

修订纪录

版本号	修订描述	作者	修订日期
V0.1	初始版本	许国明	2014年7月
V0.2	增加 XML 描述	许国明、乔恩鑫	2014年8月
V1.0	正式版本	许国明、乔恩鑫	2014年10月
V1.1	增加仅拍照 js 接口, 应用启动参数增加设备号、人员组参数	许国明、乔恩鑫	2015年3月
V1.2	增加轻应用横向屏幕支持	许国明、乔恩鑫	2015年3月
V1.3	增加 Native 应用登录接口	许国明、乔恩鑫	2015年6月
V1.4	向第三方应用传递的参数增加 ip 信息 第三方客户端应用注册增加证书信息。	许国明、乔恩鑫	2015年7月
V1.5	新增扫描二维码接口 新增注册查询代办数量接口	乔恩鑫 乔恩鑫	2015年8月 2015年8月17
V1.6	修改消息推送接口为 Webservice 接口 新增启动 activity 接口	许国明 乔恩鑫	2015年8月19 2015年10月30
V1.7	优化 System 接口 新增 url schema 协议	乔恩鑫	2016-7-12

1. 引言

1.1. 编写目的

本开发接口手册的编写目的是为南京苏迪科技有限公司的微门户平台产品的第三方移动应用供应商提供详细的开发接口描述。

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

同时也为微门户平台的最终用户提供平台集成能力的描述。

本开发接口手册的适用读者，一般为：校方 IT 部门技术人员、第三方移动应用开发人员。

1.2. 系统背景

南京苏迪科技有限公司的微门户平台为高校打造了校园内部的移动应用生态圈，为学生开发团体提供了一个移动应用的创新、创业平台，为数字化校园建设提供了移动门户平台。

平台除了提供基本的资讯、即时通讯、群组、企业通讯录等功能外，还为开发人员提供应用注册、发布申请、版本管理、测试人员管理、评论查看等功能；为校方管理人员提供发布审核、评论审核、应用统计等功能；为终端用户提供应用查询、下载安卓、评论评价等功能。

微门户平台支持多种类型的应用接入，平台为接入的应用提供多种认证接口、界面集成接口、即时消息接口、业务消息扩展接口、移动设备本地能力调用接口等多种接口。

1.3. 术语定义

- **native 应用**
native 应用是指经过编译的直接由移动设备操作系统载入运行的应用。
在安卓系统中，通过开发工具编译后生成的 apk 文件，即 native 应用；
在 ios 系统中，通过开发工具编译后生成的 ipa 文件，即 native 应用。
- **ios inhouse native 应用**
ios 编译的 ipa 应用包，一般需要放入到苹果的 appstore 审核后向使用者分发；
此外，苹果公司还提供了另外一种应用分发模式，即 inhouse 模式应用，此类应用是给企业内部员工（包括学校的师生）使用的，通过企业开发者帐号对应用签名后，不需要发布到 appstore 审核，只需要通过企业内部网站即可分发给使用者。
- **web 应用**
即适配移动设备展示界面的、采用的 BS 模式的应用。此类应用可以采用 html5 技

术，提供对移动特有功能的支持。

- **本地 html5 应用**

是苏迪微门户平台提供的一种基于 html5 的应用，此类应用的资源文件（页面、css、图片、脚本）可以打包后提前下载到微门户客户端中，启动后由页面中的脚本再从服务端获取。

由于资源文件已经在移动设备本地存在，加载资源时直接从本地加载，因此应用的执行效率要比 web 应用高，用户体验较好。

- **窗口部件**

苏迪微门户平台提供的一种界面集成方案。应用中可以注册多个窗口部件，窗口部件可以被用户选择性的加载在微门户客户端的首页。

- **消息部件**

苏迪微门户平台提供的一种消息集成的方案。应用中可以注册多个消息部件，当用户收到应用系统发来的消息后，能以这些消息部件正确的打开并处理。

- **本地能力调用**

是苏迪微门户平台为 web 应用、本地 html5 应用提供的一套 js 接口，通过这些接口，能够调用移动设备的本地能力，比如拍照、定位等功能。

2. 系统支持

- 支持的手机类型

安卓系统，版本 Android 2.3.3 及以上

ios 系统，版本 ios 6.0 及以上

- 支持的应用类型

native 应用(安卓)

native 应用(ios)

web 应用

本地 html5 应用

3. 应用的注册方式

各种应用注册部署到微门户平台基本方式为：编写一个 Manifest.xml 文件，此文件描述了应用的各种注册信息，然后把此文件与应用的其他相关文件打包后，上传到微门户平台应

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

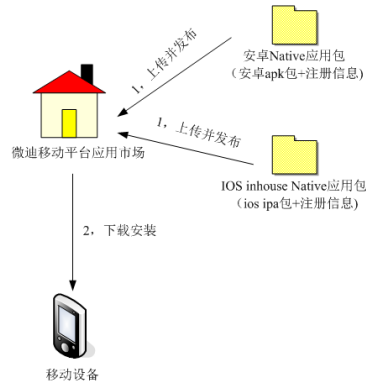
网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

用市场。

- **native 应用包注册到微门户平台**

如下图：

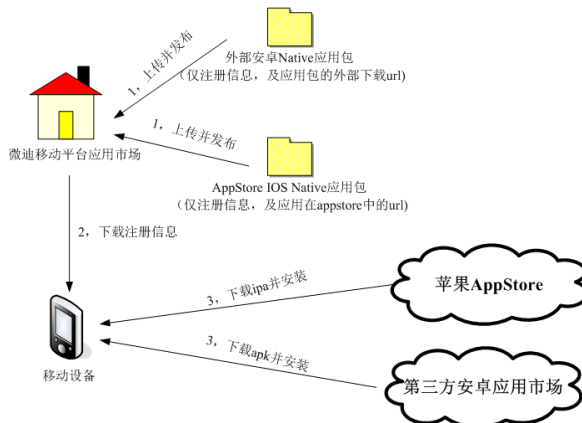


说明：

- 1) 把安卓 apk (或 ios 的 inhouse 模式 ipa) 与注册信息 Manifest.xml 文件打包后上传到微门户应用市场平台，审核并发布
- 2) 在微门户客户端中下载应用并安装

- **第三方市场 native 应用注册到微门户平台**

苏迪微门户平台还提供第三方应用市场及苹果 AppStore 的支持，如下图：

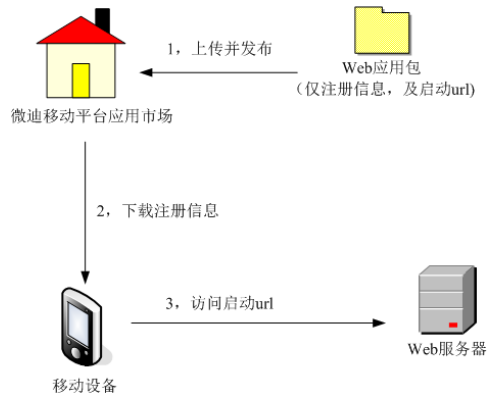


说明：

- 1) 把在第三方安卓市场的 apk 的 url (或 ios app 在 AppStore 中的 url) 写入注册信息 Manifest.xml 文件，打包后上传到微门户应用市场平台，审核并发布
- 2) 在微门户客户端中下载注册信息

3) 微门户客户端自动从第三方安卓市场及苹果 AppStore 下载并安装

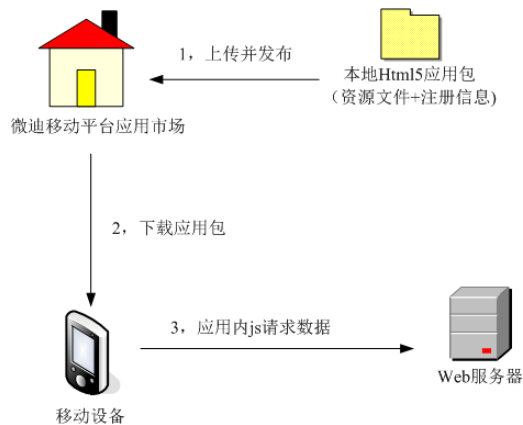
● web 应用注册到微门户平台



说明:

- 1) 把 Web 应用的启动 url 写入注册信息 Manifest.xml 文件, 打包后上传到微门户应用市场平台, 审核并发布
- 2) 在微门户客户端中下载注册信息
- 3) 微门户客户端打开应用时, 通过启动 url 访问 web 服务器。

● 本地 html5 应用注册到微门户平台



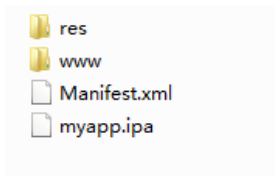
说明:

- 1) 把本地 html5 的资源文件(包括 html 页面)及注册信息 Manifest.xml 文件, 打包后上传到微门户应用市场平台, 审核并发布

- 2) 在微门户客户端中应用包
- 3) 微门户客户端打开应用时, 首先加载本地页面, 页面中的 js 访问 web 服务器获取数据后中展示给使用者。

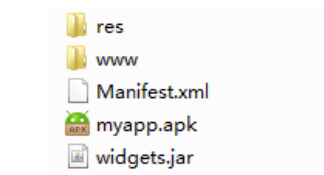
4. 应用的打包方式

部署到微门户平台中的应用, 都必须打好 zip 包上传。你个人机本地文件所在的根目录不要打包在内, 打包后 zip 文件的结构应该如下图:



以下是各种类型应用的文件结构:

■ native 应用(安卓)



说明:

res 文件夹, 必须存在, 其内放置图标、截屏等图片。

Manifest.xml, 必须存在, 是应用相关信息的描述。

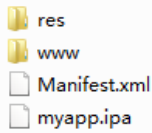
myapp.apk 文件按安卓应用程序标准方式编译打包, 文件名不能为中文。

注: 如果安装模式是“仅注册”或“外部安装”, apk 文件可以不打包在应用包中。参见“[安装模式](#)”

widgets.jar 文件可选, 文件名必须是 widgets.jar。此文件是安卓平台上微门户客户端中 native 窗口部件。

www 目录是可选的

■ native 应用(ios)



说明:

res 文件夹, 必须存在, 其内放置图标、截屏等图片。

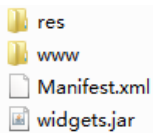
Manifest.xml, 必须存在, 是应用相关信息的描述。

myapp.ipa 文件, 按照苹果标准编译打包 ipa 应用包, 并使用\$299 企业账号签名。文件名不能是中文。

注: 如果安装模式是“仅注册”或“外部安装”, ipa 文件可以不打包在应用包中。参见“[安装模式](#)”

www 目录是可选的。

■ web 应用



说明:

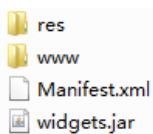
res 文件夹, 必须存在, 其内放置图标、截屏等图片。

Manifest.xml, 必须存在, 是应用相关信息的描述。

widgets.jar 文件可选, 文件名必须是 widgets.jar。此文件是 web 应用在安卓微门户客户端中 native 窗口部件。

web 应用的 www 目录是可选的。

■ 本地 html5 应用



说明:

res 文件夹, 必须存在, 其内放置图标、截屏等图片。

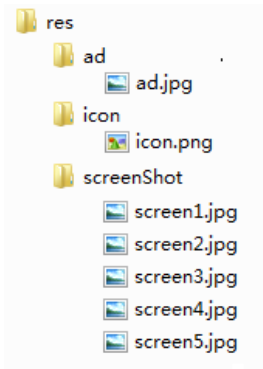
Manifest.xml, 必须存在, 是应用相关信息的描述。

www 目录, 必须存在, 其内存放本地 html5 的页面文件。

widgets.jar 文件可选, 文件名必须是 widgets.jar。此文件是 web 应用在安卓微门户客户端中 native 窗口部件。

4.1.res 文件夹

res 文件夹的结构如下:



各子文件夹及文件说明:

- icon 子文件
必须存在, 其内存放应用的图标 icon.png
- icon.png -- 应用的图标
格式: 必须是 png 格式
大小: 120x120 pix
- screenShot 子文件夹
必须存在, 其内放 1 至 5 张应用的屏幕截图。
- screenX.jpg
应用截图。至少 1 个, 最多 5 个。
文件名按顺序为 screen1.jpg, ... screen.5.jpg
格式: 必须是 jpg
大小: 宽 720 pix , 高 1280 pix
- ad 子文件夹
必须存在, 存放应用被“置顶”时的展示图片。
- ad.jp
必须存在, 文件名固定为 ad.jpg
格式: jpg
大小: 宽 640pix , 高 200 pix

4.2.www 文件夹

www 文件夹内存放本地 html 文件, 对于本地 html5 应用是必选的, 其内存放应用的程序页面, 以及本地 html 的窗口部件与消息组件。

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410

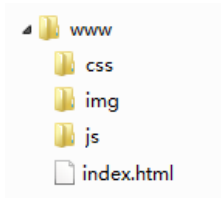
传真: 86-25-84512645 转 605

网址: <http://www.sudotech.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

www 文件夹对于其他应用是可选的。对于其他应用来说，如果存在 www 文件夹，其内存放的是本地 html5 窗口部件、及消息组件。

www 文件夹内的文件及子文件夹可以自由组织，推荐的结构如下：



4.3.widget.jar

此 jar 内是编译过的 java 代码，是微门户客户端在安卓平台中 native 窗口部件。

widget.jar 仅对安卓平台的移动设备有效。

可以选用于以下类型的应用：

- ◆ native 应用(安卓)
- ◆ web 应用
- ◆ 本地 html5 应用

参见 [界面集成-窗口部件](#)

4.4.apk 文件与 ipa 文件

apk 文件是 native 应用（安卓）的主程序包，需按安卓应用程序标准方式编译打包，文件名不能为中文。

ipa 文件是 native 应用（ios）的主程序包，需按照苹果标准编译打包 ipa 应用包，并使用\$299 企业账号签名。文件名不能是中文。

4.5.Manifest.xml

Manifest.xml 是应用的描述文件，其内包括应用的启动入口、认证方式、窗口部件、消息部件等描述。

样例：

```
<?xml version="1.0" encoding="UTF-8"?>
<Application xmlns="http://www.sudytech.com/schemas/iportal/iapp"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.sudytech.com/schemas/iportal/iapp
```

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytech.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

```
http://www.sudytech.com/schemas/iportal/iapp/manifest-def.xsd"
name="com.hello.myapp"
installMode="download"
externalUrl=""
type="local-html"
versionCode="1"
versionName="v1.0" >
<targetOS minVersion="2.2.3" name="android"/>
<targetOS minVersion="5.0" name="ios"/>
<friendlyName lang="cn" name="我的应用"/>
<keywords><![CDATA[微门户,练习]]></keywords>
<description><![CDATA[我的第一个应用]]></description>
<whatsNew>这是第一个版本</whatsNew >
<vender>张三</vender>
<author>张三</author >
<entrance xsi:type="urlEntrance" url="/index.html" hideNavigationBar="true"/>
<authentication type="unified"/>
< transactionCountUrl>http://mail.com/queryTransactionCount</ transactionCountUrl>
< injectJs><![CDATA[
var iPortal={noAjax:true};
iPortal.onReady=function(){alert("ok");}
]]</ injectJs>
<widgets>
  <widget type="native" name="map" verticalTouchevent="true" height="60">
    <entrance xsi:type="classEntrance"
      packageName="com.hello.myapp"
      className="my.test.MapView">
      <fixParam name="myParam" value="123"/>
      <fixParam name="myParam2" value="456"/>
    </entrance>
    <friendlyName lang="cn" name="地图"/>
    <description>地图部件</description>
  </widget>
  <widget type="local-html" name="news" verticalTouchevent="true" height="60">
    <entrance xsi:type="urlEntrance" url="/news/news.html"/>
    <friendlyName lang="cn" name="新闻"/>
    <description>新闻部件</description>
  </widget>
</widgets>
<jsPermissions>
  <jsPermission name="iapp.perm.Take_Photo"/>
</jsPermissions>
```

```
<jsPermission name="iapp.perm.Send_SMS"/>
</jsPermissions>
<urlScopes>
  <urlScope>http://www.hello.com</urlScope>
  <urlScope>http://www.baidu.com/maps</urlScope>
</urlScopes>
<messageComponents>
  <messageComponent type="web" bizType="1">
    <entrance xsi:type="urlEntrance"
      url="http://www.hello.com/msg/showMessage.jsp"/>
    <description>处理待办消息</description>
  </messageComponent>
  <messageComponent type="local-html" bizType="default">
    <entrance xsi:type="urlEntrance" url="/msg/default.html"/>
    <description>处理其他业务消息</description>
  </messageComponent>
</messageComponents>
</Application>
```

5. 应用基本信息

在微门户平台中注册的应用，需要在 Manifest.xml 中描述应用名、应用类型、目标平台、版本、关键词、启动入口等基本信息。

Manifest.xml 中，Application 节点的形式如下：

```
<Application
  name=" com.hello.myapp "
  type="local-html"
  installMode="download"
  versionCode="1"
  versionName="v1.0" >
```

5.1. 应用名

以 Application 节点的属性 name 表示，类型是字符串。此属性名用于鉴别不同的应用。为避免与其他应用名冲突。

- ios native 应用
此 name 的值，必须要与 ios app 的 bundle Id 保持一致。

- 安卓 native 应用
此 name 的值，必须要与 android app 的 packageName 保持一致。
- Web 应用与本地 html5 应用
此 name 的值可以任意取，但不能与系统中已注册的其它任何应用的 name 相同。建议使用类似 java 类包名的命名方式，以避免冲突。推荐应该使用格式：“反转域名+点+名称”，比如：公司域名为 sudytech.com，应用名 map。则应用的 name 应该使用：“com.sudytech.map”。

5.2. 应用类型

以 **Application** 节点的属性 **type** 表示，类型是字符串，值必须是以下之一

- "native" -- 本地应用
- "web" -- 远程 web 应用
- "local-html" -- 本地 html5 应用

5.3. 证书指纹

证书指纹是指 native 应用的代码签名证书的用 SHA1 算法得到的 hash 值，以 **Application** 节点的属性 **certSHA1** 表示，类型是字符串，值是证书的 SHA1 算法计算结果的 16 进制字符，比如：

```
<Application
  name=" com.hello.myapp "
  type=" native "
  installMode="download"
  certSHA1="BE2CC3FC8F32C1D1CAE5771D8280E7BE09EF9A3B"
  versionCode="1"
  versionName="v1.0" >
```

此属性可以为空，但是如果是安卓应用，且需要主动调用微门户的登录接口时（参见 [Native App 登录接口](#)），必须提供此值，否则微门户客户端将拒绝登录接口调用。

指纹查看方式，参考[附录 1 APK 指纹获取](#)

5.4. 目标平台与兼容性

表示本应用可以在哪种类型的移动设备使用。

以 **Application** 的子节点 **targetOS** 表示，如下：

```
<targetOS minVersion="2.2.3" name="android"/>
```

属性 **name**，目标平台的名称，字符串，必须是以下值：

"android" -- 目标平台是安卓平台，

"ios" -- 目标平台是 ios 平台

属性 **minVersion**，目标平台的版本，字符串，格式为"n.n"或"n.n.n"。

比如：

```
<targetOS minVersion="5.0" name="ios"/>
```

表示要求客户端的是 ios，最小版本是 5.0

注意：

targetOS 节点必须要有。

此节点最多能有 2 个，即 web 应用和本地 html5 应用可以指定多个目标平台。

对于 native 应用，此节点只能有一个。

5.5. 安装模式

安装模式表示，应用在微门户平台中如何安装到用户的客户端上。

以 **Application** 节点的属性 **installMode** 表示，类型是字符串，其值必须是以下之一：

"download" -- 表示需要下载安装。web 应用，本地 html5 应用只能选择此值。native 应用一般选择此值。

"register" -- 仅 native 应用可以选择此值，表示应用已经提前通过其他方式安装到了用户客户端上，仅需要在微门户平台注册应用信息。此时上传到微门户的应用包中，可以没有 apk 文件或 ipa 文件。

"external" -- 仅 native 应用可以选择此值，表示应用需要通过微门户平台以外的网站来下载安装。

■ externalUrl

当 **installMode= "external"**时，需要为 **Application** 指定另外一个属性 **externalUrl** 来指定应用的下载地址。

● 对于安卓应用

externalUrl 是一个外部 URL，指向 apk 的文件即可。

举例：

```
<Application
  name=" com.hello.myapp "
  type="native"
  installMode="external"
  externalUrl="http://www.my.com/app/hello.apk"
  versionCode="1"
```

```
versionName="v1.0" >
```

- 对于 ios 应用

如果是指向苹果的 appStore 内的应用，需要以如下方式指定：

externalUrl = "itms-apps:346703830"，"itms-apps:" 是固定的，后面的数字是 ios 应用的 bundleId。

举例：

```
<Application
  name=" com.hello.myapp "
  type="native"
  installMode="external"
  externalUrl="itms-apps:346703830"
  versionCode="1"
  versionName="v1.0" >
```

如果是指向第三方开发的 in-house 应用，则 itms 的 url 如下：

itms-services://?action=download-manifest&url= https://www.my.com/app/hello.plist

举例：

```
<Application
  name=" com.hello.myapp "
  type="native"
  installMode="external"
  externalUrl="itms-services://?action=download-manifest&url=
https://www.my.com/app/hello.plist"
  versionCode="1"
  versionName="v1.0" >
```

5.6. 版本号

应用的版本号，以 **Application** 节点的属性 **versionCode** 表示，类型是整数。如果在微门户平台已经上传应用，而现在要创建一个新版本，那么此数值必须比上一个版本的版本号大。如下：

```
versionCode="1"
```

5.7. 版本名称

用户可以看懂的版本名称，以 **Application** 节点的属性 **versionCode** 表示，类型是字符串。值可以是任意字符串。如下：

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

```
versionName="v1.0"
```

5.8. 友好名称

即给用户看的应用名称，以 **Application** 的子节点 **friendlyName** 表示，如下：

```
<friendlyName lang="cn" name="我的应用"/>
```

属性 **lang**：友好名称的语言，目前只支持“cn”，即中文。

属性 **name**：友好名称的值

目前微门户平台只支持一个中文的友好名称，即节点 **friendlyName** 只允许出现一个，其 **lang="cn"**。

5.9. 关键词

本应用的关键词描述，以“,”分割多个关键词。

以 **Application** 的子节点 **keywords** 表示，如下：

```
<keywords>[CDATA[微门户,练习]]</keywords>
```

5.10. 简介

本应用的简要介绍。

以 **Application** 的子节点 **description** 表示，如下：

```
<description>[CDATA[我的第一个应用]] </description>
```

5.11. 更新说明

如果本应用是上个版本的升级版本，应该给出更新说明。

以 **Application** 的子节点 **whatsNew** 表示，如下：

```
<whatsNew>这是第一个版本</whatsNew >
```

5.12. 供应商

描述应用的提供者，以 **Application** 的子节点 **vender** 表示，如下：

```
<vender >张三</ vender >
```

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

5.13. 查询待办数量 url

客户端通过此 url 查询应用的待办数量，并在应用图标上面显示提醒数量。微门户客户端访问 url 时，微门户客户端会根据认证接入方式向应用传递用户的认证信息

```
< transactionCountUrl>http://mail.com/queryTransactionCount</ transactionCountUrl>
```

此 url 接口规范：

- 1，此 url 要求为使用 http 协议的 get 方式。
- 2，传入参数参考[认证接入方式](#)
- 3，返回内容格式如下：

```
{  
  "transactionCount":1  
}
```

5.14. 支持 web 应用 js 注入

对于 web 应用，可以编写一段注入 js。移动平台每次打开这个 url，或者当 WebView 的 url 发生改变的时候，会在页面加载完成的时候，注入此 js。以便于对一些第三方提供对网页做二次定制。

```
< injectJs><![CDATA[  
var iPortal={noAjax:true};  
iPortal.onReady=function(){alert("ok");}  
]]></ injectJs>
```

6. 应用的启动入口

除了 native 应用在安装后可以从移动设备操作系统直接启动外；web 应用、本地 html5 应用都必须从微门户客户端启动。native 应用也可以选择从微门户客户端启动。

从微门户客户端启动时，微门户客户端会根据需要向应用传递用户的认证信息。

应用必须在 Manifest.xml 中注册应用的启动入口，以 Application 节点的子节点 entrance 表示。

有两种类型：url 入口与本地代码的类(class)入口，分别如下：

- url 入口

```
<entrance xsi:type="urlEntrance" url="/index.html" hideNavigationBar="true" />
```

说明:

web 应用、本地 html5 应用、native 应用 (ios) 必须使用 url 入口。

属性: `xsi:type="urlEntrance"` , 固定格式, 表示 此 entrance 节点是 url 入口

属性: `url` 是应用的启动入口

属性: `hideStatusBar` 是否隐藏状态栏, 默认 false,

属性: `hideNavigationBar` 表示打开 url 时是否隐藏导航条, 默认 false

属性: `hideCloseButton` 隐藏关闭按钮, 默认 false。本属性在 `hideNavigationBar=false` 时有效。

属性: `hideBackButton` 隐藏返回按钮, 默认 false。本属性在 `hideNavigationBar=false` 时有效。

属性: `title` 导航条标题, 默认是应用名称。本属性在 `hideNavigationBar=false` 时有效。

属性: `orientation` 默认是 portrait, 取值: portrait(竖屏), landscape(横屏), sensor (感应)

✓ 对于 web 应用

url 是绝对地址, 形如: `http://myserver.com/myapp/index.php`.

✓ 本地 html5 应用

url 是相对地址, 指向应用包中 www 文件夹下的文件, 比如 `"index.html"`, 表示入口页面是 www 文件夹的 `index.html` 文件。

✓ native 应用(ios)

url 是 ios app 的 schema 协议, 如: `mm-qq://user/123456`。如何给 ios app 注册 schema 协议, 参考 ios 开发手册。

备注: 由于 ios7 及以上, StatusBar 没有独立高度, 必须悬浮在导航条上的, 如果应用使用自己的导航条, 必须判断当前的系统版本 (判断方式参考 Device 接口), 如果是 ios7 及以上, 必须将自己的导航条预留 20px 的高度用来显示状态栏。

■ class 入口

```
<entrance xsi:type="classEntrance"
  packageName="com.hello.myapp"
  className="my.test.myapp.MainActivity">
  <fixParam name="myParam" value="123"/>
  <fixParam name="myParam2" value="456"/>
</entrance>
```

说明:

仅用于 native 应用 (安卓)

属性: `xsi:type="classEntrance"`, 固定格式, 表示 此 entrance 节点是 url 入口

属性: `packageName="com.hello.myapp"`, 安卓应用的包名

属性: `className="my.test.myapp.MainActivity"`, 安卓应用的主启动 Activity。

注意: 安卓开发环境编译应用时, 启动 Activity 的在安卓应用的 `AndroidManifest.xml` 中, `intent-filter` 必须如下形式

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

子节点: `<fixParam name="myParam" value="123"/>`

此节点可以有多个, 从微门户客户端启动 native 应用(安卓)时, 会把这些参数传递给 MainActivity。MainActivity 可以调用 `this. getIntent()` 来获取传递过来的参数。如何使用 Intent, 请参见安卓开发文档。

7. 认证接入方式

任何类型的应用, 如果要在微门户中启动, 可以指定启动时的认证接入方式。微门户平台提供以下几种认证接入方式: 匿名、未认证的身份、统一认证、伪认证。

应用必须在 Manifest.xml 中注册应用的认证接入方式, 以 `Application` 节点子节点 `authentication` 表示。如下:

```
<authentication type="unified"/>
```

说明:

属性: `type` 表明了认证接入方式, 其值必须是以下之一:

- ✓ anonymous
- ✓ trustedAnonymous
- ✓ notVerified
- ✓ unified
- ✓ proxy

7.1. 匿名访问

以 `<authentication type="anonymous"/>` 注册时, 表示应用启动时, 不需要任何身份信息。微门户客户端在启动应用时, 不会向其传递任何用户信息。

以匿名方式启动应用时, 因为微门户客户端不需设置任何用户登录凭证以及签名信息, 启动速度比较快。因此, 建议对完全不需要登录用户信息的应用, 应该使用此设置。

7.2. 可信的匿名访问

以`<authentication type="trustedAnonymous"/>`注册时，表示应用启动时，不需要任何身份信息，但是应用需要确认请求是来自于微门户客户端的。此时微门户客户端在启动应用时，仅会向其传递微门户签名信息。

- native 应用(安卓)

微门户启动 native 应用（安卓）时，将为 Intent 设置如下参数：

```
putExtra("iportal.timestamp", "1408355477445"); //签名时间戳
putExtra("iportal.nonce", "8472482512"); //签名随机数
putExtra("iportal.signature", "AE0F37BC98D28B057"); //签名
```

应用启动后可以调用 `Intent.getStringExtra("iportal.signature")` 来获取微门户的签名，并验证请求来自与微门户客户端。验证算法参见：[验证请求的合法性](#)。

- native 部件(安卓)

微门户在初始化 native 部件时传入的 context 中包含如下扩展信息。微门户启动类，需要实现

```
ExtraContext ec = (ExtraContext)context;
ec.getExtra("iportal.timestamp"); //签名时间戳
ec.getExtra("iportal.nonce"); //签名随机数
ec.getExtra("iportal.signature"); //签名
```

参见[安卓 native 窗口部件](#)

- native 应用(ios)

通过应用注册的 schema 入口 url 启动，附加如下参数：

如：注册的入口 url 为 `mqq://user/1234`

```
mqq://user/1234?iportal.timestamp=1408355477445 & iportal.nonce=8472482512 &
iportal.signature=AE0F37BC98D28B057&iportal.backUrl=iportal%3A%2F%2Fback
```

参数说明：

iportal.backUrl: 从 ios 返回到微门户客户端的 schema，ios 应用可以此参数表示的 shema 返回到微门户客户端。

其他参数：ios 应用获取时间戳、随机数、签名后，可以验证请求是否来自于微门户客户端。验证算法参见：[验证请求的合法性](#)。

- web 应用/部件

假设 web 应用（以及 web 部件）的启动 url 为：`http://www.my.com/myapp`

微平台启动将如此启动 url：

```
http://www.my.com/myapp? iportal.timestamp =1408355477445 & iportal.nonce  
=8472482512 & iportal.signature = AE0F37BC98D28B057
```

web 应用的服务端，获取时间戳、随机数、签名后，可以验证请求是否来自与微门户客户端。验证算法参见：[验证请求的合法性](#)。

■ 本地 html5 应用/部件

本地 html5 应用（以及本地 html5 部件）被启动后，页面的 Dom 中将存在一个全局对象 iPortal，启动页面的脚本可以按如下方法获得时间戳、随机数、签名。

```
var timeStamp = iPortal.getContext().getExtra("iportal.timestamp");  
var nonce = iPortal.getContext().getExtra("iportal.nonce");  
var signature = iPortal.getContext().getExtra("iportal.signature");
```

验证算法参见：[验证请求的合法性](#)。

js 调用参见：[本地能力 JS 调用](#)

7.3. 微门户认证的身份

以<authentication type="notVerified"/>注册时，表示应用启动时，需要从微门户客户端获知当前的登录用户 id 与姓名，并能验证请求来自于微门户客户端，但应用不需把此身份到第三方认证系统进行验证。

微门户客户端在启动应用时，会向其传递用户 id、姓名、微门户签名等基本信息，但不会向其传递任何用户在第三方认证系统的登录凭证信息。

■ native 应用(安卓)

微门户启动 native 应用时，将为 Intent 设置如下参数：

```
putExtra("iportal.uid","123"); //登录用户 id  
putExtra("iportal.uxid","abcd456"); //登录用户在统一认证系统中的 id  
putExtra("iportal.uname","zhangsang"); //登录用户姓名
```

```
putExtra("iportal.timestamp","1408355477445"); //签名时间戳  
putExtra("iportal.nonce","8472482512"); //签名随机数  
putExtra("iportal.signature","AE0F37BC98D28B057"); //签名
```

```
putExtra("iportal.group","教职工,领导"); //用户所在的全局组  
putExtra("iportal.device","a123b91c8371e"); //用户的设备号  
putExtra("iportal.signature2","DA0837B39CD35A1D6"); //签名 2
```

```
putExtra("iportal.ip","172.18.10.18"); //客户端 ip 地址，如果是 ipv4，则以  
“172.18.10.18”形式表示，如果是 ipv6，则以“2001:DB8:2de::e13”形式表示
```

```
putExtra("iportal.signature3","1CB6D7C30135F4B9"); //签名 3
```

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

应用启动后可以调用 `Intent.getStringExtra("iportal.uid ")` 来获取登录用户 id。
调用 `Intent.getStringExtra("iportal.uxid ")` 来获取登录用户在统一认证系统中的 id
时间戳、签名随机数、签名用于验证请求的合法性，验证算法参见：[验证请求的合法性](#)。

■ native 部件(安卓)

微门户在初始化 native 部件时传入的 context 中包含如下扩展信息。

```
ExtraContext ec = (ExtraContext)context;  
ec.getStringExtra("iportal.uid");//登录用户 id  
ec.getStringExtra("iportal.uxid");//登录用户在统一认证系统中的 id
```

```
ec.getStringExtra("iportal.uname");//登录用户姓名  
ec.getStringExtra("iportal.timestamp");//签名时间戳  
ec.getStringExtra("iportal.nonce");//签名随机数  
ec.getStringExtra("iportal.signature");//签名  
ec.getStringExtra("iportal.group");//用户所在的全局组  
ec.getStringExtra("iportal.device");//用户的设备号  
ec.getStringExtra("iportal.signature2");//签名 2
```

```
ec.getStringExtra("iportal.ip");//客户端 ip 地址，如果是 ipv4，则以“172.18.10.18”形式表示，如果是 ipv6，则以“2001:DB8:2de::e13”形式表示  
ec.getStringExtra("iportal.signature3");//签名 3
```

验证算法参见：[验证请求的合法性](#)。

另，[安卓 native 窗口部件](#)

■ native 应用(ios)

通过应用注册的 schema 入口 url 启动，附加如下参数：

```
如：注册的入口 url 为 mqq://user/1234  
mqq://user/1234?iportal.uid=123&iportal.uxid  
=abcd456&iportal.uname=zhangshan&iportal.timestamp  
=1408355477445&iportal.nonce=8472482512&iportal.signature=  
AE0F37BC98D28B057&iportal.group=teacher,student&iportal.device=  
a123b91c8371e&iportal.signature2=DA0837B39CD35A1D6&iportal.backUrl=  
iportal%3A%2F%2Fback&iportal.ip=172.18.10.18  
&iportal.signature3=1CB6D7C30135F4B9
```

参数含义：

iportal.uid 登录用户的 id
iportal.uxid 登录用户在统一认证系统中的 id
iportal.username:用户的姓名
iportal.group:用户所在的全局组
iportal.device:用户的设备号
iportal.backUrl: 从 ios 返回到微门户客户端的 schema, ios 应用可以通过此表示的 schema 返回到微门户客户端。
iportal.ip: 客户端 ip 地址,如果是 ipv4, 则以“172.18.10.18”形式表示, 如果是 ipv6, 则以“2001:DB8:2de::e13”形式表示

另外时间戳、签名随机数、签名用于验证请求的合法性, 验证算法参见: [验证请求的合法性](#)。

■ web 应用/部件

假设 web 应用 (以及 web 部件) 的启动 url 为: `http://www.my.com/myapp`

微平台启动将如此启动 url: `http://www.my.com/myapp? iportal.uid =123&iportal.uxid =abcd456&iportal.username=zhangsan&iportal.timestamp =1408355477445&iportal.nonce=8472482512&iportal.signature= AE0F37BC98D28B057&iportal.group= 教 职 工 , 领 导 & iportal.device= a123b91c8371e& iportal. signature2= DA0837B39CD35A1D6&iportal.ip=172.18.10.18 &iportal.signature3=1CB6D7C30135F4B9`

传递参数含义如下:

iportal.uid: 用户的 id

iportal.uxid 登录用户在统一认证系统中的 id

iportal.username:用户的姓名, 值将以 utf-8 格式编码后, 再将经过 urlEncode 转码后传递。

iportal.group:用户所在的全局组

iportal.device:用户的设备号

iportal.ip: 客户端 ipv4 地址,如果是 ipv4, 则以“172.18.10.18”形式表示, 如果是 ipv6, 则以“2001:DB8:2de::e13”形式表示

另外时间戳、签名随机数、签名用于验证请求的合法性, 验证算法参见: [验证请求的合法性](#)。

■ 本地 html5 应用/部件

本地 html5 应用 (以及本地 html5 部件) 被启动后, 页面的 Dom 中将存在一个全局对象 iPortal, 启动页面的脚本可以按如下方法获得当前登录用户 id 与用户姓名, 已经用户在统一认证系统中的 id。

```
var userId = iPortal.getContext().getExtra("iportal.uid");
```

```
var userXid = iPortal.getContext().getExtra("iportal.uxid");
```

```
var userName = iPortal.getContext().getExtra("iportal.username");
```

```
var userGroups = iPortal.getContext().getExtra("iportal.group"); //用户组
var deviceId = iPortal.getContext().getExtra("iportal.device "); //设备号
var ip = iPortal.getContext().getExtra("iportal.ip"); //客户端 ip 地址,如果是 ipv4, 则以
“172.18.10.18” 形式表示, 如果是 ipv6, 则以“2001:DB8:2de::e13”形式表示
```

按如下方法获得时间戳、随机数、签名。

```
var timeStamp = iPortal.getContext().getExtra("iportal.timestamp");
var nonce = iPortal.getContext().getExtra("iportal.nonce");
var signature = iPortal.getContext().getExtra("iportal.signature");
var signature2 = iPortal.getContext().getExtra("iportal.signature2");
var signature3 = iPortal.getContext().getExtra("iportal.signature3");
```

验证算法参见：[验证请求的合法性](#)。

js 调用参见：[本地能力 JS 调用](#)

7.4. 统一认证

以<authentication type="unified"/>注册时, 表示应用启动时, 应用需要以安全的方式, 从校方的统一身份认证系统获知当前客户端上的登录用户。

微门户客户端在启动应用时, 除了向其传递用户 id、姓名、微门户签名等基本信息, 对于 web 应用, 本地 html5 应用, 还会通过 cookie 隐式传递用户在统一认证系统中的登录凭证。

应用系统服务端得到客户端发来的请求后, 需要向统一认证系统验证用户的登录信息。

对于 native 应用/部件, 微门户平台出于安全考虑, 暂不向其传递用户登录凭证, 也就是说, 微门户平台暂不支持 native 应用向统一认证系统验证用户信息。但是 native 应用可以通过签名验证, 验证请求确实是来自于微门户客户端的。验证算法参见：[验证请求的合法性](#)。

■ native 应用(安卓)

微门户启动 native 应用时, 将为 Intent 设置如下参数:

```
putExtra("iportal.uid","123"); //登录用户 id
putExtra("iportal.uxid","abcd456"); //登录用户在统一认证系统中的 id
putExtra("iportal.uname","zhangsan"); //登录用户姓名
```

```
putExtra("iportal.timestamp","1408355477445"); //签名时间戳
putExtra("iportal.nonce","8472482512"); //签名随机数
putExtra("iportal.signature","AE0F37BC98D28B057"); //签名
```

```
putExtra("iportal.group","教职工,领导"); //用户所在的全局组
putExtra("iportal.device","a123b91c8371e"); //用户的设备号
putExtra("iportal.signature2","DA0837B39CD35A1D6"); //签名 2
```

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

```
putExtra("iportal.ip","172.18.10.18"); //客户端 ip 地址,如果是 ipv4,则以  
“172.18.10.18”形式表示,如果是 ipv6,则以“2001:DB8:2de::e13”形式表示  
putExtra("iportal.signature3","1CB6D7C30135F4B9"); //签名 3
```

应用启动后可以调用 `Intent.getStringExtra("iportal.uid")` 来获取登录用户 id。

调用 `Intent.getStringExtra("iportal.uxid")` 来获取登录用户在统一认证系统中的 id 时间戳、签名随机数、签名用于验证请求的合法性,验证算法参见: [验证请求的合法性](#)。

■ native 部件 (安卓)

微门户在初始化 native 部件时传入的 context 中包含如下扩展信息。

```
ExtraContext ec = (ExtraContext)context;
```

```
ec.putExtra("iportal.uid");//登录用户 id
```

```
ec.putExtra("iportal.uxid");//登录用户在统一认证系统中的 id
```

```
ec.putExtra("iportal.uname");//登录用户姓名
```

```
ec.putExtra("iportal.timestamp");//签名时间戳
```

```
ec.putExtra("iportal.nonce");//签名随机数
```

```
ec.putExtra("iportal.signature");//签名
```

```
ec.putExtra("iportal.group");//用户所在的全局组
```

```
ec.putExtra("iportal.device");//用户的设备号
```

```
ec.putExtra("iportal.signature2");//签名 2
```

```
ec.putExtra("iportal.ip");//客户端 ip 地址,如果是 ipv4,则以“172.18.10.18”形式表示,如果是 ipv6,则以“2001:DB8:2de::e13”形式表示
```

```
ec.putExtra("iportal.signature3");//签名 3
```

验证算法参见: [验证请求的合法性](#)。

另参见, [安卓 native 窗口部件](#)

■ native 应用(ios)

通过应用注册的 schema 入口 url 启动,附加如下参数:

如:注册的入口 url 为 `mqq://user/1234`

```
mqq://user/1234?iportal.uid=123&iportal.uxid
```

```
=abcd456&iportal.uname=zhangshan&iportal.timestamp
```

```
=1408355477445&iportal.nonce=8472482512&iportal.signature=
```

```
AE0F37BC98D28B057&iportal.group= 教 职 工 , 领 导 & iportal.device=
```

```
a123b91c8371e&iportal.signature2= DA0837B39CD35A1D6&iportal.backUrl =
```

```
iportal%3A%2F%2Fback&iportal.ip=172.18.10.18
```

```
&iportal.signature3=1CB6D7C30135F4B9
```

参数含义:

iportal.uid 登录用户的 id
iportal.uxid 登录用户在统一认证系统中的 id
iportal.uname:用户的姓名
iportal.group:用户所在的全局组
iportal.device:用户的设备号
iportal.backUrl: 从 ios 返回到微门户客户端的 schema, ios 应用可以通过此表示的 shema 返回到微门户客户端。
iportal.ip: 客户端 ip 地址, 如果是 ipv4, 则以“172.18.10.18”形式表示, 如果是 ipv6, 则以“2001:DB8:2de::e13”形式表示

另外时间戳、签名随机数、签名用于验证请求的合法性, 验证算法参见: [验证请求的合法性](#)。

■ web 应用/部件

假设 web 应用 (以及 web 部件) 的启动 url 为: `http://www.my.com/myapp`
微平台启动将如此启动 url: `http://www.my.com/myapp? iportal.uid =123&iportal.uxid =abcd456&iportal.uname=zhangsan&iportal.timestamp =1408355477445&iportal.nonce=8472482512&iportal.signature= AE0F37BC98D28B057& iportal.group=教职工、领导& iportal.device= a123b91c8371e& iportal. signature2= DA0837B39CD35A1D6&iportal.ip=172.18.10.18 &iportal.signature3=1CB6D7C30135F4B9`

传递参数含义如下:

iportal.uid: 登录用户的 id
iportal.uxid 登录用户在统一认证系统中的 id
iportal.uname:用户的姓名, 值将以 utf-8 格式编码后, 再将经过 urlEncode 转码后传递。
iportal.group:用户所在的全局组
iportal.device:用户的设备号
iportal.ip: 客户端 ip 地址, 如果是 ipv4, 则以“172.18.10.18”形式表示, 如果是 ipv6, 则以“2001:DB8:2de::e13”形式表示

另外时间戳、签名随机数、签名用于验证请求的合法性, 验证算法参见: [验证请求的合法性](#)。

注: 在统一认证模式下, 参数 iportal.uid、iportal.uxid、iportal.uname 仅供参考, 微门户客户端会在浏览器中通过 cookie 隐式传递用户在统一身份认证系统的登录凭证, 应用的服务的按统一认证系统原有的协议即可以验证用户登录信息。

✓ 会话失效处理

服务端发现会话失效时，应该按统一认证系统原有的协议向客户端发出重定向即可。

■ 本地 html5 应用/部件

本地 html5 应用（以及本地 html5 部件）被启动后，页面的 Dom 中将存在一个全局对象 iPortal，启动页面的脚本可以按如下方法获得当前登录用户 id 与用户姓名，已经用户在统一认证系统中的 id。

```
var userId = iPortal.getContext().getExtra("iportal.uid");
var userXid = iPortal.getContext().getExtra("iportal.uxid");
var userName = iPortal.getContext().getExtra("iportal.uname");
var userGroups = iPortal.getContext().getExtra("iportal.group"); //用户组
var deviceId = iPortal.getContext().getExtra("iportal.device "); //设备号
var ip = iPortal.getContext().getExtra("iportal.ip"); //客户端 ip 地址，如果是 ipv4，则以“172.18.10.18”形式表示，如果是 ipv6，则以"2001:DB8:2de::e13"形式表示
```

按如下方法获得时间戳、随机数、签名。

```
var timeStamp = iPortal.getContext().getExtra("iportal.timestamp");
var nonce = iPortal.getContext().getExtra("iportal.nonce");
var signature = iPortal.getContext().getExtra("iportal.signature");
var signature2 = iPortal.getContext().getExtra("iportal.signature2");
var signature3 = iPortal.getContext().getExtra("iportal.signature3");
```

验证算法参见：[验证请求的合法性](#)。

js 调用参见：[本地能力 JS 调用](#)

参数 iportal.uid、iportal.uxid、iportal.uname 仅供参考，可以在客户端无网络情况下使用。本地 html 如需访问网络，由于客户端会保存统一认证的 cookie。服务端安装原有的方式对接同一认证即可。

✓ 会话失效处理

服务端发现会话失效时，应该按统一认证系统原有的协议向客户端发出重定向即可。

7.5. 代理认证

批注 [eq1]: no

应用系统具备自己的认证体系，并且无法与校方的统一认证系统对接。这种情况下，从微门户客户端中启动应用时，如果需要避免输入用户在应用系统的用户名口令，那么可以采用代理认证的方案。

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytech.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

采用代理认证方案时，微门户首先会记录用户在应用系统中的用户名与口令，当从微门户客户端中启动应用时，微门户客户端会向应用系统传递已记录的用户名与口令。

代理认证按如下模式注册：

■ native 应用(安卓)

注册时，按以下模式注册：

```
<authentication type="proxy" />
```

微门户启动 native 应用时，将为 Intent 设置如下参数：

```
putExtra("iportal.uid","123");//登录用户 id
```

```
putExtra("iportal.uxid","abcd456");//登录用户在统一认证系统中的 id
```

```
putExtra("iportal.uname","zhangsan");//登录用户姓名
```

```
putExtra("iportal.proxyUid","zs");//用户在微门户中设置的自己在 native 应用系统中的登录名
```

```
putExtra("iportal.proxyCredit","..");//用户在微门户中设置的自己在 native 应用系统中的登录口令；
```

```
putExtra("iportal.timestamp","1408355477445");//签名时间戳
```

```
putExtra("iportal.nonce","8472482512");//签名随机数
```

```
putExtra("iportal.signature","AE0F37BC98D28B057");//签名
```

```
putExtra("iportal.group","教职工,领导");//用户所在的全局组
```

```
putExtra("iportal.device","a123b91c8371e");//用户的设备号
```

```
putExtra("iportal.signature2","DA0837B39CD35A1D6");//签名 2
```

```
putExtra("iportal.ip","172.18.10.18");//客户端 ip 地址，如果是 ipv4，则以“172.18.10.18”形式表示，如果是 ipv6，则以“2001:DB8:2de::e13”形式表示
```

```
putExtra("iportal.signature3","1CB6D7C30135F4B9");//签名 3
```

说明：

iportal.uid, iportal.uxid 与 iportal.uname 是用户的 id 以及在统一认证系统中的用户 id 及用户名，iportal.group 是用户所在的全局组，iportal.device 是设备号，这些信息仅供 native 应用参考

iportal.proxyUid 与 iportal.proxyCredit 用户在 native 应用系统中的用户名与口令，native 应用应该需要验证其合法性后才输出内容给用户。

另外时间戳、签名随机数、签名用于验证请求的合法性，验证算法参见：[验证请求的合法性](#)。

✓ 会话失效处理

native 应用可以在内存中持有启动时传递过来的凭证，会话失效时，利用此凭证重新到服务端验证。

■ native 部件(安卓)

微门户在初始化 native 部件时传入的 context 中包含如下扩展信息。

```
ExtraContext ec = (ExtraContext)context;
```

```
ec.getExtra("iportal.uid");//登录用户 id
```

```
ec.getExtra("iportal.uxid");//登录用户在统一认证系统中的 id
```

```
ec.getExtra("iportal.uname");//登录用户姓名
```

```
ec.getExtra("iportal.proxyUid");//用户在微门户中设置的自己在 native 应用系统中的登录名
```

```
ec.getExtra("iportal.proxyCredit");//用户在微门户中设置的自己在 native 应用系统中的登录口令
```

```
ec.getExtra("iportal.timestamp");//签名时间戳
```

```
ec.getExtra("iportal.nonce");//签名随机数
```

```
ec.getExtra("iportal.signature");//签名
```

```
ec.getExtra("iportal.group");//用户所在的全局组
```

```
ec.getExtra("iportal.device");//用户的设备号
```

```
ec.getExtra("iportal.signature2");//签名 2
```

```
ec.getExtra("iportal.ip");//客户端 ip 地址, 如果是 ipv4, 则以“172.18.10.18”形式表示, 如果是 ipv6, 则以“2001:DB8:2de::e13”形式表示
```

验证算法参见: [验证请求的合法性](#)。

另参见: [安卓 native 窗口部件](#)。参见 native 窗口/消息部件(安卓)的开发接口。

■ native 应用(ios)

通过应用注册的 schema 入口 url 启动, 附加如下参数:

如: 注册的入口 url 为 mqq://user/1234

```
mqq://user/1234?iportal.uid=123&iportal.uxid
```

```
=abcd456&iportal.uname=zhangshan&iportal.timestamp
```

```
=1408355477445&iportal.nonce=8472482512&iportal.signature= AE0F37BC98D28B057&
```

```
iportal.proxyUid=zhangshan&iportal.proxyCredit=sssee& iportal.group=教职工,领导&
```

```
iportal.device= a123b91c8371e& iportal.signature2=
```

```
DA0837B39CD35A1D6&iportal.backUrl =
```

```
iportal%3A%2F%2Fback&iportal.ip=172.18.10.18
```

```
&iportal.signature3=1CB6D7C30135F4B9
```

■ web 应用/部件

web 应用如果采用代理认证, 则在注册时需要注册认证页面的相关参数, 微门户客

户端打开此应用时,将以 post 或 get 方式向此页面提交用户的认证信息,注册方式如下:

```
<authentication type="proxy" />
  <proxyDef url="http://myapp/login.php"
    method="post"
    encoding="utf-8"
    userNameField="loginName"
    passwordField="password"
  >
    <fixParam name="myParam" value="123"/>
    <fixParam name="myParam2" value="456"/>
  </proxyDef>
</authentication>
```

◆ proxyDef 节点说明:

属性: url -- 认证页面的 url

属性: method -- 向认证页面 post/get 数据时,采用 POST 还是 GET

属性: encoding -- 编码方式,向认证页面提交数据时,采用的编码方式

属性: userNameField-- 用户名字段名,向认证页面 post/get 用户名的表单字段名。

属性: userPasswordField-- 口令字段名,向认证页面 post/get 用户口令的表单字段名

子节点 fixParam -- 固定参数 -- 这些参数将原样 post/get 给认证页面

◆ 微平台启动将如此启动 url:

```
http://myapp/login.php? iportal.uid =123 &iportal.uxid =abcd456&iportal.uname= zhangsan
&iportal.timestamp =1408355477445&iportal.nonce=8472482512&iportal.signature=
AE0F37BC98D28B057& iportal.group=教职工,领导& iportal.device= a123b91c8371e&
iportal.signature2= DA0837B39CD35A1D6
&iportal.ip=172.18.10.18&iportal.signature3=1CB6D7C30135F4B9
```

如果是 GET 方式,同时还将传递:

```
loginName=zs& password=abcdef& myParam=123&myParm2=456
```

如果是 POST 方式,loginName,password,myParam 将以 POST 方式传递。

传递参数含义如下:

iportal.uid: 用户的 id

iportal.uxid: 用户在统一认证系统中的 id

iportal.uname:用户的姓名,值将以 utf-8 格式编码后,再将经过 urlEncode 转码

后传递。

iportal.group:用户所在的全局组

iportal.device:用户的设备号

iportal.ip: 客户端 ip 地址,如果是 ipv4,则以“172.18.10.18”形式表示,如果是 ipv6,则以“2001:DB8:2de::e13”形式表示

`iportal.uid`、`iportal.uxid` 与 `iportal.uname` 是用户 id 以及用户在统一认证系统中的 id 及用户名，`iportal.group` 是用户所在的全局组，`iportal.device` 是设备号，这些信息仅供 native 应用参考。

另外时间戳、签名随机数、签名用于验证请求的合法性，验证算法参见：[验证请求的合法性](#)。

✓ 会话失效处理

代理认证模式下，web 应用服务端发现会话失效时，不应该向客户端输出登录界面。而是应该向客户端输出一段 js，此 js 调用微门户客户端提供的系统 js 来登录，此系统 js 将向认证页面发送用户登录信息，js 登录完成后，重新访问前一个业务页面。

■ 本地 html5 应用/部件

本地 html5 应用的服务端，应该同普通 web 应用一样，在服务端提供一个认证页面，微门户客户端打开本地页面时，首先将以 post 方式向此页面提交用户的认证信息，从而在客户端与服务端直接产生已认证的会话。认证页面的参数参见“[web 应用代理认证](#)”

✓ 会话失效处理

由于“本地 html5 应用”，一般的模式是在页面中通过 js 向服务端获取数据，然后再浏览器中通过 js 绘制界面。因此服务端服务端数据服务接口中，不应该直接发出重定向指令，因为此时 js 是要从服务端获取数据，而不会执行重定向后向用户展示登录页面。因此，会话失效时，应用服务端应该向客户端的 js 返回错误信息，客户端的 js 得到错误信息后，调用微门户客户端提供的系统 js 重新登录。

7.6. Native app 登录接口

接入到苏迪微门户的第三方 native APP 应用，除了能从微门户客户端内启动时获得登录用户信息外，苏迪微门户也支持 native APP 独立启动时获得用户登录凭证。

采用的方案是，微门户客户端提供登录接口由 native APP 启动时调用，当微门户客户端获得用户登录凭证后，将把凭证返回给 native APP。

7.6.1. ios 登录接口

在 ios 下，第三方应用使用 schema 方式（注：schema 的取值将会客户方的微门户客户端不同而不同），调用微门户的登录接口，路径为 `login`，参数值使用 `urlencode` 方式编码，编码字符集使用 `utf-8`，参数如下：

`backUrl` //当用户在微门户登录成功后，返回到app的schema url。

微门户客户端响应：

微门户客户端在用户登录成功后，将通过请求参数中的 `backUrl` 参数指定的 schema 返

回第三方应用。并将登录结果作为参数附加在 backUrl 地址中，参数值使用 urlencode 方式编码，编码字符集使用 utf-8，传递以下参数：

```
iportal.uid //用户在微门户中的 id
iportal.uxid //用户在统一身份认证系统中的 id
iportal.uname //用户名称
iportal.signature //签名
iportal.group: //用户所在的全局组
iportal.device: //用户的设备号
iportal.signature2: //签名 2
iportal.ip://客户端 ip 如果是 ipv4，则以“172.18.10.18”形式表示，如果是 ipv6，
则以“2001:DB8:2de::e13”形式表示
iportal.signature3: //签名 3
```

第三方应用接收 backUrl 的回调后，应该把收到的参数传回应用后台进行验证，验证算法参见：[验证请求的合法性](#)。

示例：

```
NSString *url = @"seuiportal://login?
backUrl=imap%3A%2F%2Fback%3Fsid%3DzWpi53";
[[UIApplication sharedApplication] openURL:[NSURL URLWithString:url]];
```

7.6.2. android 登录接口

安卓 native app 通过 startActivityForResult() 来调用微门户登录接口。通过 Intent 设置正确的 ComponentName 的包名和类名，com.sudytech.iportal.api.LoginActivity 是固定的类名，注意：包名随客户方的微门户客户端不同而不同。

微门户客户端响应：

```
Intent data = new Intent();
data.putExtra("iportal.uid", "123");//用户在微门户中的id
data.putExtra("iportal.uxid", "1111");//用户在统一身份认证系统中的id
data.putExtra("iportal.uname", "张三");//用户名称
data.putExtra("iportal.timestamp", "1408355477445");//签名时间戳
data.putExtra("iportal.nonce", "8472482512");//签名随机数
data.putExtra("iportal.signature", "AE0F37BC98D28B057");//签名

data.putExtra("iportal.group", "教职工,领导");//用户所在的全局组
data.putExtra("iportal.device", "a123b91c8371e");//用户的设备号
data.putExtra("iportal.signature2", "DA0837B39CD35A1D6");//签名2
```

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytech.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

```
data.putExtra("iportal.ip","172.18.10.18");//客户端ip地址,如果是ipv4,则以“172.18.10.18”形式表示,如果是ipv6,则以“2001:DB8:2de::e13”形式表示
data.putExtra("iportal.signature3","1CB6D7C30135F4B9");//签名3
setResult(0, data);
```

第三方应用接收回调参数后, **应该把参数传回应用后台进行验证**, 验证算法参见: [验证请求的合法性](#)。

代码示例, 如: 东大微门户

```
Intent intent = new Intent();
ComponentName component = new ComponentName("cn.edu.seu.iportal",
"com.sudytch.iportal.api.LoginActivity");//注意:“cn.edu.seu.iportal”
是东大的微门户客户端对应的包名,不同学校不一致;第二个参数登录的activity
intent.setComponent(component);
startActivityForResult(intent, 128);
```

7.7. 验证请求的合法性

除了“匿名访问”的应用, 微门户客户端启动应用时, 会向其传递签名信息, 第三方可以通过以下方式验证请求“确实”是来自于微门户客户端的。验证方案如下:

首先, 在应用供应商在“微门户开发者平台”上传应用包注册应用后, 可以为应用设置一个安全密钥 (SecurityKey)。参见《微门户平台开发者使用手册》。

当应用首次启动时, 可以获取到 `iportal.uid`, `iportal.uxid`, `iportal.timestamp`, `iportal.nonce`, `iportal.signature`, `iportal.group`, `iportal.device`, `iportal.signature2`, `iportal.ip`, `iportal.signature3` 这几个参数。

参数说明:

- ✓ **iportal.uid**
登录用户 id。注: 在“可信的匿名访问”模式下, 无此参数。
- ✓ **iportal.uxid**
登录用户在统一身份认证系统中的 id。注: 在“可信的匿名访问”模式下, 无此参数。
- ✓ **iportal.timestamp**:
时间戳。长整数, 是 UNIX 时间的毫秒数, 即产生签名时的时间距 1970 年 1 月 1 日 零时的毫秒数的十进制字符串。

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

此时间戳代表了签名 (iportal.signature) 的时效性, 如果此时间与当前时间差大于某个值 (建议不超过 20 分钟) 时, 第三方应用服务端应当认为此签名已经失效。

✓ **iportal.nonce:**

随机数。

✓ **iportal.signature**

微门户客户端的签名, 是 16 进制的字符串表示。(注: 此参数是为了兼容老版本, 新开发的应用应当使用 iportal.signature2)。

其算法如下:

首先, 拿第三方应用设置的 SecurityKey, iportal.uid, iportal.uxid 当前的 iportal.timestamp 以及 iportal.nonce 按字符串从小到大顺序排序。比如排序的结果是: iportal.nonce, iportal.uxid, SecurityKey, iportal.timestamp, iportal.uid

排序后, 把四个字符串拼接成一个新的字符串 tmpStr = iportal.nonce + iportal.uxid + SecurityKey + iportal.timestamp + iportal.uid。(注意: 在“可信的匿名访问”模式下无 iportal.uid 与 iportal.uxid)

最后对 tmpStr 按 SHA1 算法求 hash 值, 即

iportal.signature = sha1(tmpStr)

✓ **iportal.group:**

用户在微门户平台中所属的全局组, 如果有多个组, 则将以逗号“,” 隔开传递。

✓ **iportal.device:**

用户打开应用时的设备号。此设备号不是移动设备的 IMEI 号, 是用户设备在微门户平台注册的设备号。

✓ **iportal.signature2:** 微门户客户端的签名 2, 算法同 iportal.signature。但是 iportal.group, iportal.device 也参与 hash 运算。

✓ **iportal.ip:** 客户端 ip 地址, 如果是 ipv4, 则以“172.18.10.18”形式表示, 如果是 ipv6, 则以“2001:DB8:2de::e13”形式表示

✓ **iportal.signature3:** 微门户客户端的签名 3, 算法同 iportal.signature。但是 iportal.group, iportal.device, iportal.ip 也参与 hash 运算。

■ 第三方应用验证方法

第三方应用收到客户端首次发过来的请求后, 从参数中获取 iportal.uid, iportal.uxid (注: “可信的匿名访问”模式下没有这两个参数)、iportal.timestamp、iportal.nonce 这几个值, 并使用第三方应用自己掌握的 SecurityKey, 按上述算法求签名值, 如果求出的签名值与参数中的 iportal.signature 一致, 并且 iportal.timestamp 表示的时间在有效范围内(建议 20 分钟以内), 则表示客户端的请求是合法。

如果同时要验证 iportal.group、iportal.device 信息是否有效, 则用 iportal.signature2 按上述算法进行校验。

如果同时要验证 `iportal.ip` 信息是否有效, 则用 `iportal.signature3` 按上述算法进行校验。

第三方应用对首次请求验证通过后, 可以与客户端建立自己的会话, 通过会话保持后续请求的合法性。

8. 界面集成--窗口部件

窗口部件是应用的一部分, 是应用提供的能在微门户客户端中直接展示界面的组件。一个应用可以注册多个部件。

窗口部件也是有类型, 微门户客户端支持的部件类型如下:

- ◆ native 窗口部件(安卓)
- ◆ web 窗口部件
- ◆ 本地 html5 窗口部件

窗口部件的类型, 可以与主体应用程序不一致, 比如: web 应用可以注册 native 部件(安卓)。

注: 微门户客户端暂不支持 ios 的 native 窗口部件。各应用类型可以注册的消息部件如下:

- ◆ native 应用(ios)
 - ✓ web 窗口部件
 - ✓ 本地 html5 窗口部件
- ◆ native 应用(安卓)
 - ✓ native 窗口部件 (安卓)
 - ✓ web 窗口部件
 - ✓ 本地 html5 窗口部件
- ◆ web 应用
 - ✓ web 窗口部件
 - ✓ 本地 html5 窗口部件
 - ✓ native 窗口部件 (安卓)
- ◆ 本地 html5 应用
 - ✓ web 窗口部件
 - ✓ 本地 html 窗口部件
 - ✓ native 窗口部件 (安卓)

8.1. 窗口部件的注册

窗口部件在 Manifest 中以 `Widget` 节点描述, 所有窗口部件的 `Widget` 节点都必须包含在 `Widgets` 节点中, 如下:

```
<widgets>
  <widget type="native" name="map" verticalTouchevent="true" height="60">
    ...
  </widget>
</widgets>
```

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

```
</widget>
<widget type="local-html" name="news" verticalTouchEvent="true" height="60">
    ...
</widget>
</widgets>
```

下面按窗口部件的类型分别说明:

8.1.1. native 窗口部件(安卓)

样例:

```
<widget type="native" name="map" verticalTouchEvent="true" height="60">
    <entrance xsi:type="classEntrance"
        className="my.test.MapView">
        <fixParam name="myParam" value="123"/>
        <fixParam name="myParam2" value="456"/>
    </entrance>
    <friendlyName lang="cn" name="地图"/>
    <description>地图部件</description>
</widget>
```

说明:

- **widget** 节点的属性:
 - ✓ **type**: 部件的类型, 值固定为"native"。表示此窗口部件是本地代码。注: 目前只支持 android 的 native 窗口部件。
 - ✓ **name**: 窗口部件的名称, 值是字符串。
 - ✓ **verticalTouchEvent**: 值"ture"或"false", 表示此部件在首页中展示时, 是否需要响应用户的上下滑动事件。
 - ✓ **height**: 窗口部件的初始化高度, 以 pix 为单位。窗口部件被启动后, 可以调用接口改变自己的高度。
- **entrance** 子节点
entrance 子节点描述了 native 窗口部件的启动入口, 属性如下:
 - ✓ **xsi:type**: 入口的类型, native 窗口部件的入口类型固定为"classEntrance"
 - ✓ **className**: 字符串, native 窗口部件的启动类的全名, 此类必须继承于 com.sudytech.mobile.widget.WidgetActivity
 - ✓ 子节点 **fixParam**, 固定参数, 可以多个, native 窗口部件的启动类在启动时, 这些参数将被原样传递给它。
- **friendlyName** 子节点
窗口部件的友好名称, 参见[应用的友好名称](#)。
- **description** 子节点

窗口部件的描述信息。

8.1.2. web 窗口部件

样例:

```
<widget type="web" name="news" verticalTouchEvent="true" height="60">
  <entrance xsi:type="urlEntrance" url="http://myapp.com/news/news.php" />
  <friendlyName lang="cn" name="新闻" />
  <description>新闻部件</description>
</widget>
```

说明:

- **widget** 节点的属性:
 - ✓ **type**: 部件的类型, 对于 web 窗口部件, 值固定为"web"。表示此窗口部件是 web 服务器输出的页面。
 - ✓ **name**: 窗口部件的名称, 值是字符串。
 - ✓ **verticalTouchEvent**: 值"ture"或"false", 表示此部件在首页中展示时, 是否需要响应用户的上下滑动事件。
 - ✓ **height**: 窗口部件的初始化高度, 以 pix 为单位。窗口部件被启动后, 可以调用接口改变自己的高度。
- **entrance** 子节点
entrance 子节点描述了 web 窗口部件的启动入口, 属性如下:
 - ✓ **xsi:type**: 入口的类型, web 窗口部件的入口类型固定为"urlEntrance"
 - ✓ **url**: 字符串, web 窗口部件的启动 url
- **friendlyName** 子节点
窗口部件的友好名称, 参见[应用的友好名称](#)。
- **description** 子节点
窗口部件的描述信息。

8.1.3. 本地 html5 窗口部件

样例:

```
<widget type="local-html" name="news" verticalTouchEvent="true" height="60">
  <entrance xsi:type="urlEntrance" url="/news/news.html" />
  <friendlyName lang="cn" name="新闻" />
  <description>新闻部件</description>
</widget>
```

说明:

- **widget** 节点的属性:

- ✓ **type**: 部件的类型, 对于本地 html5 窗口部件, 值固定为" **local-html** "。表示此窗口部件的页面是本地 html5 页面。
- ✓ **name**: 窗口部件的名称, 值是字符串。
- ✓ **verticalTouchEvent**: 值"ture"或"false", 表示此部件在首页中展示时, 是否需要响应用户的上下滑动事件。
- ✓ **height**: 窗口部件的初始化高度, 以 pix 为单位。窗口部件被启动后, 可以调用接口改变自己的高度。
- **entrance** 子节点
entrance 子节点描述了 web 窗口部件的启动入口, 属性如下:
 - ✓ **xsi:type**: 入口的类型, web 窗口部件的入口类型固定为" **urlEntrance** "
 - ✓ **url**: 字符串, 指向应用包中 www 文件夹下的文件, 比如"/news/news.html ", 表示入口页面是 www 文件夹下 news 子文件夹中 news.html 文件。
- **friendlyName** 子节点
窗口部件的友好名称, 参见[应用的友好名称](#)。
- **description** 子节点
窗口部件的描述信息。

8.2. 窗口部件的开发接口 api

8.2.1. 安卓 native 窗口部件

安卓的 native 窗口部件的代码不是打包在 apk 中的, 而是必须按照微门户客户端的提供的规范打包在 widget.jar 中, 与应用包一起上传。

开发步骤:

- 1, 新建一个 Android-Lib 项目应用。引用 sudytech-widget-sdk.jar。
- 2, 新建一个启动入口类, 微门户在初始化 widget 的时候, 需要调用入口类的 initView 方法, 入口类必须实现 com.sudytech.mobile.widget.MainView 接口。initView 方法返回的 view 可以在微门户主页中以部件方式展示。详见示例。编码及打包方式

导出使用原生 java-jdk 编译的 jar 包, 然后使用 dx 命令转换 java 格式的 jar 文件为 dalvik 格式的 jar 文件。

转换命令: dx --dex --output=D:\widget.jar destination.jar

- ✓ 如何打开全屏 Activity
Context.startActivity(intent); //参考 android 原生程序开发,intent 指示的 activity 的 class 必须继承 com.sudytech.mobile.widget.WidgetActivity
- ✓ 如何回到微门户客户端主界面
Active.finish();//参考 android 原生程序开发
- ✓ 如何调整部件界面的高度
部件是一个 android 的 view。参考 android 原生程序开发

注: 微门户的 widget 不支持使用布局文件创建窗口应用。应用需要使用代码方式绘制

应用界面。

示例：

```
package com.ss;

import com.sudytech.mobile.widget.ExtraContext;
import com.sudytech.mobile.widget.MainView;
import android.content.Context;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.RelativeLayout;

public class TestView implements MainView {

    @Override
    public View initView(final Context context) {

        // 获取参数
        ExtraContext ec = (ExtraContext) context;
        String userName = ec.getExtra("iportal.uname");
        String userId = ec.getExtra("iportal.uid");

        // 创建一个widgetView
        RelativeLayout widgetView = new RelativeLayout(context);
        Button btn = new Button(context);
        btn.setText("start new activity");
        btn.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                //创建一个启动一个activity。activity需要继承
                //com.sudytech.mobile.widget.WidgetActivit
                Intent itent = new Intent(context, MyActivity.class);
                context.startActivity(itent);
            }
        });

        // 设置或改变高度
        LayoutParams params = new LayoutParams(320, 100);
        widgetView.setLayoutParams(params);
    }
}
```

```
return widgetView;  
  
}  
}
```

8.2.2. web 部件/本地 html5 部件

web 部件/本地 html5 部件的页面加载时需要引入 iPortalInit.js。iPortalInit.js 由附件提供。使用方式参考[本地能力 JS 调用](#)章节

- ✓ 如何打开全屏页面
iPortal.openWindow(url, callBack);
- ✓ 如何关闭全屏界面回到微门户客户端主界面
iPortal.closeWindow(result);
- ✓ 如何调整部件界面的高度
iPortal.getWidgetHeight();
iPortal.getWidgetWidth();
iPortal.setWidgetHeight(height, isPersistent);

8.3. 窗口部件的注意事项

微门户客户端的首页展示窗口部件时，会屏蔽右侧的 5pix 区域的上下滑动事件，此域是首页自己上下滚动时使用，即用户在首页边缘上下滑动时，首页本身会上下滚动，而 Widget 将接受不到此事件，如下图：



上图中，右侧黄色区域的上下滑动时间将被屏蔽，因此窗口部件应该避免在此区域响应上下滑动事件。

9. 消息部件

微门户平台中，支持应用系统向客户端发送自定义的消息，这些消息的处理界面将由应用系统自己给出。微门户平台中称这些消息为“**扩展消息**”；称应用系统自己提供的扩展消息的处理模块叫“**消息部件**”。

消息部件被微门户客户端打开时，微门户客户端将为其传递额外的消息参数。每个应用在微门户开发者平台上注册时，可以预注册多个消息部件。

消息部件也是有类型，微门户客户端支持的消息部件类型如下：

- ◆ native 消息部件(安卓)
- ◆ native 消息部件(ios)
- ◆ web 消息部件

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

◆ 本地 html5 消息部件

消息部件的类型，可以与主体应用程序不一致，比如：native 应用（安卓）可以注册 web 消息部件。

注意：因为 native 消息部件的代码是应用的主程序 apk（安卓）或 ipa（ios）中的，故 web 应用、本地 html5 应用无法注册 native 的消息部件。

因此各应用类型可以注册的消息部件如下：

- ◆ native 应用(ios)
 - ✓ native 消息部件 (ios)
 - ✓ web 消息部件
 - ✓ 本地 html5 消息部件
- ◆ native 应用(安卓)
 - ✓ native 消息部件 (安卓)
 - ✓ web 消息部件
 - ✓ 本地 html5 消息部件
- ◆ web 应用
 - ✓ web 消息部件
 - ✓ 本地 html5 消息部件
- ◆ 本地 html5 应用
 - ✓ web 消息部件
 - ✓ 本地 html5 消息部件

9.1. 消息部件的注册

消息部件在 Manifest 中以 `messageComponent` 节点描述，所有消息部件的 `messageComponent` 节点都必须包含在 `messageComponents` 节点中，如下：

```
<messageComponents>
  <messageComponent type="web" bizType="1">
    <entrance xsi:type="urlEntrance"
      url="http://www.hello.com/msg/showMessage.jsp"/>
    <description>处理待办消息</description>
  </messageComponent>
  <messageComponent type="local-html" bizType="default">
    <entrance xsi:type="urlEntrance" url="/msg/default.html"/>
    <description>处理其他业务消息</description>
  </messageComponent>
</messageComponents>
```

下面按消息部件的类型分别说明：

9.1.1. native 消息部件(安卓)

样例:

```
<messageComponent type="native" bizType="1">
  <entrance xsi:type="classEntrance"
    packageName="com.hello.myapp"
    className="my.test.MessageActivity">
    <fixParam name="myParam" value="123"/>
    <fixParam name="myParam2" value="456"/>
  </entrance>
  <description>处理待办消息</description>
</messageComponent>
```

说明:

■ **messageComponent** 节点的属性:

- ✓ **type**: 部件的类型, 对于 native 消息部件, 值固定为"**native**". 表示此消息部件的代码是本地代码。

注: 与 native 窗口部件不一样, **native 消息部件 (安卓)** 的代码在应用的主程序 apk 中, 是以 **Intent** 方式启动的。所以, **native 消息部件 (安卓)** 的应用类型也必须是 **native (安卓)** 的。

- ✓ **bizType**: 字符串, 表示此消息部件要能处理的哪个业务类型的消息。

消息的业务类型是应用系统自己定义的业务概念上的类型, 参见“[消息接口](#)”。如果值为"**default**", 表示此消息部件缺省模块, 即能处理应用发出的扩展消息中找不到 **bizType** 对应的消息部件的消息。

■ **entrance** 子节点

entrance 子节点描述了 native 消息部件的启动入口, 属性如下:

- ✓ **xsi:type**: 入口的类型, native 消息部件(安卓)的入口类型固定为"**classEntrance**"
- ✓ 属性: **packageName="com.hello.myapp"**, 安卓应用的包名
- ✓ **className**: 字符串, **native 窗口部件**的启动类的全名, 此类必须继承于 xxx
- ✓ 子节点 **fixParam**, 固定参数, 可以多个, **native 消息部件**的启动类在启动时, 这些参数将被原样传递给它。

■ **description** 子节点

消息部件的描述信息。

9.1.2. native 消息部件(ios)

样例:

```
<messageComponent type="native" bizType="1">
  <entrance xsi:type="urlEntrance" url="mm-qq://user/123456"/>
  <friendlyName lang="cn" name="新闻"/>
  <description>新闻部件</description>
  <description>处理待办消息</description>
</messageComponent>
```

说明:

- **messageComponent** 节点的属性:
 - ✓ **type**: 部件的类型, 对于 native 消息部件, 值固定为"native"。表示此消息部件的代码是本地代码。
注: 与 native 窗口部件不一样, native 消息部件 (ios) 的代码在应用的主程序 ipa 中, 是以 ios 的 schema 协议方式启动的。所以, native 消息部件 (ios) 的应用类型也必须是 native (ios) 的。
 - ✓ **bizType**: 字符串, 表示此消息部件要能处理的哪个业务类型的消息。
消息的业务类型是应用系统自己定义的业务概念上的类型, 参见“[消息接口](#)”。如果值为"default", 表示此消息部件缺省模块, 即能处理应用发出的扩展消息中找不到 bizType 对应的消息部件的消息。
- **entrance** 子节点
entrance 子节点描述了 native 消息部件的启动入口, 属性如下:
 - ✓ **xsi:type**: 入口的类型, native 消息部件(ios)的入口类型固定为"urlEntrance "
 - ✓ **url**: 是 ios app 的 schema 协议, 如: mm-qq://user/123456。如何给 ios app 注册 schema 协议, 参考 ios 开发手册。
- **description** 子节点
消息部件的描述信息。

9.1.3. web 消息部件

web 消息部件的入口类型是 **urlEntrance**, 参见“[native 消息部件\(ios\)](#)”。

9.1.4. 本地 html5 消息部件

本地 html5 消息部件的入口类型是 **urlEntrance**, 参见“[native 消息部件\(ios\)](#)”。

9.2. 消息的传递

微门户客户端在启动时任意消息部件时，除了会传递基本的用户信息、签名信息、用户登录凭证外（参见“[认证接入方式](#)”），还会额外传递消息相关的参数，分述如下：

■ web 消息部件

微门户客户端启动其 url 时传递以下参数：

`iportal.im.imid` – 微门户平台发送给手机的消息的 id

`iportal.im.recvUid` – 接收者在微门户中的 id，注：如果接收者是“应用消息盒”，则无此参数。

`iportal.im.recvUxid` – 接收者在统一认证系统中的 id，注：如果接收者是“应用消息盒”，则无此参数。

`iportal.im.recvBoxId` – 接收消息盒的 id，注：如果接收者是人员，则无此 id。

`ucp_sendMsgId` – 应用系统调用微门户平台消息接口发送消息时，产生的[发件消息](#)的 ID

消息体中的由应用系统设置的 `attributes` 都将作为参数传回，key 是参数名，value 是参数值。

■ native 消息部件 (ios)

微门户客户端启动其 url 时传递以下参数：

`iportal.im.imid` – 微门户平台发送给手机的消息的 id

`iportal.im.recvUid` – 接收者在微门户中的 id，注：如果接收者是“应用消息盒”，则无此参数。

`iportal.im.recvUxid` – 接收者在统一认证系统中的 id，注：如果接收者是“应用消息盒”，则无此参数。

`iportal.im.recvBoxId` – 接收消息盒的 id，注：如果接收者是人员，则无此 id。

`ucp_sendMsgId` – 应用系统调用微门户平台消息接口发送消息时，产生的[发件消息](#)的 ID

消息体中的由应用系统设置的 `attributes` 都将作为参数传回，key 是参数名，value 是参数值。

■ native 消息部件 (安卓)

微门户客户端在启动安卓应用的 native 消息部件 (安卓) 时，在 Intent 中传递以下参数：

`putExtra("iportal.im.imid","123");` // 微门户平台发送给手机的消息的 id

`putExtra("iportal.im.recvUid","456")` – 接收者在微门户中的 id，注：如果接收者是“应用消息盒”，则无此参数。

`putExtra("iportal.im.recvUxid","abc456")` – 接收者在统一认证系统中的 id，注：如果接收者是“应用消息盒”，则无此参数。

`putExtra("iportal.im.recvBoxId","b998")` – 接收消息盒的 id，注：如果接收者是人员，则无此 id。

`putExtra("ucp_sendMsgId","0A3E98")` – 应用系统调用微门户平台消息接口发送消息时，

批注 [g2]: V1.6

批注 [g3]: V1.6

此参数由 UCP 的 IM 通道调用 IM 接口时，以扩展属性给出。

微门户客户端只要原样传递给消息部件即可。

批注 [g4]: V1.6

产生的发件消息的 ID

批注 [g5]: V1.6

另外消息体内的 attributes 都将作为参数传回，如下

```
putExtra("KEY1","VALUE1");//消息体内的 attritues 中的第一个 key-value  
putExtra("KEY2","VALUE2");//消息体内的 attritues 中第二个 key-value
```

■ 本地 html5 消息处理模块

微门户客户端启动本地 html5 消息部件时，本地 html5 中的 js 可以调用以下脚本获取消息相关参数。

```
iPortal.getContext().getExtra("iportal.im.imid");//获取微门户平台发送给手机的消息的消息 id, 返回 Sting
```

`iPortal.getContext().getExtra("iportal.im.recvUId")` – 接收者在微门户中的 id, 注: 如果接收者是“应用消息盒”，则无此参数。

`iPortal.getContext().getExtra("iportal.im.recvUxid")` --接收者在统一认证系统中的 id, 注: 如果接收者是“应用消息盒”，则无此参数。

`iPortal.getContext().getExtra("iportal.im.recvBoxId")` – 接收消息盒的 id, 注: 如果接收者是人员，则无此 id。

`iPortal.getContext().getExtra("ucp_sendMsgId")` – 应用系统调用微门户平台消息接口发送消息时，产生的发件消息的 ID。

批注 [g6]: V1.6

另外消息体内的 attributes 都将作为参数传回，获取方式如下：

```
iPortal.getContext().getExtra("KEY1");//获取消息的附加属性 KEY1 的值
```

10. 本地能力 JS 调用

微门户客户端对 web 应用、本地 html5 应用提供 JS API，应用利用这些 JS API 能够调用移动设备的本地能力，比如发短信、打电话、拍照等功能。

10.1. 本地能力权限声明

web 应用/本地 html5 应用需要在注册时声明将调用哪些本地能力，审核通过后，才能调用相应的系统 js。调用未经授权的 js，系统会抛出异常。

注册样例：

```
<jsPermissions>  
  <jsPermission name="iapp.perm.Take_Photo" />  
  <jsPermission name="iapp.perm.Send_SMS" />  
</jsPermissions>
```

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：http://www.sudytch.com

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

说明:

所有 JS 权限声明的 `jsPermission` 节点必须包含在 `jsPermissions` 节点中。
`jsPermission` 节点的属性 `name`, 表明了需求申请哪种权限。

以下是 `jsPermission` 的 `name` 的列表与权限说明:

jsPermission 的 name 值	描述
<code>iapp.perm.Call_Phone</code>	启动拨打电话窗口
<code>iapp.perm.Send_SMS</code>	启动发送短信窗口
<code>iapp.perm.Send_Email</code>	启动发送邮件窗口
<code>iapp.perm.Take_Photo</code>	拍照并获取照片,或选取相册中的图片
<code>iapp.perm.Record_Audio</code>	录音
<code>iapp.perm.Read_Friends</code>	获取好友列表及好友分组列表
<code>iapp.perm.Open_IM</code>	打开 IM 聊天窗口
<code>iapp.perm.Access_Location</code>	获取地理位置
<code>iapp.perm.Vibrate</code>	振动

10.2. 本地能力 JS API

微门户启动时, 在浏览器的 `window` 对象上绑定了一个 `iPortal` 对象。`window.iPortal` 是 `iPortal` 对象的唯一实例。微门户的所有本地能力需要通过此对象访问。

10.2.1. 初始化

调用本地能力 `js` 接口, 必须在 `html` 中引入 `iPortalInit.js` 库文件(注此文件作为 `sdk` 另行提供), 并且需要在 `iPortalInit.js` 初始化之后, 才可以正确使用。本地化 `html` 应用应该将 `iPortalInit.js` 随应用一起打包, 并且在需要调用本地能力的页面中引用此 `js`。web 方式的需要在服务端放入 `iPortalInit.js`, 并且在需要调用本地能力的页面中引用此 `js`。

`iPortalInit.js` 的初始化, 是在 `Dom` 加载完毕后自动初始化的, 应用程序可以在 `Dom` 加载完毕之前, 为 `iPortal` 对象挂载初始化回调函数, 来感知 `iPortal` 的初始化完毕。也可以查看 `iPortal.isReady` 属性来判断是否已经初始化完毕。

`iPortal` 对象内的除 `isReady`, `onReady` 之外的其他任何属性、对象、方法都必须在初始化之后才能使用。

下例给出了初始化过程。

完整示例:

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012
电话: 86-25-86641410 传真: 86-25-84512645 转 605
网址: <http://www.sudyeach.com>
版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script type="text/javascript" src="iPortalInit.js"></script>

<script type="text/javascript">
var iPortalInit = false;
var debug = true;
//初始化
var myInit = function(succeeded) {
    //执行初始化
    iPortalInit = succeeded;
    if(debug && !succeeded){ //调试用
        alert("iPortal初始化失败");
    }
};
if(!iPortal.isReady){
    iPortal.onReady=myInit; //挂载初始化回调函数
}else{
    myInit();
}
function takePhoto(){
    if(iPortalInit){ // 或者 if(iPortal.isReady)
        iPortal.getCamera().getPicture(function (picture) { //调用拍照功能
            document.getElementById("pic").src=picture;
        },
        function (err) {
            alert(err);
        },
        1);
    }
    else{
        alert("iportal尚未初始化");
    }
}
</script>
</head>
<body>
```

```
<img id="pic" src="" />
<button onclick='takePhoto()'>拍照</button>
</body>
</html>
```

10.2.2. 微门户事件。

事件绑定, `window.addEventListener(name: String, func: Function);`

`func: Function` 形如: `function(e: Event);`

解除绑定, `window.removeEventListener(func:Function);`

事件类型, `iPortalClose`, `iPortalBack`

`iPortalClose`, 点击关闭按钮时触发, 默认关闭当前窗口

`iPortalBack`, 点击后退按钮时触发, 默认值执行 `history.go(-1)`; 如果没有历史, 就关闭当前窗口。

示例:

```
window.addEventListener('iPortalClose', function(e){
    console.log('窗口关闭');
    event.cancelBehavior = false;
});
```

对象: Event

属性:

名称	类型	描述
type	String	事件类型, 如: <code>iPortalClose</code> , <code>iPortalBack</code>
cancelBehavior	boolean	是否取消默认行为, 默认: <code>false</code>

10.2.3. iPortal 微门户根对象 IPortal 的实例。

对象: IPortal

属性:

名称	类型	描述
----	----	----

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利



onReady	Func	iPortal 环境初始化完成调用此方法。 回调函数形如 function(): void
isReady	boolean	iPortal 初始化完毕后, 此值会被标记为 true

方法:

getContext(): Context	
描述	返回环境信息
返回	Context 上下文对象
参数	
openWindow (url:String, callback: func, options: Json): void	
描述	打开一个的浏览器窗口
返回	void
参数	
url:String	新窗口访问的地址 url。
callback: func	新窗口关闭时回调此方法, 回调函数形如 function(result: String):void; 其中 result 的内容来自 closeWindow (result: String): void 方法的参数
options: Json	新窗口的属性, hideNavigationBar 是否隐藏导航条, 默认 false。hideStatusBar 是否隐藏状态栏, 默认 false, hideCloseButton 隐藏关闭按钮, 默认 false。hideBackButton 隐藏返回按钮, title 导航条标题, 默认是应用名称 {hideNavigationBar:false,hideStatusBar:false,hideCloseButton:false,hideBackButton:false,title:'oa 办公'}
closeWindow (result: String): void	
描述	关闭当前窗口
返回	void
参数	
result: String	发送给窗口启动者的内容。
getWidgetHeight(): number	
描述	获取 widget 的当前高度, 仅当窗口处于 widget 状态时, 可用
返回	widget 的当前高度
参数	
getWidgetWidth(): number	
描述	获取 widget 的当前宽度, 仅当窗口处于 widget 状态时, 可用
返回	widget 的当前宽度
参数	
setWidgetHeight(height: number, isPersistent: boolean): void	
描述	改变当前 widget 的高度, 仅当窗口处于 widget 状态时, 可用

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410

传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利



返回	void
参数	
height: number	变更的高度值
isPersistent: boolean	当前高度是否需要存储。如果 isPersistent 为 true, 系统将存储此高度, 下次初始化此 widget 时, 将使用改变后的高度。如果 isPersistent 为 false, 则临时改变此 widget 的高度, 系统不做存储。
openApp(schemaUrl:String): void	
描述	打开 native 应用
返回	void
参数	
schemaUrl:String	应用入口的字符串描述。 Android 的描述格式: “应用包名/类名?参数 1=参数值&参数 2=参数值”。如: com.sudytech.iportal/com.sudytech.iportal.index.Welcome?userId=332&token=24422 IOS 的描述格式 (遵循 ios 打开第三方应用的规范): 应用的 schema://路径?参数 1=参数值 &参数 2=参数值。如: myapp://open?userId=1&token=233 注: 参数值使用 urlencode 的 utf-8 编码。
getDevice(): Device	
描述	获取设备描述信息, 参考 Device 对象描述设备的硬件和软件。
返回	获取 Device 对象。
参数	
getSNS(): SNS	
描述	获取社交对象, 参考 SNS 对象提供对微门户平台用户 sns 信息访问
返回	获取 SNS 对象。
参数	
getIM(): IM	
描述	获取即时通讯对象, 参考 IM 对象提供对微门户平台即时通信的调用。
返回	获取 IM 对象
参数	
getSMS(): SMS	
描述	获取短信对象, 参考 SMS 对象提供发送短信的能力
返回	获取 SMS 对象
参数	
getPhone(): Phone	
描述	获取电话对象对象, 参考 Phone 对象提供引对电话的访问



返回	获取 Phone 对象
参数	
getAudio(): Audio	
描述	获取音频录制对象, 参考 Audio 对象提供音频录制的功能
返回	获取 Audio 对象
参数	
getLocation(): Location	
描述	获取位置对象, 参考 Location 对象提供访问位置的能力
返回	获取 Location 对象
参数	
getVibrator(): Vibrator	
描述	获取振动器对象, 参考 Vibrator 对象提供对震动能力的访问
返回	获取 Vibrator 对象
参数	
getEmail(): Email	
描述	获取邮件发送对象, 参考 Email 对象提供对利用本地邮箱发送邮件的能力
返回	获取 Email 对象
参数	
getCamera(): Camera	
描述	获取拍照对象, 参考 Camera 对象提供对设备默认摄像头应用程序的访问。
返回	获取 Camera 对象
参数	
getNews(): News	
描述	获取拍照对象, 参考 News 对象提供访问本地资讯的能力
返回	获取 News 对象
参数	
getLog(): Log	
描述	获取日志保存对象, 参考 Log 对象提供保存日志能力
返回	获取 Log 对象
参数	
getUser(): User	

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利



描述	获取操作用户信息对象，参考 User 对象提供获取用户信息能力
返回	获取 User 对象
参数	
getScan(): Scan	
描述	获取扫描外部信息对象，参考 Scan 对象提供获取扫描外部信息能力
返回	获取 Scan 对象
参数	
getSystem(): System	
描述	获取操作系统对象，参考 System 提供与操作系统相关的能力
返回	获取 System 对象
参数	
getCryptor(): Cryptor	
描述	获取加解密对象，参考 Cryptor 提供加密解密能力
返回	获取 Cryptor 对象
参数	

对象：Context

属性：

名称	类型	描述

方法：

getExtra (key: String): String	
描述	根据关键字获取环境变量的值
返回	变量值
参数	
Key	变量名称关键字

10.2.4. Camera 对象提供对设备默认摄像头应用程序的访问。

对象：Camera

属性：

名称	类型	描述
number	int	相机的编号

方法：

takePicture(success: func, error: func): void	
描述	选择使用摄像头拍照，或从设备相册中获取一张照片。图片以 base64 编码的字符串返回，需要权限： iapp.perm.Take_Photo
返回	Void
参数	
success: func	回调函数形如 function(data: JsonObject): void 。 data 格式 { pic:"data:image/gif;base64,waferwre==", fileName: 'head.jpg' }
error: func	回调函数形如 function(error: String): void
takePhoto (success: func, error: func): void	
描述	选择使用摄像头拍照，图片以 base64 编码的字符串返回，需要权限： iapp.perm.Take_Photo
返回	Void
参数	
success: func	回调函数形如 function(data: JsonObject): void 。 data 格式 { pic:"data:image/gif;base64,waferwre==", fileName: 'head.jpg' }
error: func	回调函数形如 function(error: String): void

10.2.5. Device 对象描述设备的硬件和软件。

对象：Device

属性：

名称	类型	描述

方法：

getPlatform(): String	
描述	获得设备使用的操作系统名称
返回	描述操作系统信息的字符串，如 IOS 6.0, Android 4.1
参数	
getUuid (): String	
描述	获得设备的全球唯一标识符（UUID）。
返回	设备的全球唯一标识符
参数	

10.2.6. SNS 对象提供对微门户平台用户 sns 信息访问

对象：SNS

批注 [eq7]: no

属性：

名称	类型	描述

方法：

getCategories(success:func, error:func):void	
描述	获取用户的好友分组信息

返回	void
参数	
success: func	回调函数形如 function(categories: Category[]): void
error: func	回调函数形如 function(error:String);
getCategoryFriends(categoryId: String, success: func, error: func): void	
描述	获取好友分组下的好友列表
返回	void
参数	
categoryId: String	分类 id
success: func	回调函数,如下 success(categories: Friend[]):void
error: func	回调函数形如 function(error:String);
getBlacklists(success: func, error: func): void	
描述	获取黑名单列表
返回	void
参数	
success: func	回调函数,如下 success(blacklists: Friend[]):void
error: func	回调函数形如 function(error:String);

对象：Category

属性:

名称	类型	描述
id	number	唯一 id
name	String	分类名称

方法:

对象：Friend

属性:

名称	类型	描述
----	----	----

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利



id	number	好友对象 id
noteName	String	备注名称
userId	number	好友的用户 id
categoryId	number	好友的分类 id

方法:

10.2.7. IM 对象提供对微门户平台即时通信的调用。

对象: IM

属性:

名称	类型	描述

方法:

openChat(address: String, success: func, error: func): void	
描述	打开一个聊天窗口
返回	void
参数	
address: String	聊天目标地址。 格式一, u:321 格式说明 u 表示用户, 321 表示用户 id。 格式二, b:123 格式说明 b 表示消息盒, 123 表示消息盒 id
success: func	回调函数形如 function(): void
error: func	回调函数形如 function(error: String): void

10.2.8. Phone 对象提供引对电话的访问

对象: Phone

属性:

名称	类型	描述

方法:

call(phoneNumber: String, success: func, error: func): void	
描述	启动拨打电话窗口
返回	void
参数	
phoneNumber: String	拨打的电话号码
success: func	回调函数形如 function(): void
error: func	回调函数形如 function(error: String): void

10.2.9. SMS 对象提供发送短信的能力

对象: SMS

属性:

名称	类型	描述

方法:

sendMsg(phoneNumber: String, msg: String, success: func, error: func): void	
描述	启动发送短信窗口
返回	void
参数	
phoneNumber: String	电话号码
success: func	回调函数形如 function(): void
error: func	回调函数形如 function(error: String): void

10.2.10. Audio 对象提供音频录制的能力

批注 [eq8]: no

对象: Audio

属性:

名称	类型	描述

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

 网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

方法:

openRecorder (success: func, error: func, [option: number]): void	
描述	录制一段录音, 获取从设备中选择一段录音
返回	void
参数	
success: func	回调函数形如 function(audio: String): void 。audio 的内容为 Base64 编码的照录音文件
error: func	回调函数形如 function(error: String): void

10.2.11. Location 对象提供访问位置的能力

对象: Location

属性:

名称	类型	描述

方法:

getCurrentPosition(success: func, [error: func], [option: number]): void	
描述	返回一个 Position 对象表示设备的当前位置。
返回	void
参数	
success: func	回调函数形如 function(position: Position): void
error: func	回调函数形如 function(error: String): void

对象: Position

属性:

名称	类型	描述
latitude	number	以十进制表示的纬度。
longitude	number	以十进制表示的经度。
altitude	number	位置相对于椭圆球面的高度, 单位为米。
accuracy	number	以米为单位的纬度和经度坐标的精度水平。
altitudeAccuracy	number	以米为单位的高度坐标的精度水平。
heading	number	运动的方向, 通过相对正北做顺时针旋转的角度指定。
speed	number	以米/秒为单位的设备当前地面速度。

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytech.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利

方法:

10.2.12. Vibrator 对象提供对震动能力的访问

批注 [eq9]: no

对象: Vibrator

属性:

名称	类型	描述

方法:

vibrate(milliseconds: number): void	
描述	使设备震动指定的时长。
返回	void
参数	
milliseconds: number	震动的时间, 以毫秒为单位的设备震动时长

10.2.13. Email 对象提供对利用本地邮箱发送邮件的能力

对象: Email

属性:

名称	类型	描述

方法:

sendEmail(address: String, content: String): void	
描述	启动发送 email 的编辑框
返回	void
参数	
address: String	目标邮箱地址
content: String	邮件内容

10.2.14. Notification 提供通知能力。

批注 [eq10]: no

对象：Notification

属性：

名称	类型	描述

方法：

notice(id: number, title: String, content: String, [clickUrl: String]): void	
描述	立即发出一个通知
返回	void
参数	
id: number	通知的唯一标识，相同标识只显示最新的。
title: String	通知标题
content: String	通知内容
clickUrl: String	点击通知时，微门户使用 webView 访问此 url
scheduleNotice(id: number, title: String, content: String, firstTime: String, period: number, [clickUrl: String]): void	
描述	周期性发出通知
返回	void
参数	
id: number	通知的唯一标识，相同标识只显示最新的。
title: String	通知标题
content: String	通知内容
firstTime: String	第一次通知的时间，日期格式 yyyy-MM-dd hh:mm:ss
period: number	通知间隔，单位：小时
clickUrl: String	点击通知时，微门户使用 webView 访问此 url
cancelNotice(id: number): void	
描述	取消通知
返回	void
参数	
id: number	通知的唯一标识。

10.2.15. News 对象提供访问本地资讯的能力

对象：News

属性：

名称	类型	描述

方法：

showColumnList(siteId: String, columnId: String, columnName: String): void	
描述	展示资讯栏目列表
返回	void
参数	
siteId: String	站点 id
columnId: String	栏目 id
columnName: String	栏目名称

showNewsDetail (siteId: String,siteName: String,articleId: String,articleTitle: String, url: String): void	
描述	展示栏目列表
返回	void
参数	
siteId: String	站点 id
siteName: String	站点名称
articleId: String	文章 id
articleTitle: String	文章标题
url: String	文章地址

10.2.16. Log 对象提供保存日志能力

对象：Log

属性：

名称	类型	描述

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

方法:

log(content: String): void	
描述	保存日志接口
返回	void
参数	
content: String	日志内容

10.2.17. User 对象提供获取用户信息能力

对象: User

属性:

名称	类型	描述

方法:

showUserDetail (userId: Number,username: String): void	
描述	查看用户的详情
返回	void
参数	
userId: Number	用户 id
username: String	用户名称

10.2.18. Scan 对象提供获取扫描外部信息能力

对象: Scan

属性:

名称	类型	描述

方法:

scan2DBarcode(success: func, error: func): void	
--	--



描述	扫描二维码
返回	void
参数	
success: func	回调函数形如 function(code:String); 回调参数 code 表示二维码识别后的字符串
error: func	回调函数形如 function(error:String);

10.2.19. System 提供与操作系统相关的能力

对象：System

属性:

名称	类型	描述

方法:

startActivity (intent:Json, [error: func, success:func]): void	
描述	启动一个 activity, 仅支持 android 系统
返回	void
参数	
intent:Json	Json 键值对格式 { packageName:String,//包名 className:String,//类名 action:String, data:String, type:String, extras:Json //扩展参数 }
error: func	回调函数形如 function(error:String);
success:func	回调函数形如 function();
openSchema (schema:String, [error: func, success:func]): void	
描述	通过 schame 启动外部应用, 支持 android 系统 ios 系统
返回	void
参数	
schema:String	打开外部应用的 schema 入口。
error: func	回调函数形如 function(error:String);
success:func	回调函数形如 function();

批注 [eq11]: 新增 error 参数, 当 app 未安装或者启动失败时, 返回错误信息

批注 [eq12]: 使用 js 方式启动外部应用

10.2.20. Cryptor 提供加密解密能力

对象：Cryptor

属性：

名称	类型	描述

方法：

RSA (data:String, pubKey:String): void	
描述	启动一个 activity，仅支持 android 系统
返回	void
参数	
data:String	被加密的字符串
pubKey:String	加密用的公钥
success: func	回调函数形如 function(encode:String); 回调参数 encode 表示加密的字符串
error: func	回调函数形如 function(error:String);

11. url schema 协议

批注 [eq13]:

微门户的 webview 支持通过 url schema 协议启动外部应用，对于符合 schema 的 url 请求，可以打开外部应用。

微门户保留的 schema 协议 iportal-app。通过此协议，可以启动在微门户安装的应用，如：

iportal-app.com.sudytech.mapp?param1=value1¶m2=value2

解释：com.sudytech.mapp 表示在微门户创建的应用名称。

param1=value1¶m2=value2 为希望想在打开应用时传递的附加参数。微门户平台除了附带此参数，还会附加微门户特有的参数，附加参数参考[认证接入方式](#)。

12. web 访问范围声明

出于安全考虑，微门户客户端会对 Web 应用以及本地 html 应用的远程 http 请求进行审核与限制。

Web 应用以及本地 html 应用在注册时首先要声明应用需要访问哪些远程服务，审核通过后，Web 应用及本地 html 应用才能访问到这些远程服务，超出声明范围外的远程请求将

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytech.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

被限制。

注册样例:

```
<urlScopes>
  <urlScope>http://www.hello.com</urlScope>
  <urlScope>http://www.baidu.com/maps</urlScope>
</urlScopes>
```

说明:

每个范围什么节点 `urlScope` 必须包含在 `urlScopes` 节点中。

`urlScope` 节点的内容, 表明了一个访问范围, 比如: `http://www.hello.com` 表示本应用将访问所有以 `http://www.hello.com` 开头的 url。

13. 消息接口

微门户平台向第三方应用提供 `webservice api`, 使第三方应用服务端可以向目标用户推送消息。第三方应用在能调消息推送的 `api` 之前, 需要提出申请, 管理员审核通过后为分配消息盒帐号, 第三方应用使用其消息盒帐号来发送消息。

13.1. 消息的地址编码方式

在微门户平台中, 会为用户, 群组, 业务系统等分别分配一个消息盒, 发送消息时, 是利用自己的消息盒, 向目标消息盒发送消息, 目标(用户或群组)从他们的消息盒的收件箱中检索出新到的消息。

消息的地址编码方式, 就是如何指定消息接受者编码的方式, 微门户平台提供两类的地址编码方式:

一类是范围地址, 即接受人是一个范围, 可以指定的范围有: 组织机构(可递归)、全局组

另一类是单个地址, 可以指定到具体的人员, 也可以指定到某个消息盒。

这两类地址的由 `WebService` 的 `Address` 对象来描述, `Address` 有两个属性: 地址类型、详细地址, 详细地址的格式随地址类型不同而不同, 见下表:

地址类型	地址格式	说明
uc_o	(ordId)orgName[:r]	表示接受人员是系统内某机构内的所有人员 ordId -- int, 微门户平台内部的机构id orgName -- string, 机构名称, 可略 [:r]--表示":r"可以不出现, 如果出现,

地址: 南京市雨花台区软件大道 106-2 号, 雨花软件园 A 栋 6 层 邮编: 210012

电话: 86-25-86641410 传真: 86-25-84512645 转 605

网址: <http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司, 保留所有权利



		则表示机构是递归的，即发送给所有子机构内人员。 举例： (127)宣传部:r
uc_ox	(ordXid)orgName[:r]	当使用第三方统一身份认证系统，表示接受人员是统一身份认证系统内某机构内的所有人员 ordXid -- String ,统一身份认证系统内的机构 id orgName -- string,机构名称，可略 [:r]--表示":r"可以不出现，如果出现，则表示机构是递归的，即发送给所有子机构内人员。 举例： (idso_573)宣传部:r
uc_g	(groupId)groupName	表示接受人员是微门户平台内部的全局组内的所有人员 groupId -- int , 微门户平台内部的组 id groupName -- string,组名称，可略 举例： (105)教师组
uc_u	(userId)userName	表示向某个系统内人员发送消息。 userId -- int, 微门户平台内部的人员 id userName -- string, 姓名，可略。 举例： (23005)张三
uc_ux	(userXid)userName	当使用第三方统一身份认证系统时，表示向统一身份系统内人员发送消息。 userXid -- string, 统一身份人员 id userName -- string, 姓名，可略。 举例： (idsu_9908631)张三
uc_b	(boxId)boxName	表示向系统内的某个消息盒发送消息。 boxId -- int,消息盒 id boxName -- string, 消息盒名称，可略 举例： (57)组织部

		(99)OA 系统
sms	phoneNum	表示向某个手机号发消息 举例： 13611555888 注：用户方必须购置苏迪科技的统一通信平台后，微门户才能支持短信发送
mail	emailAddress	表示向某个邮件地址发消息 举例： zhangsan@google.com 注：用户方必须购置苏迪科技的统一通信平台后，微门户才能支持邮件发送

13.2. 发件消息与收件消息

发件消息，是指通过消息盒发出的消息，发件消息被保存在发送者的消息盒内，调用微门户平台的发送接口，返回的 `SendResult` 内的消息 `id`，即是发件消息的 `id`。

收件消息，是指接收方收到的消息，收件消息保存在接收方的消息盒内。

由于微门户平台的支持范围收件地址，因此一个发件消息可能会产生很多收件消息。

13.3. 发送扩展消息

微门户平台的消息有 3 种类型：普通消息、扩展消息、系统消息；其中“系统消息”为内部消息，第三方应用无法发出系统消息。

普通消息，接受者收到后，按客户端默认的打开方式处理。

扩展消息，微门户客户端将调用应用的消息部件或者消息内指定的外链来打开消息。

第三方应用发送此类消息时，需要：

- 指定消息的类型为 1 - 扩展消息
- 为消息设置扩展属性 `im_linkUrl`，如果此属性不设置，系统将根据消息扩展属性 `ucp_bizType` 的值来决定调用哪个消息部件，如果消息部件不存在，系统将认为是普通消息。
- 为消息设置扩展属性 `ucp_bizType`，客户端用于决定此消息用哪个消息部件打开（参见 [消息部件注册](#)）。如果消息中存在 `im_linkUrl`，系统将优先以此 `url` 打开消息。

举例，假设应用注册了 3 个消息部件：

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

messageComponent 节点内的 bizType 属性值	打开界面	说明
1	Com.sudy.pend.MsgActivity	业务系统中 bizType=1 表示“待办”微门户客户端发现消息的扩展属性 ucp_bizType=1 时，使用此模块打开
2	Com.sudy.supervisor.MsgActivity	业务系统中 bizType=2 表示“督办”消息的扩展属性 ucp_bizType=2 时，使用此模块打开
default	Com.sudy.DefaultMsgActivity	微门户客户端找不到“扩展消息”有对应业务类型的注册模块时，缺省使用此模块打开

如果消息中的 businessType == 1， 则打开 Com.sudy.pend.MsgActivity

如果消息中的 businessType == 2， 则打开 Com.sudy.supervisor.MsgActivity

如果消息中的 businessType 为其他值，则打开 Com.sudy.DefaultMsgActivity

13.3.1. 为扩展消息添加应用系统自己的属性

应用系统可以为消息添加自己的属性，这些属性中微门户客户端打开消息部件或外链时，将原样传回。

参见 [“消息的传递”](#)

参见 [“Message 对象”](#)

13.3.2. 为扩展消息指定“已读”方式

默认情况下，当微门户客户端收到了多个消息时，如果用户浏览了最后一个消息，那么前面的消息会自动设置为已读；第三方应用可以指定某个消息用户必须点击后才能

设置方法为，为消息设置扩展属性 im_readType=1

13.4. 消息的内容格式

目前，微门户客户端只支持第三方应用发送存文本的消息内容。

13.5. WebService 接口描述

13.5.1. 对象介绍

13.5.1.1. Address

```
public class Address {  
    /**  
     * 地址类型  
     */  
    protected String type; //即消息地址的编码方式表内指定的类型  
    /**  
     * 地址的值  
     */  
    protected String address;  
}
```

说明：Address 描述一个收件地址。

13.5.1.2. Message

```
public class Message {  
    /**  
     * 主送地址  
     */  
    private Address[] to;  
    /**  
     * 标题  
     */  
    private String subject;  
    /**  
     * 内容  
     */  
    private String content;  
    /**  
     * 消息类型，默认为 0.  
     * 0 - 普通消息  
     */  
}
```

```
* 1—扩展消息
*/
private int msgType;
/**
 * 消息扩展属性
 * 以 key-value 的键值对
 */
private Map<String, String> properties;
}
```

应用系统可以为消息设置扩展属性，方法为在 `Message.properties` 中放入自己的扩展属性。这些属性将在客户端打开消息部件或外链时原样传回。

此外，应用系统设置的自己扩展属性的 `key`，不能以“`im_`”“`ucp_`”“`sys_`”开头，以这三个前缀开头的扩展属性将不会被传回。另外，附加属性的 `key` 的字符串中不能有“`.`”“`$`”符号。

系统定义的扩展属性列表：

`im_linkUrl` – 消息的外链

`ucp_bizType` – 消息的业务类型的值

`ucp_bizTypeName` – 消息的业务类型名称。注：微门户客户端目前尚未使用此值

`im_readType` – 扩展消息的已读方式

13.5.1.3. SendResult

```
public class SendResult {
/**
 * 是否成功
 */
private boolean succeeded;
/**
 * 有错误时的错误描述
 */
private String errorInfo;
/**
 * 发件消息的编号，即消息 id
 */
private String messageId;
/**
 * 发件地址中错误的地址列表
 */
}
```

批注 [g14]: V1.6

郭雷

第三方应用不传递 `msgType`，即为 0 时，调用 `uc-api` 创建 `Type_Normal` 的消息

第三方应用不传递 `msgType=1` 时，调用 `uc-api` 创建 `Type_ExtUrl` 的消息

应用传递其他值，抛出错误

```
private WrongAddress[] wrongAddresses;  
}
```

说明：SendResult 是调用发送方法后，如果存在错误的发件地址，将返回这些错误地址，详见具体发送方法。

13.5.1.4. WrongAddress

```
public class WrongAddress {  
    /**  
     * 错误类型  
     * 0 -- 整个地址错误  
     * 1 -- 范围地址内有部分地址没权限，比如指定机构地址：科技处[递归]，但是发送者  
     * 对科技处的子机构没有权限  
     * 2 -- 指定的地址或地址范围正确且具备基本权限，但是某些地址在通道上无权限。  
     * 如果一次发送中，仅存在错误类型为 2 的地址，系统将继续发送，但会返回这些地  
     * 址信息。  
     */  
    private int errorType;  
    /**  
     * 错误的地址，即 Message 中的发件地址  
     */  
    private Address address;  
    /**  
     * 错误描述  
     */  
    private String errorInfo;  
    /**  
     * 错误的子地址，即 address 中展开的地址  
     * 为空时，表示整个主地址 address 无权限。  
     * 如果主地址 address 是地址簿时，其中的内部人员直接作为 UserAddress 返回，  
     * 非内部人员作为 ContactRecordAddress 返回  
     */  
    private Address[] subAddresses;  
    /**  
     * 无权限的通道  
     */  
    private Channel[] forbiddenChannels;  
}
```

13.5.1.5. ChannelDeliverState

```
public class ChannelDeliverState {
    /**
     * 产生此收件消息的原始地址
     * 由于本系统支持范围地址，系统将把范围地址解析展开为多个实际的地址来发送消
     * 息。
     * 因此，此属性表明了本收件消息是由哪个发件地址产生的。
     */
    private Recipient recipientBy;
    /**
     * 实际的最终收件地址
     */
    private Address sendTo;
    /**
     * 通道号
     */
    private int channelId;
    /**
     * 发送状态
     * 0 -- 未发送
     * 1 -- 发送中
     * 2 -- 已发送
     * 3 -- 重试发送
     * -1 -- 发送失败
     */
    private int state;
    /**
     * 错误信息
     */
    private String errorMessage;
    /**
     * 发送时间
     */
    private Date sendTime;
    /**
     * 在通道上发送时，通道返回的 id
     */
    private String chanelDeliverId;
    /**

```

```
* 发送批次
*/
private int frameNo;
/**
 * 计费数量
 */
private int billCount;
/**
 * 收件消息的 ID
 */
private String rcvMessageId;
}
```

说明:

此对象用于检查发件消息的每个实际发送状态的返回结果。详见 Webservice 方法：
getMessageDelivers

13.5.2. Webservice 方法

13.5.2.1. Webservice 调用安全验证方式

如果同时使用南京苏迪科技有限公司的统一身份认证产品，第三方应用系统需要先从统一身份认证系统认证 App 的身份，获取一个 serviceToken 后再调用 webservice 方法。

如果使用其他公司的统一身份认证产品，第三方应用系统必须使用客户端证书模式的 https 调用统一通信平台的 webservice 方法，此时所有调用可以不传递 serviceToken。

13.5.2.2. sendMessage

```
/**
 * 直接发送消息
 * @param boxId 分配给应用系统的，用于发送消息的消息盒编号
 * @param message 要发送的消息内容
 * @param channels 指定在哪些通道发送，系统支持的通道有
 * 1 - 短信通道，即此消息同时通过短信发送给用户
 * 2 - 电子邮件通道，即此消息同时通过电子邮件发送给用户
 * 3 - 微门户 IM 通道，即此消息通过微门户客户端推送给用户
 * 注：1 短信通道、2 电子邮件通道，用户方必须购置苏迪科技的“统
```

地址：南京市雨花台区软件大道 106-2 号，雨花软件园 A 栋 6 层 邮编：210012

电话：86-25-86641410 传真：86-25-84512645 转 605

网址：<http://www.sudytch.com>

版权所有 (C) 南京苏迪科技有限公司，保留所有权利

```
*          “一通信平台”后才能被支持。
* @param usesSignature 是否使用默认签名
* @param plannedTime 计划发送时间
* @param serviceToken 服务令牌
* @return SendResult 发送结果
*          如果地址中存在错误地址（整个地址错误，或范围地址内一部分地址无
权限），消息将不发送，还是处于草稿状态，
*          此时 SendResult.succeeded=false;
*          如果地址完全正确，或者仅部分地址在某个通道上无权限，消息将发送
成功，将处于发送状态。
*          此时 SendResult.succeeded=true;
*          SendResult.wrongAddressess 中包含了的错误地址列表，这些地址是原始
的发件地址，不是对“范围地址”展开后的最终的地址。
* @throws MessageException
*/
public SendResult sendMessage(int boxId, Message message, int[] channels, boolean
usesSignature, Date plannedTime, String serviceToken) throws MessageException{

}
```

13.5.2.3. sendIndivMessage

```
/**
* 个性化批量发送
* @param boxId 分配给应用的消息盒编号
* @param message 要发送的消息内容模板，在内容可以使用${fieldName}标签实现动
态内容替换
* @param channels 指定在哪些通道发送，系统支持的通道有
*          1 – 短信通道，即此消息同时通过短信发送给用户
*          2 – 电子邮件通道，即此消息同时通过电子邮件发送给用户
*          3 – 微门户 IM 通道，即此消息通过微门户客户端推送给用户
*          注：1 短信通道、2 电子邮件通道，用户方必须购置苏迪科技的“统
一通信平台”后才能被支持。
* @param usesSignature 是否使用默认签名
* @param solution 个性化方案，目前支 1
* @param datasrc 发送的地址及动态替换内容列表。是以 json 表示的二维表。
*          比如 {rows:{{"phone":"13611555888",
"score":95},{"phone":"13956782233", "score":87}...}}
* @param individualParams datasrc 中的地址列名，比如"phone"
```

```
* @param plannedTime 计划发送时间
* @param serviceToken 服务令牌
* @return SendResult 发送结果
*     如果地址中存在错误地址（整个地址错误，或范围地址内一部分地址无
权限），消息将不发送，还是处于草稿状态，
*     此时 SendResult.succeeded=false;
*     如果地址完全正确，或者仅部分地址在某个通道上无权限，消息将发送
成功，将处于发送状态。
*     此时 SendResult.succeeded=true;
*     SendResult.wrongAddressess 中包含了的错误地址列表，这些地址是原始
的发件地址，不是对“范围地址”展开后的最终的地址。
* @throws MessageException
*/
public SendResult sendIndivMessage(int boxId,Message message,
    int[] channels, boolean usesSignature,
    int solution, String datasrc, String idividualParams, Date plannedTime,String
serviceToken) throws MessageException{

}
```

13.5.2.4. getMessageDeliverCount

```
/**
* 查询本次消息在指定通道内实际产生的收件消息数量
* @param messageId 调用 sendMessage 产生的发件消息 id
* @param channel 指定的通道号
* @param serviceToken 服务编号
* @return
*/
public int getMessageDeliverCount(String messageId, int channel,String serviceToken){
}
```

13.5.2.5. getMessageDelivers

```
/**
* 分页查询本次消息在指定通道内实际参数消息状态
* @param messageId 消息 id
* @param channel 指定的通道号
* @param beginIndex 需要返回条目的开始位置，第一条为 0
```

```
* @param count 需要返回的数量。0 返回全部
* @param serviceToken 服务编号
* @return 通道内发送的消息的状态
*/
public ChannelDeliverState[] getMessageDelivers(String messageId, int channel, int
    beginIndex, int count, String serviceToken){
}
```

13.6. 发送带外链的扩展消息 Demo

```
Message m = new Message();
m.msgType=1; //扩展消息
m.properties.put("im_readType","1"); //设置此消息需要点击后才被置为已读
m.properties.put("im_linkUrl","http://my.com/foo"); //设置点击此消息后打开的外链
m.properties.put("ucp_bizType","1"); //消息的业务类型为 1
m.properties.put("ucp_bizTypeName","待办"); //业务类型名称
m.properties.put("my_id","abcd"); //业务系统自定义属性
Address address = new Address();
address.type = "uc_ux"; //统一认证内的人员
address.address = "(12345)张三"; //发给张三
m.to = new Address[]{address};
m.subject = "会议通知"; //主题，注：可以不填
m.content = "请与 3 日下午参加会议"; //内容

int myBoxId;
SendResult result = sendMessage(myBoxId //提前分配给业务系统消息盒 id
    ,m //要发送的消息
    , new int[]{3} //通过微门户 IM 通道通知用户
    , false //不使用签名
    , null //无计划时间，立即发送
    , null //无安全凭证
    )

String sendMsgId = result.messageId; //刚发的消息的 id
doSaveMsgId(); //保存 sendMsgId
```

14. 附件一：Manifest.xsd

15. 附件二：iPortalInit.js

16. 附件三 sudytech-widget-sdk.zip

内部包含 sudytech-widget-sdk.jar 与相关 javadoc

17. 附件四 UcpWebServ.wsdl、 UcpWebServ.xsd

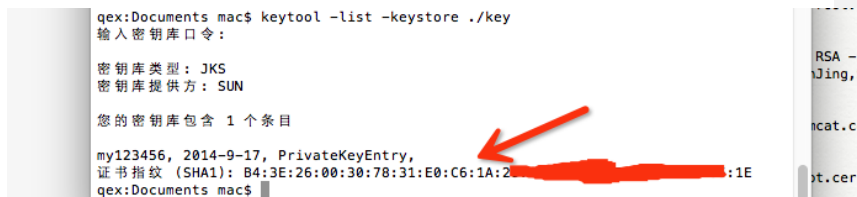
此两个附件是消息的 WebService 定义文件。

18. 附录 1 APK 指纹获取

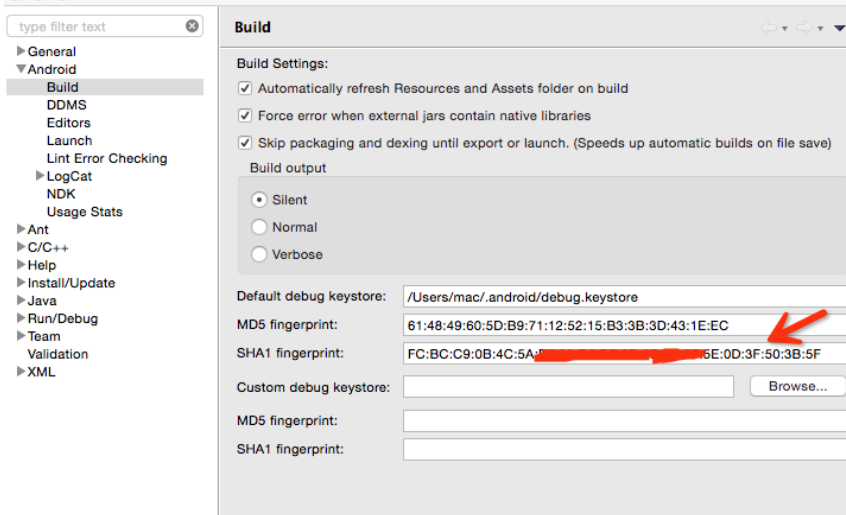
用来签名 apk 的证书库 keystore 中。可以通过 keytool 命令查看证书的指纹。

命令：`keytool -list -keystore ./keystore`

```
qex:Documents mac$ keytool -list -keystore ./key
输入密钥库口令：
密钥库类型：JKS
密钥库提供方：SUN
您的密钥库包含 1 个条目
my123456, 2014-9-17, PrivateKeyEntry,
证书指纹 (SHA1): B4:3E:26:00:30:78:31:E0:C6:1A:2[REDACTED]:1E
qex:Documents mac$
```



adt 开发环境中的指纹查看方式如下，注意发布的 apk 指纹必须使用 keystore 文件中的指纹。



19. 注意事项

使用 html 开发在手机应用时，不要使用同步的 ajax 请求，同步的 ajax 请求会导致整个界面被卡死，影响用户体验。