

# 个推 SDK 文档

getui.com

October 2016

## 目录

https 接口说明	2
1. 概述	2
2. IGtPush 接口类型定义	2
2.1 函数说明	2
2.2 参数说明	2
3. https 使用方法	2
4. https 接口实例	3

## https 接口说明

### 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. IGtPush 接口类型定义

#### 2.1 函数说明

```
IGtPush(String host, String appKey, String masterSecret)

IGtPush(String appKey, String masterSecret, boolean useSSL)
```

#### 2.2 参数说明

参数名	类型	必需	默认值	参数描述
host	string	否	无	推送 os 域名.host 可选填，如果 host 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	string	是	无	用于鉴定身份是否合法
masterSecret	string	是	无	第三方客户端个推集成鉴权码，用于验证第三方合法性。在服务端推送鉴权时使用
useSSL	boolean	否	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）

### 3. https 使用方法

方法一：

```
// 此处true为https域名，false为http，默认为false。java推荐使用此方法，
// 没有特别需求不需要指定host。
IGtPush push = new IGtPush(appkey, master, true);
```

方法二：

```
// host为域名，优先使用host进行连接，根据域名协议来区分，将代码中host
// 的值变为"https://api.getui.com/apiex.htm";所有语言均支持此方法。
IGtPush push = new IGtPush(host, appkey, master);
```

## 4. https 接口实例

```
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.base.impl.SingleMessage;
import com.gexin.rp.sdk.base.impl.Target;
import com.gexin.rp.sdk.exceptions.RequestException;
import com.gexin.rp.sdk.http.IGtPush;
import com.gexin.rp.sdk.template.LinkTemplate;
public class Demo {
    static String appId = "b03c5cfef65ed30108f0a3fd82c3f6b41";
    static String appkey = "110000";
    static String master = "a02a76119b20d4e31620d7597a3b4f35";
    static String CID = "873ffc4fec7bfd43d4705639eacb41d0";
    static String host = "https://api.getui.com/apiex.htm";
    public static void main(String[] args) throws Exception {
        // https连接
        IGtPush push = new IGtPush(appkey, master, true);
        // 此处true为https域名, false为http, 默认为false。Java语言推荐使用此方式
        // IGtPush push = new IGtPush(host, appkey, master);
        // host为域名, 根据域名区分是http协议/https协议
        LinkTemplate template = linkTemplateDemo();
        SingleMessage message = new SingleMessage();
        message.setOffline(true);
        // 离线有效时间, 单位为毫秒, 可选
        message.setOfflineExpireTime(24 * 3600 * 1000);
        message.setData(template);
        message.setPushNetWorkType(0); //
        // 可选, 判断是否客户端是否wifi环境下推送, 1为在WIFI环境下, 0为不限制网络环境。
        Target target = new Target();
        target.setAppId(appId);
        target.setClientId(CID);
        // 用户别名推送, cid和用户别名只能2者选其一
        // String alias = "个";
        // target.setAlias(alias);
        IPushResult ret = null;
        try {
            ret = push.pushMessageToSingle(message, target);
        } catch (RequestException e) {
            e.printStackTrace();
            ret = push.pushMessageToSingle(message, target, e.getRequestId());
        }
        if (ret != null) {
            System.out.println(ret.getResponse().toString());
        } else {
            System.out.println("服务器响应异常");
        }
    }
}
```

```
    }  
}  
  
public static LinkTemplate linkTemplateDemo() {  
    LinkTemplate template = new LinkTemplate();  
    // 设置APPID与APPKEY  
    template.setAppId(appId);  
    template.setAppkey(appkey);  
    // 设置通知栏标题与内容  
    template.setTitle("请输入通知栏标题");  
    template.setText("请输入通知栏内容");  
    // 配置通知栏图标  
    template.setLogo("icon.png");  
    // 配置通知栏网络图标，填写图标URL地址  
    template.setLogoUrl("");  
    // 设置通知是否响铃，震动，或者可清除  
    template.setIsRing(true);  
    template.setIsVibrate(true);  
    template.setIsClearable(true);  
    // 设置打开的网址地址  
    template.setUrl("http://www.baidu.com");  
    return template;  
}
```