

第三方厂商门禁接入方案

文档编号：
撰写人：郭铃
日期：2015-06-23

目录

1.	总体功能概述.....	4
1.1	总体功能需求.....	4
1.2	需求场景.....	4
1.2.1	第三方厂商云服务器直连卡卡兔服务器.....	4
2.	第三方厂商接入流程.....	4
2.1	第三方厂商接入流程.....	4
2.1.1	第三方厂商申请接入信息.....	4
2.1.2	第三方厂商开发通信协议.....	4
2.1.3	第三方厂商接入测试服务器.....	4
2.1.4	第三方厂商接入正式服务器.....	4
3.	门禁通信协议.....	5
3.1	AMQP.....	5
3.1.1	什么是 AMQP.....	5
3.1.2	为什么要用 AMQP.....	5
3.1.3	AMQP 服务器.....	5
4.	第三方厂商接入.....	5
4.1	接入说明.....	5
4.2	接入细节.....	5
4.2.1	申请门禁密钥.....	6
4.2.2	回复门禁密钥.....	6
4.2.3	申请门禁二维码.....	6
4.2.4	推送门禁二维码.....	6
4.2.5	开门指令.....	7
4.2.6	开门结果反馈.....	7
4.2.7	禁止指令.....	8
4.2.8	使能指令.....	8
4.2.9	重启指令.....	8
4.2.10	心跳.....	8

4.2.11	告警.....	9
4.2.12	蓝牙开门.....	9
4.2.13	取消告警.....	10
4.2.14	发卡.....	10
4.2.15	取消卡.....	10
4.2.16	刷卡(开门请求).....	10
5.	代码示例.....	11
5.1.1	Java.....	11
5.1.2	PHP.....	11
5.1.3	C.....	11

版本号	变动
1.03	加入中控对接方案
1.02	加入脱机门禁指令：申请密钥，回复密钥
1.01	加入禁止指令,使能指令和告警指令
1.00	初始化文档，第三方厂商接入协议基础内容

1. 总体功能概述

1.1 总体功能需求

- 手机 APP 扫描二维码开门，提高用户出入的便捷性，无需带卡出行
- 智能自动化减少物业业务，节约运营成本

1.2 需求场景

1.2.1 第三方厂商云服务器直连卡卡兔服务器

2. 第三方厂商接入流程

2.1 第三方厂商接入流程

2.1.1 第三方厂商申请接入信息

第三方厂商向我方申请接入账号密码，我方将提供以下数据

厂商用户名

厂商密码

预定义厂商密钥

测试服务器

正式服务器

2.1.2 第三方厂商开发通信协议

2.1.3 第三方厂商接入测试服务器

我方将提供测试服务器用以进行接入调试

接入服务器地址

接入 app 微唐测试版

2.1.4 第三方厂商接入正式服务器

接入服务器地址 1

接入服务器地址 2

接入 app 微唐正式版

3. 门禁通信协议

门禁通信协议基于 AMQP 通信协议

3.1 AMQP

3.1.1 什么是 AMQP

AMQP，即 Advanced Message Queuing Protocol, 一个提供统一消息服务的应用层标准高级消息队列协议, 是应用层协议的一个开放标准, 为面向消息的中间件设计。基于此协议的客户端与消息中间件可传递消息，并不受客户端/中间件不同产品，不同开发语言等条件的限制。Erlang 中的实现有 RabbitMQ 等。

3.1.2 为什么要用 AMQP

我们的目标是实现一种在全行业广泛使用的标准消息中间件技术，以便降低企业和系统集成的开销，并且向大众提供工业级的集成服务。

我们的宗旨是通过 AMQP，让消息中间件的能力最终被网络本身所具有，并且通过消息中间件的广泛使用发展出一系列有用的应用程序。

3.1.3 AMQP 服务器

自主开发的 AGUA SYSTEM 1.0 集群系统

4. 第三方厂商云服务接入

4.1 接入说明

我方为第三方厂商预分配 amqp 服务器用户名，密码，通道和队列。第三方厂商云服务器通过 AMQP 协议接入我方服务器，模式为 Direct

- 1 在 qrcode 队列申请二维码
- 2 每隔 25s 向 heartbeat 队列发送下挂门禁的心跳信息
- 3 监听开门队列开门指令，完成开门动作后向 door_response 队列反馈开门状态（成功/失败）
- 4 备份，第三方服务器应同时连接我方两个正式服务器，如果出现其中一个服务器断开情况，可以使用第二个服务器作为备份。同时，还继续尝试连接第一个服务器。我方开门指令会同时发送到两个服务器，如果两个服务器都工作的情况下，仅有一台服务器会向外发送消息。

4.2 接入细节

厂商通道/队列：以厂商预分配密钥命名的队列和通道

4.2.1 申请门禁密钥

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥
我方向第三方服务器申请指令	70#流水号#厂商设备地址#用户 ID#有效期	
第三方服务器回复到 keys 通道	71#预分配厂商密钥#流水号#密钥	

我方服务器向脱机门禁厂商申请门禁密钥

密钥会通过 keys 通道回复

4.2.2 回复门禁密钥

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
keys	keys	keys
第三方服务器回复指令	71#预分配厂商密钥#流水号#密钥	

第三方服务器向我方发送门禁密钥

4.2.3 申请门禁二维码

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
qrcode	qrcode	预分配厂商密钥
第三方服务器申请指令	30#预分配产生密钥#厂商设备地址	
我方服务器回复到厂商通道	31#厂商设备地址#二维码字符串	

仅对带有显示屏的智能门口机有效

门禁设备启动后，应主动向服务器申请最新二维码信息

二维码会通过厂商通道下发，厂商对比设备地址，确认正确情况下可以将二维码字符串转换成二维码用于展示

4.2.4 推送门禁二维码

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥

我方服务器推送指令	31#厂商设备地址#二维码字符串
第三方服务器确认	无

仅对带有显示屏的智能门口机有效

后台修改门禁二维码后，我方服务器会推送最新二维码字符串到第三方厂商云服务器

第三方服务器也可以主动申请查询门禁二维码

厂商对比设备地址，确认正确情况下可以将二维码字符串转换成二维码用于展示

4.2.5 开门指令

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥
我方服务器发送	00#开门标示符#厂商设备地址#门号	

说明

开门标示符为 一个唯一标示本次开门操作的数字

门号为多路控制器使用，范围 [1-最大门限]

举例： 四路控制器， 范围为 [1-4]

如果是单路控制器，请忽略这个域

4.2.6 开门结果反馈

目标交换机(exchange)	目标队列（持久化 true）	目标关键字(routing key)
door_response	door_response	door_response
第三方服务器发送	50#厂商设备地址#开门标示符#开门结果#理由	

在收到开门指令并执行开门动作后，第三方厂商应该发送开门反馈至我方服务器

50#开门标示符#开门结果#理由

说明

开门标示符为 开门指令中的开门标示符

开门结果 0 成功

1 失败

理由 描述开门成功或者失败的原因

4.2.7 禁止指令

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥
我方服务器发送	02#厂商设备地址	

说明

禁止指定门禁设备连接到我方服务器

4.2.8 使能指令

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥
我方服务器发送	03#厂商设备地址	

说明

允许指定门禁设备连接到我方服务器，配合禁止指令

4.2.9 重启指令

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥
我方服务器发送	01#厂商设备地址	

说明

重启门禁设备

4.2.10 心跳

目标交换机(exchange)	目标队列（持久化 true）	目标关键字(routing key)
heartbeat	heartbeat	heartbeat
发送指令	99#厂商预定义密钥#上线设备个数#异常设备个数#	

注

每 200 秒第三方服务器应该发送心跳到我方服务器，我方服务器无消息回送

4.2.11 告警

目标交换机(exchange)	目标队列（持久化 true）	目标关键字(routing key)
heartbeat	heartbeat	heartbeat
发送指令	98#厂商设备地址#问题原因	

注

通知我方服务器指定门禁设备出问题以及问题原因

4.2.12 蓝牙开门

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
secrets	secrets	secrets
第三方厂商发送蓝牙开门申请	#72#标识码#蓝牙地址#临时密码 标识码由门禁主机本地产生，用来表示一次开门	
我方回复到主机队列	#73#标识码#0 （允许开门）	
	#73#标识码#1 （禁止开门,无权限）	
	#73#标识码#2 （禁止开门,密码已经过期）	

4.2.13 取消告警

目标交换机(exchange)	目标队列（持久化 true）	目标关键字(routing key)
heartbeat	heartbeat	heartbeat
发送指令	97#厂商设备地址	

注

通知我方服务器指定门禁设备已经恢复正常

4.2.14 发卡

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥
我方服务器发送	88#厂商设备地址#授权卡号#起始时间#结束时间	

时间格式 YYYYMMDDHHIISS

20150908125959

4.2.15 取消卡

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
预分配厂商密钥	预分配厂商密钥	预分配厂商密钥
我方服务器发送	87#厂商设备地址#授权卡号	

4.2.16 刷卡(开门请求)

目标交换机(exchange)	目标队列（持久化 false）	目标关键字(routing key)
open	open	预分配厂商密钥
第三方服务器申请指令	80#预分配产生密钥#厂商设备地址#卡号	
我方服务器回复到厂商通道	81#厂商设备地址#卡号#状态 (0 无效 1 允许)	

用户刷卡后,门禁读头读取门禁卡卡号,通过中控服务器向门禁后台请求该卡号是否有开门权

限,门禁后台返回后,根据状态值进行操作.

5. 代码示例

5.1.1 Java

<http://www.rabbitmq.com/api-guide.html>

- **RabbitMQ Java client** and the **API-guide**
- **RabbitMQ JMS client** (commercial)
- Lyra, **highly available RabbitMQ Java client**
- **camel-rabbitmq**, an Apache Camel component for interacting with RabbitMQ. This ships as part of Camel 2.12 or later.
- **Banyan**, a RabbitMQ-backed message bus with a tree topology.

5.1.2 PHP

- **php-amqplib** a pure PHP, fully featured RabbitMQ client
- **RabbitMqBundle** incorporates RabbitMQ messaging with the Symfony2 web framework
- **PECL AMQP library** built on top of the **RabbitMQ C client**
- **VorpaiBunny** a PHP client library using **rabbitmq jsonrpc channel**
- **Thumper** a library of messaging patterns
- **CAMQP** an extension for the Yii framework providing a gateway for RabbitMQ messaging

5.1.3 C

- **RabbitMQ C client**
- **SimpleAmqpClient**, a C++ wrapper around rabbitmq-c
- **amqpcpp**, a C++ message library for RabbitMQ
- **AMQP-CPP**, a C++ RabbitMQ client