; // 是否是一呼万应广播消息,1:一呼万应

UINT64 dwSrcMsgID; // 一呼万应源消息ID

INT8 aszTitle[MSG\_TITLE\_LEN]; //标题

UINT16 dwMsgLen; //消息长度

INT8 aszMsgContent[MSG\_BROAD\_LEN]; //广播内容

};

## 广播发送信息应答

struct send\_broad\_resp

{

UINT8 sMsgid[MSG\_ID\_LENGTH]; //系统MSGID

UINT16 iRetcode; //返回码

};

## 广播消息通知

struct broad\_notice\_req

{

UINT8 sMsgid[MSG\_ID\_LENGTH]; //系统MSGID

UINT32 dwSrcUserID; //源号码

UINT32 dwDstUserID; //目的号码

UINT8 cTerminalType; //信息接收方类型 1:PC端与移动端均发 2:只发送移动端 3:只发送给PC端

UINT64 dwMsgID; //消息ID

UINT32 dwSendTime; //发送时间

UINT8 cMsgType; //0:文字 1:图片 2:语音

UINT8 cAllReply; / /是否是一呼万应 0:广播消息,1:一呼万应,3:轻应用下发

UINT64 dwSrcMsgID; // 一呼万应源消息ID

INT8 aszTitle[MSG\_TITLE\_LEN]; //标题

UINT16 dwMsgLen; //消息长度

INT8 aszMsgContent[MSG\_BROAD\_LEN]; //广播内容

};

## 广播消息通知应答

struct broad\_notice\_resp

{

UINT8 sMsgid[MSG\_ID\_LENGTH]; //系统MSGID

UINT16 iRetcode; //返回码

}

# 代码名称，功能

## 处理短信消息类

class WdInterface

|  |  |
| --- | --- |
| 函数名 | 功能 |
| string GetHostByName(const char\* domain) | 解析主机名字到IP |
| static int InitListenWcfSocket(struct sockaddr\_in\* sSocketAddr) | 初始化监听端口 |
| static int DealWithMsgFromWcf(char\* buff, int nSocket); | 处理来自wcf平台的消息 |
| static int SendMsgToWcf(WDMSG \* pMsg, int nSocket) | 组织json和http请求消息 |
| int ReturnBatchResp(SMSbatch\* pBatch,SMS\_receipt \*pSmsReceipt = NULL) | 返回短信批次结果给客户端 |
| int SendMsgToClient(WDMSG \* pRecvMsg, WDMSG \*pRamMsg) | 将消息转发给客户端 |
| static int ConnectToWcf(int nType,WDMSG \* pMsg) | 根据客户端请求产生报文，发送给个短信平台 |
| static int SendRespToWcf(WD\_respmsg\* pMsg, int nSocket,int nFlag) | 根据短信平台请求产生应答消息报文发送给短信平台 |
| static void\* thRecvThread(void\* pParam) | 接收短信平台发送过来的消息 |
| static void\* thDealThread(void\* pParam) | 对收到短信平台的消息进行处理 |
| static void\* thTimeOut(void\* pParam) | 对等待短信平台回执超时进行处理 |
| int SendData(int nSocket, char \*pszData, unsigned int dwSize) | 发送数据 |
| int WDRecvMsg(int nSockFd, char\* pBuffer, unsigned int uBufLen); | 接收数据 |

## Udp类

**class CUdp**

|  |  |
| --- | --- |
| 函数名 | 功能 |
| int Init(deal\_recvdata\_func nrecvfunc,const RamTable \*pRamTable,char\* dumpPath,char\* logPath,UINT32 nPort = 8083) | 初始化udp |
| static void\* thSendThread(void\* pParam); | 发送消息到核心模块线程 |
| static void\* thRecvThread(void\* pParam) | 接收核心模块发送过来数据线程 |
| static void\* thFetchRecvData(void\* pParam) | 分析处理数据线程 |
| static void\* thDeaiWithTimeOut(void\* pParam) | 超时处理线程 |
| int FindDstInfoByNetid(int netid,terminal \*pTerminal) | 读取接入终端网元信息 |
| int FindDstInfoByServerType(int type,terminal \*pTerminal) | 读取核心模块终端网元信息 |
| int SendToTerminal(int sockfd,const sys\_data\_packet \*data,int sendlen) | 发送消息给客户端 |

## 内存数据表类

class RamTable

|  |  |
| --- | --- |
| 函数名 | 功能 |
| int InitTable(RamTableInfo \*pTableInfo, RamKeyInfo \*pKeyInfo, const char \*sDumpPath=DEFAULT\_DUMP\_PATH, const char \*sLogPath=DEFAULT\_LOG\_PATH) | 初始化内存表 |
| int OperateRecord(int nOPMode, char \*pRecord, const char \*pKey, int nKeyIndex, char \*sCursor=NULL) | 操作内存表函数 |
| int InsertRecord(const char \*pRecord) | 插入数据到内存表 |
| int UpdateRecord(const char \*pRecord, char \*sCursor) | 更新内存表数据 |
| int DeleteRecord(char \*sCursor) | 删除内存表数据 |
| int SelectRecord(int nOPMode, char \*pRecord, const char \*pKey, int nKeyIndex, char \*sCursor=NULL) | 单选数据 |

## 数据库类

class DBConnPool

|  |  |
| --- | --- |
| 函数名 | 功能 |
| void init(const char \*usr, const char \*pwd, const char \*db, const char \*host, int port, const char\* chrset) | 初始化mysql连接 |
| MYSQL\* getConnection() | 连接数据库 |
| void close(MYSQL\* conn) | 关闭连接 |

## 终端参数类

class CCfgParam

|  |  |
| --- | --- |
| 函数名 | 功能 |
| INT32 ReadParamFromFile() | 读取配置文件终端参数 |
| void PrintCfgParam() | 打印数据 |

## 队列类

class CQueue

|  |  |
| --- | --- |
| 函数名 | 功能 |
| int GetIterm(char \*mIterm, int\* mLen) | 获取队列中的数据 |
| void AddIterm(char \* mIterm, int mLen) | 添加数据到队列中 |

# 数据库交互设计

## 读取终端网元信息

从数据库terminal表中把终端网元信息读取到terminal结构对象中，然后把terminal结构对象插入到终端内存表g\_oRamTerminal中

//网元信息结构

struct terminal

{

UINT32 lNetid; //消息接收方

UINT16 iTerminalType; //终端类型1 CORE 2SERVICE 3MANAGER

UINT16 iMajorFlag; //主备标识

INT8 strIP[IP\_STR\_LENGTH]; //IP地址

UINT32 lPort; //端口

UINT32 dwUdpPort; //udp监听端口

UINT8 cIsAlive; //活跃时间

UINT32 dwAliveTime; //活跃时间

};

## 读取用户简要信息

从数据库user表中把用户简要信息读取到user\_info\_fetch结构对象中，然后把user\_info\_fetch对象插入到 g\_oRam\_UserBasicInfo内存表中。

//用户简要信息

typedef struct \_user\_info\_fetch

{

UINT32 dwUserID; //用户ID

INT8 aszUserName[USERNAME\_LEN]; //姓名

INT8 email[EMAIL\_STR\_LEN];

INT8 szPwd[PWD\_STR\_LEN];

UINT32 dwCompID;

UINT8 cMsgSynType; //同步类型，针对于移动客户端。设置同步

后，移动端会收到PC端同步的消息

UINT8 cUserSpecial; //特殊用户标识，不允许被发起会话，

发送短信，发起广播，创建群组

UINT8 code[USERCODE\_LEN]; //工号

} user\_info\_fetch;