

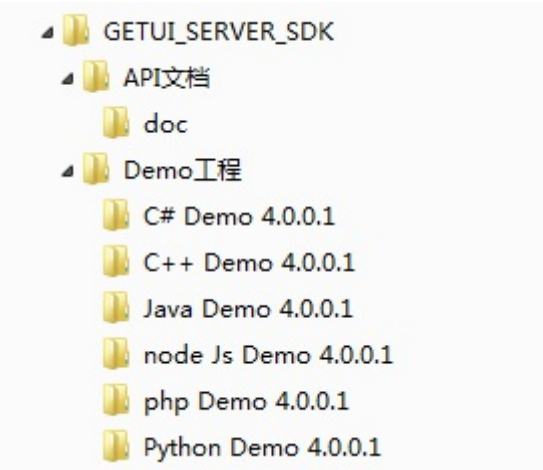
1. Python 开发准备

1.1 前期准备

- 【1】请确保您已经初始化客户端SDK，并成功取得ClientID
- 【2】文档以Python程序为例介绍了如何调用OpenService提供的HTTP服务接口，采用json作为数据传输格式
- 【3】HTTP上行的Json消息体最大长度为25KB
- 【4】请确保工程编码格式为UTF-8，否则可能导致报错：“result=sign_error”

1.2 环境搭建

第一步： 下载服务端SDK开发工具包，目录结构如下图所示:
(下载地址：http://www.getui.com/download/server/GETUI_SERVER_SDK.zip)



第二步： 安装并调试Python环境

2. Python https接口说明

2.1 概述

http 协议是明文传输，内容不需要解密直接可读，https 在 http 基础上添加 SSL/TLS 加密 层，保证数据安全。相对而言，https 协议更安全，http 协议效率更高。

2.2 IGtPush接口类型定义

接口定义	
IGtPush(host, appKey, masterSecret, useSSL)	

参数说明

参数名	默认值	参数描述
host	无	推送os域名：Ip可选填，如果Ip不填程序自动检测用户网络，选择最快的域名连接下发。getui服务url地址，可使用 https url 或者http url（服务端SDK版本号3.4.0.0及以上版本支持该功能）
appKey	无	用于鉴定身份是否合法

masterSecret	无	第三方客户端个推集成鉴权码，用于验证第三方合法性。在客户端集成SDK时需要提供
useSSL	false	是否使用http接口调用:true为https， false为http。默认为http。(服务端SDK版本号4.0.1.0及以上版本支持该功能)

2.3 https接口实例

```
# -*- coding: utf-8 -*-
from igt_push import *
from igetui.template import *
from igetui.template.igt_base_template import *
from igetui.template.igt_transmission_template import *
from igetui.template.igt_link_template import *
from igetui.template.igt_notification_template import *
from igetui.template.igt_notypopload_template import *
from igetui.igt_message import *
from igetui.igt_target import *
from igetui.template import *
import os

APPKEY = "请输入您的APPKEY"
APPID = "请输入您的APPID"
MASTERSECRET = "请输入您的MASTERSECRET"
CID = "请输入您的CID"

def pushMessageToSingle():
    push = IGeTui(None, APPKEY, MASTERSECRET, True)
    template = NotificationTemplateDemo()
    message = IGtSingleMessage()
    message.isOffline = True
    message.offlineExpireTime = 1000 * 3600 * 12
    message.data = template
    message.pushNetWorkType = 1
    target = Target()
    target.appId = APPID
    target.clientId = CID

    try:
        ret = push.pushMessageToSingle(message, target)
        print ret
    except RequestException, e:
        requestId = e.getRequestId()
        ret = push.pushMessageToSingle(message, target, requestId)
        print ret

#透传模板动作内容
def NotificationTemplateDemo():
    template = NotificationTemplate()
    template.appId = APPID
    template.appKey = APPKEY
    template.transmissionType = 1
    template.transmissionContent = u"请输入您要透传的内容"
    template.title = u"请输入通知栏标题"
    template.text = u"请输入通知栏内容"
    template.logo = "icon.png"
    template.logoURL = ""
    template.isRing = True
    template.isVibrate = True
    template.isClearable = True
    return template

if __name__ == "__main__":
    pushMessageToSingle()
```

3. Python 推送模板

本章介绍了PYTHON API的推送模板实例，推送模板结合推送方式即可完成整套推送方案。

3.1 点击通知打开应用模板

###模板说明

pushType	说明	描述
NotificationTemplate	点击通知启动应用	在通知栏显示一条含图标、标题等的通知，用户点击后激活您的应用(iOS不推荐使用该模板，详情请看iOS模板说明)

###模板参数说明

|类名|成员和方法名|长度|必填|说明| |NotificationTemplate|appId|-|是|设定接收的应用| |appKey|-|是|用于鉴定身份是否合法| |title|40中/英字符|是|通知标题| |text|600中/英字符|是|通知内容| |logo|40中/英字符|是|通知的图标名称，包含后缀名（需要在客户端开发式嵌入），如“push.png”| |isRing|-|否|收到通知是否响铃：true响铃，false不响铃。默认响铃。| |isVibrate|-|否|收到通知是否振动：true振动，false不振动。默认振动。| |isClearable|-|否|通知是否可清除：true可清除，false不可清除。默认可清除。| |transmissionType|4Byte|是|收到消息是否立即启动应用：1为立即启动，2则广播等待客户端自启动| |transmissionContent|2048中/英字符|是|透传内容，不支持转义字符| |setAPNInfo|-|否|iOS推送使用该字段，详见iOS模板推送说明|

###NotificationTemplate实例

```
def NotificationTemplateDemo():
    template = NotificationTemplate()
    template.appId = APPID
    template.appKey = APPKEY
    template.transmissionType = 1
    template.transmissionContent = u"请填入透传内容"
    template.title = u"请填入通知标题"
    template.text = u"请填入通知内容"
    template.logo = "icon.png"
    template.logoURL = ""
    template.isRing = True
    template.isVibrate = True
    template.isClearable = True
    return template
```

3.2 点击通知打开网页模板

###模板说明

pushType	说明	描述
LinkTemplate	点击通知打开网页	在通知栏显示一条含图标、标题等的通知，用户点击可打开您指定的网页（iOS不推荐使用该模板，详情请看iOS模板说明）

###模板参数说明

类名	成员和方法名	长度	必填	说明
LinkTemplate	appId	-	是	设定接收的应用
	appKey	-	是	用于鉴定身份是否合法
	title	40中/英字符	是	通知标题
	text	600中/英字符	是	通知内容
	logo	40中/英字符	是	通知的图标名称，包含后缀名（需要在客户端开发时嵌入），如“push.png”
	isRising	-	否	收到通知是否响铃：true响铃，false不响铃。默认响铃。
	isVibrate	-	否	收到通知是否振动：true振动，false不振动。默认振动。
	isClearable	-	否	通知是否可清除：true可清除，false不可清除。默认可清除。
	url	200中/英字符	是	点击通知后打开的网页地址
	setAPNInfo	-	否	iOS推送使用该字段，详见iOS模板说明

###LinkTemplate代码实例

```
def LinkTemplateDemo():
    template = LinkTemplate()
    template.appId = APPID
    template.appKey = APPKEY
    template.title = u"请填入通知标题"
    template.text = u"请填入通知内容"
    template.logo = ""
    template.url = "http://www.baidu.com"
    template.transmissionType = 1
    template.transmissionContent = ''
    template.isRing = True
    template.isVibrate = True
    template.isClearable = True
    return template
```

3.3 点击通知栏弹框下载模板

###模板说明

pushType	说明	描述
NotyPopLoadTemplate	通知栏弹框下载模版	在通知栏显示一条含图标、标题等的通知，用户点击后弹出框，用户可以选择直接下载应用或者取消下载应用。（iOS不支持该模板）

###模板参数说明

类名	成员和方法名	长度	必填	说明
NotyPopLoadTemplate	notyIcon	40中/英字符	是	通知栏图标
	notyTitle	40中/英字符	是	通知栏标题
	notyContent	600中/英字符	是	通知栏内容
	isClearable	—	否	通知栏是否可以清除（默认是）
	isRing	—	否	是否响铃（默认是）
	isVibrate	—	否	收到通知是否振动： true 振动， false 不振动。默认振动。
	popTitle	40中/英字符	是	弹出框标题
	popContent	600中/英字符	是	弹出框内容
	popImage	200中/英字符	是	弹出框图标
	popButton1	4中/英字符	是	弹出框左边按钮名称
	popButton2	4中/英字符	是	弹出框右边按钮名称
	loadIcon	40中/英字符	是	下载图标
	loadTitle	40中/英字符	是	下载标题
	loadUrl	200中/英字符	是	下载地址
	isAutoInstall	—	否	是否自动安装（默认否）

	isActive	—	否	安装完成后是否自动启动应用程序（默认否）
	android	40中/英字符	否	包名，比如：com.sina.weibo
	symbia	40中/英字符	否	UID，比如：0x27000000
	ios	40中/英字符	否	URL Schema，用于唤醒对应的iOS应用

###NotyPopLoadTemplate代码实例

```
def NotyPopLoadTemplateDemo():
    template = NotyPopLoadTemplate()
    template.appId = APPID
    template.appKey = APPKEY
    template.notyIcon = "icon.png"
    template.logoUrl = ""
    template.notyTitle = u"通知弹框下载功能标题"
    template.notyContent= u"通知弹框下载功能内容"
    template.isRing = True
    template.isVibrate = True
    template.isClearable = True
    template.popTitle = u"弹框标题"
    template.popContent = u"弹框内容"
    template.popImage = ""
    template.popButton1 = u"下载"
    template.popButton2 = u"取消"
    template.loadIcon = "file://icon.png"
    template.loadTitle = u"下载内容"
    template.loadUrl = "http://gdown.baidu.com/data/wisegame/c95836e06c224f51/weixinxinqing_5.apk"
    template.isAutoInstall = True
    template.isActive = False
    return template
```

3.4 透传消息模板

###模板说明

pushType	说明	描述
TransmissionTemplate	透传（payload）	数据经SDK传给您的客户端，由您写代码决定如何处理展现给用户

###模板参数说明

类名	成员和方法名 长度	必填	说明
TransmissionTemplate	appId	—	是
	appKey	—	是
	transmissionContent	2048中/英字符	是
	transmissionType	4byte	是
	setAPNInfo	—	否

###TransmissionTemplate代码实例

```
def TransmissionTemplateDemo():
    template = TransmissionTemplate()
    template.transmissionType = 1
```

```
template.appId = APPID
template.appKey = APPKEY
template.transmissionContent = '请填入透传内容'
alertMsg = SimpleAlertMsg()
alertMsg.alertMsg = ""
apn = APNPayload();
apn.alertMsg = alertMsg
apn.badge = 2
apn.sound = ""
apn.addCustomMsg("payload", "payload")
    #apn.contentAvailable=1
    #apn.category="ACTIONABLE"
template.setApnInfo(apn)
/*iOS 推送需要对该字段进行设置具体参数详见iOS模板说明(PYTHON)*/
return template
```

3.5 iOS模板说明

###描述 ###对应接口

def setApnInfo(self, payload):

参数描述

setAPNInfo具体参数含义详见苹果APNs文档：

<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html>

参数说明

类名	方法名	说明
APNPayload	setBadge	应用icon上显示的数字
	setContentAvailable	推送直接带有透传数据
	setSound	通知铃声文件名
	setCategory	在客户端通知栏触发特定的action和button显示
	addCustomMsg	增加自定义的数据

类名	方法名	说明
APNPayload.DictionaryAlertMsg	setBody	通知文本消息字符串
	setActionLocKey	(用于多语言支持) 指定执行按钮所使用的Localizable.strings
	setLocKey	(用于多语言支持) 指定Localizable.strings文件中相应的key
	addLocArg	如果loc-key中使用的占位符，则在loc-args中指定各参数
	setLaunchImage	指定启动界面图片名
	setTitle	通知标题
	setTitleLocKey	(用于多语言支持) 对于标题指定执行按钮所使用的Localizable.strings,仅支持iOS8.2以上版本
	addTitleLocArg	对于标题, 如果loc-key中使用的占位符，则在loc-args中指定各参数,仅支持iOS8.2以上版本

代码示例：

```
def TransmissionTemplateDemo():
    template = TransmissionTemplate()
    template.transmissionType = 2
    template.appId = APPID
    template.appKey = APPKEY
    template.transmissionContent = '请填入透传内容'
    template = TransmissionTemplate()
    #设置APNs信息
    apnpayload = APNPayload()
    apnpayload.badge = 4
    apnpayload.sound = "test1.wav"
    apnpayload.contentAvailable = 1
    apnpayload.category = "ACTIONABLE"

    #简单类型如下设置
    alertMsg = SimpleAlertMsg()
    alertMsg.alertMsg = "alertMsg";
    #字典类型如下设置
    #alertMsg = DictionaryAlertMsg()
    #alertMsg.body = 'body'
    #alertMsg.actionLockKey = 'actionLockey'
    #alertMsg.lockKey = 'lockey'
    #alertMsg.locArgs=['loc-args']
    #alertMsg.launchImage = 'launchImage'
    # iOS8.2以上版本支持
    #alertMsg.title = 'Title'
    #alertMsg.titleLocArgs = ['TitleLocArg']
    #alertMsg.titleLockKey = 'TitleLockKey'

    #可以设置字典类型AlertMsg和简单类型AlertMsg其中之一
    apnpayload.alertMsg=alertMsg

    template.setApnInfo(apnpayload)
```


4. Python 推送方式

本章介绍了PYTHON API的推送方式实例，推送模板结合推送方式即可完成整套推送方案。

4.1 对单个用户推送接口

###接口说明

接口名称	支持推送类型	说明
pushMessageToSingle	透传（payload）、点击通知启动应用、点击通知打开网页等	对单个用户（ClientID）推送

 接口名称正常推送不需要传requestId，如果发生异常重试时将requestId传入，具体用法详见示例代码。

```
def pushMessageToSingle(self, message, target, requestId=None)
```

###pushMessageToSingle代码实例

```
__author__ = 'wei'
# -*- coding: utf-8 -*-
#更新时间为2013年11月07日
#增加了ios的离线消息推送,iOS不支持IGtNotyPopLoadTemplate模板
#更新时间为2013年02月24日
#1.增加了通知弹框下载模板
#2.统一了toapp的接口时间, 单位为毫秒
#3.允许ios用户离线状态下的APNs转发message字段为空,
#4.增加了查询用户状态接口,
#5.任务停止功能,
#6.增加ToList接口每个用户返回用户状态的功能
#
#
from igt_push import *
```

```

from igetui.template import *
from igetui.template.igt_base_template import *
from igetui.template.igt_transmission_template import *
from igetui.template.igt_link_template import *
from igetui.template.igt_notification_template import *
from igetui.template.igt_notypopload_template import *
from igetui.igt_message import *
from igetui.igt_target import *
from igetui.template import *

APPKEY = "请输入您的APPKEY"
APPID = "请输入您的APPID"
MASTERSECRET = "请输入您的MASTERSECRET"
CID = "请输入您的CID"
HOST = 'http://sdk.open.api.igexin.com/apiex.htm'

def pushMessageToSingle():
    push = IGeTui(HOST, APPKEY, MASTERSECRET)
    #push = IGeTui("", APPKEY, MASTERSECRET) // 此方式可通过获取服务端地址列表判断最快域名后进行消息推送，每10分钟检查一次最快域名
    #消息模版：
    #1.TransmissionTemplate:透传功能模板
    #2.LinkTemplate:通知打开链接功能模板
    #3.NotificationTemplate: 通知透传功能模板
    #4.NotyPopLoadTemplate: 通知弹框下载功能模板
    #template = NotificationTemplateDemo()
    #template = LinkTemplateDemo()
    template = TransmissionTemplateDemo()
    #template = NotyPopLoadTemplateDemo()

    message = IGtSingleMessage()
    message.isOffline = True
    message.offlineExpireTime = 1000 * 3600 * 12
    message.data = template
    message.pushNetWorkType = 1 // 设置是否根据WIFI推送消息，1为wifi推送，0为不限制推送
    target = Target()
    target.appId = APPID
    target.clientId = CID

    try:
        ret = push.pushMessageToSingle(message, target)
        print ret
    except RequestException, e:
        requestId = e.getRequestId()
        ret = push.pushMessageToSingle(message, target, requestId)
        print ret

#透传模板动作内容
def TransmissionTemplateDemo():
    template = TransmissionTemplate()
    template.transmissionType = 1
    template.appId = APPID
    template.appKey = APPKEY
    template.transmissionContent = '请填入透传内容'
    #
    # iOS setAPNInfo
    #
    apnpayload = APNPayload()
    #
    apnpayload.badge = 4
    #
    apnpayload.sound = "sound"
    #
    apnpayload.addCustomMsg("payload", "payload")
    # #
    # apnpayload.contentAvailable = 1
    # #
    # apnpayload.category = "ACTIONABLE"
    #
    #
    #
    alertMsg = DictionaryAlertMsg()
    #
    alertMsg.body = 'body'
    #
    alertMsg.actionLockKey = 'actionLockey'
    #
    alertMsg.lockKey = 'lockey'
    #
    alertMsg.locArgs=['locArgs']
    #
    alertMsg.launchImage = 'launchImage'
    #
    # iOS8.2以上版本支持
    # #
    # alertMsg.title = 'Title'
    # #
    # alertMsg.titleLocArgs = ['TitleLocArg']
    # #
    # alertMsg.titleLockKey = 'TitleLockKey'
    #
    apnpayload.alertMsg=alertMsg
    #
    template.setApnInfo(apnpayload)

```



```
// 设置通知定时展示时间，结束时间与开始时间相差需大于6分钟，消息推送后，客户端将在指定时间差内展示消息（误差6分钟）
begin = "2015-03-04 17:40:22";
end = "2015-03-04 17:47:24";
template.setDuration(begin, end)
return template

pushMessageToSingle()
```

##4.2 对指定用户列表推送接口

###接口说明

接口名称	支持推送类型	说明
pushMessageToList	透传（payload）、点击通知启动应用、点击通知打开网页等	（通过ClientID列表）群推，可查看ClientID列表中每个用户的在线状态

###pushMessageToList代码实例

```
__author__ = 'wei'
# -*- coding: utf-8 -*-
#更新时间为2013年11月07日
#增加了ios的离线消息推送,ios不支持IGtNotyPopLoadTemplate模板
#更新时间为2013年02月24日
#1.增加了通知弹框下载模板
#2.统一了toapp的接口时间，单位为毫秒
#3.允许ios用户离线状态下的APNs转发message字段为空，
#4.增加了查询用户状态接口，
#5.任务停止功能，
#6.增加ToList接口每个用户返回用户状态的功能
#
#
from igt_push import *
from igetui.template import *
from igetui.template.igt_base_template import *
from igetui.template.igt_transmission_template import *
from igetui.template.igt_link_template import *
from igetui.template.igt_notification_template import *
from igetui.template.igt_notypopload_template import *
from igetui.igt_message import *
from igetui.igt_target import *
from igetui.template import *
#toList接口每个用户返回用户状态开关,true: 打开 false: 关闭
os.environ['gexin_pushList_needDetails'] = 'false'
APPKEY = "请输入您的APPKEY"
APPID = "请输入您的APPID"
MASTERSECRET = "请输入您的MASTERSECRET"
CID = "请输入您的CID"
HOST = 'http://sdk.open.api.igexin.com/apiex.htm'

def pushMessageToList():
    push = IGeTui(HOST, APPKEY, MASTERSECRET)
    #push = IGeTui("",APPKEY,MASTERSECRET)//此方式可通过获取服务端地址列表判断最快域名后进行消息推送，每10分钟检查一次最快域名
    #消息模版：
    #1.TransmissionTemplate:透传功能模板
    #2.LinkTemplate:通知打开链接功能模板
    #3.NotificationTemplate: 通知透传功能模板
    #4.NotyPopLoadTemplate: 通知弹框下载功能模板
    #template = NotificationTemplateDemo()
    template = LinkTemplateDemo()
    #template = TransmissionTemplateDemo()
    #template = NotyPopLoadTemplateDemo()
    message = IGtListMessage()
    message.data = template
    message.pushNetworkType = 1//设置是否根据WIFI推送消息，1为wifi推送，0为不限制推送
    message.isOffline = True
    message.offlineExpireTime = 1000 * 3600 * 12
    target1 = Target()
    target1.appId = APPID
    target1.clientId = CID
    #target2 = Target()
```

```
#target2.appId = APPID
#target2.clientId = CID
targets = [target1]
contentId = push.getContentId(message)
ret = push.pushMessageToList(contentId, targets)

#通知链接模板动作内容
def LinkTemplateDemo():
    template = LinkTemplate()
    template.appId = APPID
    template.appKey = APPKEY
    template.title = u"请填入通知标题"
    template.text = u"请填入通知内容"
    template.logo = ""
    template.url = "http://www.baidu.com"
    template.transmissionType = 1
    template.transmissionContent = ''
    template.isRing = True
    template.isVibrate = True
    template.isClearable = True
    #    iOS setAPNInfo
    #    apnpayload = APNPayload()
    #    apnpayload.badge = 4
    #    apnpayload.sound = "sound"
    #    apnpayload.addCustomMsg("payload", "payload")
    # #    apnpayload.contentAvailable = 1
    # #    apnpayload.category = "ACTIONABLE"
    #
    #
    #    alertMsg = DictionaryAlertMsg()
    #    alertMsg.body = 'body'
    #    alertMsg.actionLocKey = 'actionLockey'
    #    alertMsg.locKey = 'lockey'
    #    alertMsg.locArgs=['locArgs']
    #    alertMsg.launchImage = 'launchImage'
    #    # iOS8.2以上版本支持
    # #    alertMsg.title = 'Title'
    # #    alertMsg.titleLocArgs = ['TitleLocArg']
    # #    alertMsg.titleLocKey = 'TitleLocKey'
    #    apnpayload.alertMsg=alertMsg
    #    template.setApnInfo(apnpayload)

    // 设置通知定时展示时间，结束时间与开始时间相差需大于6分钟，消息推送后，客户端将在指定时间差内展示消息（误差6分钟）
    begin = "2015-03-04 17:40:22";
    end = "2015-03-04 17:47:24";
    template.setDuration(begin, end)
    return template
print ret

pushMessageToList()
```

4.3 对指定应用群推接口

###接口说明

接口名称	支持推送类型	说明
pushMessageToApp	透传（payload）、点击通知启动应用、点击通知打开网页等	（通过应用AppID）群推，给所有符合条件的客户端用户推送

###pushMessageToApp代码实例

```
__author__ = 'wei'
# -*- coding: utf-8 -*-
#更新时间为2013年11月07日
#增加了iOS的离线消息推送,iOS不支持IGtNotyPopLoadTemplate模板
#更新时间为2013年02月24日
#1.增加了通知弹框下载模板
#2.统一了toapp的接口时间，单位为毫秒
#3.允许ios用户离线状态下的APNs转发message字段为空，
#4.增加了查询用户状态接口，
```

```

#5.任务停止功能，
#6.增加ToList接口每个用户返回用户状态的功能
#
#
from igt_push import *
from igetui.template import *
from igetui.template.igt_base_template import *
from igetui.template.igt_transmission_template import *
from igetui.template.igt_link_template import *
from igetui.template.igt_notification_template import *
from igetui.template.igt_notypopload_template import *
from igetui.igt_message import *
from igetui.igt_target import *
from igetui.template import *

APPKEY = "请输入您的APPKEY"
APPID = "请输入您的APPID"
MASTERSECRET = "请输入您的MASTERSECRET"
CID = "请输入您的CID"
HOST = 'http://sdk.open.api.igexin.com/apiex.htm'

def pushMessageToApp():
    push = IGeTui(HOST, APPKEY, MASTERSECRET)
    #push = IGeTui("", APPKEY, MASTERSECRET)//此方式可通过获取服务端地址列表判断最快域名后进行消息推送，每10分钟检查一次最快域名
    #消息模版：
    #1.TransmissionTemplate:透传功能模板
    #2.LinkTemplate:通知打开链接功能模板
    #3.NotificationTemplate: 通知透传功能模板
    #4.NotyPopLoadTemplate: 通知弹框下载功能模板
    template = NotificationTemplateDemo()
    #template = LinkTemplateDemo()
    #template = TransmissionTemplateDemo()
    #template = NotyPopLoadTemplateDemo()
    message = IGtAppMessage()
    message.setSpeed(100)//设置消息推送速度，单位为条/秒，例如填写100，则为100条/秒。仅支持对指定应用群推接口。
    message.data = template
    message.pushNetWorkType = 1//设置是否根据WIFI推送消息，1为wifi推送，0为不限制推送
    message.isOffline = True
    message.offlineExpireTime = 1000 * 3600 * 12
    message.appIdList.extend([APPID])
    #message.phoneTypeList.extend(["ANDROID", "IOS"])
    #message.provinceList.extend(["浙江", "上海"])
    #message.tagList.extend(["开心"])
    ret = push.pushMessageToApp(message)
    print ret

#通知透传模板动作内容
def NotificationTemplateDemo():
    template = NotificationTemplate()
    template.appId = APPID
    template.appKey = APPKEY
    template.transmissionType = 1
    template.transmissionContent = u"请填入透传内容"
    template.title = u"请填入通知标题"
    template.text = u"请填入通知内容"
    template.logo = "icon.png"
    template.logoURL = ""
    template.isRing = True
    template.isVibrate = True
    template.isClearable = True
    #
    # iOS setAPNInfo
    # apnpayload = APNPPayload()
    # apnpayload.badge = 4
    # apnpayload.sound = "sound"
    # apnpayload.addCustomMsg("payload", "payload")
    # # apnpayload.contentAvailable = 1
    # # apnpayload.category = "ACTIONABLE"
    #
    #
    # alertMsg = DictionaryAlertMsg()
    # alertMsg.body = 'body'
    # alertMsg.actionLockKey = 'actionLockey'
    # alertMsg.lockKey = 'lockey'
    # alertMsg.locArgs=['locArgs']
    # alertMsg.launchImage = 'launchImage'
    # # iOS8.2以上版本支持
    # # alertMsg.title = 'Title'

```

```
# # alertMsg.titleLocArgs = ['TitleLocArg']
# # alertMsg.titleLocKey = 'TitleLocKey'
# apnpayload.alertMsg=alertMsg
# template.setApnInfo(apnpayload)

// 设置通知定时展示时间，结束时间与开始时间相差需大于6分钟，消息推送后，客户端将在指定时间差内展示消息（误差6分钟）
begin = "2015-03-04 17:40:22";
end = "2015-03-04 17:47:24";
template.setDuration(begin, end)

return template

pushMessageToApp()
```

5. Python 其他接口

5.1 停止任务接口

###接口说明

接口名称	方法名	类型	说明
stop	-	String	根据taskID对正处于推送状态，或者未接收的消息停止下发

###代码实例 类名:IGtPush（调用stop方法停止任务）

```
def stopTask():
    push = IGeTui(HOST, APPKEY, MASTERSECRET)
    print push.stop("OSA-0226_50RYYPFmos9eQEHZrkAf27");
```

5.2 查询用户当前状态

###接口说明

接口名称	方法名	类型	说明
getClientIdStatus	APPID	String	设置推送的appid
	CLIENTID	String	用户唯一标识符（获取方式请参考<前期准备>第一章节）

###代码实例 类名:IGtPush(调用getClientIdStatus方法获取用户状态)

```
def getUserStatus():
    push = IGeTui(HOST, APPKEY, MASTERSECRET)
    print push.getClientIdStatus(APPID, CID)
```

5.3 获取推送结果实例

###接口说明

参数	参数说明
url	接口地址（http://sdk.open.api.igexin.com/apiex.htm）
Appkey	用于鉴定身份是否合法
masterSecret	第三方客户端个推集成鉴权码，用于验证第三方合法性。在客户端集成SDK时需要提供

taskId	唯一识别号
--------	-------

回执	参数说明
	{taskId=OSA-0820_uQ7gevLuGS7Odz8F52ZSB9, result=ok, msgTotal=59, msgProcess=0}
	taskId: 表示任务ID
	msgTotal:表示有效可下发总数
	result: OK执行成功
	result: sign_error 表示校验失败
	msgProcess: 收到消息回执总数

###代码实例

```
# -*- coding: utf-8 -*-
#数据将在30分钟之后提供查询
#最后更新时间为2013.10.22
import hashlib
import urllib, urllib2, json
def getPushResult(url,appKey,masterSecret,taskId):
    params = {}
    params["action"] = "getPushMsgResult"
    params["appkey"] = appKey
    params["taskId"] = taskId
    sign = createSign(params, masterSecret)
    params["sign"] = sign
    rep = httpPost(url, params)
    return rep
def createSign(params, masterSecret):
    sign = masterSecret
    for (k,v) in params.items():
        sign = sign+k+v
    return hashlib.md5(sign).hexdigest()
def httpPost(url, params):
    data_json = json.dumps(params)
    req = urllib2.Request(url, data_json)
    res_stream = urllib2.urlopen(req, timeout = 60)
    page_str = res_stream.read()
    page_dict = eval(page_str)
    return page_dict
rep =
getPushResult("http://sdk.open.api.igexin.com/apiex.htm", "tpDVam96sY8pxhwBupJ462", "TBokfpttQJ6aHIhBE9y867", "GT_1017_gJs4GvJxZV77gdgBKsuv09")
print rep
```

5.4 获取单日用户数据

###描述 调用此接口可以获取某个应用单日的用户数据（用户数据包括：新增用户数，累计注册用户总数，在线峰值，日联网用户数）（目前只支持查询1天前的数据） ###接口 **接口名称： **def queryAppUserDataByDate(self, appld, date) 接口参数：

字段	说明
appld	应用ID
date	查询的日期（格式：yyyyMMdd）

返回值

字段	上级	含义
----	----	----

result	-	成功：ok
data	-	查询数据对象
appld	data	请求的AppId
date	data	查询的日期（格式：yyyyMMdd）
newRegistCount	data	新注册用户数
registTotalCount	data	累计注册用户数
activeCount	data	活跃用户数
onlineCount	data	在线用户数

代码实例

```
push = IGeTui(HOST, APPKEY, MASTERSECRET)
res = push.queryAppUserDataByDate(APPID, "20150605")
print(res)
```

5.5 获取单日推送数据

###描述 调用此接口可以获取某个应用单日的推送数据（推送数据包括：发送总数，在线发送数，接收数，展示数，点击数）（目前只支持查询1天前的数据）

###接口 **接口名称: **def queryAppPushDataByDate(self, appld, date) 接口参数

字段	说明
appld	应用ID
date	查询的日期（格式：yyyyMMdd）

返回值

字段	上级	含义
result	-	成功：ok
data	-	查询数据对象
appld	data	请求的AppId
date	data	查询的日期（格式：yyyyMMdd）
sendCount	data	发送总数
sendOnlineCount	data	在线发送数
receiveCount	data	接收数
showCount	data	展示数
clickCount	data	点击数

代码实例

```
push = IGeTui(HOST, APPKEY, MASTERSECRET)
res = push.queryAppPushDataByDate(APPID,"20150605")
print(res)
```
