

目录

1 安装.....	4
1.1 配置探针.....	5
1.1.1 方法 1: 通过 WEB 页面配置(适合单机配置).....	5
1.1.2 方法 2: 通过安卓 APP 配置(适合批量).....	6
1.2 验证配置.....	7
1.3 部署.....	7
1.3.1 如何划分区域.....	8
1.3.2 密集网状部署.....	9
1.3.3 固定探针.....	10
2 云平台.....	10
2.1 登录.....	11
2.2 配置.....	11
2.2.1 建筑物.....	11
2.2.2 楼层.....	12
2.2.3 区域.....	12
2.2.4 探针.....	13
2.2.5 算法.....	13
2.3 仪表盘.....	15
2.4 热力图.....	16
2.4.1 多探针热力图.....	16
2.4.2 单探针热力图.....	17
2.5 实时分布.....	18
2.6 MAC 轨迹.....	18
2.7 监控.....	18
2.7.1 人员详细.....	18
2.7.2 人员汇总.....	20
2.7.3 物品.....	21
2.8 实时定位.....	22
2.8.1 显示位置.....	22
2.8.2 显示列表.....	23
2.9 导出.....	23
3 转发接口.....	24
3.1 数据流.....	24
3.2 访问控制.....	24
3.3 术语说明.....	24
3.4 如何通过工具验证下接口.....	25
3.5 协议.....	26
3.6 推送类.....	26
3.6.1 接收全部数据 mac 探针信息(wifi 上报方式).....	27
3.6.2 接收全部数据 mac 探针信息(gprs 上报方式).....	27
3.6.3 接收全部数据 接收全部商品触碰信息.....	28
3.6.4 当顾客临近监测区域推送.....	28

3.7 查询类.....	29
3.7.1 view/heatmap 查询区域热度.....	29
3.7.2 view/stastatus 查询探针状态.....	29
3.7.3 view/stacustomer 查询经过探针周围顾客情况.....	30
3.7.4 view/customersummary 查询顾客经过, 进入指定场所情况(deprecated).....	30
3.7.5 view/shopcurrent 查询实时分布情况(统计数量).....	31
3.7.6 view/shopdetail 查询实时分布情况(统计从哪来, 去哪了).....	31
3.7.7 view/dashboard 仪表盘查询.....	32
3.7.8 view/customerlist 查询指定场所顾客明细.....	35
3.7.9 view/customer 查询 mac 轨迹情况.....	35
3.7.10 view/customermonitor 查询监控人员信息.....	36
3.7.11 view/current 查询当前指定区域所探测到的顾客数.....	36
3.7.12 view/currentdetail 查询指定条件探测到的所有 mac.....	37
3.8 配置类.....	38
3.8.1 config/buildinglist 查询建筑物属性.....	38
3.8.2 config/floorlist 查询楼层属性.....	38
3.8.3 config/shoplist2 查询区域属性.....	39
3.8.4 config/shoplist 查询区域属性(deprecated, 推荐使用 shoplist2).....	40
3.8.5 config/shopupdate 配置区域属性.....	41
3.8.6 config/stalist 查询探针属性.....	41
3.8.7 config/staupdate 更新探针属性.....	42
3.8.8 config/urllist 查询转发 url.....	42
3.8.9 config/urlinsert 插入转发 url.....	42
3.8.10 config/urlupdate 更新转发 url.....	43
3.8.11 config/urldel 删除转发 url.....	43
3.8.12 config/approacharealist 查询监测区域.....	43
3.8.13 config/approachareainsert 设定特定监测区域.....	44
3.8.14 config/approachareadel 删除监测区域.....	44
3.8.15 config/approachareauupdate 更新监测区域.....	44
4 直报接口(wifi 上报).....	45
4.1 数据流.....	45
4.2 配置探针.....	45
4.3 协议格式.....	46
4.4 探针说明.....	46
4.5 接口定义.....	46
4.5.1 获取服务器时间.....	46
4.5.2 接收 wifi a 信息.....	47
4.5.3 接收 wifi b 信息.....	50
4.5.4 接收 flash 存储信息.....	50
4.5.5 接收 ap 扫描信息.....	50
4.5.6 接收 wifi b 心跳信息.....	50
4.5.7 接收 wifi a 配置信息.....	51
4.5.8 接收手机与 mac 地址关联信息.....	51
4.5.9 接收商品标签信息.....	52

4.6 代码例子.....	52
5 直报接口(gprs 上报).....	54
5.1 数据流.....	54
5.2 配置探针.....	54
5.3 探针说明.....	55
5.4 协议格式.....	55
5.4.1 上报.....	55
5.4.2 返回.....	55
5.5 代码例子.....	56
5.5.1 Go 语言.....	56
6 硬件规格.....	59
6.1 智隐测 T1 型 (仅探测人流,WIFI 上报方式).....	59
6.2 智隐测 T2 型 (仅探测人流,GPRS 上报方式).....	60
6.3 智隐测 T3 型 (仅探测人流,GPRS, GPS 定位).....	60
6.4 智隐测 T4 型 (探测人流与探测商品触碰).....	61
6.5 智隐测 T5 型 (探测人流,网线上报,POE 供电).....	62
6.6 智隐测 C1 型 (触碰标签).....	63
6.7 智隐测 C2 型 (定位标签).....	64
7 经常问的问题.....	65
7.1 为什么需要上报数据?.....	65
7.2 如何评估一个探针?.....	65
7.3 这个探针既能探测又能当做 wifi 上网么?	65
7.4 探针防水么?	65
7.5 你们的后台程序能够共享源代码么?	65
7.6 你们的后台程序能部署在客户的服务器上么?	65
7.7 如何供电, 如何安装?	65
7.8 已经连接了 ap 的手机能探测么?.....	66
7.9 苹果能探测么?	66
7.10 苹果黑屏到底多久能够抓取到包?.....	67
7.11 手机黑屏时多久发射一次信号.....	67
7.12 探测率到底是多少比例.....	67
7.13 客流准确度如何?.....	68
7.14 定位精度如何?.....	68
7.15 不同型号有什么区别?.....	68
7.16 如何购买?.....	68
7.17 能发送广告么?	69
7.18 只买硬件有用么?	69
7.19 方案如何收费合作如何开始?.....	69
7.20 苹果采集率到底有多高?.....	69
7.21 如何获取探针完整 mac 地址.....	72
7.22 如何根据 mac 地址推送?	72
8 探针高级配置.....	73
8.1 以下针对 A 模块配置.....	73
8.2 以下针对 B 模块配置.....	74

1 安装

我司系统属于专业级别的客流采集和定位系统，功能丰富全面，需要客户认真阅读手册，也可以通过目录指引，在需要的时候参考。

智隐测探针因型号和使用方式不同，安装方式也不同，以下逐一说明。如果是 T2,T3 型号，可以直接插电后（一定要先插电）跳转到章节[2.云平台]登陆后台系统。

1.1 配置探针

探针探测到数据后需要周期上报到服务器端，我们需要配置某些方式让探针能够上报，T1,T4 型号探针通过连接其他 wifi 上报(目前必须是 2.4G wifi 才可以)数据，上报频率快，可以最快 1 秒上报一次，T2,T3 型探针通过 GPRS 上报，因为有流量的限制,GPRS 上报最快 1 分钟 1 次(如果有特殊需求，可以和我们沟通，也可以设置为 20 秒,但流量费就高了)。T2,T3 不需要配置,插电即可使用,因此可以直接跳转到” 1.2 验证配置”，当然，针对 T2,T3 如果需要配置上报服务器地址等操作，也是可以的，请阅读“探针高级配置”中关于 B 模块的配置方法或者看看“直报接口”。以下仅介绍 T1,T4 型号探针配置方式，注意，商场那种商业 wifi 一般是通过微信或者手机短信认证，不能直接使用，可以让 wifi 部署方开设一个用密码接入的新 ssid 用于探针来连接，商业 wifi 一般都支持配置多个 ssid。

1.1.1 方法 1：通过 WEB 页面配置(适合单机配置)

道理和配置家庭路由器很像，探针通电，红灯亮起，大约 1 分钟后熄灭，一定要等熄灭后，通过电脑搜索 wifi,会看到 fk-xxxxxx 的 wifi 名称，其中 xxxxxx 为你设备外壳上的字母，点击它连接，电脑不要设置为固定 IP,要设置为通过 dhcp 获取 ip，提示密码时输入 aaaaaaaa （8 个小写的 a),如果连接不成功，就多尝试几次，尝试 10 次，肯定能成功，如果还不行就通过安卓 app 的办法配置。然后打开浏览器，输入 http://192.168.4.1/config

注意，一般情况下只要修改“热点名称”和“热点密码”，(热点就是你的 wifi,比如你手机连接 wifi 上网，就是那个 wifi 的名称和密码,没有密码的 wifi 不能用)其他不要修改，尤其“上报地址”不要修改，查看探测的数据，在配置好后要看章节 2 云平台，而不是修改这里。

热点名称

sec

热点密码

f %

探针密码

aaa

上报地址(只能为ip,如果是TCP连接，请不要修改斜线后面的URI内容，只要改ip和端口即可)

1. /upload3

上报频率(单位:100毫秒)

10

状态

41:online[http/tcp]

提交

热点名称和热点密码改为你 wifi 的对应名称和密码（需要为 2.4G），然后点击提交(其他内容不了解的话，不要修改)，大约 10 秒后探针会进入工作状态，指示灯约几秒闪烁 1 次即表示正常，不闪烁一定不正常，闪烁了一定是正常，这里的闪烁是指几秒以内一次，配置 1 次即可，以后更改 wifi 后重新配置，不改不用配置，重启也没关系。

注：如需获知探针本身 mac 地址，请参考常见问题章节。

1.1.2 方法 2: 通过安卓 APP 配置(适合批量)

以下配置是通过手机 app 把指定的 wifi 的 ssid 和密码告知探针, 这样探针以后就通过指定的 wifi 来上报数据, 配置一次即可。wifi 要设置一个密码, 密码不要以*##开头, 另外就是 wifi 要启用 2.4 G 频段, 目前不支持 5G 频段。

- ✓ 用安卓手机扫描下面二维码下载配置工具, 在手机连接 wifi(必须)的情况下打开 app, **要确保你手机连接的 wifi 的 ssid 是 2.4G 不是 5G, 否则连不上的**, 在界面中的“密码”输入框里面输入该 wifi 的密码, **先不要点击“确认”**, 注意手机和设备都尽量距离手机连的 wifi 近些。需要先让手机连接上 wifi 再打开 app, 这个顺序很重要。



如果扫描二维码不能安装 app, 也可以通过链接下载

<http://be.zhiyince.com/download/apk/smartconfig.apk>

如果还是报解析错误, 一般是 apk 还没有下载完就安装了, 可以在电脑上下载 apk 后 copy 到手机上安装, 比如 360 手机助手, 能很方便安装。

或者在我司云盘目录的下载目录里面下载。

- ✓ 把探针的电源线连接好, 连接后 10 秒左右, 与探针电源插入口对称的地方会有指示灯长亮 (注意是与电源插口位置对称的指示灯, 如果是防水外壳, 需要把外壳下方的罩拆下才能清晰看到指示灯), 这时候点击第一步中 app 的“确认” (如果较长时间后 app 报“搜索不到”, 不用理会, **只要指示灯 3-5 秒熄灭就表示成功**) 大约 3-5 秒后指示灯熄灭表示配置成功, 如果一切正常, 大约 40 秒后指示灯开始闪烁, 大约几秒钟闪烁一次, 每闪烁一次表示上传一次数据。点击手机的返回键可以关闭 app 弹出的对话框来开始新的配置。

如果配置不成功, 要检查 wifi 是否是 2.4G, 密码是否正确, 手机是否连上了这个 wifi, 要在探针灯亮时立刻点击确认 (提前输入好密码), 几百个人都连接成功, 因此不成功一定要多检查下。

配置的主要目的是通过手机 app 把你选定的 wifi 的 ssid 和密码广播出来, 而探针会扫描这些广播的包, 截取 ssid 和密码, 然后探针会存储下来, 这样下次探针启动时就知道要连接那个 wifi 来上网, 上报数据了, 因此配置只是一次性配置, 以后只要 wifi 不换, 就不用配置。

通过在密码输入框输入命令行能够做高级配置, 比如恢复出厂配置, 设置 ip 端口等, 具体参考章节: 探针高级配置

注: 如需获知探针本身 mac 地址, 请参考常见问题章节。

1.2 验证配置

只要探针几秒钟以内闪烁 1 次(T1,T4)或者几十秒到 10 分钟闪烁一次(T2,T3)，即表示配置成功，如果探针指示灯在 app 上按了确认后但仍旧亮持续 30 秒以上,表示配置没有成功(要检查路由器是否是 5G 的,需要 2.4G 的，也需要检查手机是否连接了路由器而不是连接了探针，通过 ssid 名字可以区分)，探针在找不到上报通路时会自动重启，这是为什么即便配置不成功，探针也会每几分钟亮起 1 次。另外，可以通过如下链接查看探针上报的原始数据来确认上报结果。

<http://be.zhiyince.com/dataserver/web/data/view?sta=07ccaa>

很多人经常把上面链接里面的 data 输入为 date,注意不要输入错误，如果的确没错，但还提示探针找不到，一定是过期了，一般购买硬件后我们送 1 个月的平台使用期限，过了这个期限就要续费了，否则会查不到。在浏览器中输入上述 url，其中黄色部分是设备外壳标签上的字母，输入设备上贴的标签（6 个字母），即可搜索到该设备上报的信息(探针的 mac 地址为 12 个字母,一般设备外壳上只会标示后 6 个字节,通过该页面能找到对应设备的完整的 12 个字节的 mac 地址),如果打不开连接核实下收入是否正确,http 后面没有 s,是 sta= 不是下面的 tag=

注: 如果您购买的是能够识别商品触碰的探针(t4)

可以查看商品标签原始数据。

<http://be.zhiyince.com/dataserver/web/data/view?tag=8c0000&sta=07ccaa>

只要把黄色部分替换为标签上粘贴的序列号，红色部分替换为探针上的序列号即可，注意上述链接和之前看探针数据的链接不同。

1.3 部署

部署和我们的应用场景有密切关系，探针的基本原理是探测手机无线信号，因此能够接收到手机的 mac 地址和信号强度，信号强度又能大致换算为距离，也就是说我们不但能探测到探针周围有多少台手机设备，也能探测到手机设备与探针的大致距离。使用场景有很多，比如有的客户通过人员携带，进行流动采集，有的客户部署在室内间隔 10 米密集部署，有的客户在室外间隔 50 米以上稀疏部署。系统内部署的架构是，探针属于区域，区域属于楼层，楼层属于建筑，和真实生活场景匹配，探针采集生成数据后会汇总到区域，区域汇总到楼层，楼层汇总到建筑，一个建筑的数据就是其叶子节点采集的数据去重后的结果。

部署方式	描述	注意事项	适用探针类型
流动采集	探针通过充电宝供电，人员携带流动采集	可以通过后台系统创建多个区域，每个区域对应一个探针，通过查看区域的数据情况了解探针采集的数据，多台流动探针也可以一起合并管理。	T1,T2,T3,T4,T1 需要配合随身 wifi,推荐 T2 或者 T3
稀疏网状部署	一般部署间隔大于 30 米	主要采取吸附法定位，也就是说每台探针探测到的 mac 地址多少即表示该探针所在区域的人数多少，无法生成多探针热力图，但有分布图和单探针热力图，定位精度取决于探针的位置，只要创建相应数量的区域对应到探针即可，不需要严格的坐标。	T1,T2,T3,T4 都可以，T1 需要 wifi 来上报数据
单探针部署	一个几十平米的店铺部署一台	一般建议探针部署在店的中心点，计算店内客流时，算法上以探针位置为圆心，设定的距离为半径，探测的信号强度落入这个区域就算做进店，不需要配置坐	T1,T2,T3,T4 都适合。

		标。	
密集网状部署	几百平米的区域内,间隔 10-15 米部署探针	需要严格计算探针坐标,系统通过多探针定位,能够较为准确的给出被探测设备的位置,以及生成漂亮的多探针热力图。	T1,T4, GPRS 上报类型的探针因为上报周期长,不适用此种方式
防楼上楼下串部署	天花板和地面都部署	有些建筑物的楼板钢筋含量低,因此不同楼层可能串信号,因此采取天花板和地面(或贴近地面的墙壁)都部署的方式	T1,T2,T3,T4 都可以, T1 需要 wifi 来上报数据

部署操作和云平台配置页面内容紧密相关,可以参考阅读。其他部署方式较为简单,下文重点讲述“密集网状部署”的方法。

1.3.1 如何划分区域

比如你想在办公室部署,那么在系统的配置页面可以创建一个建筑(默认已经创建),就是你们办公室所在大楼,然后建筑下面创建一个楼层(默认已经创建),然后楼层下面创建一个区域(默认已经创建),这个区域就是你们办公室,当然如果你们办公室占用整个楼层,这个区域的大小就等于你们楼层了。如果你们有多个区域,比如办公室横跨楼道两端,也可以考虑创建多个区域。

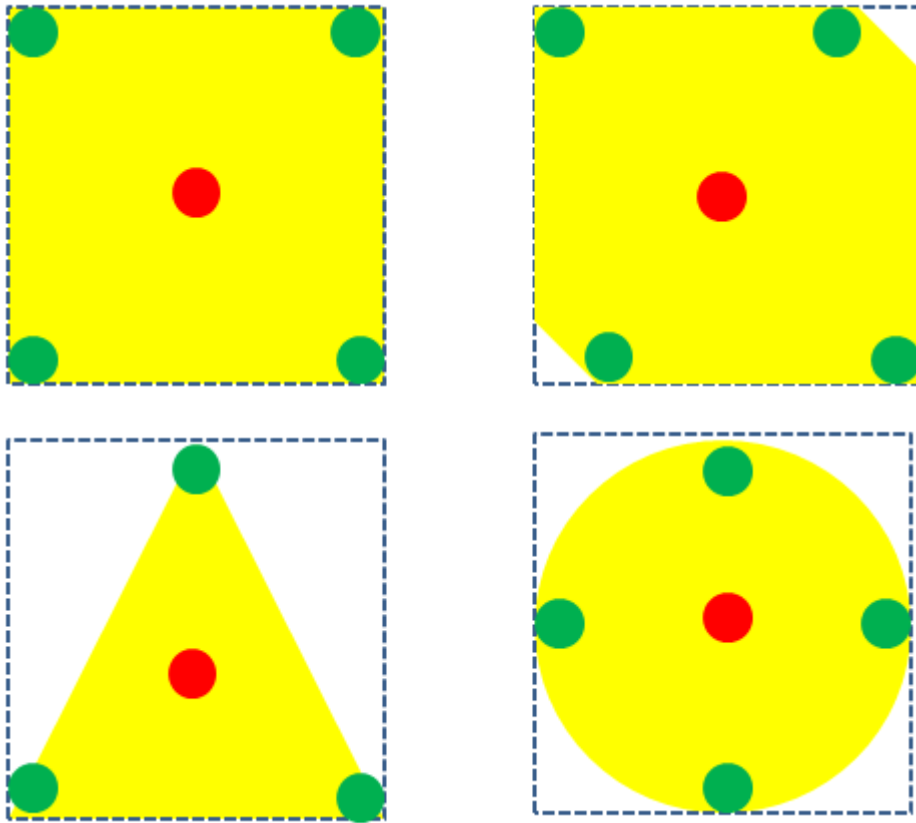
举 4 个例子

黄色部分为办公室区域,虚线代表在系统内设置的区域大小,可以看到如果区域恰好是长方向,那都很吻合,但如果黄色区域是不规则形状,系统内也只能按虚线的范围设定为规则的长方形,并且空白处也会被系统认为是区域的有效范围,这一点是因为系统目前只支持长方形。

如果希望做多探针热力图(密集网状部署),那么至少四个探针按照绿色圆点所指示的位置安装,并且区域建议面积在 100-200 平米之间,如果过小不建议做多探针热力图,因为误差较大,如果过大,建议切割为多个区域。

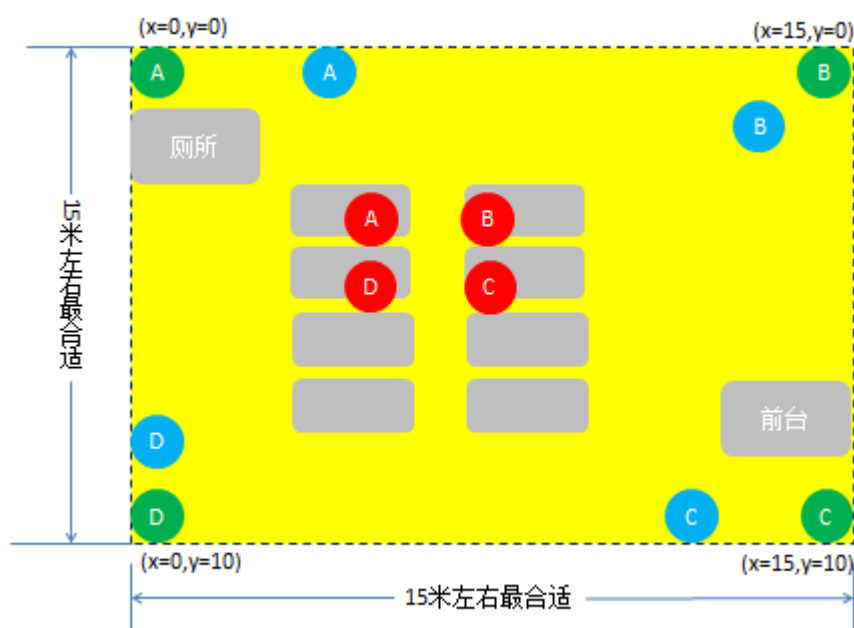
如果不做多探针热力图,一个区域只装一个探针,那么建议安装在红点所示位置。

如果你的区域较大, 并且很不规则,可以考虑把区域切割为多个区域。



1.3.2 密集网状部署

我们需要仔细测量探针所在的坐标，然后以米为单位，两位小数点填写进系统，以下图为例，系统以屏幕的左上角为坐标原点，向下为 y 轴，向右为 x 轴，例如如果希望厕所的位置出现在屏幕左上角，那么要以厕所附近为原点来测量。



- A,B,C,D 为探针，绿色为最合适的放置位置，蓝色可以接受，红色不能接受，总体原则是探针要尽量分散放置，绿色就是正好放到长方形的4个点，左上角的探针坐标就是(0,0)，这样的效果要好很多，即便比蓝色也要好很多，因此要尽量按绿色安装。
- 如果以绿色方式放置，系统登记坐标就如图所示。
- 热力图上呈现的平面图为 jpg 格式，一般要小于 300K，如果没有合适的图，可以如上图一样绘制个示意图。
- 图上传前要严格按坐标线来裁剪，不然比例就不协调，如上图，需要按虚线处裁剪，也即是图的尺寸要匹配实际坐标，实际场所坐标外的区域不能出现在图上。
- 图的上传和坐标录入，请阅读配置页面说明来了解如何操作。
- 区域在楼层内，一个区域在楼层的什么位置通过配置页面来定义，也就是区域的起点(左上角)x 和 y，这时用楼层的坐标系，但定义探针在区域的位置，那就要用区域的坐标系，比如区域左上角在楼层的(10,4)位置，而一个探针正好在该区域的左上角，那么探针的坐标是 (0,0)

1.3.3 固定探针

普通外壳探针可以通过玻璃胶粘贴，如果没有插座，供电可以通过 220v 电灯电线与插头的铜片相连，防水外壳通过螺丝(螺丝客户需要根据情况自备)与底座固定。探针要尽量远离金属，尤其不要被金属物体包裹，尽量装在天花板上以防止被踢掉。

2 云平台

默认情况下探针上报数据到云平台，我司给客户开通云平台账号，客户就可以登录后进行配置或者查看数据，探针也可以上报到客户的服务器，但需要客户自己技术人员开发一套后台系统，或者购买我司后台系统为客户私有搭建。

本系统按建筑物，楼层，区域的结构来划分实际的线下场所，比如建筑物可以是一幢楼，楼层可以是1楼，区域可以是1楼的一个房间或者一个指定的区域。一个楼层下可以有1个或者多个区域，在区域内会安装探针等硬件设备。对应的英文分别为，建筑物（building），楼层(floor)，区域（shop），探针(sniffer)。

2.1 登录

登录地址如下，登录用户名一般为购买者的淘宝账号名，中文部分以拼音代替，密码为 234567，推荐使用 chrome 浏览器，其他浏览器可能有异常或者乱码。

http://be.zhiyince.com/dataserver/web/be/login

2.2 配置

2.2.1 建筑物

点击左下角的配置即可进入配置页面



探针发货前会默认为客户创建一个建筑物，一个楼层，一个区域，探针会创建在该区域下，客户不需要再添加探针



点击添加可以添加新的建筑物, 点击编辑可以编辑建筑物信息, 点击查看在下方会展示该建筑物下的楼层, 如果做了定位算法调整可以点击清缓来清除缓存。

名称：可以任意取;排序：当有多个建筑物时，在其他页面下拉列表显示的顺序由排序控制，数字越小越靠上;只显示监控人员：当为 1 时在实时定位页面只显示在监控页面录入的人员，如果为 0 则显示所有探测到的 mac 地址；上班时间，下班时间为监控页面用于考勤的上下班时间；最小工作小时：在考勤算法中，只有一个人持续被探测时间累积超过最小工作小时才满足要求。

2.2.2 楼层

点击每行建筑物右边的绿色查看按钮，会在楼层区域展示该建筑物下的所有楼层。

楼层

点击右侧每一行楼层对应的绿色查看按钮来展开下方的区域

ID	名称	楼层x长(米)	楼层y长(米)	图片路径	状态
	<input type="text" value="1楼"/>	<input type="text" value="10"/>	<input type="text" value="10"/>		<button>添加</button>
101	1楼	7.67	10.33	floor_101.jpg	1 <button>编辑</button> <button>删除</button> <button>查看</button> <button>上传</button>

名称：任意取；楼层 x 长和 y 长：楼层的长和宽，如果我们把楼层的平面图放在电脑屏幕查看，屏幕的横向对应楼层的 x 长，纵向对应楼层的 y 长，单位为米，精确到小数点两位；点击上传按钮可以为该楼层上传一张平面图，jpg 格式，大约 300k 大小。

2.2.3 区域

点击每行楼层右边的绿色查看按钮，会在区域位置展示该楼层下的所有区域。

区域

点击右侧每一行区域对应的绿色查看按钮来展开下方的探针

ID	名称	x长(米)	y长(米)	起点x	起点y	热力图	产品触碰	定位类型	全部存储	标签信号标准	标签信号次数	A参数	N参数	投票探针数	进入信号强度	图片路径	场景类型	临近信号强度	状态
	<input type="text" value="中心区域"/>	<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="75"/>	<input type="text" value="2"/>	<input type="text" value="52"/>	<input type="text" value="24"/>	<input type="text" value="1"/>	<input type="text" value="62"/>		<input type="text" value="2"/>	<input type="text" value="80"/>	<button>添加</button>
132	中心区域	7.67	10.33	0	0	1	0	6	0	75	3	52	24	3	70	shop_132.jpg	2	85	1 <button>编辑</button> <button>删除</button> <button>查看</button> <button>上传</button>

起点 x,起点 y 是相对于所在楼层，该区域原点也就是左上角的坐标，一个楼层内有多个区域，每个区域所处的位置由这两个值来定义；热力图：是否使用热力图；产品触碰：是否使用产品触碰（仅对 t4 类型探针有效）；全部存储：是否全部存储原始数据，需要管理员才能设置；标签信号标准等参数在算法章节阐述，场景类型：1 为商业场景,2 为办公场景，影响系统如何识别工作人员的算法，同样可以上传一张该区域的平面图。

2.2.4 探针

点击每行区域右边的绿色查看按钮，会在探针位置展示该区域下的所有探针。

探针

ID	mac	区域	类型	名称	nparam	aparam	x	y	探针 rssi 过滤值	到期日	在线	状态	
			1		23	46	2.1	3.6	99			添加	
372	5ccf7fd4	3	132	1	t1	24	52	6.07	8.63	99	2017-01-31	在 1 线	编辑 查看

mac:探针的 mac 地址，为 12 字节，其后 6 字节对应探针外壳上的 6 个字母;x,y 为探针在所属区域的坐标；区域在楼层内，一个区域在楼层的什么位置通过配置页面来定义，也就是区域的起点(左上角)x 和 y，这时用楼层的坐标系,但定义探针在区域的位置,那就要用区域的坐标系,比如区域左上角在楼层的(10,4)位置，而一个探针正好在该区域的左上角，那么探针的坐标是 (0,0),探针 rssi 过滤值:探测到的信号如果弱于该信号则信号被抛弃，数值越大越弱;nparam,aparam:暂时无用;查看：点击查看可以看到探针原始数据页面，该页面的数据是探针报上来的所有数据，未经过滤，主要是调试探针状态时使用。

2.2.5 算法

算法主要配置在区域上,分为探针算法和标签算法两个方面。一个被探测的信号如何经过探针到达云平台，并最终被汇总呈现呢？大致过程是，首先探针会实时扫描空气中所有 wifi 信号，信号会很多，有手机，电脑的也有其他 ap 和无线设备的，探针根据自身配置情况（可以查看高级配置来配置探针自身参数）对信号进行第一层次的过滤，比如是否抛弃苹果的随机地址，是否抛弃路由器的 mac 地址，是否抛弃信号特别弱的等，经过第一层次的筛选，探针周期性（1 秒到几分钟）的通过网络把数据发送到服务器，服务器首先按探针在服务器上配置的强度（探针 rssi 过滤率值）过滤，弱于这个信号的就抛弃，然后根据探测的 mac 地址查看是否属于移动终端，如果不属于也会抛弃，然后根据探针所属区域上的配置属性进行进店判断和位置判断等。

2.2.5.1 探针算法

针对探测的信息，我们首先判断探测到的设备是否进入了探针所在的区域，如果判定为进入，进而会判定其位置，整个判定过程 10 秒钟重复一次。

区域

点击右侧每一行区域对应的绿色查看按钮来展开下方的探针

ID	名称	x长 (米)	y长 (米)	起点x	起点y	热力图	产品触碰	定位类型	全部存储	标签信号标准	标签信号次数	A参数	N参数	投票探针数	进入信号强度	图片路径	场景类型	临近信号强度	状态
	中	10	10	0	0	1	0	1	0	75	2	52	24	1	62		2	80	添加
132	中心区域	7.67	10.33	0	0	1	0	6	0	75	3	52	24	3	70	shop_132.jpg	2	85	1 编辑 删除 查看 上传

A 参数：为设备距离探针 1 米处的信号强度，一般不用改变；N 参数：为衰减因子，一般也不用改变；临近信号强度：探针能探测的距离空旷半径超过 100 多米，因此我们需要抛弃一部分数据，否则探测到很多无意义的数数据，如果信号弱于该值（绝对值大于该值），则不算临近，直接抛弃掉，在计算进店率时，临近的数目算作分母，进入的数目算作分子；数值与距离大致对应关系如下，但因为信号的波动，数值只能作为参考。

强度	距离(米)	强度	距离(米)	强度	距离(米)	强度	距离(米)
56	1.47	67	4.22	78	12.12	89	34.81
57	1.62	68	4.64	79	13.34	90	38.31
58	1.78	69	5.11	80	14.68	91	42.17
59	1.96	70	5.62	81	16.16	92	46.42
60	2.15	71	6.19	82	17.78	93	51.09
61	2.37	72	6.81	83	19.57	94	56.23
62	2.61	73	7.50	84	21.54	95	61.90
63	2.87	74	8.25	85	23.71	96	68.13
64	3.16	75	9.09	86	26.10	97	74.99
65	3.48	76	10.00	87	28.73	98	82.54
66	3.83	77	11.01	88	31.62	99	90.85

同样，进入信号强度是指信号强于这个值才算进店；投票探针数：是指必须有多少个探针投票认为设备进入了区域才算真的进入区域。

定位类型

如果探针能够准确的放在四个点上，建议采用平均算法定位，如果不能恰好放在四个点上，可以采用三角定位算法。

- 1:当指定的探针都满足强度要求或者三角定位坐标在指定区域算作进入区域，之后通过三角定位计算坐标
解释：算法首先判断是否有满足投票探针数的探针都采集到了强于进入信号的信号，如果有，就算进店了，如果没有，再根据三角定位判断是否进店，如果进入了就算进入了，之后通过三角定位算法跟踪设备的位置。
- 2:至少一个探针满足强度要求算作进入区域，之后通过三角定位和平均算法计算坐标
解释：该算法忽略投票探针数，只要一个探针认可进店，即算进店，之后通过两个算法跟踪定位。
- 3:至少一个探针满足强度要求算作进入区域，之后通过平均算法计算坐标
- 4:当指定的探针都满足强度要求或者三角定位坐标在指定区域算作进入区域，之后通过三角定位和平均算法计算坐标
- 5:当指定的探针都满足强度要求算作进入区域，之后通过三角定位计算坐标
- 6:当指定的探针都满足强度要求算作进入区域，之后通过平均算法计算坐标
- 7:当指定的探针都满足强度要求算作进入区域，之后通过平均算法计算坐标,可以 1 个探针
解释：平均算法一般要求 3 个以上探针，该类型允许一个探针计算位置，虽然误差较大。

2.2.5.2 标签算法

C1 型(触碰标签)

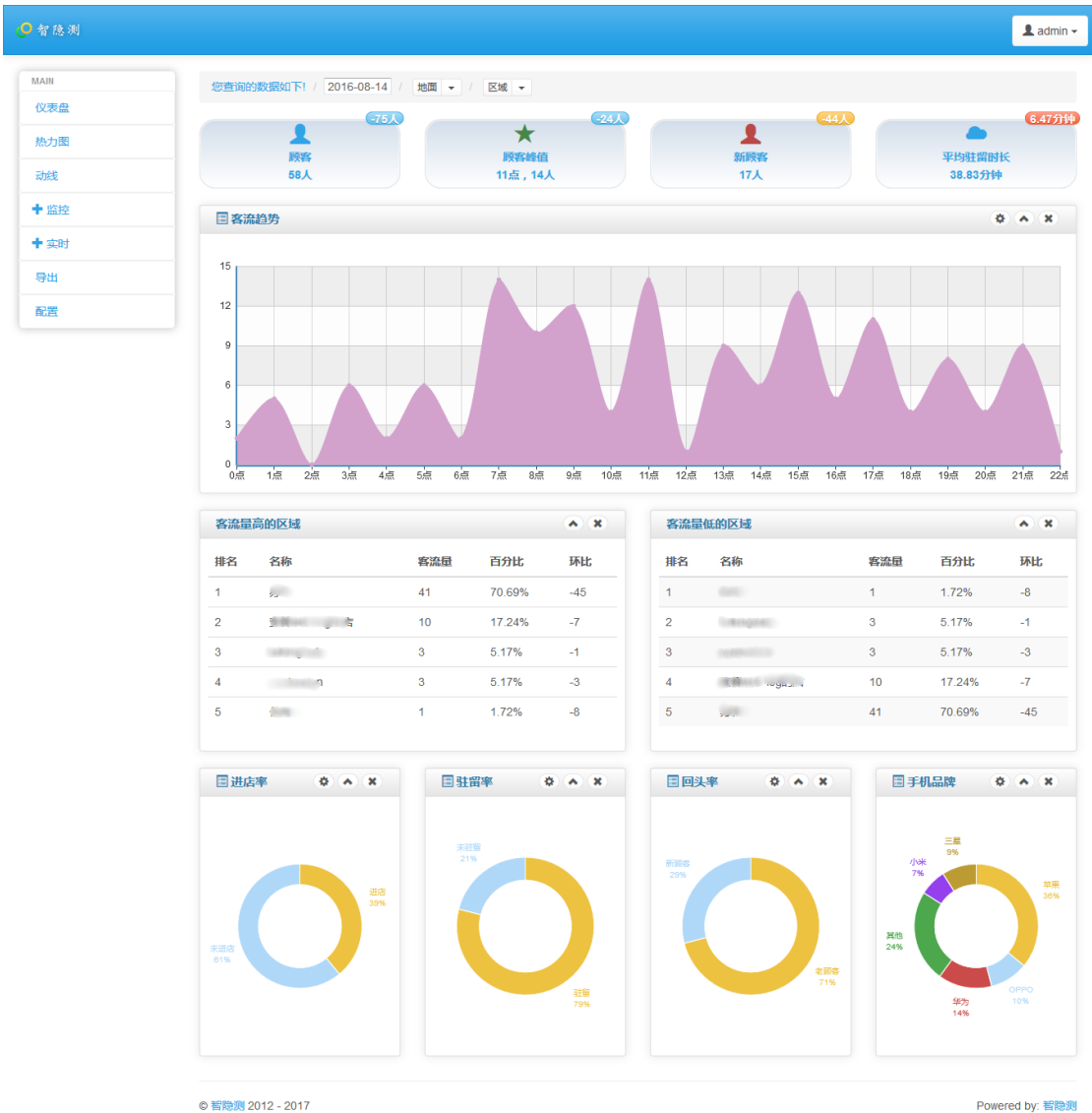
仅使用标签信号次数参数，是指在一个计算周期，一般为 40 秒，系统检测到 C1 型标签信号次数大于等于该值即表示该标签被触碰一次，值越大，标签越迟钝。

C2 型(蓝牙标签)

当在检测到的信号强于标签信号标准值，并且次数大于等于标签信号次数，则判定该标签接近了该探针。

2.3 仪表盘

仪表盘反应了总体信息，属于汇总信息，汇总信息一般延后约 30 分钟显示。



1. 日期选择，在页面上方，点击日期后可以根据日历选择开始日期，页面下方数据会随之刷新，数据范围为所选择的日期至今的数据。
2. 楼层选择，日期选择框的右边下拉列表可以选择不同楼层，选择楼层后下方数据会刷新为该楼层的数据。
3. 区域选择，楼层选择右面是区域选择下拉列表，选择区域后下方数据会刷新为该区域的数据。
4. 顾客，是指所选择的条件内检测到的不同 mac 地址的数据，也就是不同的顾客数目，如果一个顾客一天内光顾多次，系统也会智能区分，只会记录为一次，而且记录的进入时刻是当天第一次探测到的时刻。

5. 顾客峰值，一个时间区段内还有高峰和低谷，比如可能 9 点左右针对办公室来说，上班人员进来多，就是峰值，峰值反应了区段内那些时间段顾客最多。根据所选范围的时间长度，系统智能区分按小时，天还是月来估算峰值，比如一天内可能说某个小时是峰值，一百天可能说某个月是峰值。
6. 新顾客，是指该选择条件内首次采集到的顾客数目。
7. 平均驻留时长，驻留时长是指顾客离开时间减去进来时间的差值，平均是指多人的平均值，如果一个顾客一天内来多次，时长是最后一次的时间减去第一次的时间。

以上内容右上方的+-变化量反应了环比情况，比如选择的是今天的数据，那么其变化量是和昨天相比。

8. 客流趋势，反应一个时间段内不同时间点的顾客量，系统自动按条件显示按小时，天和月的趋势。
9. 客流量高的区域，如果一幢楼分为不同楼层，一个楼层分为不同区域，那么不同区域的人流不会相同，客流量高的区域列表是按所选条件内，客流量来进行的排名。区域占比是指该区域的人流除以所选条件的总人流，如果区域的占比相加超过 100%也是正常的，因为所选区域总人流经过了去重。比如一个楼层有两个区域 A 和 B,10 个人都去过 A 和 B 区域，条件选择查看该楼层，那么楼层总人流是 10,A 区域也是 10,B 区域也是 10，因此每个区域的占比都是 100%
10. 客流量低的区域，与客流量高的区域相反。
11. 进店率，无线信号的发射能够识别的距离很远，也能隔墙接收，因此能够探测到未进店的人群，进店率的分母就是探测到的总人数，分子是通过算法识别判定为进店的人数。实际中反应了人流与进店的关系。如何判定临近和进店，看算法章节。
12. 驻留率，一个顾客可能进店后立刻就离开了，也可能逛了很长时间离开，驻留率的分母就是所有判定为进店的人数，分子是指超过一定停留时间的人数。目前是超过 30 秒算驻留。
13. 回头率，是指老顾客的比率，如果一个顾客在所选区间内之前的时间曾经被探测到过，那么本次就是回头客。
14. 手机品牌比例，是指顾客所持有的智能设备的品牌比例。

2.4 热力图

2.4.1 多探针热力图

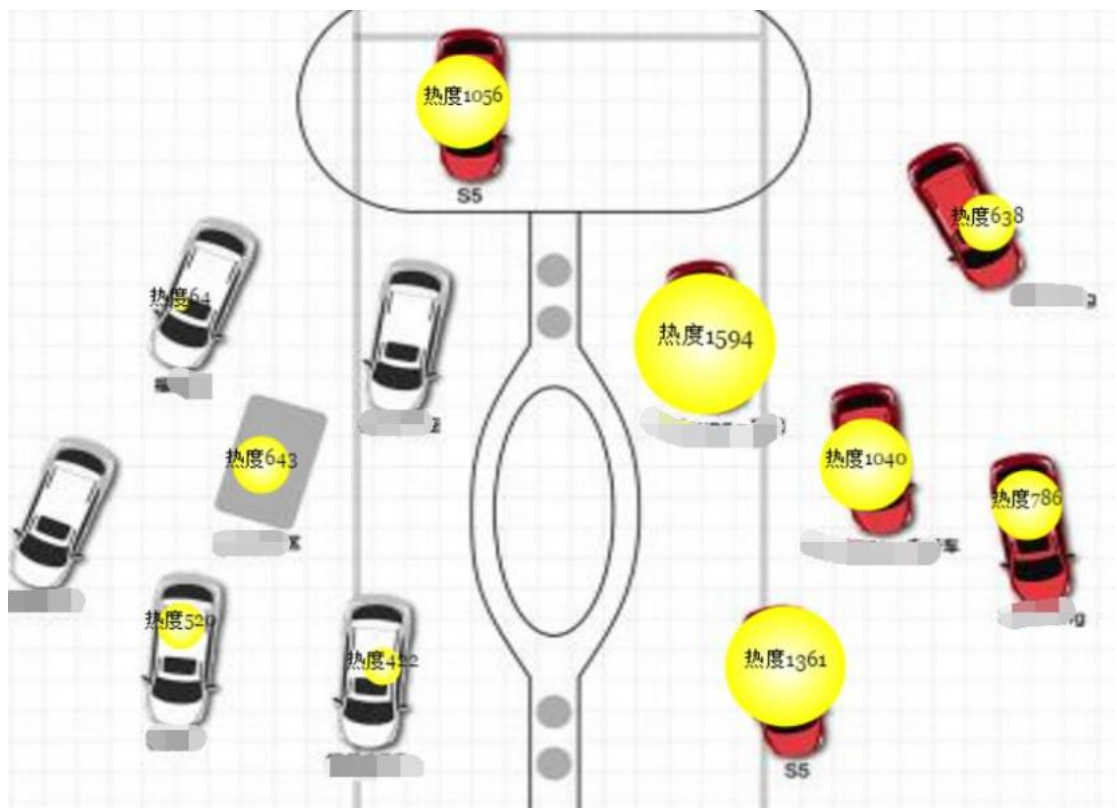
如果你的区域大于 100 平米，并且希望展示较为漂亮的热力图，效果如下，但需要探针做密集网状部署，至少 4 个探针。如果区域面积超过 200 平米，那么需要更多探针，按照探针间距 10-15 米部署，



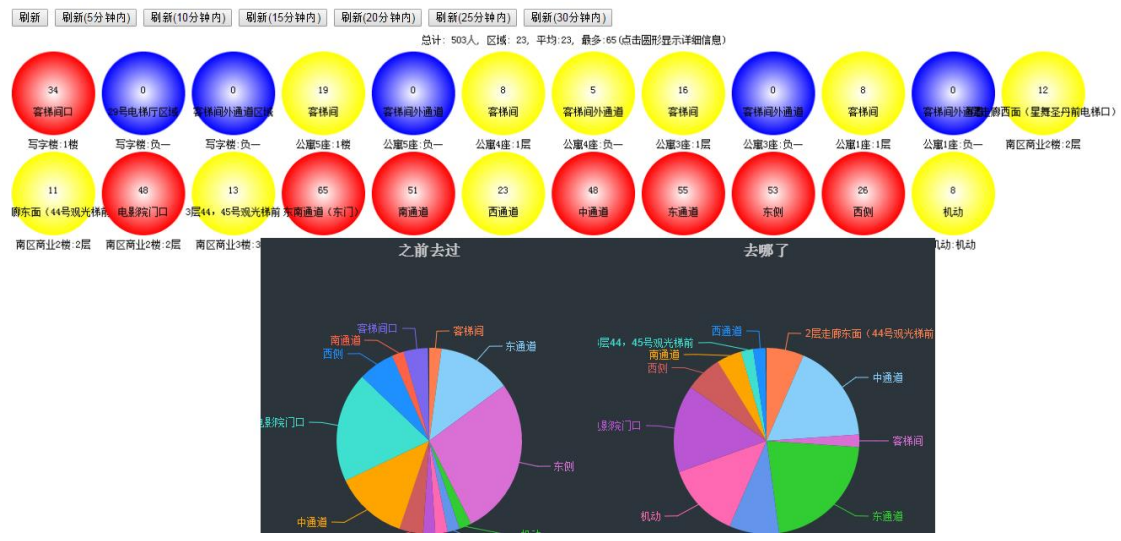
在一个区域内，一定时间积累后，某些区域光顾的人会多于其他区域，光顾的人越多的区域，其颜色越发红色，因此热力图反应了一个区域内人流的密集度，越密集的地方越发红色同样，可以选择时间，楼层和区域分别进行查看。这里的图片是在配置页面上传的jpg文件。

2.4.2 单探针热力图

如下为整个楼层，每个车的位置代表一个区域，每个区域中间放置一个探针，通过调节区域的进入和邻近信号强度控制探针探测的范围(同时要在配置页面仔细调节每个区域的大小以及起点坐标),黄色圆形的大小表示该位置的热度，热度值不等于人数或者人次，而死探测信号的次数，所以数值较大。



2.5 实时分布



该页面实时显示人流在各个区域的分布，不能显示历史数据。红色表示人多，黄色次之，蓝色人少。之前去过和去哪了都比较容易理解，不过多阐述。

2.6 MAC 轨迹

MAIN

- 仪表盘
- 热力图
- 实时分布
- MAC轨迹**
- + 监控
- + 实时定位
- 导出
- 配置

MAC

0016...4541

操作

查询

MAC到过区域

MAC	品牌	首次探测	进入	最后探测	建筑	楼层	区域	日期
0016603c4541	酷派	17:19:15		17:23:41	公寓5座	1楼	客梯间	2017-01-04
0016603c4541	酷派	17:18:02		17:18:02	南区商业2楼	2层	2层走廊东面（44号观光梯前）	2017-01-04
0016603c4541	酷派	17:17:50		17:17:50	内街	1层	南通道	2017-01-04
0016603c4541	酷派	17:17:36	17:17:36	17:18:08	南区商业2楼	2层	2层走廊西面（星舞圣丹前电梯口）	2017-01-04

输入 12 个字母数字的 mac 地址，查询该 mac 地址曾经去过的区域，可以查询历史数据。

2.7 监控

监控主要用于考勤，人员状态查询等目的。

2.7.1 人员详细

探针能够探测到的人员很多，但出于监控等目的，管理者不需要查看所有人的信息，比如公司的人事只关注员工的考勤而不关注来往客户的信息，因此通过该页面可以创建需要关注的人。

人员详细

人员汇总

导出

配置

人员管理

姓名

手机

MAC

是否排除统计

操作

0

添加

人员监控[导出] [点击全屏显示]

姓名	MAC	排除统计	首次探测时间	最后探测时间	持续时长	迟到	早退	满足最低时长	长外出(工作时间)	估计位置	操作
刘	780218	0									<div>编辑删除查看</div>
常	1c914c	0	01-07 12:17	01-07 23:11	10.90小时	迟到	正常	正常	正常	办公室(1.00米,1.00米)	<div>编辑删除查看</div>
霍	689671	0	01-07 15:02	01-07 18:02	3.00小时	迟到	早退	不满足	正常	办公室(1.00米,1.00米)	<div>编辑删除查看</div>
江	9cf387	0									<div>编辑删除查看</div>
梁	0c3021	0									<div>编辑删除查看</div>

智慧测

31到期

MAIN

仪表盘

监控

人员详细

人员汇总

导出

配置

开始2016-12-27

结束2017-01-05

李

在上面选择日期后,数据刷新为所选日期至今的数据,鼠标停留在旷工,迟到等天数上显示明细。长外出天数,是指有多少天,该员工曾经离开较长时间,累计时长没有扣除曾经较长时间离开时长。

人员监控[导出]

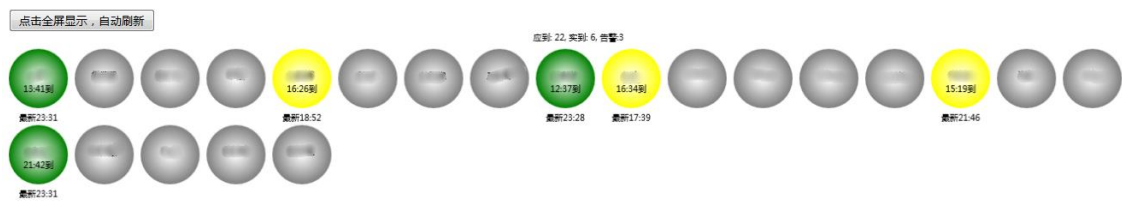
姓名	MAC	工作日数	实到天数	旷工天数	迟到天数	早退天数	长外出天数	累计时长
刘	780218	8	7	1	7	1	6	48.43小时
常	9cf387	8	4	4	1	1	3	30.12小时
霍	689671	8	7	1	0	0	7	64.25小时
江	1c914c	8	7	1	0	0	7	67.53小时
梁	0c3021	8	5	3	2	0	3	36.22小时
黄	5cf7e6	8	3	5	2	0	3	27.00小时
王	649abc	8	7	1	0	0	1	62.93小时
张	68dbc2	8	9	1	2	1	7	71.55小时
李	68dbc2	8	7	2	4	0	6	58.10小时
于	c0eefb	8	7	1	3	0	5	52.83小时
刘	9c99ac	8	3	5	0	0	0	27.72小时
姚	8863df	8	7	1	0	1	6	55.95小时
孙	dc415f	8	4	4	0	0	2	40.30小时
李	683e3	8	5	3	4	1	3	53.52小时
石	b4effa	8	8	0	7	1	3	57.83小时
李	148fcf	8	7	1	0	1	7	60.62小时
孙	20a2e	8	2	6	0	0	2	16.33小时
王	244b8	8	7	1	1	0	4	64.85小时
赵	48e9f	8	7	1	2	1	7	62.47小时
王	ec358	8	7	1	0	0	4	76.28小时
许	0cd6b	8	7	1	0	0	7	70.93小时
石	5c969	8	7	1	2	0	5	56.08小时
朴	f8279c	8	0	8	0	0	0	0.00小时
李	a85b7c	8	0	8	0	0	0	0.00小时
曹	78d75f	8	7	1	0	0	0	57.25小时

© 智慧测 2012 - 2017

Powered by: 智慧测

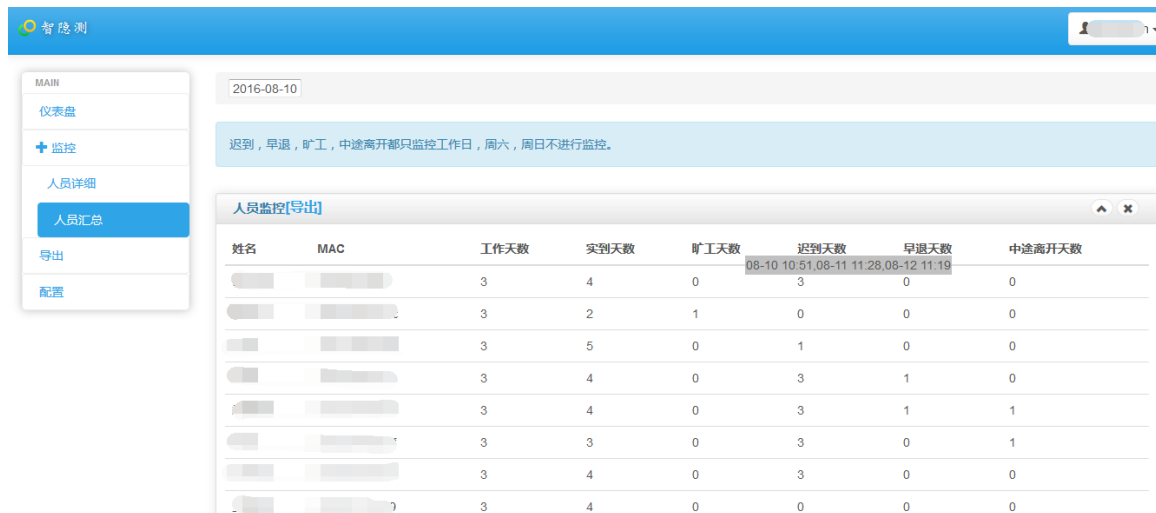
- 日期,通过上方的日期选择可以查看不同日期的数据。为单天数据。
- 是否排除统计,1表示虽然监控此人但人流报表不计算该人的次数,主要用于排除营业场所的员工。
- 姓名,手机,MAC,输入三个内容,点击右面的添加即可增加对该人的管理,添加实时生效。

- 首次探测时间，是指在选定日期内，该人被首次探测到的时间，比如上学或者上班的时间。
- 最后探测时间，是指在选定的日期内，该人被最后一次探测到的时间，比如放学或者下班。
- 持续时长，是指最后探测到的时间减去首次探测到的时间。
- 迟到，是指首次探测时间晚于系统设置的上班时间（在配置内进行配置上班时间）。
- 早退，是指最后探测时间早于系统设置的下班时间，如果是当天，并且时间还没到下班时间，系统会进行区分，不标注为迟到。
- 满最低时长，是指最后探测时间减去最早探测时间大于系统配置的最低工作时间。
- 中途离开，当探针在 50 分钟内不能探测到某一个人的信号时，系统会判定该人离开，因此一天内累计的离开时间总和即是中途离开时间，比如员工在工作时间内外出。
- 估计位置，是指员工当前被判定的位置。如果只安装了一台探针，那么位置即是该探针的位置，如果安装了多台探针，系统会根据算法自动判定该人所处的位置。
- 编辑，可以修改该人的信息。
- 删除，删除该人的信息。
- 查看，查看该人探测到的具体信息。
- 导出，信息可以导出为 excel 格式，方便二次加工。
- 点击全屏显示，显示内容如下，绿颜色表示最近极短时间内探测到过该人信号，灰色表示选定日期内未探测到该人信号，黄色表示虽然选定日期内探测到该人信号，但最近几十分钟内没有再次探测到信号。



2.7.2 人员汇总

人员汇总反应了一段时间内的数据。



- 日期选择, 选择日期后, 表格数据会更新为该日期至今的数据。
- 工作日说明, 系统只针对工作日进行汇总, 比如周六, 周日虽然探测不到, 但不判定为旷工。
- 工作天数, 是指所选时间段内非周六, 周日的天数。
- 实到天数, 是指探测到该人的天数, 比如该人周六加班, 那么实到天数会超过工作天数。
- 旷工, 迟到, 早退, 中途离开天数容易理解。鼠标停留在天数上会显示具体哪些天的什么时间旷工, 迟到, 早退和中途离开。
- 导出, 可以导出为 excel 格式的数据。

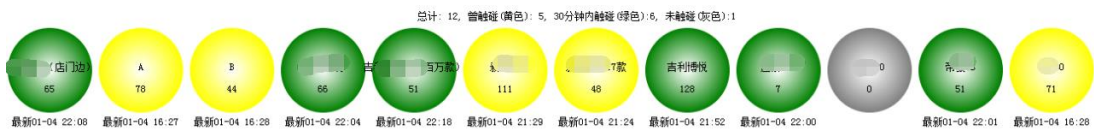
2.7.3 物品

物品反应了商品触碰标签的数据情况。



- 日期选择, 通过日历选择日期, 日期范围为所选日期全天。
- 名称, 编码, 可以通过该输入框新建商品标签, 名称为客户给标签取的名称, 编码为标签外壳上贴的标签号码。点击添加即可添加, 添加实时生效。

- 强制使用探针，当标签周围有很多探针时，系统会计算，那个探针累计的数据最多，就自动选择那个探针的数据，如果希望使用某一个特定探针，可以指定探针，这样系统就固定用那个探针的数据，探针需要为12个字母，如果取消指定，只要编辑，清空后保存即可。
- 首次探测时间，是指该标签选定日期内第一次被触碰的时间。
- 最近探测时间，是指该标签选定日期内最后一次被触碰的时间。
- 触碰次数，是指该标签选定日期内被触碰的总次数，标签无论两秒内被触碰摇晃多少次，标签只会发射2次信号，服务器接收到信号后会暂存起来，每40秒（可修改）进行一次检测，如果该周期内识别到至少3次(可修改，参见算法部分)信号，系统会认为标签被有效触碰1次。
- 估计位置，该标签通过系统定位算法判定的位置。
- 编辑，及时生效。
- 点击全屏显示，显示内容如下

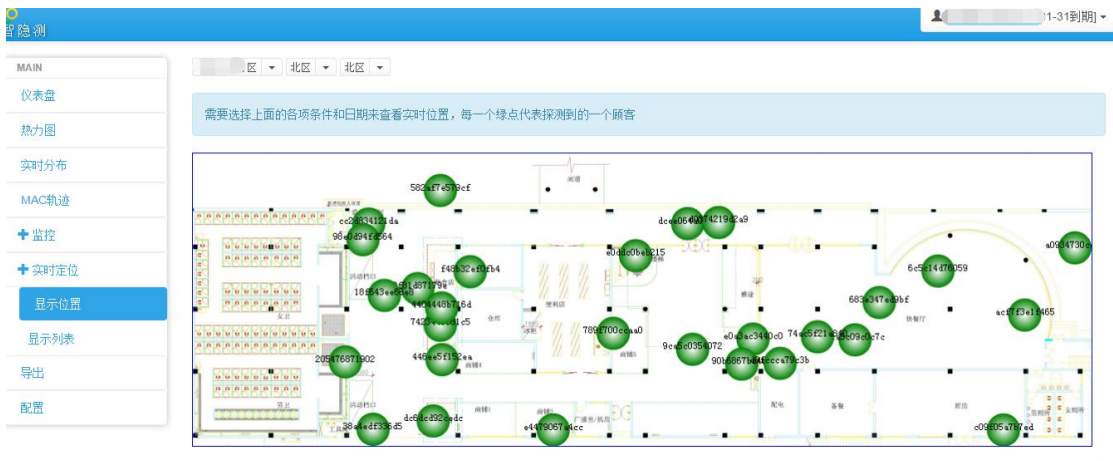


选定日期内没有触碰记录的为灰色，30分钟内有触碰记录的为绿色，有触碰记录，但30分钟内没有被触碰的为黄色。

2.8 实时定位

实时数据为当前数据情况，探针约一秒上报一次数据，系统约20秒判定一次位置信息，因此数据相对较为实时。定位主要依赖于t1,t4探针，t2,t3探针不适合实时定位。需要每个区域安装至少4台探针才能准确显示实时定位。

2.8.1 显示位置



- 每一个绿色的圆形代表此时此刻定位到的一个设备。
- 楼层，区域，选择后系统反应选定楼层和区域的信息。

- 只显示监控人员，在配置中，配置建筑物时如果选择了只显示监控人员，这里就只显示输入的被监控人员，否则会显示所有探测到的人员。
- 消失时间，如果该人 15 分钟内曾被定位在该区域内，该人会以绿色球显示出来，如果曾经配置过该人的信息则显示名称，如果没有配置过则显示其 MAC 地址。

2.8.2 显示列表

实时信息以列表形式显示出来

The screenshot shows the '智隐测' (Zhi Yin Ce) web interface. On the left is a sidebar with navigation options: MAIN, 仪表盘 (Dashboard), 热力图 (Heatmap), 实时分布 (Real-time Distribution), MAC轨迹 (MAC Trajectory), + 监控 (Monitor), + 实时定位 (Real-time Location), 显示位置 (Show Location), 显示列表 (Show List), 导出 (Export), and 配置 (Configure). The main content area has a header with filters for '北区' (North District) and a message: '需要选择上面的各项条件和日期来查看实时列表。' (Need to select the above conditions and dates to view the real-time list). Below this is a table titled '人员实时列表' (Real-time Personnel List).

MAC	品牌	首次探测时间	最后探测时间	估计位置
6044f1762	三星	01-04 22:06	01-04 22:15	(未成功定位)
f42b9ad	华为	01-04 22:00	01-04 22:25	(未成功定位)
f43a51d	三星	01-04 22:05	01-04 22:12	(未成功定位)
f43dbe86	苹果	01-04 22:16	01-04 22:24	(未成功定位)
446c152ea	华为	01-04 21:59	01-04 22:13	(26.34米, 21.82米)
dceet0931	华为	01-04 22:19	01-04 22:26	(52.31米, 7.42米)
e0dc3215	vivo	01-04 21:35	01-04 22:26	(48.69米, 1.64米)
6c53059	OPPO	01-04 21:54	01-04 22:26	(84.15米, 9.37米)

未成功定位: 根据算法不同, 有时我们虽然判定了一个设备进入了区域, 但没有足够的数据对该设备定位, 因此会显示为: 未成功定位。

2.9 导出

The screenshot shows the '智隐测' (Zhi Yin Ce) web interface. On the left is a sidebar with navigation options: MAIN, 仪表盘 (Dashboard), 热力图 (Heatmap), 实时分布 (Real-time Distribution), MAC轨迹 (MAC Trajectory), + 监控 (Monitor), + 实时定位 (Real-time Location), 显示位置 (Show Location), 显示列表 (Show List), 导出 (Export), and 配置 (Configure). The main content area has a header with filters for '2017-01-04', '全部' (All), '楼层' (Floor), and '区域' (Area). Below this is a message: '首先从上面选择日期, 楼层和区域, 然后点击下面对应的导出按钮, 系统会导出从选定日期至今的数据。' (First select the date, floor, and area from above, then click the corresponding export button below, the system will export data from the selected date to the present). Below this is a table with export options.

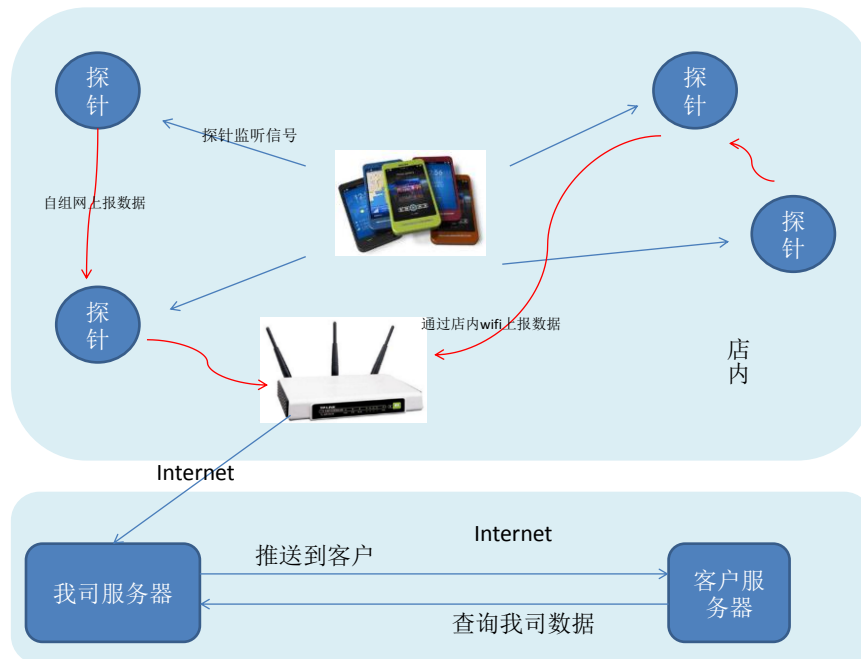
导出类型	说明	导出按钮
进店数据(不去重)	当单个或者多个探针(因部署方式不同, 算法不同), 判定该mac地址为进店时, 数据会被存储下来, 数据存储延时约30分钟左右, 就是说当前判定的数据要在30分钟后才能导出。数据不去重。	导出
进店数据(去重)	当单个或者多个探针(因部署方式不同, 算法不同), 判定该mac地址为进店时, 数据会被存储下来, 数据存储延时约30分钟左右, 就是说当前判定的数据要在30分钟后才能导出。数据去重。	导出
临近数据(不去重)	当单个或者多个探针(因部署方式不同, 算法不同), 判定该mac地址为临近时, 临近的顾客包括哪些进店的以及哪些在店附近, 但没有进店的顾客, 数据会被存储下来, 数据存储延时约30分钟左右, 就是说当前判定的数据要在30分钟后才能导出。数据不去重。	导出
临近数据(去重)	当单个或者多个探针(因部署方式不同, 算法不同), 判定该mac地址为临近时, 临近的顾客包括哪些进店的以及哪些在店附近, 但没有进店的顾客, 数据会被存储下来, 数据存储延时约30分钟左右, 就是说当前判定的数据要在30分钟后才能导出。数据去重。	导出
探针状态	导出探针在建筑, 楼层和区域的分布以及运行状态。	导出

© 智隐测 2012 - 2017 Powered by: 智隐测

导出页面的描述已经很清晰, 可以根据情况使用, 因为数据量很大, 一次只能最多导出一天的数据。

3 转发接口

3.1 数据流



探针探测店内客户手机 mac 地址和 rssi，探针通过无线的方式连接店内 wifi 并上报数据到我司转发服务器，转发服务器负责推送信息到客户服务器，以及响应客户服务器查询。

Q:为什么要转发？

A:原始数据处理异常复杂，大致包含区分包类型，虚假地址过滤，对应到手机品牌，过滤掉非手机设备 mac 地址，检测是否是 ap 地址，数据清洗，定位处理。而且数据并发量非常大处理不好系统容易宕机，我司员工都是来自 BAT 知名企业，善于设计大数据量和并发的系统。经过处理转发后的数据非常简单，便于客户处理。因此我们推荐这种方式。

Q:如果一定坚持，是否可以直接上报到客户服务器？

A:可以的，只要配置探针就可以更改上报的服务器 ip 和端口，目前支持 http 以及 socket 方式。但上报的数据为底层数据，具体格式可以和店主咨询。

3.2 访问控制

依据不同接口对性能的影响，系统会对接口访问频次进行控制，最高访问频次从几秒到几分钟，如果超过限制，系统返回的 status 字段会提示错误信息。

3.3 术语说明

探针的安装是符合客观安装环境，系统内先要创建大楼，然后大楼下面创建楼层，楼层下面创建区域，区域下面创建探针。大楼就是 building,楼层就是 floor,区域就是 shop，探针就是 sta/sniffer,通过登录系统后台，点击配置可以看到自己账户下的 building,floor,shop 和 sta 的 id

仪表盘

热力图

动线

+ 监控

+ 实时

探针管理

导出

配置

建筑物

重要, 重要, 点击右侧每一行建筑物对应的绿色查看按钮来展开下方的楼层, 然后楼层下面是区域, 区域下面是探针, 一层层从上到下点开, 停留在文字上面会提示

ID	名称	只显示监控人员(0/1)	上班时间(09:00:00)	下班时间(18:00:00)	最少工作小时	状态(0/1)	操作
12	新漕河泾大ss厦	0	09:00:00	18:00:00	9	1	编辑 删除 查看 清除

楼层

点击右侧每一行楼层对应的绿色查看按钮来展开下方的区域

ID	名称	宽(对应x,米)	高(对应y,米)	图片路径	状态
11	12楼	4.5	3.7		1

区域

点击右侧每一行区域对应的绿色查看按钮来展开下方的探针

ID	名称	宽(对应x,米)	高(对应y,米)	x	y	热力图	产品触碰	定位类型	全部存储	标签信号标准	标签信号次数	A参数	N参数	投票探针数	mac信号标准	图片路径	场景类型	足够临近标准	状态
21	test1	4.5	3.7	3	4	1	1	1	1	75	2	41	24	1	62	2	80	1	编辑 删除 查看 上传

探针

ID	mac	区域	类型	名称	nparam	aparam	x	y	rssifilter	到期日	在线	状态
294	5c:5d:24:1b:25-21	24	1	ibc	23	46	2.1	3.6	99	2017-01-31	不在线	1

3.4 如何通过工具验证下接口

可以通过 <http://coolaf.com/> 提交数据, 这是一个发送 post 请求的在线工具

接口:

<http://139.196.107.200:9090/api/view/customerlist?accessToken=2...&loginname=houpuxuan>

例: <http://coolaf.com/m?&a=xx&b=xx> get参数直接加在url后就行。用户本地web版本提供下载, 用户可以访问本地域名, 下载链接

post参数:

`{"shop":"77","date":"20161215","start":"0"}`

参数例: `a=b&c=d&f=e` get和head方式除外

隐藏高级功能

cookie的格式: key=value;key2=value2;key3=value3,可以如图所示获取: 图示

header:

Content-Type:application/json ← 设置为json

header的格式: key:value的格式,一行一条。如图: 图示

POST ← 设置为post

UTF-8 --接口输出的

提交

生成文档

Response Headers

```
{
  "data": [
    {
      "firstfound": "20161215135655",
      "closeenough": "20161215135656",
      "lastfound": "20161215135656",
      "mac": "e8bb...262cd"
    },
    {
```

数据

3.5 协议

以下都是通过 http 协议的 json 格式推送和接收,不能用其他非 json 的 encode 方法。

3.6 推送类

推送类是指客户预先设定一些条件,我司把数据发送给客户的服务器,因此,接收端在客户方实现。另外,如果客户服务器出现故障,我司转发服务器无法成功发送,转发会自动停止,但 20 分钟后转发服务器会再次检查是否网络通畅,如果通畅会恢复转发。

报文头格式举例,

POST /xxx/filter/microprobe/upMacRecord HTTP/1.1.

Host: xxx.yyy.com.

User-Agent: Go-http-client/1.1.

Content-Length: 488.

Content-Type: application/json;charset=utf-8.

Accept-Encoding: gzip.

.

{json content}

那么服务器会把数据推送到客户服务器的什么地址呢,可以参看下面的config/urlinsert urlupdate等接口,就是首先客户要调用上述接口配置用于接收我司服务器推送的信息的url,url为两个,url内容不需要带token和用户名,因为是我司调用客户的url,一个是alldata,用于接收所有较为原始的数据,也就是下述的"接收全部数据 mac探针信息","接收全部数据 接收全部商品触碰信息",另外一个approacharea,用于接收经过我司

服务器进行定位处理的数据,也就是” 当顾客临近监测区域推送”,两个url可以一样,也可以不一样。alldata类型的信息工作原理为,探针设备上报数据到我司服务器,服务器接收到后立刻处理,然后立刻转发到客户服务器,也就是其频率由设备的上报频率决定。approacharea是服务器周期性的对收到的所有数据进行定位,发现某些mac地址被定位后落入了监测的区域,服务器会把这些mac地址以及坐标推送给客户服务器,因此其周期由我司服务器决定,大约几十秒。

3.6.1 接收全部数据 mac 探针信息(wifi 上报方式)

描述	我司系统转发所有经过整理的探测数据给客户服务器,客户服务器需要实现约定的接口。
举例	<code>{"sta":"5ccf7f07ce5a","time":"20160604123043","type":"probe","data":[{"brand":"14200","mac":"28c2dd2918d1","rssi":["32","30"]},{ "brand":"18452","mac":"a4d1d25cdf4e","rssi":["93","93"]}]}</code>
协议	http post
time	因为 wifi 上报方式频率很高,可以认为上报时间就是采集时间
brand	被探测的 mac 所属设备的品牌 id, 数值类型, id 与品牌对应关系可以向我司索取。如果 brand 为-1,说明是属于被判定为垃圾包,比如路由器的包或其他非移动设备的包,可以忽略即可。
type	值为 probe, 固定字母,接收端可以根据 type 区分接收到了什么数据包。
mac	被探测设备的 mac 地址。
rssi	被探测设备的信号强度,这里是 rssi 的绝对值, 越小越强(绝对值越小表示被探测设备距离探针越近)。在一个周期内针对一个 mac 地址可以采集到很多数据包,每一个包有一个 rssi,而 rssi 因外界干扰会跳动,因此我们采取传递一个 rssi 数组的方式,这样可以把多个包的 rssi 都传递到应用层,探针默认最多针对一个 mac 传递 5 个 rssi,至于这些 rssi 应用如何使用和算法相关,最简单的是取平均, wifi 定位的难度就是化解 rssi 跳跃问题,因此需要好好思考算法来取得稳定的值。一个周期内的很多数据包,探针会选择 rssi 绝对值最小的两个来上报。如果周期设置为 1 秒,那么不同周期上报的数据可以用来衡量被探测设备大致移动趋势。
sta	上报数据的探针 mac 地址。
备注	需要预先配置接收 URL, 具体参考配置类接口。

3.6.2 接收全部数据 mac 探针信息(gprs 上报方式)

描述	我司系统转发所有经过整理的探测数据给客户服务器,客户服务器需要实现约定的接口。
举例	<code>{"csq":"31","data":[{"lat":"312429866","lon":"1212088181","mac":"74ac5f25d76a","rssi":"94","time":"20161111201859"},{"lat":"312429783","lon":"1212088170","mac":"c09467ce80a6","rssi":"91","time":"20161111201718"}],"sta":"5ccf7fd20ea9","time":"20161111202013","type":"gprs"}</code>
协议	http post
type	值为 gprs, 固定字母,接收端可以根据 type 区分接收到了什么数据包。
mac	被探测设备的 mac 地址。
rssi	被探测设备的信号强度,这里是 rssi 的绝对值, 越小越强。

lat,lon	nema 格式经纬度，如果不存在表示探针没有选配 gps 模块，如果存在但值为 0,表示信号不好，还未接收到卫星信息。经纬度格式转换可以参考云盘中经纬度转换文档。lat=纬度，lon=经度
sta	上报数据的探针 mac 地址。
time(外层)	数据接收时间
time(内层)	相应 mac 地址的采集时间
备注	需要预先配置接收 URL，具体参考配置类接口。

3.6.3 接收全部数据 接收全部商品触碰信息

描述	我司系统转发所有经过整理的探测数据给客户服务器，客户服务器需要实现约定的接口。
举例	<code>{"data":["1a3d895a","1a3d885b"],"sta":"18fe34eeec6d0","time":"20160615192836","type":"tag433"}</code>
协议	http post
data	前六位表示商品标签的 id,后两位表示探测到标签的信号强度。都是 16 进制的表示方式。
type	tag433 表示为触碰标签，tagble 表示为蓝牙标签，接收端可以根据 type 区分接收到了什么数据包。
sta	上报数据的探针 mac 地址。
备注	需要预先配置接收 URL，具体参考配置类接口。

3.6.4 当顾客临近监测区域推送

描述	根据预定义监测区域(监测区域通过 config/approacharea*等接口进行增删改查)，我司在数据符合条件时推送数据，“监测区域”是指在我们系统结构中(结构为：建筑物包含多个楼层，楼层包含多个区域)区域内的一块，一个区域可以划分为几块，也可以一个区域只划分为一块，相当于监测区域和区域相同，那等同于接收区域内所有定位到的数据。比如一个楼层是 100*100 米，你在该楼层建立了一个 100*100 的区域，如果你也设定了 100*100 米的监测区域，那意味着服务器定位到 mac 地址落入该区域，就推送信息给客户服务器，相当于客户服务器接收到了所有在该楼层被定位的 mac 信息以及坐标。当然监测区域也可以只为区域的部分，比如你在区域内选定了 5*20 米作为观测区域，那么只有被定位在这个区域的 mac 地址才会被推送。
举例	<code>{"status":"ok","type":"approacharea","data":[{"area":5,"mac":{"mac":"0c1dafced653","x":"2","y":"4","brand":"123","firstfoundtime":"20160615192836","closeenoughtime":"20160615192836","comeintime":"20160615192836","lastfoundtime":"20160615192836"}}]}</code>
协议	http post
shop	客户所定义的部署探针的场所 id。
mac	被探测设备的 mac 地址。
area	客户所定义的监测区域 area id。
x,y	被探测设备的位置 x,y 坐标
几个时间	格式都为 yyyyMMddHHmmss,如果不出现说明该时间还不存在，firstfoundtime 手册

	探测到的时间，closeenoughtime 首次探测的信号强于邻近信号的时间，comeintime 首次系统判定该 mac 进入了区域（shop 对应的区域）的时间，lastfoundtime 最近一次探测到的时间。
备注	1) 监测区域需要预先配置，具体参考配置类接口。 2)推送延迟大约 10 秒左右。

3.7 查询类

客户根据需要查询我司服务器，我司服务器返回数据。接口在我司实现。

3.7.1 view/heatmap 查询区域热度

描述	返回指定日期，指定场馆区域热度情况，返回内容为某个区域被探测到的人次。如果某些手机设备被探测到频率高于特定情况，其次数会被降权，因为我们不能因为一个手机发射信号比其他活跃就认为热度升高。
输入	http://be.zhiyince.com:9090/api/view/heatmap?accesstoken=234567&loginname=xxx { "shop": "2", "date": "20160227" }
输出	{ "max": 11, "name": "北区", "status": "ok", "roomwidth": 95, "roomheight": 31, "data": [{ "times": "1", "x": "53", "y": "1" }, { "times": "4", "x": "15", "y": "23" }, { "times": "1", "x": "82", "y": "2" }] }
协议	http post
shop	客户所定义的部署探针的场所 id。
date	指定的日期
name	shop 对应的名称
roomwidth, roomheight	shop 的宽度和长度，实际单位，因此绘制在屏幕时要做变换
x, y, times	在 (x, y) 坐标对应的 1 米方格子内被定位的人次数为 times。x, y 为真实环境的单位，因此在屏幕上绘制时要进行变更转换。
max	所有记录中 times 的最大值
备注	1) 需要预先配置，具体参考配置类接口。 2) accesstoken 为系统登录密码, loginname 为系统登录名 3) 如果出现错误返回 { "status": "具体错误信息" }

3.7.2 view/stastatus 查询探针状态

描述	返回探针的状态信息，可选输入建筑物，楼层和区域 id, 如果都不输入返回账号下所有探针，如果只输入建筑物，返回该建筑物下所有探针，如果只输入楼层，返回该楼层下所有探针等等。
输入	http://be.zhiyince.com:9090/api/view/stastatus?accesstoken=asf2321a&loginname=xxx { "type": "status", "building": "118", "floor": "124", "shop": "154" }
输出	{ "data": [{ "timea": "20170411024101", "shop": "154", "floor": "124", "mac": "5ccf7f26eec2", "building": "118" }, { "status": "ok" }] }
协议	http post

输入 type	固定传输字符串 status
输入 shop	客户所定义的部署探针的区域 id, 可选输入
输入 floor	客户所定义的部署探针的楼层 id, 可选输入
输入 building	客户所定义的部署探针的楼层 id, 可选输入
输出 building,floor,shop	探针所属的建筑物, 楼层和区域的 id
输出 mac,timea,timeb	该探针最近 a,b 模块上报的时间。如果没有改列数据, 该列不出现。
输出 timeap	探针最近做一遍扫描周围 ap 的时间。如果没有改列数据, 该列不出现。
备注	1)有 20 分钟左右延迟。 2) accesstoken 为系统登录密码,loginname 为系统登录名 3)如果出现错误返回{"status":"具体错误信息"}

3.7.3 view/stacustomer 查询经过探针周围顾客情况

描述	返回指定探针周围的客流情况。
输入	http://be.zhiyince.com:9090/api/view/stacustomer?accesstoken=asf2321a&loginname=xxx { "mac": "5ccf7f094f2f", "seconds": "200" }
输出	{ "far": 10, "near": 10, "status": "ok" }
协议	http post
mac	探针的 mac 地址, 如果输入没指定 sta, 返回 shop 下所有探针数据。
near, far	距离探针较近和较远的被探测到的设备数, near 和 far 可以针对不同探针做定义。
seconds	表示查询最近 seconds 秒内的数据, 如不输入, 返回全部。
备注	1)只能查询当前情况, 不存储历史数据。 2)如果一个设备 30 分钟内没有新的信号发出, 其数据会被移除。 3) accesstoken 为系统登录密码,loginname 为系统登录名 4)如果出现错误返回{"status":"具体错误信息"}

3.7.4 view/customersummary 查询顾客经过, 进入指定场所情况(deprecated)

描述	返回指定场馆, 指定日期经过和进入顾客汇总信息。Deprecated, 不推荐再接入。
输入	http://be.zhiyince.com:9090/api/view/customersummary?accesstoken=asf2321a&loginname=xxx { "shop": "11", "date": "20160601" }
输出	{ "status": "ok", "data": [{ "count": "10", "hour": "0" }, { "count": "20", "hour": "1" }] }
协议	http post
shop	客户所定义的部署探针的场所 id。
count	该小时进店的顾客。小时是 1-24。
备注	1) 可以查询历史信息。 2) 如果查询当前数据, 数据有 30 分钟延迟。 3) accesstoken 为系统登录密码,loginname 为系统登录名 4) 4)如果出现错误返回{"status":"具体错误信息"}

3.7.5 view/shopcurrent 查询实时分布情况(统计数量)

描述	返回每个区域的实时情况，对应页面功能菜单：实时分布。
输入	http://be.zhiyince.com:9090/api/view/shopcurrent?accesstoken=333321&loginname=xx xx { "type": "shopcurrent", "building": "99", "floor": "98", "shop": "128", "minutes": "5" }
输出	{ "avgnum": 6, "maxnum": 6, "minutes": 5, "shopinfo": [{ "floorid": "98", "buildingname": "写字楼", "shopname": "客梯间口", "num": "6", "width": "10.0", "x": "0.0", "y": "0.0", "shopid": "128", "floorname": "1 楼", "buildingid": "99", "height": "10.0" }], "status": "ok" }
协议	http post
type	固定输入 shopcurrent
输入 building,floor,shop,minu tes	building,floor,shop 都为可选输入，如果不输入则查询所有，minutes 为必选项，为距离当前时间的分钟数，如为 5，表示返回的是最近 5 分钟内探测的数据情况，一般不能大于 30 分钟。
avgnum,maxnum	每个区域的平均数以及最大数。
备注	1) 几乎无延迟 2) accesstoken 为系统登录密码,loginname 为系统登录名 3) 如果出现错误返回 {"status": "具体错误信息"}

3.7.6 view/shopdetail 查询实时分布情况(统计从哪来，去哪了)

描述	返回每个区域内客流从哪里来的到哪里去了，对应实时分布功能。
输入	http://be.zhiyince.com:9090/api/view/shopdetail?accesstoken=345678&loginname=xxxx { "type": "shopdetail", "building": "99", "floor": "98", "shop": "127", "shopidcheck": "128", "minutes": "5" }

输出	<pre>{ "shopto": [{ "num": "1", "name": "2 层走廊西面（星舞圣丹前电梯口）", "id": "163" }], "shopfrom": [{ "num": "7", "name": "客梯间口", "id": "128" }], "status": "ok" }</pre>
协议	http post
type	固定输入 shopdetail
输入 building,floor,shop,minu tes	building,floor,shop 都为可选输入，表示在这个范围内查询来自和去哪的关系，如果不输入则查询所有，minutes 为必选项，为距离当前时间的分钟数，如为 5，表示返回的是最近 5 分钟内探测的数据情况，一般不能大于 30 分钟。
shopidcheck	检查这个区域的人流情况，就是这个区域的人流从哪里来到哪里去。
num	人数。
备注	1) 几乎无延迟 2) accesstoken 为系统登录密码,loginname 为系统登录名 3) 如果出现错误返回{"status":"具体错误信息"}

3.7.7 view/dashboard 仪表盘查询

描述	返回几乎与我司 web 系统仪表盘内容一样丰富的内容，用户可以方便的构造自己的仪表盘
输入	http://bc.zhiyince.com:9090/api/view/dashboard?accesstoken=asf2321a&loginname=xx x {"start":"20170104","end":"20170104","shop":"1","floor":"2","building":"3"}
输出	<pre>{ "status": "ok" //状态信息 "allcustomer": 2556, //进店数目 "allcustomerdiff": -8450, //环比变化量，比如查的区间是今天，数目为 40, 昨天数据是 100，那么就是环比减少 60 "newcustomer": 409, //新顾客，就是在所选时间内，第一次出现的顾客 "newcustomerdiff": -3103, "top": "8 点, 995 人", //高峰人数，系统根据所选的时间范围自动判断高峰是按小时，天还是月，详细见输入的描述解释。 }</pre>

	<pre>"topdiff": -69, "stay": "23.00",//停留的分钟数 "staydiff": "-88.76", "longstayrate": "0.30",//超过 30 秒以上算驻留，驻留除以进店的比例 "oldcustomerrate": "0.84",//老客户占比 "newrate": "0.16",//新客户占比 "inrate": "0.25",//进店率=进店/(进店+经过) "bottomshop": [//排名靠后的区域，最大显示 10 个 { "num": 6, "numdiff": "-37", "name": "公寓 3 座-负一-客梯间外通道", "shopid": 153, "percent": "0.23%"//该区域进店的人数除以所选范围的总人数，一个 范围可能包含多个区域，一个顾客可能逛多个区域，因此范围总人数会进行去重， 比如 a 区域 1 人，b 区域 1 人，人是同样的那个人，那么整个楼层就是 1 人而不是 两人，所占比例 a 区域就是 100%,b 区域也是 100%,加起来超过 100% }, { "num": 8, "numdiff": "-25", "name": "公寓 1 座-负一-客梯间外通道", "shopid": 155, "percent": "0.31%" }], "trend": [//趋势，可能按小时，天或者月，具体看输入的解释。 { "num": "215", "time": "0 点" }, { "num": "90", "time": "1 点" }], "buildinglist": [{ "name": "南区外走廊", "id": "130" }, { "name": "南区商业 2 楼", "id": "126"</pre>
--	--

	<pre> }], "topshop": [{ "num": 516, "numdiff": "-870", "name": "写字楼-1 楼-客梯间口", "shopid": 128, "percent": "20.19%" }, { "num": 365, "numdiff": "-1171", "name": "南区外走廊-1 层-东侧", "shopid": 172, "percent": "14.28%" }], "brand": { // 品牌占比 "苹果": "0.27", "vivo ": "0.09", "OPPO": "0.10", "酷派": "0.03", "华为": "0.14", "魅族": "0.05", "其他": "0.07", "小米": "0.19", "三星": "0.05" } } </pre>
协议	http post
start,end,shop,floor,building	<p>start 为开始日期，end 为结束日期，start 不能大于 end,start 和 end 最大相差不能超过 90 天,start 如和 end 相等，趋势按小时显示，如果相差小于 62 小时，按天来显示，如果超过，则按月来显示；shop,floor,building 为对应的结构为配置页面相应实体的 ID 值，这三个为可选输入，如果不输入，则返回该账号下所有实体的信息。如果想查询 shop,就要对应的 shop,floor,building 都要输入，如果查询 floor 就要连同对应的 building 也要输入，如果查询全部，就 shop,floor,building 都不输入。</p>
备注	<ol style="list-style-type: none"> 1) 可以查询历史信息。 2) 如果查询当前数据，数据有 30 分钟延迟。 3) accesstoken 为系统登录密码,loginname 为系统登录名 4) 如果出现错误返回{"status":"具体错误信息"}

3.7.8 view/customerlist 查询指定场所顾客明细

描述	返回指定场馆，指定日期经过，进店顾客列表。
输入	http://be.zhiyince.com:9090/api/view/customerlist?accessToken=asf2321a&loginname=xxx 格式 1，查询一天数据 {"shop":"11","date":"20160601","start":"0"} 格式 2，查询一个时间段数据,时间段不超过 24 小时， {"shop":"11","date":"20160601120001","date2":"20160601130001","start":"0"}
输出	{"status":"ok","data":[{"firstfound":"20160601222241","closeenough":"20160601222241","lastfound":"20160601222721","mac":"f8a45fc767e1"}]}
协议	http post
shop	客户所定义的部署探针的场所 id。
date,date2	主要考虑新老版本兼容，这两个参数设计的有点不太友善，见谅。如果只传递 date，那格式一定是 yyyyMMdd，这种情况下系统查询一整天数据。如果同时传递 date 和 date2，其格式一定为 yyyyMMddHHmmss，其中 date 表示开始时间，date2 表示结束时间，具体到秒，相差不能超过 24 小时。
firstfound,closeenough,comein,lastfound	该顾客第一次被探测到时间，距离该店足够近（系统智能定义）的时间，进店时间，最后被探测到时间。firstfound,lastfound,closeenough 一定存在，不够 closeenough 的记录不返回。如果 comein 存在表示进店过，如果不存在表示没有进店过。如果一天内，某一个顾客进出多次，返回多条记录。
start	最多返回 500 条，读取不完可以分页，按 order by closeenough 排序。
备注	1) 可以查询历史信息。 2) 如果查询当前数据，数据有 30 分钟延迟。 3) accessToken 为系统登录密码,loginname 为系统登录名 4) 如果出现错误返回 {"status":"具体错误信息"}

3.7.9 view/customer 查询 mac 轨迹情况

描述	返回指定顾客历史经过，进入指定区域情况，只返回大约最近的 50 多条数据。
输入	http://be.zhiyince.com:9090/api/view/customer?accessToken=asf2321a&loginname=xxx {"building":"116","floor":"121","shop":"151","startdate":"20170429110919","enddate":"20170429110920","mac":"289afa1b6c1a"}
输出	{"data":[{"floorid":"121","shop":"客梯间外通道","comein":"20170429110919","firstfound":"20170429110919","closeenough":"20170429110919","shopid":"151","floor":"负一","lastfound":"20170429110919","building":"公寓 4 座","buildingid":"116"}],"status":"ok"}
协议	http post
building,floor,shop,start date,enddate	都为可选输入，如果不输入则查询所有，如果设定时间范围，开始和结束时间都要设置，格式为 yyyyMMddHHmmss
mac,firstfound,closeenough,comein,lastfound	该顾客进过和进入该商店的历史记录。分别是该访问的首次探测时间，足够靠近时间，进入时间，最后探测时间。其中 comein 有可能不出现，表示本次只经过，没有进入。其他时间一定出现。按 order by closeenough desc 返回。

备注	4) 可以查询历史信息 5) 几乎无延迟 6) accesstoken 为系统登录密码,loginname 为系统登录名 7) 如果出现错误返回{"status":"具体错误信息"}
----	--

3.7.10 view/customermonitor 查询监控人员信息

描述	可以通过系统页面创建监控人员，本功能用于查询创建的监控人员
输入	http://bc.zhiyince.com:9090/api/view/customermonitor?accesstoken=asf2321a&loginname=xxx {"type":"customermonitor","building":"118","name":"z111","mac":"289afa1b6c1a"}
输出	<pre>{ "data": [{ "name": "z111", "id": "271", "isexcluded": "false", "mac": "289afa1b6c1a", "building": "118" }], "status": "ok" }</pre>
协议	http post
type	固定输入 customermonitor
输入 building,name,mac	都为可选输入，如果不输入则查询所有
输出	Isexcluded 是否排除统计，building 监控人员所属的建筑物
备注	1) 可以查询历史信息 2) 几乎无延迟 3) accesstoken 为系统登录密码,loginname 为系统登录名 4) 如果出现错误返回{"status":"具体错误信息"}

3.7.11 view/current 查询当前指定区域所探测到的顾客数

描述	返回指定区域当前探测到的顾客数
输入	http://bc.zhiyince.com:9090/api/view/current?accesstoken=asf2321a&loginname=xxx {"shop":"11","seconds":"30","comein":"0"}
输出	{"all":4,"status":"ok"}
协议	http post
shop	客户所定义的部署探针的区域 id。
comein	可选，如果为 0 表示只要探测到的 mac 就计数，如果为 1，表示不但需要探测到 mac,也需要系统通过算法计算该 mac 是否的确进入了场所才计数，默认是 0.
seconds	限定的时间范围，如果指定 30 秒，则返回在该场所最近 30 秒内有探测到信号的顾客（去重后）数，如果一个顾客最近一次被探测的时间是 40 秒前，则不被统计

	来。
all	数目
备注	1) 方法经过优化,能够高性能返回当前状况,不包含历史信息,seconds 最大值为 30*60 秒 2) accesstoken 为系统登录密码,loginname 为系统登录名 3) 如果出现错误返回{"status":"具体错误信息"}

3.7.12 view/currentdetail 查询指定条件探测到的所有 mac

描述	根据选定条件返回所探测到的 mac 信息,该信息都较为实时,从服务器内存获取,因此速度很快,最多可获取三十分钟内数据。可选输入建筑物,楼层和区域 id,如果都不输入返回账号下所有探针,如果只输入建筑物,返回该建筑物下所有探针,如果只输入楼层,返回该楼层下所有探针等等。
输入	http://bc.zhiyince.com:9090/api/view/currentdetail?accesstoken=asf2321a&loginname=xxx {"monitortype":"0","building":"118","floor":"124","shop":"154","seconds":"30","comein":"0"}
输出	{"data":[{"shop":"128","list":[{"rssi":"37","closeenoughtime":"20170427205955","firstfoundtime":"20170427205955","mac":"b8e856438040","name":"王三","comeintime":"20170427205955","lastfoundtime":"20170427205955"}]}],"status":"ok"}
协议	http post
输入 monitortype	固定传输字符串 0:全部人员, 1:仅监控的人员, 2: 仅未监控的人员
输入 shop	客户所定义的部署探针的区域 id, 可选输入
输入 floor	客户所定义的部署探针的楼层 id, 可选输入
输入 building	客户所定义的部署探针的楼层 id, 可选输入
rssi	如果一个区域下有多个探针,该 rssi 是最近获取到信号的探针的 rssi,比如 a,b,cd 四个探针都在同一个区域,都获取到该 mac 的信号,但该 rssi 是按时间来看最近的那个探针对应的 rssi
comein	可选,如果为 0 表示只要探测到的 mac 就计数,如果为 1,表示不但需要探测到 mac,也需要系统通过算法计算该 mac 是否的确进入了场所才计数,默认是 0.
seconds	限定的时间范围,如果指定 30 秒,则返回在该场所最近 30 秒内有探测到信号的顾客(去重后)数,如果一个顾客最近一次被探测的时间是 40 秒前,则不被统计进来。
data	数据主体,内容为列表,每一个元素代表一个区域的内容,区域 id 后面是具体 mac 信息,其中如果能够输出 name 则表示该人员是被监控的人员
备注	1) 方法经过优化,能够高性能返回当前状况,不包含历史信息,seconds 最大值为 30*60 秒 2) accesstoken 为系统登录密码,loginname 为系统登录名 3) 如果出现错误返回{"status":"具体错误信息"}

3.8 配置类

3.8.1 config/buildinglist 查询建筑物属性

描述	返回账号下所创建的建筑物信息
输入	http://be.zhiyince.com:9090/api/config/buildinglist?accesstoken=asf2321a&loginname=xxx { "type": "buildinglist", "building": "118" }
输出	{ "data": [{ "name": "公寓 1 座", "id": "118", "order": "6" }], "status": "ok" }
协议	http post
输入 type	固定输入 buildinglist。
输入 building	可选，如果不输入则返回所有建筑物。
order	显示设定的顺序。
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回 { "status": "具体错误信息" }

3.8.2 config/floorlist 查询楼层属性

描述	返回账号下所创建的楼层信息
输入	http://be.zhiyince.com:9090/api/config/floorlist?accesstoken=asf2321a&loginname=xxx { "type": "floorlist", "building": "118", "floor": "124" }
输出	{ "data": [{ "width": "10.0", "name": "1 层", "id": "124", "building": "118", "height": "10.0" }], "status": "ok" }
协议	http post

输入 type	固定输入 floorlist。
输入 building,floor	可选，如果不输入则返回所有。
name,id,building	楼层名称，id 以及所属的建筑物 id。
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回{"status":"具体错误信息"}

3.8.3 config/shoplist2 查询区域属性

描述	返回账号下所创建的区域信息
输入	http://be.zhiyince.com:9090/api/config/shoplist2?accesstoken=asf2321a&loginname=xx x { "type": "shoplist", "building": "99", "floor": "98", "shop": "128" }
输出	{ "data": [{ "positiontype": "single", "width": "10.0", "name": "客梯间口", "id": "128", "locatinginrssi": "68", "floor": "98", "closeenoughrssi": "80", "locatinginnum": "1", "building": "99", "height": "10.0", "leavingthreshold": "40" }], "status": "ok" }
协议	http post
输入 type	固定输入 shoplist。
输入 building,floor,shop	可选，如果不输入则返回所有。
name,id,building	楼层名称，id 以及所属的建筑物 id。
width,height	场所的宽度和高度，界面的呈现从屏幕的左上角出发，横向是 x 轴，纵向是 y 轴，宽度对应 x，高度对应 y。单位为米，精确到两位小数点。
positiontype	可以为 single,multi,mixed。如果探针密度布置为间隔 10 米左右的高密形式，并且数量至少 4 个，推荐选 multi，定位时采用多探针的三角定位。如果布置很稀疏，推荐选 single，以单探针定位。如果一个场地，有些地方密集，有些地方稀疏，可以选 multi，但需要针对每个探针进行定位类型设置。
locatinginrssi	只有探测到的一个顾客手机信号小于这个值(小代表信号强)，才认为这个顾客进店，值越大判定为进店的顾客越多，房间小建议调整小些，房间大墙后建议调整大些
closeenoughrssi	探测到的信号小于(强于)这个值算作临近，会影响进店率

locatinginnum	通过多探针判断一个顾客是否进店，这个数字表示至少多少个探针投票认为顾客进店才判定为进店，数字越大判定进店的顾客越少
leavingthreshold	多少分钟探测不到某个 mac 信号，认为该 mac 已经离开
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回{"status":"具体错误信息"}

3.8.4 config/shoplist 查询区域属性(deprecated,推荐使用 shoplist2)

描述	配置场馆的参数，包括面积，定位算法等。
输入	http://be.zhiyince.com:9090/api/config/shoplist?accesstoken=asf2321a&loginname=xxx { "shop": "123" }
输出	{ "positiontype": "single", "width": "10.0", "name": "客梯间口", "id": "128", "locatinginrssi": "68", "floor": "98", "closeenoughrssi": "80", "locatinginnum": "1", "building": "99", "height": "10.0", "leavingthreshold": "40", "status": "ok" }
协议	http post
shop	客户所定义的部署探针的场所 id。
width,height	场所的宽度和高度，界面的呈现从屏幕的左上角出发，横向是 x 轴，纵向是 y 轴，宽度对应 x，高度对应 y。单位为米，精确到两位小数点。
positiontype	可以为 single,multi,mixed。如果探针密度布置为间隔 10 米左右的高密形式，并且数量至少 4 个，推荐选 multi，定位时采用多探针的三角定位。如果布置很稀疏，推荐选 single，以单探针定位。如果一个场地，有些地方密集，有些地方稀疏，可以选 multi，但需要针对每个探针进行定位类型设置。
locatinginrssi	只有探测到的一个顾客手机信号小于这个值(小代表信号强)，才认为这个顾客进店，值越大判定为进店的顾客越多，房间小建议调整小些，房间大墙后建议调整大些
closeenoughrssi	探测到的信号小于(强于)这个值算作临近，会影响进店率
locatinginnum	通过多探针判断一个顾客是否进店，这个数字表示至少多少个探针投票认为顾客进店才判定为进店，数字越大判定进店的顾客越少
leavingthreshold	多少分钟探测不到某个 mac 信号，认为该 mac 已经离开
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回{"status":"具体错误信息"}

3.8.5 config/shopupdate 配置区域属性

描述	配置场馆的参数，包括面积，定位算法等。
输入	http://be.zhiyince.com:9090/api/config/shopupdate?accesstoken=asf2321a&loginname=xxx {"shop":"123","positiontype":"multi","width":"4.5","locatinginrssi":"62","closeenoughrssi":"80","locatinginnum":"1","height":"3.7","leavingthreshold":"40"}
输出	{"status":"ok"}
协议	http post
shop	客户所定义的部署探针的场所 id。
width,height	场所的宽度和高度，界面的呈现从屏幕的左上角出发，横向是 x 轴，纵向是 y 轴，宽度对应 x，高度对应 y。单位为米，精确到两位小数点。
locatinginrssi	只有探测到的一个顾客手机信号小于这个值(小代表信号强)，才认为这个顾客进店，值越大判定为进店的顾客越多，房间小建议调整小些，房间大墙后建议调整大些
closeenoughrssi	探测到的信号小于(强于)这个值算作临近，会影响进店率
locatinginnum	通过多探针判断一个顾客是否进店，这个数字表示至少多少个探针投票认为顾客进店才判定为进店，数字越大判定进店的顾客越少
leavingthreshold	多少分钟探测不到某个 mac 信号，认为该 mac 已经离开
positiontype	可以为 single,multi,mixed。如果探针密度布置为间隔 10 米左右的高密形式，并且数量至少 4 个，推荐选 multi，定位时采用多探针的三角定位。如果布置很稀疏，推荐选 single，以单探针定位。如果一个场地，有些地方密集，有些地方稀疏，可以选 multi，但需要针对每个探针进行定位类型设置。
备注	3) accesstoken 为系统登录密码,loginname 为系统登录名 4) 如果出现错误返回 {"status":"具体错误信息"}

3.8.6 config/stalist 查询探针属性

描述	查询探针参数，包括坐标，算法，技术指标和所属场馆参数等。
输入	http://be.zhiyince.com:9090/api/config/stalist?accesstoken=asf2321a&loginname=xxx {"type":"stalist","building":"118","floor":"124","shop":"154","mac":"5ccf7f26ecc2"}
输出	{"data":[{"name":"zddd","x":"5.0","y":"5.0","id":"617","mac":"5ccf7f26ecc2"}],"status":"ok"}
协议	http post
type	输入固定字符 stalist
building,floor,shop,mac	都是可选输入，如果输入则返回该实体下的探针信息，如果都不输入，返回该账号下的所有探针。
x,y	探针的坐标，界面的呈现从屏幕的左上角出发，横向是 x 轴，纵向是 y 轴，宽度对应 x，高度对应 y。单位为米，精确到两位小数点。
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回 {"status":"具体错误信息"}

3.8.7 config/staupdate 更新探针属性

描述	配置探针参数，包括坐标，算法，技术指标和所属场馆参数等。
输入	http://be.zhiyince.com:9090/api/config/staupdate?accessToken=asf2321a&loginname=xxx { "mac": "5ccf7f07ce5a", "x": "10.11", "y": "20.22" }
输出	{ "status": "ok" }
协议	http post
mac	需要配置的探针的 mac 地址。
x,y	探针的坐标，界面的呈现从屏幕的左上角出发，横向是 x 轴，纵向是 y 轴，宽度对应 x，高度对应 y。单位为米，精确到两位小数点。
备注	1) accessToken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回{"status": "具体错误信息"}

3.8.8 config/urllist 查询转发 url

描述	查询客户配置的接收数据的 url。
输入	http://be.zhiyince.com:9090/api/config/urllist? accessToken=asf2321a&loginname=xxx { "type": "all" }
输出	{ "status": "ok", "alldata": "http://139.196.107.200/forward1", "approacharea": "http://139.196.107.200/forward2" }
协议	http post
alldata	用于接收所有数据的 url。
approacharea	用于接收进去监测区域的 url。
备注	1) accessToken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回{"status": "具体错误信息"}

3.8.9 config/urlinsert 插入转发 url

描述	新增客户接收数据的 url。
输入	http://be.zhiyince.com:9090/api/config/urlinsert?accessToken=asf2321a&loginname=xxx { "alldata": "http://139.196.107.200/forward1", "approacharea": "http://139.196.107.200/forward2" }
输出	{ "status": "ok" }
协议	http post
alldata	用于接收所有数据的 url，数据量非常大，慎用。如果设置了之后想清除，可以调用 urldel,然后再次插入。设置后大约 2 小时后缓存过期生效。
approacharea	用于接收进去监测区域的 url，设置后立刻生效，但需要再设置监测区域才能有真正的数据推送。如果设置了之后想清除，可以调用 urldel,然后再次插入。
备注	1) accessToken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回{"status": "具体错误信息"}

3.8.10 config/urlupdate 更新转发 url

描述	配置客户接收数据的 url。
输入	http://be.zhiyince.com:9090/api/config/urlupdate?accessToken=asf2321a&loginname=xxx { "alldata": "http://139.196.107.200/forward1", "approacharea": "http://139.196.107.200/forward2" }
输出	{ "status": "ok" }
协议	http post
alldata	用于接收所有数据的 url。
approacharea	用于接收进去监测区域的 url。
备注	1) accessToken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回 { "status": "具体错误信息" }

3.8.11 config/urldel 删除转发 url

描述	删除客户配置的接收数据的 url。
输入	http://be.zhiyince.com:9090/api/config/urldel? accessToken=asf2321a&loginname=xxx { "type": "all" }
输出	{ "status": "ok" }
协议	http post
备注	1) accessToken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回 { "status": "具体错误信息" }

3.8.12 config/approacharealist 查询监测区域

描述	查询配置的监测区域。
输入	http://be.zhiyince.com:9090/api/config/approacharealist?accessToken=asf2321a&loginname=xxx { "shop": "123" }
输出	{ "status": "ok", "data": [{ "area": "12345", "name": "一个名字", "xfrom": "1", "xto": "4", "yfrom": "3", "yto": "6" }] }
协议	http post
shop	所配置的 shop id。
xfrom,xto,yfrom,yto	四个点确定的监测区域。
area	返回生成的 area id。这里是监测区域的 id,和区域 (shop) 的 id 不是同一个概念,可以理解为监测区域是区域 (shop) 的一部分。
备注	1) accessToken 为系统登录密码,loginname 为系统登录名 2) xfrom,yfrom,xto,yto 单位都为米,没有小数点。 3) 如果出现错误返回 { "status": "具体错误信息" }

3.8.13 config/approachareainsert 设定特定监测区域

描述	配置监测区域，一旦有被探测的顾客落入该监测区域就触发信息推送。请注意这里的监测区域和上述的区域（shop）是不同的概念，可以理解为监测区域是区域（shop）的一部分。
输入	http://be.zhiyince.com:9090/api/config/approachareainsert?accesstoken=asf2321a&loginname=xxx { "shop": "123", "name": "name", "xfrom": "1", "xto": "1", "yfrom": "3", "yto": "6" }
输出	{ "status": "ok", "area": "12345" }
协议	http post
shop	所配置的 shop id。
name	为数字和字母，不能为中文
xfrom,xto,yfrom,yto	四个点确定的区块,为区域内的绝对值，比如一个 shop/区域的大小是 100*100 米，xfrom=5,xto=8,yfrom=2,yto=7，当在该区域被定位的 mac 坐标为(6,5)因为 6 介于 xfrom 和 xto 之间，5 介于 yfrom,yto 之间，因此被认为落入该监测区域，就会推送信息。
area	返回生成的 area id。
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) xfrom,yfrom,xto,yto 单位都为米，没有小数点。 3) 如果出现错误返回 { "status": "具体错误信息" }

3.8.14 config/approachareadel 删除监测区域

描述	删除配置的监测区域。
输入	http://be.zhiyince.com:9090/api/config/approachareadel?accesstoken=asf2321a&loginname=xxx { "area": "123" }
输出	{ "status": "ok" }
协议	http post
area	area id。这里是监测区域的 id,和区域（shop）的 id 不是同一个概念，可以理解为监测区域是区域（shop）的一部分。
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) 如果出现错误返回 { "status": "具体错误信息" }

3.8.15 config/approachareaunder 更新监测区域

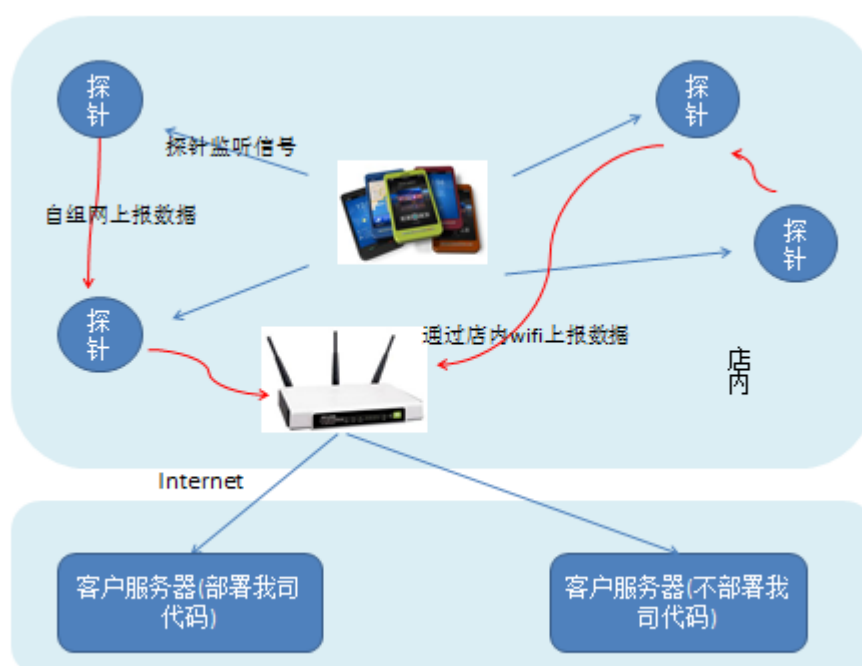
描述	修改配置的监测区域。
输入	http://be.zhiyince.com:9090/api/config/approachareaunder?accesstoken=asf2321a&loginname=xxx { "area": "123456", "name": "name", "xfrom": "1", "xto": "1", "yfrom": "3", "yto": "6" }
输出	{ "status": "ok" }
协议	http post
name	为数字和字母，不能为中文

area	area id。这里是监测区域的 id,和区域（shop）的 id 不是同一个概念，可以理解为监测区域是区域（shop）的一部分。
备注	1) accesstoken 为系统登录密码,loginname 为系统登录名 2) xfrom,yfrom,xto,yto 单位都为米，没有小数点。 3) 如果出现错误返回{"status": "具体错误信息"}

4 直报接口(wifi 上报)

原则上我们建议客户通过转发接口，因为在转发接口内我司已经做了大量的工作，如果顾客坚持探针信息直接报给客户的服务器，数据不经过我司，可以通过如下接口。

4.1 数据流



在直报模式下，仍旧有两种方式可选，一是客户直接处理底层数据，一是客户部署我司提供的 sdk 来处理数据，我司 sdk 能够适用大数据并发处理，提供室内定位算法并且提供丰富的上层接口供使用，能够节省客户大量时间，上层接口的使用同转发接口的说明，本文档重点介绍不部署我司代码时底层接口数据上报的处理方式。

4.2 配置探针

通过我司提供的 app(参考章节“通过安卓 APP 配置(适合批量)”)可以配置探针，详见探针高级配置说明，这里做简要介绍。正常情况下通过 app 配置密码，配置其他参数只要在 app 配置页面的密码栏目输入特殊字符串就可以配置探针，其他和配置密码一样，配置页面的 SSID 内容不变，按如下说明输入，要以*##开头，其实容易理解，这个字符开头探针就认为是在配置参数而不是配置密码。注意这里不是通过电脑或者手机连接探针，然后浏览器打开的页面中输入，是必须通过安卓 app 的配置功能输入。如同配置探针密码一样,只是在密码输入栏里面输入下述内容,也需要断电,插电,待灯亮后点击确认。

##*2139.196.107.200

表示将上报的 ip 地址更改为 139.196.107.200

注：为了性能考虑，暂不支持服务器为域名，如果需要可以讨论。

##*180

表示将上报的端口更改为 80

##*o/data/upload3

表示把 uri 设置为 /data/upload3

##*a

表示恢复出厂设置，任何时候感觉配置乱了，都可以恢复出厂设置。

以上例子也是系统默认配置，表示探针以 http post 协议上报到 http:// 139.196.107.200:80/data/upload3

4.3 协议格式

探针以 http post 方式进行上报，header 具体格式如下

POST /data/upload3 HTTP/1.1\r\nUser-Agent: zyc0.1\r\nHost: 139.196.107.200:80\r\nAccept:

*/\r\nContent-Type: application/x-www-form-urlencoded;charset=utf-8\r\nContent-Length:789\r\nConnection:

keep-alive\r\n\r\n

建议客户的 web 服务器开启 keep-alive,这会提升上报性能。

客户服务器接收到请求后要进行 response，返回 ok 两个字母即可(注意为小写,同时 http response status 为 200).接收到 response 后探针指示灯会闪烁一次,如果没有 response 或者 response 不包含 ok 字符,探针会重发上次数据,数据包的 time 参数值不变,如果长时间不能接收到 response,探针会自动重启。以下数据类型很多,对于不感兴趣的类型只要返回 ok 字母即可,等于抛弃,对于感兴趣的类型进行接收并返回 ok。

4.4 探针说明

我司探针集成了两个 wifi 芯片(定义为 wifi a 与 wifi b)，一个蓝牙芯片（可选），一个商品标签信号接收芯片（可选）。从特性上来说，我们能够接收 probe 包，也能够接收其他非 probe 包，比如其他管理包，控制包以及数据包等。另外，为了更好识别探针周围环境，探针会定期扫描周围 ap 并上传。从监控的角度，可以按不同信道轮询等，具体参看探针高级配置说明。除了 wifi 相关的，可选模块能够发射 ibeacon 信标，也能够接收我司商品标签的信号。

4.5 接口定义

其中 a 接口为必须实现接口,其他接口可以选择实现,需要那种数据类型就实现那种接口即可.探针在启动时会上报一些启动 log,log 不符合如下的约定，因此凡是不符合如下约定的内容都可以忽略。

4.5.1 获取服务器时间

sta=5ccf7f26f057&shop=1&token=123232jd&r=9c216a8416ea&cid=3f53e337&type=unixtime&data=testconnection

探针进行联网模式时要首先获取服务器时间,请求 type 类型为 unixtime,服务器返回的内容要包含如下黄色内容之一

内容之一 ok-timestart-1477628006-timeend

(其中 1477628006 为以秒为单位的 unix 时间戳,返回上述内容探针认为联网成功并正确配置探针的时间,注意这里的时间戳是以秒为单位,如果不知道如何获取可以看看这个页面
<http://tool.chinaz.com/Tools/unixtime.aspx> 这个页面下面有不同编程语言获取时间戳的例子)
内容之二 ok
(仅仅返回 ok 探针仍旧认为联网成功,但不会设置本地时间)

4.5.2 接收 wifi a 信息

sta=18fe34ee6d0&shop=1&token=123232jd&r=8134ad506c4c&id=12a2c334&type=probea&data=2123\18c34fd3dad70\60\61\62\17c34fd3dad70\60\61\62\1ac25

(以上信息接收后如果按字符串直接打印,那些转义字符,就是紫色和红色内容打印出来是乱码的,因为是不可见 ascii 符,可以参考下面的 java 例子来处理)

- “sta” 为上报信息的探针 mac 地址, 12 个字母或数字组合。
- “shop” 是该探针上设置的 shop id 数值, 比如安装在不同店的探针 shop 可以配置不同值, 这样可以根据 shop 进行服务器端负载分发。
- “token” 是探针上设置的密钥, 可以通过 app 进行配置修改。
- “id” 是探针上报的 id 序列, 如果探针没有接收到正确的服务器返回值,探针会重复上传本次数据,重复上传的数据包的 id 相同, 自探针上电开始的计数, 单位为微妙, 该序列不是具体的时间值, 不能作为时间使用, 因为探针一秒上报一次, 因此时间取服务器时间即可, 该时间序列用于表示不同上报的相对时间,类型为 16 进制字符串,8 个字节。每次探针断电后 time 被重置,因此断电前所发送的 time 之间可以比较,不同断电间无比较意义。
- “type” 该上报的类型,probea 表示 wifi a 芯片探测到的 mac 信息。
- “data” 表示上报的具体数据, 每一个 mac 的信息通过数字 1 (\1, 红色部分) 进行分割(注意例子中蓝色部分,由于字符串切割原因,两头可能出现,但大部分情况不会出现正好被切割部分的情况,因此程序要处理,可以直接忽略), 注意这里的数字 1 不是字母“1”, 数字 1 和字母“1” 在 ascii 字码表的数值不同。 \1 表示 ascii 字码表数值为 1 的字码, 因此 \1 代表一个字节, 该字节数值为 1。绿色部分表示探测到的不同 mac 地址, 紫色部分是探测到该 mac 地址时的 rssi 强度, 其中 \60 表示数值为 60 的一个字节的 ascii 字码。因为在一个上报周期内可能探测到同一个 mac 地址的多个包, 每一个包在发射时有一个 rssi, 因此一次上报, 同一个 mac 地址可能会有多个 rssi 值, 具体最多多少个 rssi, 可以通过 app 配置探针。注意 \ 表示的转义字符符合 c 语言格式说明, 例如 \60 只是为了这里说明所写, 其实是一个数值为 60 的不能展示的 ascii 字码, 在比如 java 语言中占用一个 byte, 如果用抓包原件或者直接在接收程序里面按字符串打印, 你会发现有乱码或者奇怪的字符出现, 这是因为传输的有些字节是按 acsii 的值传递, 不能直接打印, 因为是不可见字符或者混乱字符。具体 ascii 字码可以参考文章来理解:

http://baike.baidu.com/link?url=FjYGR97AghZQGLaR6XtrayhHVwlBAehJnOkVIroWBw5kYpzcYQKF7UVKasO8oG3DLZ71YspihWYk8-CIMQOwh_
比如我们用的分隔符, 习惯的写法 \1 来表示数值为 1 的不可见 ascii 码, 就是如下的内容。

标准表

Bin(二进制)	Oct(八进制)	Dec(十进制)	Hex(十六进制)	缩写字符	解释
0000 0000	0	0	00	NUL(null)	空字符
0000 0001	1	1	01	SOH(start of headline)	标题开始

再比如, 如果一个 rssi 值为 69,我们传递到服务器的一个字节是值为 69,一般习惯的转义写法为 \69 也就是下面的 ascii 码, 69 恰好对应可见字符 E,一般在 java 语言等里面通过 ' E ' 定义字符 E, 其 ascii 字码也就是 69,

但我们这里的意图是传递一个数值 69,而不是一个字母 E,但抓包软件或者一般的 printf 都会打印出来 E 这个字母而不是 69 这个字符串。

01000101	105	69	45	E	大写字母E
----------	-----	----	----	---	-------

以 java 语言为例,一般 `byte[] databytes = data.getBytes();`//data 为 String 类型,这里把 string 转为 byte 的数组
然后通过 databytes 的数组就可以引用对应的值,类似 `databytes[5]`,如果其位置恰好是我们要传递的 rssi 的话,那么 `databytes[5]` 的值就是 69,我们也可以把这个值赋值给一个 int 类型,比如 `int rssi= databytes[5]`,也可以这样打印 `System.out.println(“rssi is:” +rssi);`打印结果会是 rssi is:69

rssi 为什么是多个值

被探测设备的信号强度,这里是 rssi 的绝对值,越小越强。在一个周期内针对一个 mac 地址可以采集到很多数据包,每一个包有一个 rssi,而 rssi 因外界干扰会跳动,因此我们采取传递一个 rssi 数组的方式,这样可以把多个包的 rssi 都传递到应用层,探针默认最多针对一个 mac 传递 5 个 rssi,至于这些 rssi 应用如何使用和算法相关,最简单的是取平均,wifi 定位的难度就是化解 rssi 跳跃问题,因此需要好好思考算法来取得稳定的值。

其他参数容易解读,这里给一个解读 data 参数值的代码例子

```
java
String delimiter = "\\1";//分隔符号为数字 1 不是字母 1,因此需要斜线转义
int startp = data.indexOf(delimiter);// 正确的数据以分隔符开始,如果第一个字母不是分隔符,就去掉,从找到的第一个分隔符开始
if(startp!=-1)
{
    // 正确的数据以分隔符结束,如果最后一个字母不是分隔符,就去掉,从最后一个分隔符结束
    int endp = data.lastIndexOf(delimiter);
    if(((endp-startp)-1)>12)// 因为 mac 地址为 12 字母,加上至少一个 rssi,因此长度要大于 12
    {
        String datatrimed = data.substring(startp+1, endp);// 去除掉两边可能的脏数据
        String[] datasplits = datatrimed.split(delimiter);
        for(int i=0;i<datasplits.length;i++)
        {
            if(datasplits[i].length()>12)// 风险防控,正确的数据长度肯定大于 12
            {
                String mac=datasplits[i].substring(0, 12);// 每一段的前 12 个字符为 mac 地址
                //转为 byte 处理,不同语言方法不同
                byte[] datasplitbytes = datasplits[i].getBytes();
                //从第 13 个 byte 开始,每一个 byte 表示该 mac 地址的一个 rssi 值
                for(int j = 12;j<datasplitbytes.length;j++)
                {
                    int rssi=datasplitbytes[j];
                    //正确的 rssi, 进行处理即可,rssi 应该是负数,但我们为了减少传输字节,用的是绝对值,rssi 是一个 9 到 99 的数值
                    if((rssi>9)&&(rssi<100))
                    ;
                }
            }
        }
    }
}
}
```

php


```

$arr = explode("\1",$content['data']);
foreach ($arr as $v) {
    if(strlen($v)<13 || strlen($v)>100){
        continue;
    }
    $macstr = substr($v,0,12);
    if($macstr) {
        $macbytes = get_bytes($v);
        $rssiArr = [];
        for ($i = 12; $i < count($macbytes); $i++) {
            $rssi = $macbytes[$i];
            if($rssi>9 && $rssi<100){
                $rssiArr[] = $rssi;
            }
        }
    }
}

/**
 * 转换一个 String 字符串为 byte 数组
 * @param $str
 * @return array
 */
function get_bytes($str) {
    $len = strlen($str);
    $bytes = array();
    for($i=0;$i<$len;$i++) {
        if(ord($str[$i]) >= 128){
            $byte = ord($str[$i]) - 256;
        }else{
            $byte = ord($str[$i]);
        }
        $bytes[] = $byte ;
    }
    return $bytes;
}

dotnet
char[] macSplitChars = new char[] { (char)0x01, '' };
var macPackets = data.Split(macSplitChars, StringSplitOptions.RemoveEmptyEntries);
foreach (var mac in macPackets)
{
    var macAddr = mac.Substring(0, 12);    // 前 12 个字符为 mac 地址
    for (int i = 12; i < mac.Length; i++)
    {
        int rssi = mac[i];
        if (rssi>9 && rssi<100)    // 有效性检查
        {
            // 用户自定义处理...
        }
    }
}
}

```

“r”黄色部分为探针上报数据时连接的热点的 mac 地址,该项信息视探针的固件而定,有的版本没有黄色部分数据,有的版本有。因此程序接收时要做好黄色不存在的处理逻辑。

4.5.3 接收 wifi b 信息

```
sta=18fe34eee6d0&shop=1&token=123232jd&cr=8134ad506c4c
&id=12a2c334&type=probeb&data=\18c34fd3dad70\60\61\62\17c34fd3dad70\60\61\62\1
```

基本定义同上,只是 type 为 probeb,表示由 wifi b 上报的 mac 探测信息。

4.5.4 接收 flash 存储信息

当探针处于 flash 存储模式与联网模式自动切换模式时,一旦脱离网络,探针会在设备的本地 flash 进行存储,当重新联网时探针会第一时间将本地存储的信息上报,上报完毕后探针会清空本地存储。默认探针设置为不进行本地存储,可以通过配置 `*#*v1` (开启本地存储) 和 `*#*200` (每 10 分钟存储一次,如果是 20 分钟就设置为 200),具体如何配置参考本手册高级配置章节。

```
sta=18fe34eee6d0&shop=1&token=123232jd&cr=8134ad506c4c
&id=0012a2c334&type=flash&data=\18c34fd3dad705a00212c24c6\17c34fd3dad705a00c1232d56\1
```

绿色部分表示 mac 地址,灰色部分表示时间

当探针从服务器未正确接收到时间戳时,探针上报的数据意义如下

5a00212c24c6, 5a(1,2 位)表示 rssi 值, 3,4 位(本例为 00)为 00 表示未获取服务器真实时间,后续以探针计数为时间戳, 212c24c6 表示从探针开机开始(重新上电后数字被重置为 0)到现在累计的数字,以 16 进制打印,在探针中是 4 字节的 uint32,单位为秒,

当探针从服务器正确接收到时间戳时,探针上报的数据意义如下

5aff212c24c6, 5a 表示 rssi 值, ff 表示该数据为正确接收服务器时间后的数据, 212c24c6 为真实时间,为 unix 格式时间戳的 16 进制打印方式。(参考这里了解 unix 时间戳 <http://tool.chinaz.com/Tools/unixtime.aspx>)

4.5.5 接收 ap 扫描信息

```
sta=18fe34eee6d0&shop=1&token=123232jd&cr=8134ad506c4c
&id=12a2c334&type=ap&data=\1282cb2fd2b00\50\56first\19c216a8416ea05\51\57second\150bd5f2df979\51\5717*1f1\150bd5f38db13\51\5717*2f2\1
```

探针会定期扫描上报检测到的 ap 信息,基本格式同上,每一个检测到的 ap 信息通过 \1 进行分割。绿色部分为不同 ap 的 mac 地址,红色部分为分隔符,紫色部分分别为检测到该 ap 的信号 rssi 值和 ap 所在的信道(需要减 50),比如 \50\56 表示该 ap 的信号强度为 50,ap 的工作信道为 (56-50=6)信道 6,蓝色部分为每个 ap 的 ssid,也就是我们通过手机搜索 wifi 时看到的名称。

4.5.6 接收 wifi b 心跳信息

```
sta=18fe34eee6d0&shop=1&token=123232jd&cr=8134ad506c4c
&id=12a2c334&type=check&data=t572007166r0e0c0h31624hb18fe34010150aps1800api5si10em2pl1024pll3072tag20sp1
```

“check” type 表示上报的是心跳包，其 data 内容为探针的内部配置情况，具体等同于探针高级配置中的说明。处理时可以忽略，但需要 response，不然探针会误以为服务器端没有接收成功。

t:当前时间戳
r:上次启动原因
e:上次启动 epc1 寄存器值
c:上次启动原因
h:剩余内存字节数
hb:mesh group id
aps:对应高级配置项:7
api:对应高级配置项:8
si: 对应高级配置项:c
em: 对应高级配置项:d
pl: 对应高级配置项:e
pll: 对应高级配置项:f
tag: 对应高级配置项:n
sp: 对应高级配置项:p

4.5.7 接收 wifi a 配置信息

sta=18fe34eee6d0&shop=1&token=123232jd&r=8134ad506c4e
&id=12a2c334&type=configa&data=ch1cas5cat5sc5t30603970r0e0c0h47368cst36000iga1st1
“configa” type 表示 wifi a 芯片上报的配置信息。

ch:channel 配置项的配置值，具体参见高级配置了解其意义，对应高级配置项:Z
cas:内部设定 cache 缓存数组的大小，对应高级配置项:X
cat:每个 mac 最多存几个 rssi，对应高级配置项:W
sc:信道扫描间隔，具体参见高级配置了解其意义，对应高级配置项:Y。
t:探针当前时间戳
r:上次启动的原因代码
e:上次启动的 epc1 寄存器值
c:上次启动原因
h:剩余内存字节
cst:对应高级配置项:V
iga: 对应高级配置项:U
st: 对应高级配置项:T

4.5.8 接收手机与 mac 地址关联信息

探针只能探测顾客手机的 mac 地址，但我们希望 mac 地址能够与顾客的手机号，姓名等进行关联，当顾客连接我们探针时，我们弹出 portal 页面，让顾客注册信息，该信息会上报到服务器端。

sta=18fe34eee6d0&shop=1&token=123232jd&r=8134ad506c4e
&id=12a2c334&type=register&data=\1name=&phone=13621874566&qq=&email=&sex=1&age=2&mac=8c34fd3dad70\1

数据比较一目了然，每一个注册仍旧以\1进行分割，其中 mac 表示该顾客手机的 mac 地址，phone 是其手机号等。

4.5.9 接收商品标签信息

当店内商品被顾客触碰时，绑定在商品上的商品标签会发出无线信号，探针能够进行接收并进行上报。

sta=18fc34eee6d0&shop=1&token=123232jd&cr=8134ad506c4e &id=12a2c334&type=tag&data=\101000015\1

“tag” type 表示上报的是商品标签内容，其中蓝色部分表示标签的 id,绿色部分表示接收到的 rssi 强度。

4.6 代码例子

如下以 spring mvc 为框架的例子

```
@RequestMapping("/data/upload3")
public @ResponseBody String dataupload3(HttpServletRequest request){
    DataInputStream in=null;
    int len = request.getContentLength();
    byte[] contentbytes = new byte[len];
    String content=null;
    try {
        in = new DataInputStream(request.getInputStream());
        in.readFully(contentbytes);
        content = new String(contentbytes,"utf-8");
    } catch (IOException e) {
        e.printStackTrace();
    } finally
    {
        if(in!=null){
            try {
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

//探针重启是会上报一些日志，这里进行忽略的例子

```
if((content==null) || (content.length()>1600) || (!content.startsWith("sta=")))
    return "ok";

String[] contentsplits = content.split("sta=|&shop=|&time=|&type=|&data=");

String sta = contentsplits[1];
String type = contentsplits[4];
String data = contentsplits[5];

if((sta==null) || (type==null) || (data==null))
    return "ok";

long timelong = System.currentTimeMillis();
```

```

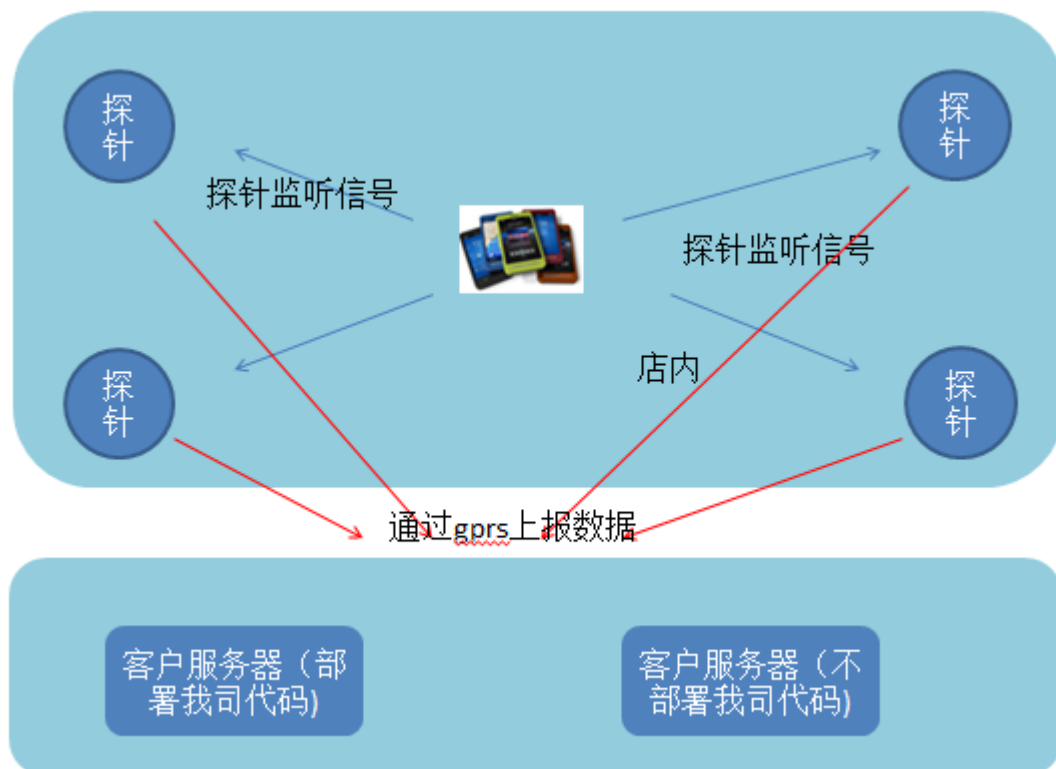
        if(type.equals("ap"))
        {
            //处理ap扫描结果
            return "ok";
        }
        if(type.equals("configa"))
        {
            //处理wifi 上报配置信息
            return "ok";
        }
        if(type.equals("check"))
        {
            //处理wifi b上报心跳信息
            return "ok";
        }
        boolean probeatrue = false;
        if(type.equals("probea"))
        {
            probeatrue = true;
        }
        boolean probebtrue = false;
        if(type.equals("probeb"))
        {
            probebtrue = true;
        }
        if(probeatrue || probebtrue)
        {
            //处理上报探测到的mac信息
            return "ok";
        }
        if(type.equals("tag"))
        {
            //处理上报探测到的商品标签信息
            return "ok";
        }
        if(type.equals("register"))
        {
            //处理上报的mac与个人信息映射
            return "ok";
        }
        return "ok";
    }
}

```

5 直报接口(gprs 上报)

原则上我们建议客户通过转发接口，因为在转发接口内我司已经做了大量的工作，如果顾客坚持探针信息直接报给客户的服务器，数据不经过我司，可以通过如下接口。

5.1 数据流



在直报模式下，仍旧有两种方式可选，一是客户直接处理底层数据，一是客户部署我司提供的 sdk 来处理数据，我司 sdk 能够适用大数据并发处理，提供室内定位算法并且提供丰富的上层接口供使用，能够节省客户大量时间，上层接口的使用同转发接口的说明，本文档重点介绍不部署我司代码时底层接口数据上报的处理方式。

5.2 配置探针

通过我司提供的 app 可以配置探针，详见探针高级配置说明，这里做简要介绍。

如同配置探针密码一样，只是在密码输入栏里面输入下述内容，也需要断电，插电，待灯亮后点击确认。

*#*2139.196.107.200

表示将上报的 ip 地址更改为 139.196.107.200

注：为了性能考虑，暂不支持服务器为域名，如果需要可以讨论。

*#*18000

表示将上报的端口更改为 8000

以上例子也是系统默认配置，表示探针以 tcp 协议上报到 139.196.107.200 的 8000 端口

因为 gprs 通讯是通过移动的服务器发送给上位服务器，因此上位服务器的地址和端口需要可以在公网访问，内部局域网是发不到的。

5.3 探针说明

该型号探针集成了 1 个 wifi 芯片 1 个 gprs 芯片,1 个 gps 芯片(T3 才具备), wifi 芯片用于探测 mac 地址, gprs 用于传输, gps(T3 才具备)芯片用于定位。

5.4 协议格式

5.4.1 上报

探针以 tcp 方式进行上报, 每一个数据包不会超过 2k 字节大小, 协议格式如下

```
sta=[6 bytes sta mac][4 bytes system time][1 byte csq][1 byte 纬度高字节 1byte 经度高字节][6 bytes mac][1 byte  
rssi][1 byte time][3 bytes 纬度低字节 3bytes 经度低字节][6 bytes mac][1 byte rssi][1 byte time][3 bytes 纬度低字  
节 3bytes 经度低字节]\r\n
```

黄色部分为固定字母 sta=, 可以用作判断包的开始位置。

绿色部分为探针的 mac 地址的 6 字节表示方式, mac 地址为 12 字节, 6 字节为压缩表示方式, 一个字节是 8bit, 每 4bit 能够表示一个 16 进制数字, 也就是 12 字节 mac 地址的一个字节。如果用 c 语言还原 mac 地址, 其打印方式为 printf(“%02x%02x%02x%02x%02x%02x”, mac[0], mac[1], mac[2], mac[3], mac[4], mac[5])

紫色部分为探针内部的时间戳, 类型为 uint32, 该时间戳不是时间, 是从探针上电开始计数的时间戳, 单位为微秒, 超过范围溢出后从 0 再次开始。

蓝色部分表示 gprs 的信号强弱, 一般要大于 19 才是比较好的信号, 但信号肯定小于 100, 如果该字节大于等于 100, 说明本包是选配了 gps 模块的设备, 因此真正的信号值还要再减去 100, 如果本身就小于 100, 说明探针没有选用 gps 模块, 也不需要减去 100。上报的第一包数据因为还未取得 csq 值, 因此会等于 0(T2 型号, 不带 GPS)或者 100(T3 型号, 带 GPS)。


亮蓝色部分, 只有选配了 gps 的探针才会出现, 经纬度通过 uint32 的 4 字节可以表示, 但一般每次上报间隔高位字节是不变的, 因为短时间内不可能移动那么远的距离, 因此该信息表示经纬度的高位的一个字节。比如 nema 格式的纬度为 3124.29185, 如果去掉其中的.号, 以 16 进制的数值形式表示这个值为 129F4A81, 那么 12 就是高位字节。


灰色部分表示探测到的 mac 地址, 前 6 个字节是探测到的 mac 地址的 6 字节表示方式, 后面的是 1 个字节的 rssi, 再后面是 1 个字节的系统时间戳, 这里的系统时间戳要解释下, 前述提到探针从上电开始维持一个 uint32(4 字节)的计数时间戳, 单位为微秒, 因此 4 字节最大计数时间为 $256*256*256*256/1000/1000/60=71.58$ 分钟, 如果我们为了减少传输字节数, 可以只取最高位的 1 个字节而忽略其他 3 个字节, 那么我们的单位就是 $256*256*256/1000/1000=16.77$ 秒, 也就是说[1 byte time]的单位是 16.77 秒。这个时间反应了该 mac 地址被采集的时间戳, 通过这个时间以及[4 bytes system time]可以计算出采集时间与传输时间的差值, 而传输时间等于服务器接收时间, 这样可以根据差值计算出 mac 地址采集的真实时间。

红色部分表示每个 mac 地址对应的经纬度的低位三字节, 以上述例子来说, 9F4A81 就是低位的三字节。经纬度表示有多种格式, 具体参考云盘中经纬度格式转换的文档。

5.4.2 返回

每次接收数据后服务器需要返回 kk 字符, 只有探针接收到 kk 字符探针才认为上报数据成功。可以通过服务器返回内容对探针进行配置, 协议规则如下。

例子 1:  kkc30121

例子 2:  kkc07121

格式: kkc[sendinterval]30[ignored_ap]1[super_ignored_ap]1[ignored_random]1

红色的 2 字节 kk 表示状态, 黄色的 1 字节字母 c 表示本次返回内容需要配置探针, 绿色的 2 字节表示探针上报时间为 30 分钟, 如果设置为 3 分钟, 前面需要补 0, 应该写为 03, 蓝色的 1 字节表示是否忽略采集到的 ap 的 mac 地址, 1 为忽略, 0 为不忽略, 紫色的 1 字节表示是否开启超强忽略算法来忽略 ap 的 mac 地址, 1 为不开启, 2 为开启, 开启后会降低一定程度的抓包率, 也就是有副作用, 会导致抓到的手机 mac 地址也减少。灰色的 1 字节表示是否忽略采集到的随机地址, 1 忽略, 0 不忽略。以上配置要严格按照字节数和顺序。带有配置信息的返回内容被探针接收到后探针会和本地配置进行比较, 如果不同, 就采用新配置, 并且重启探针, 如果相同, 就忽略本次配置信息。一般习惯的做法是配置信息连发 3 遍以保证探针肯定能接收到配置信息。

总结一下, 只返回 kk 表示数据已经接收, 如果要配置探针需要按上述格式返回。

5.5 代码例子

5.5.1 Go 语言

```
func tcpHandler(conn net.Conn) {
    fmt.Printf("%s gprs %s\n", time.Now().Format("2006-01-02 15:04:05"), conn.RemoteAddr().String())
    defer conn.Close()
    conn.SetReadDeadline(time.Time{})
    conn.SetWriteDeadline(time.Time{})
    for {
        buf := make([]byte, 2048)
        length, err := conn.Read(buf)
        if err != nil {
            conn.Close()
            //fmt.Println("Error tcp reading:", err.Error())
            return
        }

        //sta=[6 bytes sta][4 bytes system time][1 byte csq,if csq>100, gps enabled][1 byte weidu high][1 byte jingdu high][6
        bytes mac 1 byte time 1 byte ssi 3 bytes weidulow 3bytes jingdulow]
        if bytes.HasPrefix(buf, []byte("sta")) {
            now := time.Now()
            //6 bytes 字节转为12字符的探针mac地址
            sta := fmt.Sprintf("%02x%02x%02x%02x%02x%02x", buf[4], buf[5], buf[6], buf[7], buf[8], buf[9])
            var weidu = []byte{0, 0, 0, 0}
            var jingdu = []byte{0, 0, 0, 0}
            hasgps := 0

            //csq本身是小于100的, 如果该探针为选配了gps类型, 其csq=csq+100,因此只要csq大于100就表示
            是选配了gps类型的探针,探针第一包数据因为还未取得csq的数值, 因此会等于100
```



```

if buf[14]>=100 {
    hasgps = 1
    //减去100，恢复csq的真实数值
    buf[14] = buf[14] - 100
    //纬度和经度都用4字节的uint32表示，每个被探测的mac地址都会附带一个经纬度，但如果
    都带4字节的经纬度数据量就太大了，因此和mac地址绑定的经纬度都是取的4字节中的低位3字节，而高位
    的字节会在包级别附带，因为一次上传间，很少会出现高位字节变化的情况。
    //以下取得经纬度的高位字节
    weidu[3]=buf[15]
    jingdu[3]=buf[16]
}
csq := fmt.Sprintf("%d",buf[14])
fmt.Println(now.Format("2006-01-02 15:04:05 "),sta," gprs start reading, len ",length," csq ",csq)
//fmt.Println(buf)
secsnow := now.Unix()
//[4 bytes system time]，把字节数组转为uint32数值类型，该信息并不是真实的时间，而是探针上
电以来以微秒为单位的计数
sendtime := binary.LittleEndian.Uint32(buf[10:14])
result := make(map[string]interface{})
result["sta"] = string(sta)
result["time"] = now.Format("20060102150405")
result["type"] = "gprs"
result["csq"] = csq
dataresult:=make([]map[string]interface {},(length-15-hasgps*2)/(8+hasgps*6))
result["data"] = dataresult
for k:=0;(15+hasgps*2+(k+1)*(8+hasgps*6))<=length;k++ {
    macinfo := make(map[string]interface{})
    startpoint := 15+hasgps*2+k*(8+hasgps*6)
    //把被探测的mac地址的6字节转为12字母显示方式
    mac :=
fmt.Sprintf("%02x%02x%02x%02x%02x%02x",buf[startpoint],buf[startpoint+1],buf[startpoint+2],buf[startpoint+3],b
uf[startpoint+4],buf[startpoint+5])
    macinfo["mac"] = mac
    macinfo["rssi"] = fmt.Sprintf("%d",buf[startpoint+6])
    //每个mac地址都会有一个字节的时戳，本身探针的时戳是4字节以微秒为单位的uint32，
    最高位字节的单位相当于是256*256*256/1000/1000=16.77秒
    var difftime uint32
    //取得该mac地址对应的时戳
    difftime = uint32(buf[startpoint+7])
    //左移24位，到高位字节
    difftime = difftime<<24
    //探针的时戳如果未溢出，也就是未到uint32的最大值，包的时戳一定是大于等于mac
    地址的时戳的,相减得到差值
    if sendtime>=difftime {

```

```

        difftime = sendtime-difftime
    } else {
        //如果溢出，需要借位相减
        difftime = uint32(256*256*256*256-uint64(difftime) + uint64(sendtime))
    }
    //差值由微秒换算为秒
    difftime = difftime/1000/1000
    //secsnow为接收的服务器时间，减去差值基本等于mac地址被采集的时间
    mactime := time.Unix(secsnow-uint64(difftime), 0)
    macinfo["time"] = mactime.Format("20060102150405")
    if hasgps == 1 {
        if (buf[startpoint+8]==0)&&(buf[startpoint+9]==0)&&(buf[startpoint+10]==0) {
            macinfo["lat"]="0"
            macinfo["lon"]="0"
        } else {
            weidu[0]=buf[startpoint+8]
            weidu[1]=buf[startpoint+9]
            weidu[2]=buf[startpoint+10]
            jingdu[0]=buf[startpoint+11]
            jingdu[1]=buf[startpoint+12]
            jingdu[2]=buf[startpoint+13]

            //每个mac地址对应的3字节经纬度和包级别的高位经纬度合并后计算出经纬度，
            格式为nema格式，纬度ddmm.mmmmm,经度dddmm.mmmmm,这里为了传输方便，没有其中的.号
            macinfo["lat"]=fmt.Sprintf("%d",binary.LittleEndian.Uint32(weidu[0:4]))
            macinfo["lon"]=fmt.Sprintf("%d",binary.LittleEndian.Uint32(jingdu[0:4]))
        }
    }
    dataresult[k] = macinfo
}

resultbytes_ := json.Marshal(result)
fmt.Println(string(resultbytes))

responsestr := "kk"
response :=
httpsend(resultbytes,"http://"+jvaserverip+":"+jvaserverport+"/dataserver/api/data/upload2","post")
    if len(response)>0 && len(response)<20 {
        response = bytes.Trim(response,"\\");
        //kkc[sendinterval]09[ignored_ap]1[super_ignored_ap]1[ignored_random]1 [sendinterval]
        最大99 分钟上报一次,必须为两字节，如果为个位数，前面需补0 [ignored_ap] 0不忽略ap, 1忽略
        [super_ignored_ap] 1不开启超强, 2开启超强忽略ap, [ignored_random] 0不忽略随机地址, 1忽略随机地址
        //例子 kkc09111 kkc09121 等
        //其中第三字节c必须存在，表示有配置信息的意思，探针接收会进行解读，如果和本地

```

配置不一样，探针会采用新的配置，并进行自身重启，但探针并不回应配置信息的变化，因此建议返回多次以保证配置送达探针

```
        responsestr = "kkc"+string(response)
    }
    fmt.Println(responsestr)
    _, err = conn.Write([]byte(responsestr))
    if err != nil {
        fmt.Println("Error tcp writing:", err.Error())
    }
    }
}
```

6 硬件规格

6.1 智隐测 T1 型 (仅探测人流,WIFI 上报方式)

一句话描述，采用双 WIFI 模块，全信道扫描，1 秒上传一次数据，通过连接其他 wifi 来上报数据，适合室内定位。



尺寸	长 62MM 宽 62MM 高 20MM
材质	ABS
供电	Micro USB 5V
核心数	双核
探测距离	空旷地带，半径>100 米
FLASH	4MB*2
天线	3DBI *2
无线标准	802.11 b/g/n
频率范围	2.4GHz - 2.5GHz (2400M-2483.5M)
工作电流	平均值: 160mA
工作温度	-40 度-125 度
加密类型	WEP/TKIP/AES
MCU	Tensilica L106 超低功耗 32 位微型 MCU*2

组网类型	Mesh 自组网
上报方式	HTTP, TCP
联网方式	本地存储,wifi 联网

6.2 智隐测 T2 型 (仅探测人流,GPRS 上报方式)

一句话描述，全信道扫描，几分钟上传一次数据，通过 GPRS 上报数据，适合无联网环境的场所探测客流和粗略定位。



尺寸	长 62MM 宽 62MM 高 20MM
材质	ABS
供电	Micro USB 5V
核心数	1 核 WIFI, 1 核 GPRS
探测距离	空旷地带，半径>100 米
FLASH	2MB
天线	3DBI
无线标准	802.11 b/g/n
频率范围	2.4GHz - 2.5GHz (2400M-2483.5M)
工作电流	平均值: 160mA
工作温度	-40 度-125 度
加密类型	WEP/TKIP/AES
MCU	Tensilica L106 超低功耗 32 位微型 MCU*2
组网类型	无
上报方式	TCP
联网方式	GPRS 联网

6.3 智隐测 T3 型 (仅探测人流,GPRS, GPS 定位)

一句话描述，全信道扫描，几分钟上传一次数据，通过 GPRS 上报数据，搭载 GPS 模块，适合需要上报经纬度的流动采集场所。



尺寸	长 62MM 宽 62MM 高 20MM
材质	ABS
供电	Micro USB 5V
核心数	1 核 WIFI, 1 核 GPRS, 1 核 GPS
探测距离	空旷地带, 半径>100 米
FLASH	2MB
天线	3DBI
无线标准	802.11 b/g/n
频率范围	2.4GHz - 2.5GHz (2400M-2483.5M)
工作电流	平均值: 160mA
工作温度	-40 度-125 度
加密类型	WEP/TKIP/AES
MCU	Tensilica L106 超低功耗 32 位微型 MCU*2
组网类型	无
上报方式	TCP
GPS 定位	有
联网方式	GPRS 联网

6.4 智隐测 T4 型 (探测人流与探测商品触碰)

一句话描述, 全信道扫描, 既能探测 WIFI 信号, 又能探测蓝牙 BLE 信号还能探测 433Mhz 信号, 全能型探针, 通过 WIFI 秒级上报, 不能通过 GPRS 上报, 只有此类型才能探测蓝牙和触碰标签。



尺寸	长 62MM 宽 62MM 高 20MM
材质	ABS
供电	Micro USB 5V
核心数	双核
探测距离	空旷地带, 半径>100 米
FLASH	4MB*2
天线	3DBI *2
无线标准	802.11 b/g/n
频率范围	2.4GHz - 2.5GHz (2400M-2483.5M)
工作电流	平均值: 160mA
工作温度	-40 度-125 度
加密类型	WEP/TKIP/AES
MCU	Tensilica L106 超低功耗 32 位微型 MCU*2
组网类型	Mesh 自组网
上报方式	HTTP, TCP
联网方式	本地存储,wifi 联网
商品触碰探测距离	空旷,半径 10 米
iBeacon 芯片	TI cc2540

6.5 智隐测 T5 型 (探测人流,网线上报,POE 供电)

一句话描述, 全信道扫描, 1 秒上传一次数据, 通过网线上报数据, 适合室内定位。同时支持 POE 供电和 USB 供电



尺寸	长 74MM 宽 72MM 高 20MM
材质	ABS
供电	POE 或 Micro USB 5V
核心数	单核
探测距离	空旷地带, 半径>100 米
FLASH	4MB*1
天线	3DBI *1
无线标准	802.11 b/g/n
频率范围	2.4GHz - 2.5GHz (2400M-2483.5M)
工作电流	平均值: 160mA
工作温度	-40 度-125 度
加密类型	WEP/TKIP/AES
MCU	Tensilica L106 超低功耗 32 位微型 MCU*2
组网类型	无
上报方式	HTTP, TCP
联网方式	本地存储, 网线联网

6.6 智隐测 C1 型 (触碰标签)

一句话描述, 感知触碰, 有触碰就发射信号, 价格低廉适合大批量部署, 比如用于店内统计试穿试戴, 需配合 T4 使用。



尺寸	长 52MM 宽 30MM 厚 10MM
材质	塑料
供电	CR2032 纽扣电池
供电时长	>1 年
敏感度	轻微触碰
发射距离	空旷地带，半径>10 米
工作温度	-40 度-125 度

6.7 智隐测 C2 型 (定位标签)

一句话描述，基于蓝牙协议，适合作标签定位，比如保安佩戴标签后定位其行为轨迹，需要配合 T4 使用。



尺寸	多种尺寸可以选择
材质	塑料
供电	CR2032 纽扣电池
供电时长	>1 年
定位精度	1-2 米
通信协议	蓝牙 BLE
发射距离	空旷地带，半径>10 米
工作温度	-40 度-125 度

7 经常问的问题

7.1 为什么需要上报数据?

探针是探测顾客手机的 wifi 信号, 探测到的信号要上传到服务器才能被处理, 使用者也才能看到人流报告等信息, 因此探针需要一种联网的办法上报数据, 目前支持探针通过场地内 wifi 上报方式 (如同手机要连 wifi 才能访问网络一样) 也支持通过 gprs 的 sim 卡上报数据的方式。

7.2 如何评估一个探针?

主要从抓包率和稳定性两个角度评估, 抓包主要测试下手机不连接 ap, 连接 ap, 距离远近等几种情况, openwrt 改装的既能上网又能探测的探针在手机连接了其他 ap 后是无法探测到的, 因为 openwrt 一般只探测 probe 包类型。稳定性是运行今天, 看看抓到的数据时间是否连贯, 是否有中断。

7.3 这个探针既能探测又能当做 wifi 上网么?

不能, 我们是纯粹的探针。既能上网又能探测的探针探测率不能满足要求, 还是建议客户分开使用, 比如不能探测已经连上 wifi 的手机, 不能探测其他信道, 只能探测 probe 等。另外兼具功能的价格一般也较贵, 不利于高密度部署。

7.4 探针防水么?

普通外壳不防水, 可选择防水外壳。

7.5 你们的后台程序能够共享源代码么?

不能, 但我们提供了很多接口, 如果客户有其他定制要求, 我们也可以提供定制接口, 但不能共享源代码。

7.6 你们的后台程序能部署在客户的服务器上么?

可以, 这样探针可以直接报数据到部署了我们程序的后台服务器 (服务器属于客户), 然后客户可以调用上层接口进行开发, 或者定制不同的展示页面, 但要收取一次性费用。

7.7 如何供电, 如何安装?

我们采用的是 5v micro usb 供电头 (就是安卓手机的充电线口), 可以插充电宝供电, 一般的充电宝大约能用一天多。如果是固定部署可以通过插座供电, 如果不方便接插座, 可以把插电头 (变压器) 的两个插电片直接和 220v 的电线相连。

7.8 已经连接了 ap 的手机能探测么?

100%能,之所以问这个问题,肯定是您之前了解过 openwrt 改装的探针,那种即可以上网又能探测客流的探针一般是 openwrt 改装的,是没办法探测连接了 ap 的手机的 mac 地址的.因为无线工作的原理,手机连接了 AP1,手机就不会再发送 probe 类型帧,而 openwrt 改装的探针只能检测 probe 类型帧,这是为什么手机连接后,openwrt 探针探测不到,我司探针拥有多模块,全信道,全类型帧扫描,因此 100%能探测到连接了 AP 的手机.

7.9 苹果能探测么?

苹果主要有两个问题,一个是苹果 app 无法读取苹果本身的 mac 地址(如需要 hack,可以联系我司),另外一个苹果手机从 ios8 开始在一定情况下会在发送的 probe 包中填写假的 mac 地址(假 mac 地址能够 100%识别出来过滤掉,因此尽可能抓取真实 mac 地址即可),从 ios9 开始就更多的情况会发送假地址,所谓假地址就是这十分钟可能发送的 mac 是 1234,下十分钟可能发送的 mac 是 5678,苹果这么做是希望能回避被跟踪.要弄清这个问题,要理解什么是 probe 包,就是你的手机如果 wifi 没有关闭,你会发现一旦你回到家里,手机就会自动连接上你家里的 wifi,这就是 probe 包的作用,你的手机会定时发送 probe 包,不同手机发送频率不同,probe 包就是询问:周围的 wifi ap 的名字是什么?请回答.周围的 ap 回答后,你的手机会把回答的 ap 的 ssid 和之前手机存储的曾经连接过的 wifi 的 ssid 进行比对,一旦匹配,手机就会主动发起对那个 ap 的连接,这时你会发现已经连接上了 wifi.

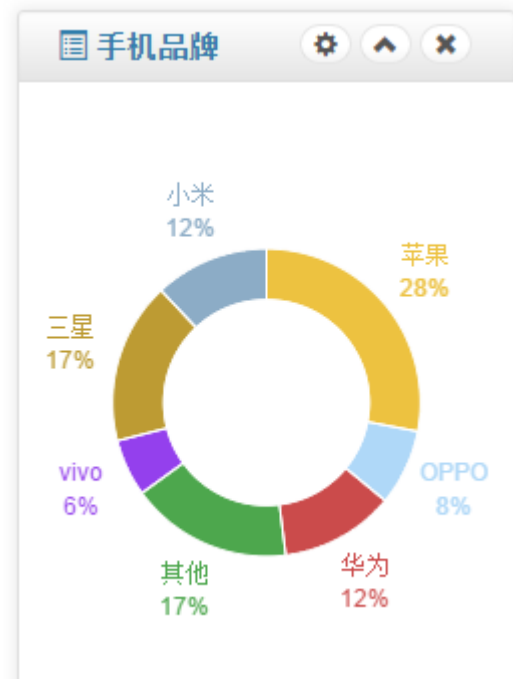
在 ap 回答了手机的 probe 包问候后,如果手机发现该 ap 曾经连过,手机会发起其他类型的包,如 association, authentication, data 等,在这些包里面即便是 ios9 也没法填写假的 mac 地址的,因此只要探针能够探测到这些包,就还是能够探测到苹果手机的真实 mac 地址的.

那问题又来了,手机什么时候才发那些包呢?简单的讲,只要你的手机曾经连接过一些 wifi,而你又走进了有这些 wifi 同样名字的环境就会发起这些包,即便密码不同,也会发起.比如你在上海连接过 cmcc 这个公用 wifi 名字,然后到了北京一个商场,哪里也有公用 wifi cmcc,你的手机就会试图连接,就会发送那些包.又比如一般员工都会连接公司的 wifi,如果你探测的场合是办公室,那探测 ios mac 地址一点问题都没有.同样道理,即便在大街上,也能探测到一定程度的 ios 真地址.但比如一个刚出厂的 ios 手机,一个 wifi 也没有连接过,这个时候一般你很难探测到它的真实地址.但具体何时发送什么类型包,情况比较复杂,但大多数时候还是会发送除了 probe 之外的包.

那什么样的探针能探测那些包呢?一般来说 openwrt 改装的,也就是既是路由器,又兼具探针能力的探针是不可能探测到那些真实的 ios mac 地址的,因为它只探测 probe 包,除非这台手机连接了该改装的 wifi,openwrt 才能收集到真实的 ios mac 地址.

我们的探针为什么能探测呢?因为我们的探针加装了多个底层模块,我们会针对不同信道,不同包类型进行扫描,因此我们能够捕捉.

总之,我司探针能够较高比例探测到 ios 手机,以下为例证,以下报表来自一个在高速路服务区部署我司产品的客户,可以看到真实 ios 地址占比达到 28%,这和该地区的 ios 手机市场占有率基本持平,一般高速服务区顾客也很少连接本地 wifi,所以我司产品能够高概率抓取 ios 地址.



不用费力气去破解随机地址，既然苹果公司发送随机地址，就没办法破解，但我司能够 100% 区别苹果发送的地址中那些是真的那些是随机的，我们把随机的丢弃掉留下真的即可，因此只要评估探针的抓包能力，无线传输过程中，每秒有非常多的无线包在传输，这里有各种手机的，电脑的，ap 的无线包被传输，如果抓包能力不强，那么苹果偶尔发的几个真地址包也会被忽略掉，因此费力气评估探针抓包率才是正解。

7.10 苹果黑屏到底多久能够抓取到包？

这是一个复杂问题，可以参考上面的问题，没有一个简单的答案，不但苹果，其他手机比如华为黑屏也很少发送数据，即便同一个手机根据你安装的 app，使用的环境不同，发送频率也不同，因此我们还是把重点放到抓包率上，要做到只要它一发包就能抓到，而不是遗漏。当然伪装 ssid 有一定作用，但也作用不大，因为环境中已经有很多不同的 ap 起到了伪装的作用。

7.11 手机黑屏时多久发射一次信号

这个很难确定，和手机的品牌，手机是否装了很多 app，手机连接过那些 wifi 都有关系，因此答案是不确定，有的手机几十秒发射一次，有的几分钟甚至更长。

7.12 探测率到底是多少比例

这是一个无法有标准答案的问题，只能自己测试体会，因为不同手机，场景结果都不一样，具体可以参考之前的问题：“已经连接了 ap 的手机能探测么”，“苹果能探测么”，“苹果黑屏到底多久能够抓取到包”，“手机黑屏时多久发射一次信号”，这个问题的解决只能靠全信道，全协议包，多硬件模块扫描，我司就是这么处理的，应该说我司产品表现不错，但仍旧无法承诺 100%

7.13 客流准确度如何?

wifi 探测客流准确度不是 100%, 考虑到有顾客不带手机,或者带多部手机,或者手机没开 wifi,或者顾客经过的时间太短手机来不及发信号,或者定位不准确,这些都是导致准确度下降的原因,一般来说能做到 70%的准确度已经是行业内水平非常高的方案,我司方案能够超过 70%

7.14 定位精度如何?

wifi 定位精度误差一般在 5-10 米,我们的精度大多 2-3 米,如果高密度部署,可以说能够做到业界最好,密度越高,精度越好,大约在 2 米.但我们为什么能做到业界最好呢,主要是我们技术团队吃透了硬件以及算法,我司方案都是我司从底层研发,包括线路板设计,固件研发,云端服务算法,大数据分析等.光凭借好的定位算法无法做到高精度,需要硬件在抓包和发包上配合,另外就是硬件我们从底层研发,能够大大降低成本,因为成本低就能高密度部署,这些因素让我们定位能够更准确.

7.15 不同型号有什么区别?

型号	特点	用途	注意事项
T1	双核全信道扫描, 全类型协议包探测 wifi 信号, 通过 wifi 上报,适合极高频率上报信息	人流, 室内多探针定位	需要有能上网的 wifi, 不能通过 gprs 上报
T2	单核全信道, 全类型协议包探测, 通过 gprs 上报, 适合无网络环境, 低频上报(分钟级别)	室外, 移动探测, 单探针定位	上报频率低, 不适合多探针定位
T3	比 T2 多搭载一颗 gps 芯片	移动探测, 同时记录 gps 地理位置, 单探针定位	gps 只能室外使用
T4	基础功能同 T1,双核探测 wifi 信号, 同时能探测蓝牙以及 433Mhz 信号, 也能作为 ibeacon 使用, 高频上报	探测人流和商品触碰, 标签定位,多探针定位	只有该款才能探测商品触碰, 不能通过 gprs 上报
T5	全信道扫描, 全类型协议包探测 wifi 信号, 通过网线上报,亦可以通过 usb 或者 poe 供电, 适合极高频率上报信息	人流, 室内多探针定位	需要有支持 dhcp 功能的路由器或者交换机, 不能通过 gprs 上报
C1	商品触碰发射标签	识别商品触碰	需要和 T4 配合使用
C2	蓝牙定位标签	识别定位	需要和 T4 配合使用

7.16 如何购买?

- 1)如果只想定位顾客的手机,购买 4 台 150 块的探针即可.
- 2)如果想同时定位顾客的手机和触碰标签,购买 4 台 250 块的和几个 20 块的即可.

3)如果不想定位,只要探测 mac 地址即可,那可以酌情购买,空旷地带探针采集半径超过 100 米

7.17 能发送广告么?

设备探测的只是顾客的手机 mac 地址, 因此无法直接发送广告, 如果要发送有两种途径。

- a. 通过一些小礼物刺激顾客在我司系统注册, 注册了后我们就让顾客填写其手机号, 这样就获取了其手机号和 mac 地址关联关系, 以后他经过周围被探测后可以推送信息, 因此注册一次即可。
- b. 和具有 mac 地址与手机号关联关系的公司合作, 获取其数据库, 但一般要收费, 而且他们也未必齐全。

7.18 只买硬件有用么?

如果贵方是技术公司, 可以只购买我司硬件来上报数据, 然后自行进行处理, 如果贵司不是技术公司, 需要使用我司的云平台, 我们按探针数目每年收取一定服务费, 费用不高。也可以协助客户搭建自己的私有云平台, 那就收取一次性费用。

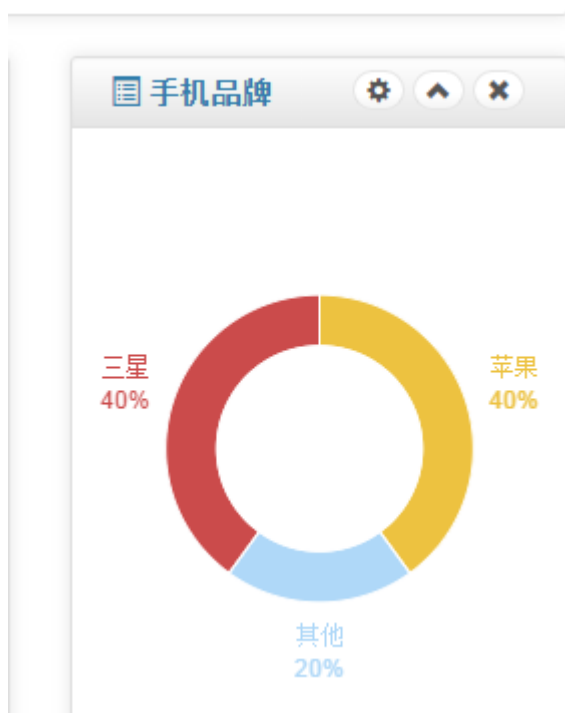
7.19 方案如何收费合作如何开始?

- 1)首先可以购买 4 台探针,可以安装在自己的办公室内,我们开设平台测试账号,你们可以体验,测试.
- 2)觉得不错时,可以多购买几台,然后装到你们客户那里,继续体验测试.
- 3)我们可以协助你们给客户解答问题,乃至定制开发一些小需求.
- 4)待客户项目基本敲定,我们可以一起协商收费模式,或租或者我们参与项目分成,或者你们一次性买断,都可以.
- 5)不建议客户只用我们硬件,自己开发服务器平台,因为真是很多坑,要过滤干扰包,要定位,要考虑性能和数据量,不太容易做,如果外包我司做,相信会成本最低和时间最快,我们可以给客户搭建专属服务器,更换域名和 logo,提供完整的 oem 方案.如果还是想试试,我们建议客户调用我们上层 api,只做展示层面.当然如果客户非要挑战下高难度,我们也非常乐意,随时会协助客户.

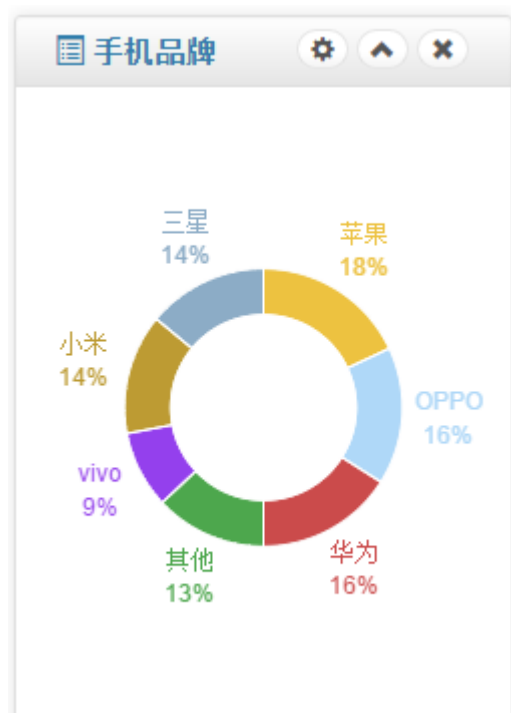
7.20 苹果采集率到底有多高?

这个是个复杂问题, 不同场景, 不同人群, 不同手机状态采集比例不同, 因此无法给出一个具体数字, 但我们给几个例子, 以下数据都是过滤掉苹果的随机地址的真实地址数据, 因此苹果的占比如果和当地苹果品牌存量手机的占比接近, 那么说明获取比例还是很高的。

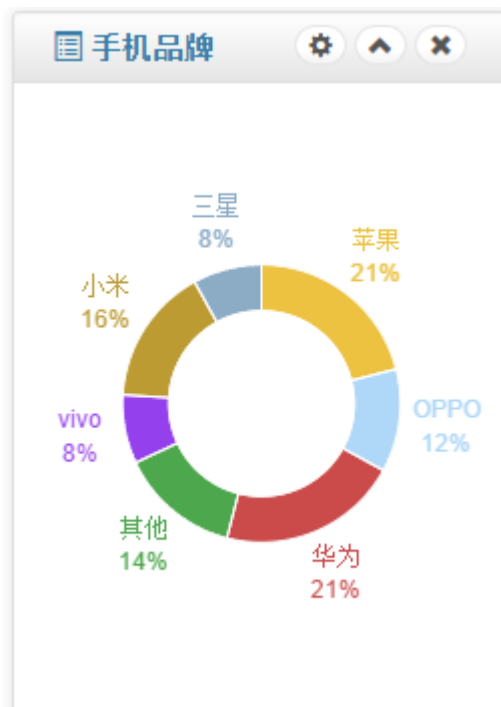
香港的一个展会



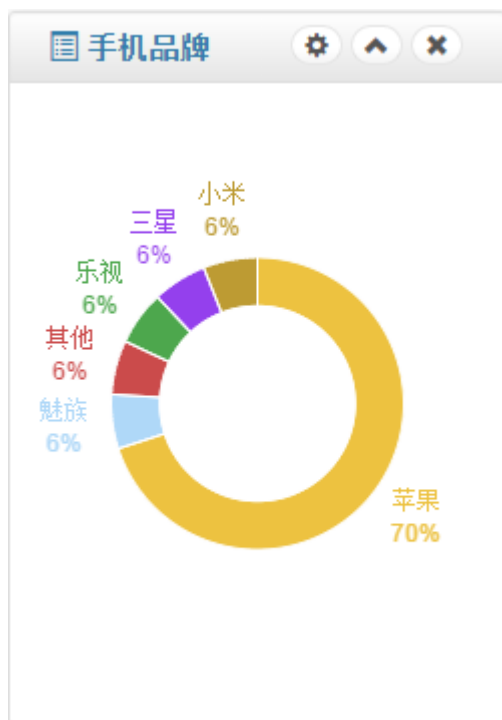
南方的一个高速公路服务区



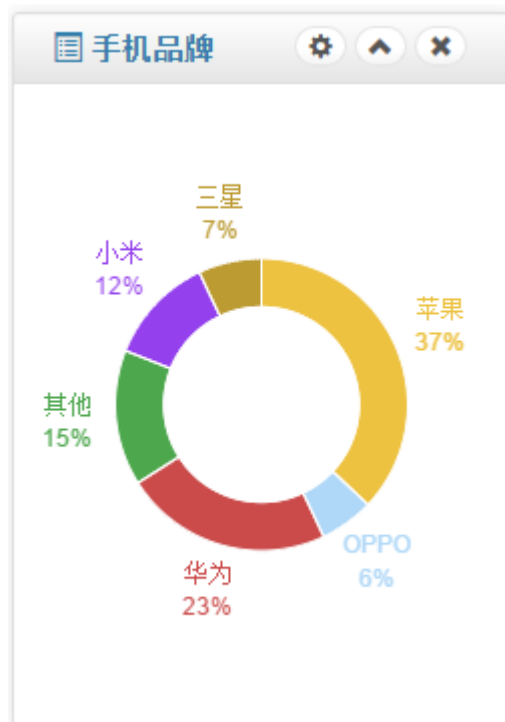
佛山的一个商场



苏州一个设计公司的办公室



上海一家中档写字楼办公室



7.21 如何获取探针完整 mac 地址

方法 1: <http://bc.zhiyince.com/dataserver/web/data/view?sta=07ccaa>

黄色部分替换为探针外壳的字母，打开的页面中就是，如果页面是空内容，有可能是探针使用过期，需要续费。

方法 2: 通过浏览器登陆我司系统，在页面左下方点击“配置”，然后注意页面右边的红色的字，按照指示一直找到探针列表，列表中的 12 字母序列号的后 6 位和外壳字母序列号一致。\\

7.22 如何根据 mac 地址推送?

探测到顾客的 mac 地址，能想到的一个场景就是给顾客推送信息，因此有人会问，能否探测到手机号，答案是不能，绝对不能，不仅仅是技术不能，法律也是不允许的。

那么如何才能做到给顾客推送信息呢？有以下几种思路，但关键点是找到顾客手机的 mac 地址和其可触达信息的关联关系。

- 1) 通过人工的方式查看顾客的手机，记录其 mac 地址，然后录入到会员系统，当再次探测到顾客 mac 地址时，可以推送短信，技术上是没障碍的，但你不给顾客送礼物，或者送个大礼物，顾客一般不肯。
- 2) 也可以通过温和些的办法，可以请顾客把手机放到探针上，我司探针可以通过算法基本确定放在探针上的手机的 mac 地址，这样营业员只要问下顾客的手机号和姓名，就可以录入到系统中。
- 3) 把上述方案变通下，让顾客连接我们探针，就如同连接 wifi 一样，然后输入 <http://192.168.4.1> 在顾客的浏览器上，然后顾客可以注册资料，也能拿到 mac 地址和触达信息对应关系，但同样需要送礼物的。
- 4) 再变通下，因为我司探针只是探针，不能用于上网，店内如果装一个商业 wifi，就是顾客想用 wifi 必须验证手机号那种，也能拿到手机和 mac 地址对应关系，道理相同，只是顾客连接 wifi 的比例一般不超过 10%，除非送礼物。当然，我们有些特殊场所，比如有的公司，连接 wifi 必须有手机号，这样的效果就不错，因为员工总要连接 wifi 的。

- 5) 那么再变通下,可以找一些数据公司,他们有多种渠道来的数据库,就是 mac 地址和手机号对应关系,向他们购买,但一般匹配率低于 15%
- 6) 还可以变通下,我们不推送短信,我们选择和各个大 app 合作,比如微信,我们探测到顾客来过你的店,然后在微信端做广告,这样顾客看微信朋友圈的时候就可以看到你的广告,因为我们告诉微信一个 mac 地址什么时间来到你的店,但顾客打开微信时,微信也能识别这个人的 mac 地址,一看,相同,就可以展示广告。
- 这里有两个障碍,一个是微信这么牛,他不愿意和你对接,其他大一些的比如今日头条,一样很牛,愿意和你对接的都是小 app,因此匹配率仍旧有问题。
- 另外一个障碍来自乔布斯,他限制了一些东西,就是说 ios 上的 app 不能读取本机的 mac 地址,那样就没法做匹配,但我司已经突破了这个技术,大部分 app 都还没突破。
- 7) 再变通下,既然 app 不愿意和我们对接,我们自己在商场放一些大的 LED 广告屏幕,这样,比如万达广场的进口出处的广告屏幕,如果探测到一个顾客之前去过该广场的某一个饭店,这次又来了,就可以在该屏幕上展示那个饭店的广告或者抵用券。

8 探针高级配置

本产品采用双模块探针, A 模块负责信道扫描监听, B 模块负责连接互联网, probe 帧监听, ap 扫描, 配置蓝牙等杂项功能。A 模块周期轮询信道, 然后以一定周期发送给 B 模块, B 模块以一定周期上报的服务器。应场景不同, 发货时的默认配置不同, 如有特殊需要请询问店主, 但客户可以通过如下办法修改配置。

高级配置方法

配置调整方法和配置探针连 wifi 密码方法相同, 操作办法也相同。就是在输入密码处输入我们规定的特殊字符*##然后跟随配置命令。**注意一定是通过安卓 app 配置(参考章节“通过安卓 APP 配置(适合批量)”), 而不是通过连接探针的浏览器配置。**

在配置 App 的密码栏输入如下内容进行配置, 但不点击确认, 给设备重新插电, 待指示灯长亮后点击确认, 待指示灯熄灭后即表示配置成功, 不用理会 app 的返回提示。每次只能配置一项, 如果需要配置其他项目需要重新断电, 但可以同时配置多台, 比如把 10 台探针插到一个接线板上, 同时上电既能同时配置。注意*##后的命令符区分大小写, 请仔细核对。

*##a

表示恢复出厂配置, 之前配置的内容包括密码, 配置托全部复位, 该命令针对 A,B 模块均有效。

8.1 以下针对 A 模块配置

*##Z25

表示配置 A 模块监听信道 5 (25-20=5), 如果是*##Z26 表示监听信道 6, 以此类推, 注意 Z 为大写字母

*##Z1

表示轮询监听信道 1,2,3,4,5,6,7,8,9,10,11,12,13., 注意 Z 为大写字母

*##Z2

表示轮询监听信道 1,6,11

*##Z3

表示轮询监听信道 1,3,5,7,9,11,13

*##Z4

表示轮询监听信道 2,4,6,8,10,12

*##Y5

表示每隔 5*100 毫秒切换一次信道, *##Y6 表示每隔 6*100 毫秒切换一次信道, 以此类推

配置值最大 256, 也就是允许最大 256*100 毫秒切换一次。

##*X5

表示 5*100 大小的 cache, 如果是##*X6 表示 6*100 大小的 cache, 以此类推, 配置值最大 256, 仅对 A 模块有效。

##*W10

表示针对一个 mac 地址, 一次采集过程中最多采集 10 个包, 11 表示 11 个包, 以此类推
配置值最大 256, 仅对 A 模块有效。

##*V200

表示每切换多少次信道 (Y 命令配置) 上报一次模块 A 的配置信息,
配置值最大 2 的 32 次方, 仅对 A 模块有效。

##*U1

是否忽略监听到的 ap 的无线包, 1 表示忽略, 0 表示不忽略。

##*S1

设置忽略监听 ap 包的等级, 0 不启用增强忽略, 1 启用增强忽略(探针需要 1 分钟初始化算法, 但不会导致丢包), 2 超强忽略(在 1 的基础上会把 ap 和手机都可能发的协议类型都忽略, 这样会导致一定程度丢包)

##*R1

是否忽略苹果手机的随机地址, 1 忽略, 0 不忽略。

##*I3

表示每切换多少次信道向 B 模块传输一次数据
最大值 256, 仅对 A 模块有效。

8.2 以下针对 B 模块配置

注: gprs 上报探针也通过 B 模块方式配置

##*17000

表示 B 模块把数据上报给服务器的 7000 端口, ##*18000 表示 8000 端口等。

##*2192.110.110.1

表示 B 模块把数据上报给服务器 ip 192.110.110.1

##*3groupid

最大值 6 个字母或数字, mesh 组网时, 只有 groupid 相同才可能组网。

##*4ssidprefixname

最大值 20 个字母或数字, mesh 组网时, 只有 ssidprefix 相同才可能组网。

##*5password

最大值 20 个字母或数字, 设置 mesh 组网的 password

以上三个参数在自组网模式下有效, 三个值相同的探针会自动组成一个局域网。

##*64

系统每间隔一段时间自动检查一次是否探针与服务器信息通讯正常, 配置值 4 表示如果发现 4 次以上不正常, 系统自动重启或者切换为本地存储模式。

最大值 256

##*730

表示 B 模块每上报 30 次数据, 做一次 ap 扫描, 从而把探针周围 ap 的 ssid 进行上报。

##*85

和上一个命令结合来看, 表示每 5%30, 表示 30 次中的第 5 次后做 ap 扫描。

最大值 256

##*9100

表示每上报 100 次做一次心跳包上报。

最大值 256

##*05

表示 5%100, 结合上面命令, 每 100 次中的第 5 次后进行心跳包上报。

最大值 256

##*b3

表示 B 模块从服务器接收到的响应如果低于发送次数的 3 次以上就重新检查启动连接或者切换为本地存储模式,如果本地存储模式打开的话.

最大值 256

##*c10

表示 10*100 毫秒, B 模块试图上报一次数据到服务器, 如果期间没有新数据要上报, 也不上报。

最大值 2 的 32 次方,但不宜设置过大

##*d0

配置值可以为 0 (启用 mesh 组网的架构, 但不真正组网) 1(启用组网, 真正组网)2 (通过 http 上报数据) 3 (通过 socket tcp 上报数据)

##*e1024

每次发送包的最大为 1024

可以设置为其他值但不能超过 1300

##*f2048

内部包的最大缓冲, 如果某些时间段数据来不及发送, 可以增大缓冲。

不要超过 8*1024

##*gfda50693a4e24fb1afcfc6eb07647825

设置 ibeacon 的 uuid,小写, 不要有分隔符号。

##*h0010

设置 ibeacon 的 major

##*i0010

设置 ibeacon 的 minor

##*j-59

设置 ibeacon 的 measured power

##*k2

设置 ibeacon 的广播间隔, 单位为 100ms

##*l0

设置 ibeacon 的发射功率, ##*l-23 ##*l-6 等

允许值为-23dbm, -6dbm, 0dbm, 4dbm

##*mbeacon01

设置 ibeacon 的 beacon name 为 beacon01

最大值为 20 个字母数字

##*n10

表示每 10 毫秒采集一次商品触碰数据

##*o/data/upload

设置接收 http 请求的服务器的 uri

##*p1

设置探针的 shop id 为 1

*#*q4567

设置探针传输数据时的 token 为 4567

*#*rfuk01

设置探针作为可以连接的 ap 时的 ssid 名称为 fuk01

*#*s12345678

设置探针作为可以连接的 ap 时的 password 为 12345678

*#*t10

设置探针作为 ap 时对外发送 beacon 帧的周期为 10*100 毫秒。

*#*u1

设置探针 ssid 为隐藏，也就是别人通过手机扫描 wifi 看不到该探针的 ssid

*#*v0

设置探针是否启动本地存储模式,0 不启动,1 当联网失败启动,2 不进行联网仅存储在本地

*#*w5

设置 B 模块采集 mac 地址时,每个 mac 地址最大抓包数目

*#*x5

设置 B 模块 mac 地址内存缓存空间大小,5 表示设置为 5*100 大小.

*#*y50

在本地存储模式运行时,每多少次(每次运行间隔由*#*c 设置)运行进行一次数据到 flash 存储.

*#*z1

探针可以被手机搜索到,当手机连接到探针上时,一般情况下手机会弹出默认页面,该配置配置弹出那个页面:

1 填写个人信息关联 mac 地址页面,2 配置探针信息页面,3 查看探针本地数据页面

*#*za0

探针能够对接一些现有公司的平台,以该公司的协议进行数据上报,0 表示对接我司云平台,其他号码表示对接特定合作方平台,只有合作方知道自己的平台号码.