SGM

B2E移动应用开发规范

目 录

[1 引言 4](#_Toc458762275)

[1.1 目的与适用范围 4](#_Toc458762276)

[1.2 相关文档 4](#_Toc458762277)

[1.2.1 B2E MDM SDK 文档 4](#_Toc458762278)

[1.2.2 B2E T-Push SDK文档 4](#_Toc458762279)

[1.2.3 B2E Data-Collect SDK文档 4](#_Toc458762280)

[1.2.4 B2D SDK 文档（包含SSO，Data-Collect 等功能） 4](#_Toc458762281)

[1.2.5 B2D T-Push SDK 文档 4](#_Toc458762282)

[1.3 相关人员通讯录 4](#_Toc458762283)

[1.4 相关术语及缩略语定义 5](#_Toc458762284)

[2 B2E应用的技术规范 8](#_Toc458762285)

[2.1 B2E应用的主要数据流动图 8](#_Toc458762286)

[2.2 金桥以外基地的数据流动图 9](#_Toc458762287)

[2.3 B2E MDM 11](#_Toc458762288)

[2.3.1 MDM SDK简介 11](#_Toc458762289)

[2.3.2 MDM SDK中的手势密码 11](#_Toc458762290)

[2.4 API Gateway 代理验证用户、管理Token及转发服务请求的机制 12](#_Toc458762291)

[2.4.1 BYOD 设备的机制 12](#_Toc458762292)

[2.4.2 SGM 设备的机制 14](#_Toc458762293)

[2.5 如何请求B2E服务 15](#_Toc458762294)

[2.5.1 服务请求URI举例 16](#_Toc458762295)

[2.5.2 API Gateway 的域名与端口 17](#_Toc458762296)

[2.5.3 后台服务的请求类型 17](#_Toc458762297)

[2.5.4 system\_tag 17](#_Toc458762298)

[2.5.5 是否公共接口 17](#_Toc458762299)

[2.5.6 请求参数 18](#_Toc458762300)

[2.5.7 请求Header里的强制性参数 18](#_Toc458762301)

[2.6 B2E Login 接口 19](#_Toc458762302)

[2.6.1 请求URI 19](#_Toc458762303)

[2.6.2 参数传入 19](#_Toc458762304)

[2.6.3 认证成功 20](#_Toc458762305)

[2.6.4 认证失败 21](#_Toc458762306)

[2.7 T-Push 消息推送 22](#_Toc458762307)

[2.7.1 安卓应用的推送流程 22](#_Toc458762308)

[2.7.2 IOS应用的推送流程 23](#_Toc458762309)

[2.8 通过Data Collect发送行为日志 24](#_Toc458762310)

[3 B2D应用的技术规范 26](#_Toc458762311)

[3.1 B2D应用的主要数据流动图 26](#_Toc458762312)

[3.2 API Gateway 代理验证用户、管理Token及转发服务请求的机制 27](#_Toc458762313)

[3.2.1 首次登陆并生成Token的过程 27](#_Toc458762314)

[3.2.2 后续请求服务的过程 28](#_Toc458762315)

[3.2.3 Token的续租 29](#_Toc458762316)

[3.3 如何请求B2D服务 29](#_Toc458762317)

[3.3.1 服务请求URI举例 29](#_Toc458762318)

[3.3.2 API Gateway 的域名与端口 30](#_Toc458762319)

[3.3.3 后台服务的请求类型 31](#_Toc458762320)

[3.3.4 system\_tag 31](#_Toc458762321)

[3.3.5 是否公共接口 31](#_Toc458762322)

[3.3.6 请求参数 31](#_Toc458762323)

[3.3.7 请求Header里的强制性参数 32](#_Toc458762324)

[3.4 B2D Login 接口 33](#_Toc458762325)

[3.4.1 请求URI 33](#_Toc458762326)

[3.4.2 参数传入 33](#_Toc458762327)

[3.4.3 认证成功 34](#_Toc458762328)

[3.4.4 失败信息 34](#_Toc458762329)

[3.5 T-Push 消息推送 35](#_Toc458762330)

[3.5.1 安卓应用的推送流程 35](#_Toc458762331)

[3.5.2 IOS应用的推送流程 36](#_Toc458762332)

[3.6 通过Data Collect发送行为日志 37](#_Toc458762333)

[4 移动应用开发的技术选型 38](#_Toc458762334)

[4.1 决定选型的四个问题 38](#_Toc458762335)

[4.2 建议采用的技术 39](#_Toc458762336)

[5 微信应用架构规范 40](#_Toc458762337)

[5.1 微信企业号运营平台 40](#_Toc458762338)

[5.1.1 运营平台简介 40](#_Toc458762339)

[5.1.2 用户同步与认证 40](#_Toc458762340)

[5.1.3 微信 JS-SDK 41](#_Toc458762341)

[5.2 微信应用开发所需的接口 41](#_Toc458762342)

[5.2.1 常量参数 41](#_Toc458762343)

[5.2.2 Redirect 接口 (原auth接口) 42](#_Toc458762344)

[5.2.3 获取单用户信息 44](#_Toc458762345)

[5.2.4 推送文字 45](#_Toc458762346)

[5.2.5 推送图片 45](#_Toc458762347)

[5.2.6 推送图文消息 46](#_Toc458762348)

[5.3 微信服务的域名与端口 47](#_Toc458762349)

[6 移动应用项目开发流程 48](#_Toc458762350)

[6.1 移动应用开发流程图 48](#_Toc458762351)

[6.1 P&D 48](#_Toc458762352)

[6.2 Design 49](#_Toc458762353)

[6.3 Constructing 49](#_Toc458762354)

[6.4 Testing 49](#_Toc458762355)

[6.5 Deployment 49](#_Toc458762356)

# 引言

## 目的与适用范围

本文档为SGM的主要目的为方便参与SGM各个移动应用项目的相关人员（供应商架构，开发团队，各移动应用服务模块的开发**和部署团队**）理解移动应用在安全、数据流程、交互接口等方面的总体规范要求。开发人员在阅读本文档之后能够更加快速的理解和使用相关的SDK，加速移动项目的开发进程，减少不必要的误解。

本文档直接目的是给开发者一个全景图，而不是作为开发者直接依赖的根据，请开发者参考相关的SDK文档以获得最直接的代码级的参考。

## 相关文档

与本规范相关的文档及其在金桥的Portal的链接如下：

### [B2E MDM](#_B2E_MDM) SDK 文档

### [B2E T-Push](#_T-Push_消息推送) SDK文档

包括供B2E APP集成的SDK以及后台T-Push服务的接口文档（B2E 后台服务模块如何发送一个消息到T-Push 服务器）

### [B2E Data-Collect](#_通过Data_Collect发送行为日志) SDK文档

### B2D SDK 文档（包含SSO，Data-Collect 等功能）

### [B2D T-Push](#_T-Push_消息推送_1) SDK 文档

包括供**B2D** APP集成的SDK以及后台T-Push服务的接口文档（B2D 后台服务模块如何发送一个消息到T-Push 服务器）

## 相关人员通讯录

下表列出的相关人员在本文所描述的技术规范中某一规范发生较大改变时需要被通知到，以避免由此而产生的在项目上的不必要的时间耽误以及人力物力上的消耗。

**此表最后更新于2016年7月8日**

|  |  |  |
| --- | --- | --- |
| 组织机构 | 姓名及职务 | 通讯方式 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

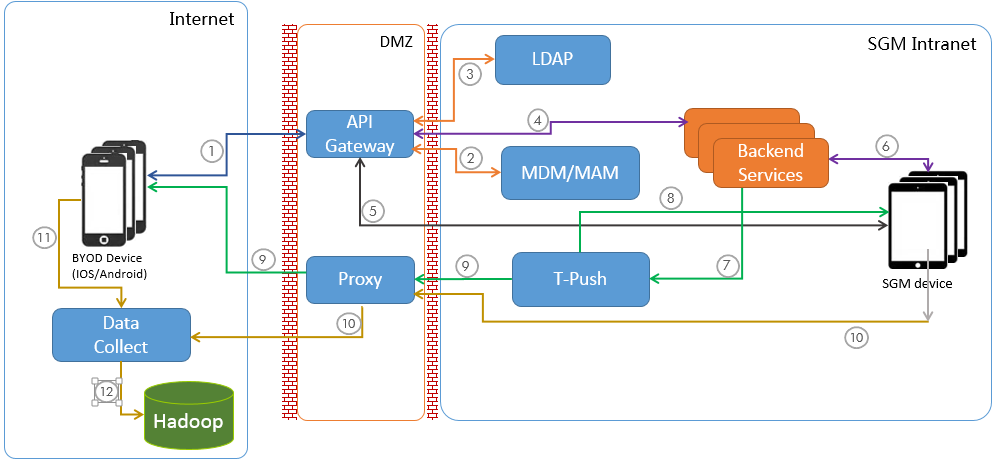
## 相关术语及缩略语定义

|  |  |
| --- | --- |
| 术语 | 定义或描述 |
| B2E | Business to Employees 的缩略语，指的是开放给SGM 员工所用的软件应用。 |
| B2Dealer | Business to Dealer 的缩略语，指的是开发给SGM 的经销商所使用的软件应用。 |
| DMZ | Demilitarized Zone，网络部署里处于外网与局域网之间的子网，对局域网里的服务提供额外一层的防护作用。 |
| SGM Device | SGM分发的移动设备，使用于SGM intranet 里，不能在公网里连接使用。 |
| BYOD Device | BYOD 为Bring Your Own Device 的缩写，指非由SGM 定制分发的自由移动设备。例如员工、经销商自带的移动设备。 |
| API Gateway | 是SGM移动架构里负责验证用户，维护用户token，校验服务合法性，转发服务请求及其响应的一个网关。在其他文档里可能被叫作Datapower。 |
| SSO | Single Sign-On 的缩写, 在本文指SGM移动开发门户里的任何一个APP登录了之后，用户可以打开其他APP而无需再次登录的机制和易用性。 |
| MDM | Mobile Device Management的缩写, 它以及 MAM（Mobile Application Management）是移动应用的管控模块。移动应用，不论是BYOD还是SGM 设备的，登录时都需要经过API Gateway进行身份验证和MDM/MAM接入，以确保该应用（及其设备）在MDM 的管控之内。 |
| LDAP | Lightweight Directory Access Protocol的缩写. 是一个开放的，中立的，工业标准的应用协议，通过IP协议提供访问控制和维护分布式信息的目录信息。本文LDAP也指部署于金桥的SGM LDAP服务器。SGM 的员工用户密码信息就保存在LDAP服务器上，并且通过LDAP协议来进行访问。 |
| Dealer Porter | SGM 的B2D 系统的认证门户 |
| APNS | Apple Push Notification Service, Apple Push服务器，是苹果公司负责推送消息的服务器 |
| Proxy | 代理或反向代理，它部署于DMZ里，在Service 发送T-Push事件给 BYOD Device、SGM设备连接到APNS时起到中间桥梁的作用。 |
|  |  |

# B2E应用的技术规范

## B2E应用的主要数据流动图

下图以数据流动的方式描述出SGM B2E移动应用全局。



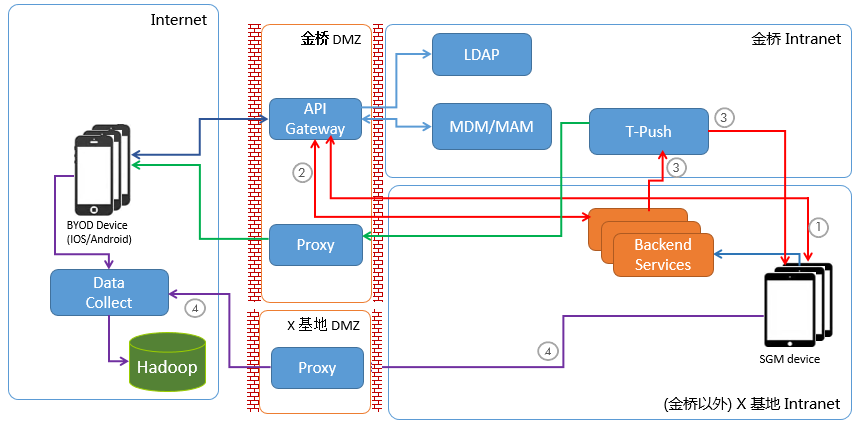
其中, 以下是每一条标有数字的连线所代表的数据流动。此处仅作出大致定义，其序号与上图的连线上的数字一一对应。

1. BYOD 设备向地处DMZ的API Gateway 发起的身份验证请求及验证结果的返回，或者是随后的服务请求及其返回。
2. 移动应用发出的身份验证请求通过API Gateway转至MDM，MDM随后通过API Gateway 发起身份验证请求。另外，在B2E设备装了MDM模块激活启用之后即受控于SGM的MDM管理，MDM后台可以主动发出命令控制即使是处于外网的BYOD设备。
3. API Gateway 收到来自MDM 的身份验证请求时，则会将传来的用户账号与密码转发给后台的LDAP以并获得验证成功与否的结果。
4. API Gateway 验证来自BYOD的服务请求之后将请求转发部署于内网的相应的Backend Services 并获得回应。
5. 处于内网的SGM设备也是通过API Gateway 去LDAP进行身份验证的。
6. 处于内网的SGM设备为了性能需要访问Backend Service 时采用内网直接连接，而无需经过API Gateway。
7. Backend Services 将需要通知设备用户的业务消息发给部署于内网的T-Push 服务器。
8. T-Push事件带有身份信息以供识别，如果是发给安卓类型的SGM设备的T-Push事件，它会被T-Push服务直接转发至响应设备上。如果设备类型是IOS，其通知方式不在本图，参见[T-Push消息推送](#_T-Push_消息推送)一章。
9. 如果是发给BYOD设备的T-Push事件，如果设备类型是安卓，它会经代理服务器推送给BYOD设备。如果设备类型是IOS，则会通过代理服务器转发到APNS，再由APNS推送给设备，详见[T-Push消息推送](#_T-Push_消息推送)一章。
10. SGM设备 的行为日志信息从设备上通过代理发送到部署于公网的Data Collect 服务器。
11. BYOD 设备的行为日志信息从设备上直接发到部署于公网的Data Collect 服务器。
12. Data Collect 将所有行为日志发送到Hadoop所开放的接口，进入Hadoop以供日后的大数据业务使用。

**请注意：本章只是一个方便读者了解全局情况的总览，下文有更加具体的流程图详述几个业务的数据流程。**

## 金桥以外基地的数据流动图

上面是SGM金桥基地为例的典型的数据流动图，其他SGM 基地的部署环境不同，会使得上图出现一些变化，具体见下图：



其数据流动的描述与“B2E应用的主要数据流动图”所描述的大体一样，不同的数据连线以数字标识，其描述如下：

1. 当处于该基地的SGM设备登录时，它通过VPN连接到金桥的API Gateway。
2. BYOD设备请求服务时，如果后台服务部署于该基地，则API Gateway是通过VPN连接到这些后台服务的。
3. 部署于该基地的后台服务有消息需要通知设备时，是通过VPN连接到部署于金桥的T-Push 服务器的，且如果是SGM安卓设备的情形，T-Push也是通过建立于VPN的长连接将消息直接推送回该基地的设备。
4. SGM设备发送行为日志消息时，是通过该基地自己的代理连接到部署于公网的Data Collect服务器，进而再转发到Hadoop 的。

注：图中红色连接部分标示出所有需要采用VPN连接的数据流动。由于VPN 已经部署而其连接在底层实现，这些连接对于这新的子系统的开发者而言是透明的，可以“假定”这些服务就像在自己内网中一样。

## B2E MDM

### MDM SDK简介

MDM 除了在用户登录时通过API Gateway向LDAP发出登录请求外，还管控所有的SGM B2E应用及其承载的移动设备。MDM的管控是通过用户设备上必须强制安装企业安全策略的一个定制的程序而实现的。该程序在手机开机之后即启动并“听从”MDM 的管控，MDM 的管理员可以在服务器的相应页面实现对任何一台B2E设备进行诸如发送通知、发出报警信号音、清除其安全管辖范畴内的所有用户数据、强制锁屏、强制关机等操控。

SGM向所有的第三方业务应用提供一套MDM的移动设备端的SDK，SDK有IOS和Android两个版本。主要功能包括：单点登录（手势密码）接口，获取用户信息接口，应用版本更新接口，等。第三方移动应用必须集成MDM平台提供的SDK，SDK的接口说明和软件库请见[B2E MDM SDK文档](#_B2E_MDM_SDK)。请B2E APP开发人员仔细阅读此文档，确保SDK的集成工作可以顺利进行。

### MDM SDK中的手势密码

从用户体验的角度考虑，SGM要求移动应用在被用户在从iwork的工作台打开的情况下，不得弹框要求用户输入手势密码，原因是在进入iwork之前已经要求用户输入过手势密码了。另外，从安全的角度考虑，SGM要求移动应用在从非iwork控制台启动的情况下必须要求用户再次输入手势密码才能进入。

具体的处理方法如下：

**Android 平台**: SGM要求所有的B2E android应用都隐藏桌面图标（安卓系统里这一需求是可实现的），所有B2E移动应用全都集成在iwork的工作台中，因此Android平台下的业务应用启动只可能从iwork工作台中启动或者从近期任务栏里启动。从iwork中启动则无需输入手势密码，反之如果从近期任务栏启动，则需要显示手势密码输入框要求用户输入手势密码再进入应用。Android应用需要通过Intent flag来判断应用是否是从近期任务栏启动。

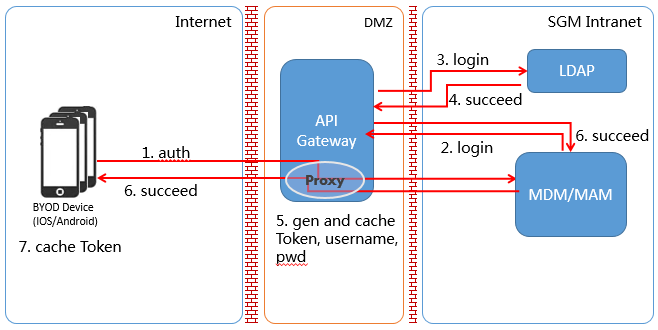
**IOS 平台:** 由于IOS强制所有移动应用都在桌面有独立的图标而不得隐藏，因此在IOS下B2E移动应用除了从iwork启动以外，还可能从桌面启动或者从后台启动（当用户按了Home键之后移动应用就进入后台）。从iwork启动时，iwork会向业务应用传入一个参数：isworkplatform=1。业务应用启动时需要对这个参数进行判断，如果发现传入这个参数并且值为1，就直接启动应用而不显示手势密码输入框。如果没有发现这个参数（说明用户是从桌面或者后台启动的该应用的）或者值不为1，则需要调用SDK中的手势密码接口，显示手势密码输入框，要求用户输入手势密码。

## API Gateway 代理验证用户、管理Token及转发服务请求的机制

### BYOD 设备的机制

**首次登陆并生成Token的过程**

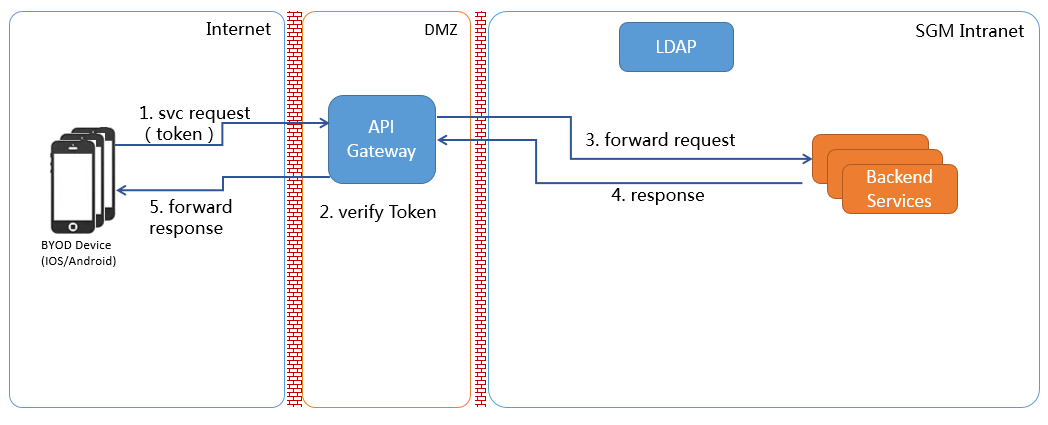
设备首次登录的过程如下图，下文的描述的序号与图中连线上的数字一一对应。



1. BYOD 设备将用户名密码通过API Gateway 发到MDM请求身份验证。
2. 收到来自MDM 的身份验证请求后，API Gateway 为当前用户向后台LDAP发起身份验证请求。
3. 如果用户名密码没有错误，LDAP 验证后返回带有成功标识的指令给API Gateway。反之如果用户名密码错，LDAP 返回的错误信息会被原路返回最终告知到BYOD 设备，由登录模块弹出窗口提示用户。
4. API Gateway为该BYOD 设备及其用户生成一个带有时间戳的Token，并在本地缓存该用户的用户名、密码等账户信息。这个Token 的有效时间为当前时间起一定的时间长度内（假设是60分钟）。
5. API gateway返回成功标识给MDM，并且附上刚刚产生的Token值。
6. MDM 将带有成功标识的指令及其Token 通过API Gateway 返回给BYOD 设备。
7. BYOD设备在本地缓存收到的Token以供后续请求服务时使用。

整个过程是同步的，即BYOD设备的客户端在发起登录请求后一直等待到成功返回并且获得Token，每个请求及返回都是即时的，整个过程发生在可以接受的较短时间内。

**后续请求服务的过程**



1. BYOD 将服务请求发至API Gateway，其中带有自己本地缓存的Token，所请求服务的类型、system\_tag 等信息。  
   注：system\_tag 是用以区分每一个独立的app及其后台服务的标签，API Gateway根据服务抵达的端口，system\_tag 等信息可以在其配置的Routing Table上映射到相应的后台服务的服务地址，使得服务请求被转发准确无误的后台服务上。B2D 服务的详细调用方法请参考 [如何请求B2E服务](#_如何请求B2E服务) 一章。
2. API Gateway 将Token 里的时间戳解析出来，并判断并证实其仍在有效期内。  
   注：有效期是可以配置的，目前通常设置为30分钟。
3. API Gateway将合法请求转发给相应的后台服务。
4. 后台服务返回其响应.
5. API Gateway 将后台服务的响应转发回发起请求的BYOD设备。

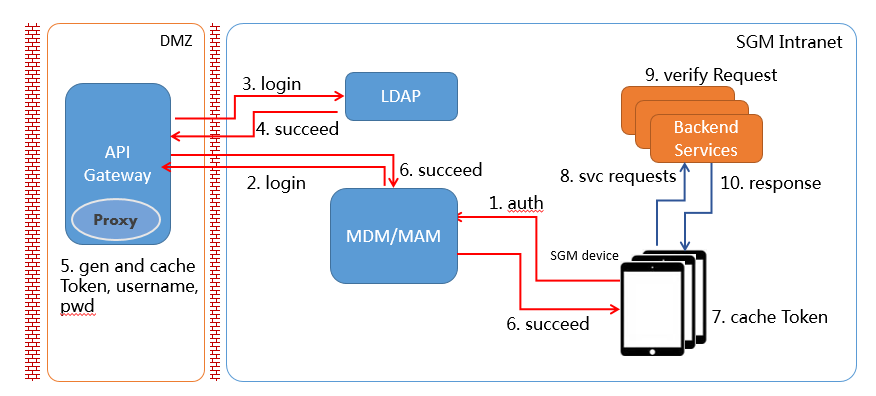
**Token的续租**

由于Token 带有时间戳且每个Token都是有时效性而不能永远使用的，客户端在请求服务之前都要检查本地的Token是否仍然有效，如果BYOD判断当前已经及其接近失效（如1分钟后就过期了）或已经失效，则为了保证服务请求被API Gateway 接受，在请求服务之前前面描述的登录会被BOYD设备再触发一次以刷新本地的Token，之后才用最新的Token 去请求服务。

以上Token续租的机制已经实现在SGM B2E移动应用的SDK 里，在登录成功之后，新移动项目的开发者只需要根据SDK 文档的描述去调用后台的服务即可，SDK 会保证 Token 在必要时被续租且总是能有效通过API Gateway 的验证。

### SGM 设备的机制

**SGM**设备处于SGM 的intranet 中，其登录和服务请求的过程相对简单，其流程如下图，下文的描述的序号与图中连线上的数字一一对应。



1. SGM 设备将用户名密码通过API Gateway 发到MDM请求身份验证。
2. 收到来自MDM 的身份验证请求后，API Gateway 为当前用户向后台LDAP发起身份验证请求。
3. 如果用户名密码没有错误，LDAP 验证后返回带有成功标识的指令给API Gateway。反之如果用户名密码错，LDAP 返回的错误信息会被原路返回最终告知到SGM 设备，由登录模块弹出窗口提示用户。
4. API Gateway为该SGM 设备及其用户生成一个带有时间戳的Token，并在本地缓存该用户的用户名、密码等账户信息。这个Token 的有效时间为当前时间起一定的时间长度内（假设是60分钟）。
5. API gateway返回成功标识给MDM，并且附上刚刚产生的Token值。
6. MDM 将带有成功标识的指令及其Token 通过API Gateway 返回给SGM 设备。
7. 虽然后续请求服务时可能不需要使用Token，SGM设备仍然在本地缓存收到的Token。
8. SGM设备在后续发送服务请求时，直接发送到相应的后台服务上（其APP内配置过了服务地址），无需经过API Gateway。
9. 后台服务返回相应给SGM 设备。

**注：**后台服务对于SGM 设备所发送的服务请求有自己的验证机制，每个项目各自实现这些机制，不受API Gateway的管控。

**另外**，取决于具体的应用系统和项目，SGM 设备也可以与BYOD设备一样通过API Gateway去请求服务，在这种情况下，SGM 设备所缓存的Token就被使用了，Token 的续租机制与BYOD情况一模一样。

## 如何请求B2E服务

由于后台服务处于SGM 的内网，外部Internet的BYOD 设备无法直接访问，只能通过API Gateway 居中进行代理。又由于SGM的后台有多个子系统，每个有各自的服务部署，因此API Gateway 通过自己所配置的映射规则来找到正确的后台服务并代理转发服务请求。

在APP登录后进行后续的服务请求时，除了前面描述的Token机制所有子系统的APP 必须符合规范之外，还需要符合如下规则才能正确调用后台服务。

### 服务请求URI举例

通过API Gateway进行服务请求首先要知道将服务发送到什么地址、什么端口、服务请求的URI 的完整格式如下：

https://[API\_Gateway\_域名]:[端口]/[请求类型]/[system\_tag]/[是否公共接口][请求参数]

上述URI 中所有以[]号起止的部分为因不同子系统、服务及接口而异的，它们之间以斜杠连接，或者直接连接。因此，开发调用服务请求的人员只需要知道API\_Gateway\_域名、端口、请求类型、system\_tag、请求参数就能够根据上述 “公式”拼凑出完整的URI去进行服务请求。在正常的情况下，API Gateway 会负责映射找出相应后台服务的内网Backend\_URL（包括IP 地址、端口、以及可能存在的入径），并将服务请求的参数原封不动的发送到该Backend\_URL 上来作为完整的内网服务的请求URI。

例1）下列URI是在生产环境下 system\_tag为 “mytest”的某个子系统的一个获得用户余额的接口的样例：

https:// b2eapi.saic-gm.com:3233/service/mytest/get\_balance?userid=0001  
[API\_Gateway\_域名]:[端口]= b2eapi.saic-gm.com:3233  
[请求类型]=service  
[system\_tag]=mytest  
[是否公共接口]=[空] （非公请求在这里留空）  
[请求参数]= get\_balance?userid=0001

例2）下列URI是在生产环境下 system\_tag为 “mytest”的某个子系统的一个获得某新闻内容的接口的样例：

https:// b2eapi.saic-gm.com:3233/service/mytest/public/get\_news?newsid=012  
[API\_Gateway\_域名]:[端口]= b2eapi.saic-gm.com:3233  
[请求类型]=service  
[system\_tag]=mytest  
[是否公共接口]=public/  
[请求参数]= get\_news?newsid=012

### API Gateway 的域名与端口

API Gateway的域名以及端口如下，通常不会变动，有变动会做另行通知。

|  |  |  |  |
| --- | --- | --- | --- |
| 软件环境 | 业务环境 | 域名 | 端口 |
| 测试环境 | B2E | apigatewayqa.sgmlink.com | 3223 |
| 生产环境 | B2E | b2eapi.saic-gm.com | 3233 |

上述例1、例2都是生产环境的例子。

### 后台服务的请求类型

目前B2E的请求类型只有“auth”和“service”两种，分别代表身份验证请求，和服务请求。用户输入用户名密码进行登录时，就调用auth 类的请求，其他多数情况都是service类的请求。上述例1、例2里的请求类型值都是“service”。

### system\_tag

SGM IT会为每个后台服务分配一个system\_tag以代表每个移动应用所隶属的子系统。API Gateway的后台配有一个system\_tag 映射到后台服务地址的映射表。通过此映射，API Gateway就能获得外网APP 发起的合法请求应该被转发到内网部署的哪个服务器及端口上。上述例1、例2里system\_tag的值是“mytest”。

### 是否公共接口

公共接口是指不必验证身份就可以调用的接口（例如获得某条新闻的内容），非公接口是指需要身份验证通过之后才能调用的接口（例如获得某用户的账户余额）。

某个后台服务接口是“公共”还是“非公”是由子系统里的业务决定的，它是在项目设计阶段由BA和项目经理以及SGM IT的负责人来协商决定的。

在服务请求URI上，公共接口的调用URL应在system\_tag 后的斜杠之后再接上“public/”。反之如果是非公接口，则服务请求URI上不会带有public字样。API Gateway据此判断一个请求所调用的接口是否为公共接口。上述例1为非公接口的调用，例2为公共接口的调用。

对于公共接口的请求，API Gateway只确认该服务接口的确是公共接口，不对发起请求的用户做身份认证即直接将请求转发到相应的服务器。

收到非公接口的调用请求时，API Gateway会去验证请求的Header里所附带的Token 是否在有效期内，如果Header里没有附带Token，或者所带Token已过期，API Gateway会拦截该请求。

### 请求参数

请求参数是整个外部服务请求URI的最后部分。对于公共接口请求参数是从public后的斜杠之后开始的字符串。对于非公接口则是system\_tag之后的斜杠之后开始的字符串。

取决于业务应用，它可以是灵活的多个参数以&连接。上述例1里的请求参数是“get\_balance?userid=0001”，例2里的请求参数是从斜杠起的 “get\_news?newsid=012”。

### 请求Header里的强制性参数

所有服务请求调用时都需要在HTTP 的header里传入 Content-Type 参数（内容格式）以标识该接口是以什么方式出入和传出参数内容的，典型的值有“application/Jason”或者“text/xml”。除了SGM 统一的接口外（如登录接口规定以JSON 形式传入和返回参数），子系统自行规定其接口的内容格式。

非公接口调用除了Content-Type还必须传入Authorization,其值即为客户端目前为止最新保存的Token值，参考一下表格中的例子。

|  |  |  |
| --- | --- | --- |
| 强制参数 | 说明 | 例如 |
| buId | 输入用户名 | buId:apptest01 |
| Content-Type | 输入接口入参出参数的格式 | 例1：Content-Type:application/json 例2：Content-Type:text/xml |
| Authorization | 在非公接口的调用则必须传入，公共接口可以不传。  这里的值为当前客户端缓存的Token，格式为:[token\_type]+空格+[access\_token].  其中的token\_type 与 access\_token 分别是调用B2E login 成功时返回的参数值。 | Authorization:Bearer AAEPQVBQU1RPUkVfQ2xpZW50jaohbx9n  PSnYn5iayUfVggB\_Pla9Hkiy7GjTU8PgY9RIv6-KqNfT\_1AbTLbclJa2jzNMLUQnv07z7Nxmb  EdixoXh\_gTUgMt892L8FoJanHE  （字符串中间的换行符是为了本文档展示效果而加进去的） |

## B2E Login 接口

B2E的Login接口也是B2E服务的一部分。但是由于它是所有后续服务的前提，且其输入输出是已经规定好了的，而不像其他后台服务那样是由第三方子系统自行规定的，所以在这里专门进行描述以方便需要的开发者。

Login即为认证用户名、密码是否正确的接口。SGM 发布的SDK里已经包含了BYOD设备的登录页面以及登录动作的所有实现，但不包括 SGM 设备的登录。因此开发BYOD设备应用的开发者只需要集成SDK，而不需要另行开发登录相关的页面及逻辑。而开发SGM 设备的移动应用，登录相关的开发需要参考本章节的指引来进行。

### 请求URI

如前所述，访问登录接口的请求URI如下，只需要将[system\_tag]替换成真实有效子系统的system\_tag 值。

测试环境：https:// apigatewayqa.sgmlink.com:3223/auth/[system\_tag]/public/login

生产环境：https:// b2eapi.saic-gm.com:3233/auth/[system\_tag]/public/login

**注：对于B2E 系统，最终始终会在SGM 的LDAP认证用户名与密码，与子系统无关，不会在子系统里验证。这里在URI里含有子系统的system\_tag 是为了保持格式上的统一。**

### 参数传入

在通过HTTPS进行登录请求时，还需要在HTTP 的Header里加上如下参数：

|  |  |
| --- | --- |
| Header参数 | 说明 |
| buId | 这里输入当前登录用户的username, 例如： buId:apptest01 注：在下述body里的JSON字符串里也要求有username 参数，目的是为了保证规范的一致性（所有接口的Header强制参数里都包括了buId，这里保持一致） |
| Content-Type | B2E的登录接口内容已经定为Json 格式的，因此这里总是使用如下字符串： Content-Type:application/json |
| Authorization | 这里请硬编码为如下字符串： Authorization:Basic QVBQU1RPUkVfQ2xpZW50OnBhc3N3MHJk |

除了Header 外，在HTTP 请求的Body里应该附上包括用户名(username)、密码(password)、授权类型（grant\_type）的整个Jason 字符串。其例子如下：

{"username":"apptest01","password":"Pass1234","grant\_type":"password"}

其中，授权类型(grant\_type)的值可以硬编码成 “password”。

### 认证成功

登录成功则API Gateway 会返回“200 OK”的成功状态码，并在HTTP返回中带有一个JSON字符串，解析后包括如下信息：

|  |  |
| --- | --- |
| 参数 | 说明 |
| token\_type | 这里返回token 的类型，由于后续服务请求如果是非公的接口则需要在Header里传入Authorization，格式为: [token\_type]+空格+[access\_token]，因此这里收到的token\_type 也需要缓存在本地以供后续服务请求使用。 |
| access\_token | 即前文提及的Token, 它是身份验证成功后API Gateway 随机生成的一个长字符串，客户端解析出来后应缓存在本地以供后续服务请求时使用。 |
| expires\_in | 上述Token在生成之时起的一定时间内是有效的，超过则需要重新验证，有效期即为expires\_in, 它是一个以秒为单位的时长数值，例如3600 标识3600秒。由于SGM 的B2E 移动应用SDK里已经实现了自动刷新（续租）Token的机制，APP 开发者可以不必处理这一信息。 |
| scope | 当前登录的用户所允许调用的服务类型，通常的“service”类的服务都是可以调用的，APP 开发者可以不必处理这一信息。 |

整个JSON字符串的例子如下：

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

{ "token\_type":"bearer","access\_token":"AAEPQVBQU1RPUkVfQ2xpZW50jaohbx9nPSnYn5iayUfVggB\_Pla9Hkiy7GjTU8PgY9RIv6-KqNfT\_1AbTLbclJa2jzNMLUQnv07z7NxmbEdixoXh\_gTUgMt892L8FoJanHE", "expires\_in":3600, "scope":"/data/\* /service/\*" }

### 认证失败

登录失败可能是由于多种原因的其中一个导致的，API Gateway 会返回HTTP 问题状态码并附上错误（error）和错误描述（error\_description）,详见下表。

|  |  |  |  |
| --- | --- | --- | --- |
| **状态码** | **response body里的错误信息** | | **说明** |
| **error** | **error\_description** |
|  | urlopen connect error | 请求服务连接错误 |
| 0x00d30003 | There is no auth to go through the gateway. | url不符合规则 |
| datapower自定义的errorCode | datapower自定义的errorMessage | datapower其它未知内部错误 |
| 401 | 0 | Auth request parse error | Auth login时，数据解析错误(login时，request body只接受JSON格式) |
| Bad auth (by user password) request, please check | Auth login时，request body中的参数不全 |
| invalid\_grant | resource owner failed to authenticate | resource owner 登录认证失败 |
| invalid\_client | Missing or invalid client\_secret | client\_secret错误或缺失 |
| invalid\_client | OAuth client is not configured | oauth client未配置 |

其中最常见的是错误为“0”的错误，它代表两种错误：

“Auth request parse error” 表示服务端所收到的应为JSON 格式的字符串无法解析，着可能是开发人员将键值对封装成JSON时做错导致。

“Bad auth (by user password) request, please check” 表示所输入的用户名、密码错误，这可能是登录用户输入错误导致。

用户名、密码填写错误导致的登录失败，其返回的例子如下：

HTTP/1.1 401 Unauthorized

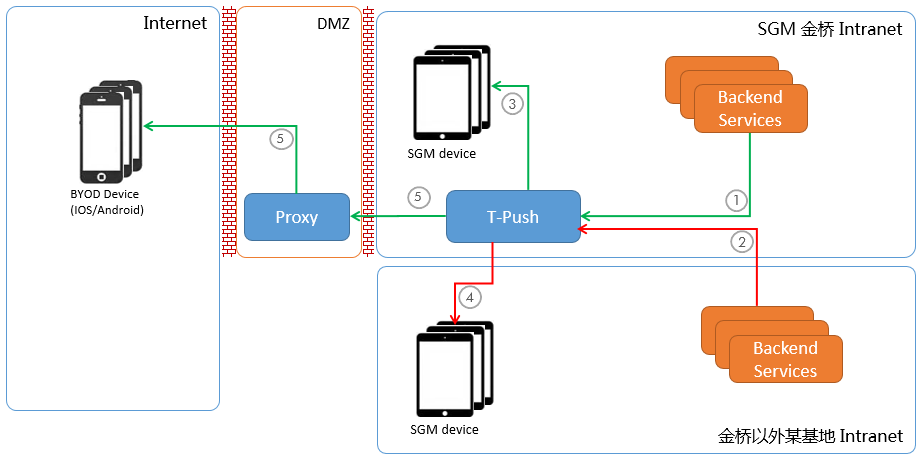
Content-Type: application/json;charset=UTF-8

{"error":"0", "error\_description":"Bad auth (by user password) request, please check"}

## T-Push 消息推送

由于软件需求的要求某些诸如待办事项、通知、审批请求遭拒等服务消息能够被推送到相应的用户的应用上。这个过程是通过部署于金桥的“T-Push”服务器来实现的。安卓与IOS有不同的消息推送机制，细节如下。

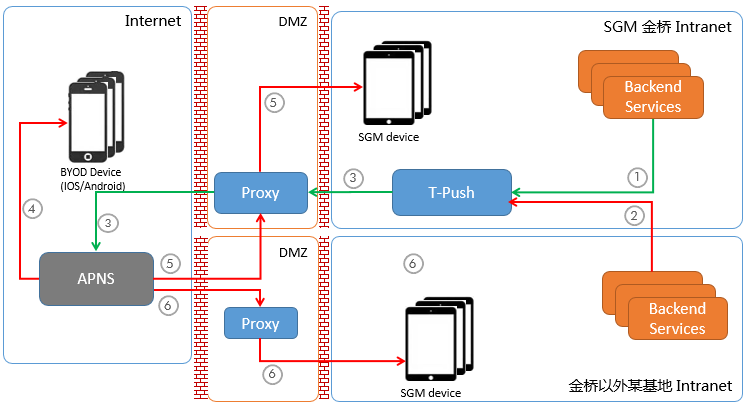
### 安卓应用的推送流程



上图是安卓应用的情况下T-Push的信息流转图。其序号与上图的连线上的数字一一对应。

1. 当部署于金桥的某个后台服务需要推送消息到某个设备时，它会将消息发送到部署于金桥的T-Push 服务器上。
2. 如果发送消息的后台服务是部署在金桥以外的其他基地的话，它会通过VPN 连接来发送消息，同样是发送到金桥的T-Push服务器上。
3. 取决于具体的应用需求， SGM的B2E移动应用在APP登录后即可建立一个到T-Push 服务的长连接。如果是处于金桥内网SGM设备，则T-Push服务会通过长连接将消息直接推送到处于金桥的SGM 设备。
4. 如果是处于金桥以外的其他基地的SGM 设备，则T-Push服务会通过基于VPN的长连接推送到该设备上。
5. 如果是处于公网的BYOD 设备，则T-Push 服务会通过反向代理将消息推送到该设备上（实际的实现方式是，处于公网的BYOD设备在登录时通过反向代理创建一个长连接到T-Push 服务器上。当T-Push 服务模块收到发给该用户的消息时则通过这个长连接即时发送消息到BYOD设备上）。

### IOS应用的推送流程



上图是IOS应用的情况下T-Push的信息流转图。其序号与上图的连线上的数字一一对应。

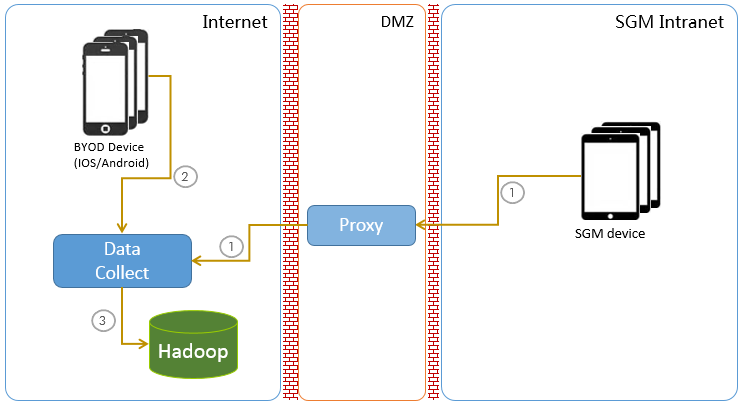
1. 当部署于金桥的某个后台服务需要推送消息到某个设备时，它会将消息发送到部署于金桥的T-Push 服务器上。
2. 如果发送消息的后台服务是部署在金桥以外的其他基地的话，它会通过VPN 连接来发送消息，同样是发送到金桥的T-Push服务器上。
3. T-Push服务收到消息之后通过代理将消息转发到苹果私有的APNS服务器上。
4. IOS有自己内建的消息推送机制使得它能够从全球部署的APNS 服务器上获得推送消息，如果是BYOD设备，它从internet上直接能获得APNS的推送消息。
5. 如果是处于金桥的SGM设备，则通过金桥的代理服务器与APNS服务器连接获得推送消息。
6. 如果是处于金桥以外其他基地的SGM设备，则通过当地代理与APNS服务器连接获得推送消息。

注：上述不同操作系统的消息推送机制都在B2E T-Push SDK 里支持。详细集成方式以及后台服务如何连接并发送消息到T-Push服务器请参考[B2E T-Push SDK文档](#_B2E_T-Push_SDK文档)。

由于SGM出于安全考虑规定SGM设备不允许直接或间接地连接到公网，因此上述推送消息可能因为这一原因无法抵达处于内网（不论是金桥还是其他基地）的IOS类型的SGM设备，因此在选型时应考虑这点以避免消息推送功能无法实现。

## 通过Data Collect发送行为日志

下图描述了移动应用日志如何通过Data Collect 发送到Hadoop数据库。



1. 如果是SGM 设备，它们通过代理将日志信息推送到部署在Internet的Data Collect服务器。
2. 如果是BYOD设备，它们直接将日志数据发送到Data Collect。
3. Data Collect 服务会定时将行为日志发送到SGM 的Hadoop 数据库以供日后大数据分析及其业务的需求使用。

**注：B2E Data-Collect SDK支持移动端发送日志的行为，集成与调用细节请参考**[**B2E Data-Collect SDK文档**](#_B2E_Data-Collect_SDK文档)**。**