

快速入门

第一步： 创建个推开发者账号

访问个推开发者中心<http://dev.getui.com>，注册个推开发者帐号，如下图所示：



第二步： 获取访问凭证

创建账号成功并登录后，可以在应用列表中看到自动生成的【个推App Demo】，然后点击【创建推送】，如下图所示：



然后点击左侧菜单栏【配置管理】->【应用配置】，就能看到关于此app的配置信息，这些配置信息就是第五步使用SDK发送消息需要的AppID,AppKey,MasterSecret，如下图所示：



第三步： 安装Demo应用到手机

在应用【个推App Demo】点击【下载应用】，下载后安装到手机上，用来接受推送消息：



第四步： 安装个推C# SDK

下载demo程序

下载个推C# Demo，下载地址

为：http://www.getui.com/download/docs/getui/server/GETUI_C#_SDK_4.1.0.0.zip，目录结构如下图所示：



第五步：使用C# SDK发送消息

下面以app推送接口为例来发送消息：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Google.ProtocolBuffers;
using com.gexin.rp.sdk.dto;
using com.igetui.api.openservice;
using com.igetui.api.openservice.igetui;
using com.igetui.api.openservice.igetui.template;
using com.igetui.api.openservice.payload;
using System.Net;

/**
 *
 * 说明：
 * 此工程是一个测试工程，所用的相关.dll文件，都已经存在protobuf
文件里，需要加载到References里。
 * 工程中还有一个System.Web.Extensions文件，这个文件是用到
Framework里V4.0版本的，
 * 一般路径如下： C:\Program Files\Reference
Assemblies\Microsoft\Framework\NETFramework\v4.0,
 * 或如下路径： C:\Program Files\Reference
Assemblies\Microsoft\Framework\v3.5没有的也可以在protobuf
文件夹里找到。如再有问题，请直接联系技术客服，谢谢！
 * GetuiServerApiSDK：此.dll文件为个推C#版本的SDK文件
 * Google.ProtocolBuffers：此.dll文件为Google的数据交换格式文件
 * 注：
 * 新增一个连接超时时间设置，通过在环境变量--用户变量中增加名为：
GETUI_TIMEOUT 的变量（修改环境变量，
```

```

    电脑重启后才能生效），值则是超时时间，如不设定，则默认为20秒。
    **/

namespace GetuiServerApiSDKDemo
{
    public class demo
    {
        //参数设置 <-----参数需要重新设置----->
        //http的域名
        private static String HOST =
"http://sdk.open.api.igexin.com/apiex.htm";

        //https的域名
        //private static String HOST = "https://api.getui.com/apiex.htm";

        //定义常量, appId、appKey、masterSecret 采用本文档 "第二步 获取访问凭证" 中获得的应用配置
        private static String APPID = "";
        private static String APPKEY = "";
        private static String MASTERSECRET = "";

        static void Main(string[] args)
        {
            //toList接口每个用户状态返回是否开启，可选
            Console.OutputEncoding = Encoding.GetEncoding(936);

            Environment.SetEnvironmentVariable("gexin_pushList_needDetails",
            "true");

            pushMessageToApp();
        }
        //pushMessageToApp接口测试代码
        private static void pushMessageToApp()
        {
            IGtPush push = new IGtPush(HOST, APPKEY, MASTERSECRET);
            // 定义"AppMessage"类型消息对象，设置消息内容模板、发送的目标App列表、是否支持离线发送、以及离线消息有效期(单位毫秒)
            AppMessage message = new AppMessage();

            TransmissionTemplate template = TransmissionTemplateDemo();

            message.IsOffline = true;           // 用户当前不在线时，是否离线存储,可选
            message.OfflineExpireTime = 1000 * 3600 * 12; // 离线有效时间，单位为毫秒，可选
            message.Data = template;
        }
    }
}

```

```

        //判断是否客户端是否wifi环境下推送，2:4G/3G/2G,1为在WIFI环境
        下，0为无限制环境
        //message.PushNetWorkType = 0;
        //message.Speed = 1000;

        List<String> appIdList = new List<string>();
        appIdList.Add(APPID);

        List<String> phoneTypeList = new List<string>(); //通知接收者的手
        机操作系统类型
        //phoneTypeList.Add("ANDROID");
        //phoneTypeList.Add("IOS");

        List<String> provinceList = new List<string>(); //通知接收者所在省
        份
        //provinceList.Add("浙江");
        //provinceList.Add("上海");
        //provinceList.Add("北京");

        List<String> tagList = new List<string>();
        //tagList.Add("中文");

        message.AppIdList = appIdList;
        message.PhoneTypeList = phoneTypeList;
        message.ProvinceList = provinceList;
        message.TagList = tagList;

        String pushResult = push.pushMessageToApp(message);
        System.Console.WriteLine("-----
        -----");
        System.Console.WriteLine("服务端返回结果: " + pushResult);
    }

    //透传模板动作内容
    public static TransmissionTemplate TransmissionTemplateDemo()
    {
        TransmissionTemplate template = new TransmissionTemplate();
        template.AppId = APPID;
        template.AppKey = APPKEY;
        //应用启动类型，1：强制应用启动 2：等待应用启动
        template.TransmissionType = 1;
        //透传内容
        template.TransmissionContent = "测试";
        //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）
        String begin = "2015-03-06 14:36:10";

```

```
String end = "2015-03-06 14:46:20";
template.setDuration(begin, end);

return template;
}
}
}
```

将上面的代码文件放入C# Demo的根目录，然后运行。如果收到下面一样的推送，恭喜你~推送成功了。

