

C# 推送模板

本章介绍了C# 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。

1. 点击通知打开应用模板

1.1 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

1.2 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（需要在客户端开发时嵌入），如“push.png”
	LogoUrl	100中/英字符	否	通知图标URL地址
	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
setDuration	—	否	消息展示时间

1.3 NotificationTemplate代码实例

```

public static NotificationTemplate NotificationTemplateDemo()
{
    NotificationTemplate template = new NotificationTemplate();
    template.AppId = APPID;
    template.AppKey = APPKEY;
    //通知栏标题
    template.Title = "请填写通知标题";
    //通知栏内容
    template.Text = "请填写通知内容";
    //通知栏显示本地图片
    template.Logo = "";
    //通知栏显示网络图标
    template.LogoURL = "";
    //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 = "请填写透传内容";
    //接收到消息是否响铃，true：响铃 false：不响铃
    template.IsRing = true;
    //接收到消息是否震动，true：震动 false：不震动
    template.IsVibrate = true;
    //接收到消息是否可清除，true：可清除 false：不可清除

```

```

template.IsClearable = true;
//设置客户端展示时间
//String begin = "2015-03-06 14:36:10";
//String end = "2015-03-06 14:46:20";
//template.setDuration(begin, end);
return template;
}

```

2. 点击通知打开网页模板

2.1 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

2.2 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”
	LogoUrl	100中/英字符	否	通知图标URL地址
	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。

2.3 LinkTemplate代码实例

```
public static LinkTemplate LinkTemplateDemo()
{
    LinkTemplate template = new LinkTemplate();
    template.AppId = APPID;
    template.AppKey = APPKEY;
    //通知栏标题
    template.Title = "请填写通知标题";
    //通知栏内容
    template.Text = "请填写通知内容";
    //通知栏显示本地图片
    template.Logo = "";
    //通知栏显示网络图标，如无法读取，则显示本地默认图标，可为空
    template.LogoURL = "";
    //打开的链接地址
    template.Url = "http://www.baidu.com";
    //接收到消息是否响铃，true：响铃 false：不响铃
    template.IsRing = true;
    //接收到消息是否震动，true：震动 false：不震动
    template.IsVibrate = true;
    //接收到消息是否可清除，true：可清除 false：不可清除
    template.IsClearable = true;
    return template;
}
```

3. 点击通知栏弹框下载模板

3.1 模板说明

PUSHTYPE	说明	描述
NotyPopLoadTemplate	通知栏弹框下载模板	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）

3.2 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
	IsCleared	—	否	通知栏是否可以清除（默认是）
	IsBelled	—	否	是否响铃（默认是）
	IsVibrationed	—	否	是否震动（默认是）
	IsVibrationed	—	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	是	弹出框右边按钮名称

	中/英字符		
LoadIcon	40 中/英字符	是	下载图标，如果是本地图标，则需要添加file://前缀；如果是网络图标，则直接填写图标网络地址
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）
IsActivated	—	否	安装完成后是否自动启动应用程序（默认否）
AndroidMark	40 中/英字符	否	安卓标识
SymbianMark	40 中/英字符	否	塞班标识
IphoneMark	40 中/英字符	否	苹果标识
setDuration	-	否	收到消息的展示时间

3.3 NotyPopLoadTemplate代码实例

```
public static NotyPopLoadTemplate NotyPopLoadTemplateDemo()
{
```

```

NotyPopLoadTemplate template = new NotyPopLoadTemplate();
template.AppId = APPID;
template.AppKey = APPKEY;
//通知栏标题
template.NotyTitle = "请填写通知标题";
//通知栏内容
template.NotyContent = "请填写通知内容";
//通知栏显示本地图片
template.NotyIcon = "icon.png";
//通知栏显示网络图标
template.LogoURL = "http://www-igexin.qiniudn.com/wp-
content/uploads/2013/08/logo_getui1.png";
//弹框显示标题
template.PopTitle = "弹框标题";
//弹框显示内容
template.PopContent = "弹框内容";
//弹框显示图片
template.PopImage = "";
//弹框左边按钮显示文本
template.PopButton1 = "下载";
//弹框右边按钮显示文本
template.PopButton2 = "取消";
//通知栏显示下载标题
template.LoadTitle = "下载标题";
//通知栏显示下载图标,可为空
template.LoadIcon = "file://push.png";
//下载地址,不可为空
template.LoadUrl =
"http://www.appchina.com/market/d/425201/cop.baidu_0/com.gexin.im.ap
k";
//应用安装完成后,是否自动启动
template.IsActivated = true;
//下载应用完成后,是否弹出安装界面, true: 弹出安装界面, false: 手
动点击弹出安装界面
template.IsAutoInstall = true;
//接收到消息是否响铃, true: 响铃 false: 不响铃
template.IsBelled = true;
//接收到消息是否震动, true: 震动 false: 不震动
template.IsVibrationed = true;
//接收到消息是否可清除, true: 可清除 false: 不可清除
template.IsCleared = true;
return template;
}

```

4. 透传消息模板

4.1 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

4.2 模板参数说明

类名	成员和方法名	长度	必填	说明
TransmissionTemplate	AppId	-	是	设定接收的应用
	AppKey	-	是	用于鉴定身份是否合法
	TransmissionContent	2048 中/英 字符	是	透传内容，不支持转义字符
	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
	setDuration	-	否	收到消息的展示时间
	setAPNInfo	-	否	iOS推送使用该字段

4.3 TransmissionTemplate代码实例


```

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 = "透传内容";

    return template;
}

```

4.4 iOS推送说明

iOS推送需要在代码中通过TransmissionTemplate的setAPNInfo接口设置相应的APNs通知参数。透传模板传输的数据最大为是2KB，APNs传输数据最大支持2KB。

4.4.1 对应接口

setAPNInfo(payload);

4.4.2 参数描述

setAPNInfo具体参数含义详见苹果APNs文档：<https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/PayloadKeyReference.html>

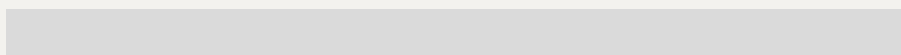
4.4.3 参数说明：

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
	setAlertMsg	通知消息体
	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
	MultiMedias	设置多媒体资源列表，单个列表最多支持3个资源

类名	方法名	说明
APNPayload.SimpleAlertMsg	SimpleAlertMsg	通知文本消息字符串
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

类名	方法名	说明
MultiMedia	rid	设置资源id
	url	设置资源url
	type	设置资源类别, MediaType.pic 图片、MediaType.audio 音频、MediaType.video 视频
	setIsOnlyWifi	设置是否在WIFI下才展示多媒体消息, 如果设置true但未使用WIFI时会展示成普通通知

4.4.4 代码示例



```

public static TransmissionTemplate TransmissionTemplateDemo()
{
    TransmissionTemplate template = new TransmissionTemplate();
    template.AppId = APPID;
    template.AppKey = APPKEY;
    template.TransmissionType = 1;    //应用启动类型，1：强制应用启动
    2：等待应用启动
    template.TransmissionContent = ""; //透传内容

    //iOS简单推送
    //APNPayload apnpayload = new APNPayload();
    //SimpleAlertMsg alertMsg = new SimpleAlertMsg("alertMsg");
    //apnpayload.AlertMsg = alertMsg;
    //apnpayload.Badge = 11;
    //apnpayload.ContentAvailable = 1;
    //apnpayload.Category = "";
    //apnpayload.Sound = "";
    //apnpayload.addCustomMsg("", "");
    //template.setAPNInfo(apnpayload);

    //APN高级推送
    APNPayload apnpayload = new APNPayload();
    DictionaryAlertMsg alertMsg = new DictionaryAlertMsg();
    alertMsg.Body = "Body";
    alertMsg.ActionLocKey = "ActionLocKey";
    alertMsg.LocKey = "LocKey";
    alertMsg.addLocArg("LocArg");
    alertMsg.LaunchImage = "LaunchImage";
    //iOS8.2支持字段
    alertMsg.Title = "Title";
    alertMsg.TitleLocKey = "TitleLocKey";
    alertMsg.addTitleLocArg("TitleLocArg");

    apnpayload.AlertMsg = alertMsg;
    apnpayload.Badge = 10;
    apnpayload.ContentAvailable = 1;
    //apnpayload.Category = "";
    apnpayload.Sound = "test1.wav";
    apnpayload.addCustomMsg("payload", "payload");
    //多媒体
    MultiMedia multiMedia = new MultiMedia();
    multiMedia.rid="xxx-1";
    multiMedia.url = "";
    multiMedia.setIsOnlyWifi(false);
    multiMedia.type = MultiMedia.MediaType.pic;

    List<MultiMedia> list = new List<MultiMedia>();

```

```
list.Add(multiMedias);
apnpayload.MultiMedias = list;
template.setAPNInfo(apnpayload);

//设置客户端展示时间
//String begin = "2015-03-06 14:28:10";
//String end = "2015-03-06 14:38:20";
//template.setDuration(begin, end);

return template;
}
```

5.5 对应接口

参数描述

参数经SDK传给您的客户端，客户端根据参数对应实现撤回机制

参数说明

成员和方法名	类型	长度	必填	默认值	说明
setAppId	String		是		设定接收的应用
setAppkey	String		是		用于鉴定身份是否合法
setOldTaskId	String		是		指定需要撤回消息对应的taskId
setForce	Boolean		否	false	客户端没有找到对应的taskId，是否把对应appid下所有的通知都撤回

RevokeTemplate代码示例

使用方式和其他模板类似

```
private static String HOST = "http://sdk.open.api.igexin.com/apiex.htm";
private static String APPID = "";
private static String APPKEY = "";
private static String MASTERSECRET = "";
private static String RevokeTASKID = "";
private static String CLIENTID1 = "";
```

```

private static String CLIENTID2 = "";

private static void pushMessageToApp()
{
    IGtPush push = new IGtPush(HOST, APPKEY, MASTERSECRET);
    AppMessage message = new AppMessage();

    RevokeTemplate template = RevokeTemplateDemo();
    message.IsOffline = false;
    message.OfflineExpireTime = 1000 * 3600 * 12;
    message.Data = template;
    List<String> appIdList = new List<string>();
    appIdList.Add(APPID);
    message.AppIdList = appIdList;

    String pushResult = push.pushMessageToApp(message);
    System.Console.WriteLine("-----
-----");
    System.Console.WriteLine(pushResult);
}

public static RevokeTemplate RevokeTemplateDemo()
{
    RevokeTemplate template = new RevokeTemplate();
    template.AppId = APPID;
    template.AppKey = APPKEY;
    template.OldTaskId = RevokeTASKID;
    template.setForce(true);

    return template;
}

```