**移动终端与服务器底层通讯接口**

本系统中，不仅有消息推送，也有服务器对终端的实时控制（终端定位和程序管理等），再加上移动终端流量和资源的限制，本系统采用如下方案进行数据通信：

* 采用TCP长连接作为终端与服务器底层通信手段。如果终端采用短连接，就需要高频的数据查询通讯，对终端的流量和电量都是不小的消耗；
* 终端使用单通道应答的方式与服务器通信；
* 通讯数据采用SSH进行加密保护，服务器要提供数字证书；
* 服务器要求能同时维持1000个终端的连接通信；
* 心跳监测，服务器和终端都采用TCP默认的心跳监测连接的有效性，空闲时间1分钟，监测次数3次，监测频率15秒；
* 数据格式说明：

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Length | Flag | Body |

命令和响应数据格式相同，并且响应数据的Code与请求命令Code一致。

服务器下发的每条命令要求终端都要有对应的响应，如果命令或响应的数据包过大，要拆成多包发送，对于多包的情况，可发送取消命令，取消下一个包的数据传输。

Code：占一个字节，标识命令。其中高4位标识命令大类（模块），低4位标识内部命令号。

Length：占四字节，为整个数据包的长度（1 + 4 + 1 + Body长度 ）。

Flag：占一个字节，标识位，其含义取决于Code命令值。

Body：具体数据内容。内容格式取决于Code命令值。

各个操作命令的详细说明如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **命令大类** | **子命令** | **Code值** | **主动命令方向** | **Flag** | **Body说明** |
| 终端设备管理(0x0\_) | 登录 | 0x01 | 上行 | 保留（0x00） | 命令：JSON数据格式  {  "info": {  "username": "用户名",  "password": "密码",  "man": "厂商",  "model": "型号",  "imei": "IMEI",  "number": "终端号码，可为NULL",  "netType": "网络类型 mobile or wifi",  "osVersion": "操作系统版本号",  "swVersion": "终端软件版本号"  }  }  响应：JSON数据格式  {  "res": {  "status": "OK or Other",  "swUrl": "升级软件URL地址",  "swSize": "升级软件包大小",  "userId": "登录用户id",  "syncAnchor": "用户信息最后修改时间"  }  }  status内容包括：  OK -> 成功；  NOBIND ->未绑定设备；  LOWSWVER ->终端软件版本过低，需升级；  LOWOSVER ->终端操作系统版本低，需升级；  ERRPW ->用户名或密码错误；  FAIL ->其他错误类型； |
| 退出 | 0x02 | 上行 | 保留（0x00） | 命令：NULL  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FAIL ->其他错误类型； |
| 终端版本更新检测 | 0x03 | 上行 | 保留（0x00） | 命令：JSON数据格式  {  "info": {  "imei": "IMEI",  "osVersion": "操作系统版本号",  "swVersion": "终端软件版本号"  }  }  响应：JSON数据格式  {  "res": {  "status": "NEWVER or Other",  "swVersion": "最新软件版本名",  "swUrl": "升级软件URL地址",  "swSize": "升级软件包大小"  }  }  status内容包括：  NEWVER ->终端已经是最新版本；  NOBIND ->未绑定设备；  LOWSWVER ->终端软件版本过低，需升级；  LOWOSVER ->终端操作系统版本低，需升级；  FAIL ->其他错误类型； |
| 终端丢失 | 0x04 | 下行 | 保留（0x00） | 命令：NULL  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功，移动终端隐私数据已清除；  FAIL ->其他错误类型； |
| 用户信息同步 | 0x05 | 上行 | 0x00：默认；  0x01:后面还有待发数据包；  0x02用户信息未改动； | 命令：JSON数据格式  {  "anchor": "上次同步anchor"  }  响应：JSON数据格式  {  "userInfo": {  "anchor": "最后修改时间",  "group": [  {  "id": "组ID",  "name": "组名",  "user": [  {  "id": "用户ID",  "name": "姓名",  "number": [  "号码1",  "号码2"  ],  "icon": "照片base64"  },  {  "id": "用户ID",  "name": "姓名",  "number": [  "号码1",  "号码2"  ],  "icon": "照片base64"  }  ]  },  {  "id": "组ID",  "name": "组名",  "user": [  {  "id": "用户ID",  "name": "用户姓名",  "number": [  "号码1",  "号码2"  ],  "icon": "照片base64"  },  {  "id": "用户ID",  "name": "姓名",  "number": [  "号码1",  "号码2"  ],  "icon": "照片base64"  }  ]  }  ]  }  }  如果待传输数据量很大，需要多包发送，数据未全部发送Flag设置为0x01,否则为0x00。  如果用户信息最后修改时间和终端上传的anchor一致则不需要下发用户数据（Body为空），Flag设置为0x02。 |
| 用户信息改动通知 | 0x06 | 下行 | 保留（0x00） | 命令：JSON数据格式  {  "newAnchor": "最新用户信息修改时间"  }  响应：NULL |
| 修改密码 | 0x07 | 上行 | 保留（0x00） | 命令：JSON数据格式  {  "changePw": {  "oldPw": "原密码，可以是九宫格密码",  "newPw": "新密码，可以是九宫格密码"  }  }  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  ERRPW ->原密码错误；  FAIL ->其他错误类型； |
| 取消 | 0x0F | 上行  下行 | 保留（0x00） | 命令：NULL  响应：NULL |
| 信息管理（0x1\_） | 信息发送 | 0x11 | 上行  下行 | 0x00：默认；  0x01:后面还有待发数据包；  0x02:该数据包为响应数据包； | 命令：JSON数据格式  {  "message": {  "threadId": "服务器会话id(终端私信无threadId)",  "id": "消息id(终端私信无id、终端回复的消息是临时id)",  "classId": "消息分类id",  "title": "消息标题（终端私信无title） ",  "description": "消息简述（终端私信无description ）",  "createTime": "消息发送时间",  "from": "发送人员id",  "to": {  "userId": "用户id1，用户id2",  "groupId": "用户组id1，用户组id2"  },  "content": "消息正文"  }  }  根据消息环境决定消息所含字段：  服务器操作人员所发起的新消息会话，携带全字段，对该会话中的消息进行回复，携带threadId、id、createTime、from、content；  终端对终端的私信消息，携带classId、createTime、from、to、content；（服务器对私信消息进行转发完成后即可删除该消息）  响应：JSON数据格式  {  "res": {  "status": "OK or Other",  "map": {  "clientId": "终端消息id",  "serverId": "服务器消息id"  }  }  }  status内容包括：  OK -> 成功；  FULL ->终端空间已满；  FAIL ->其他错误类型；  map为终端消息id和服务器消息id的映射，在服务器响应终端的非私信命令时使用。  响应数据包Flag设置为0x02。 |
| 已读回条 | 0x12 | 上行 | 保留（0x00） | 命令：JSON数据格式  {  "readMessage": {  "id": "消息id",  "isRead": "true or false"  }  }  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FAIL ->其他错误类型；  当前只针对服务器操作人员下发的消息进行已读通知。 |
| 信息删除 | 0x13 | 下行 | 保留（0x00） | 命令：JSON数据格式  {  "deleteMessage": {  "threadId": "会话id",  "id": "消息id"  }  }  其中消息id为空表示删除整个会话消息。  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FAIL ->其他错误类型；  服务器操作人员所发起的消息 会话中的所有消息（包括操作人员发的消息和移动终端发的回复消息）只能由服务器操作人员或者服务器消息保留超期删除，移动终端用户不能主动删除；  终端对终端的私信消息由各自终端用户随意删除并且终端用户对私信的删除不影响其他终端。 |
| 信息分类 | 0x14 | 上行  下行 | 0x00：默认；  0x01:后面还有待发数据包；  0x02:该数据包为响应数据包； | 信息分类JSON数据格式为：  {  "messageClass": {  "class": [  {  "id": "消息类id",  "name": "消息类名"  },  {  "id": "消息类id",  "name": "消息类名"  }  ]  }  }  消息分类命令分为终端请求获取分类和服务器主动下发分类两种。   * 终端请求获取分类格式为：   命令：NULL  响应：信息分类JSON数据格式   * 服务器主动下发分类格式为：   命令：信息分类JSON数据格式  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FAIL ->其他错误类型； |
| 取消 | 0x1F | 上行  下行 | 保留（0x00） | 命令：NULL  响应：NULL |
| 终端位置管理(0x2\_) | 启动终端定位 | 0x21 | 下行 | 保留（0x00） | 命令：JSON数据格式  {  "setting": {  "type": "1 or 2",  "interval": "最长时间间隔（分钟）"  }  }  type取值为1（实时定位）和2（历史定位），实时定位实时上报终端位置信息，历史定位粗略以interval间隔上报终端位置。interval在实时定位中无效。  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  CLOSELBS ->终端用户关闭了终端定位设置；  FAIL ->其他错误类型； |
| 停止终端定位 | 0x22 | 下行 | 保留（0x00） | 命令：JSON数据格式  {  "setting": {  "type": "1 or 2"  }  }  type取值为1（实时定位）和2（历史定位）。  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FAIL ->其他错误类型； |
| 终端位置信息上报 | 0x23 | 上行 | 保留（0x00） | 命令：JSON数据格式  {  "locationInfo": {  "type": "1（实时定位） or 2（历史定位） or Other",  "location": [  {  "longitude": "经度",  "latitude": "纬度",  "time": "时间"  },  {  "longitude": "经度",  "latitude": "纬度",  "time": "时间"  }  ]  }  }  Other可能取值为：  CLOSELBS ->终端用户关闭了终端定位设置；  对历史定位，终端可能因为暂时无法和服务器通讯而缓存了一些历史位置信息，当和服务器可通讯时将这些信息一起打包发给服务器。  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FAIL ->其他错误类型； |
| 取消 | 0x2F | 上行  下行 | 保留（0x00） | 命令：NULL  响应:NULL |
| 应用程序管理(0x3\_) | 列表 | 0x31 | 下行 | 0x00：默认；  0x01:后面还有待发数据包； | 命令：NULL  响应：JSON数据格式  {  "packages": {  "package": [  {  "label": "应用程序名",  "name": "包名",  "length": "包大小",  "md5": "安装包MD5值",  "versionCode": "版本号",  "versionName": "版本名",  "minSdkVersion": "所需系统最低版本",  "installLocation": "internal or external",  "usesPermission": [  "使用的权限",  "使用的权限",  "使用的权限"  ]  },  {  "label": "应用程序名",  "name": "包名",  "length": "包大小",  "md5": "安装包MD5值",  "versionCode": "版本号",  "versionName": "版本名",  "minSdkVersion": "所需系统最低版本",  "installLocation": "internal or external",  "usesPermission": [  "使用的权限",  "使用的权限",  "使用的权限"  ]  }  ]  }  }  package为空表示终端未安装任何应用；一条完整的程序信息只能在同一个通讯包中；服务器要保存程序的图标，对于没有的图标（使用安装包MD5判断）从终端获取； |
| 图标获取 | 0x32 | 下行 | 0x00：默认；  0x01:后面还有待发数据包； | 命令：JSON数据格式  {  "packageName": "包名"  }  响应：二进制应用程序图标，为空表示为终端无此应用。 |
| 导出 | 0x33 | 下行 | 0x00：默认；  0x01:后面还有待发数据包； | 命令：JSON数据格式  {  "packageName": "包名"  }  响应：二进制APK安装包，为空表示为终端无此应用。 |
| 安装 | 0x34 | 下行 | 保留（0x00） | 命令：JSON数据格式  {  "package": {  "name": "包名",  "url": "URL地址",  "size": "包大小"  }  }  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FULL ->终端空间已满；  FAIL ->其他错误类型； |
| 安装结果上报 | 0x35 | 上行 | 保留（0x00） | 命令：JSON数据格式  {  "installRes": {  "status": "OK or Other",  "package": {  "label": "应用程序名",  "name": "包名",  "length": "包大小",  "md5": "安装包MD5值",  "versionCode": "版本号",  "versionName": "版本名",  "minSdkVersion": "所需系统最低版本",  "installLocation": "internal or external",  "usesPermission": [  "使用的权限",  "使用的权限",  "使用的权限"  ]  }  }  }  status内容包括：  OK -> 成功；  FULL ->终端空间已满；  WAITING ->没有获取root权限，需要终端用户手动操作安装；  FAIL ->其他错误类型；  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  FAIL ->其他错误类型； |
| 卸载 | 0x36 | 下行 | 保留（0x00） | 命令：JSON数据格式  {  "packageName": "包名"  }  响应：JSON数据格式  {  "res": "OK or Other"  }  res内容包括：  OK -> 成功；  WAITING ->没有获取root权限，需要终端用户手动操作安装；  FAIL ->其他错误类型； |
| 取消 | 0x3F | 上行  下行 | 保留（0x00） | 命令：NULL  响应:NULL |

对上述表格补充说明：

* “主动命令方向”包括“上行”和“下行”，前者是指移动终端主动发起命令给服务器，后者是指服务器发起命令给移动终端；
* 对于服务器发送给终端的携带URL地址的数据包，终端要对该URL地址采用HTTPS协议提取其实际内容；
* 对于命令或响应数据的数据量较大时需要进行多包发送，单包的最大值为500K。无特殊说明时，多包发送如果后面还有数据要发送，本包的Flag设置为0x01，对端收到本个数据包后，要发送对应的继续发送命令（简单原命令）或响应以完成后继数据接收。命令发起方可以发送对应模块取消命令（0x\_F）来取消后继数据发送；
* 用户信息同步，终端设备要保存其所能获取到的所有终端用户信息，并根据需要和服务器进行用户信息同步。服务器记录最新用户信息修改时间，当用户登录时下发该修改时间，当有修改操作时对所有在线设备发送“用户信息改动通知”，由终端设备根据时间判断是否进行用户信息同步；
* 系统（服务器操作人员）发起的消息讨论只能由服务器删除；终端对终端发起的消息，服务器只负责转发，各自的终端用户自己决定删除与否；
* 为避免各个手机时间不一致，服务器将到达服务器消息的发送时间统一改为服务器收到该消息时间；
* 程序管理中以应用程序的包名作为程序的唯一标识；