

推送模板

1. 点击通知打开应用模板

1.1 描述

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。（激活后，打开应用的首页，如果只要求点击通知唤起应用，不要求到哪个指定页面就可以用此功能。

1.2 应用场景

- 场景1：针对沉默用户，发送推送消息，点击消息栏的通知可直接激活启动应用，提升应用的转化率。

1.3 对应接口(Notification Template)

接口定义

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

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
setDuration	String		否		收到消息的展示时间
transmissionContent	String	2048 中/英 字符	是		透传内容，不支持转义字符
setTransmissionType	int	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
setStyle	AbstractNotifyStyle	-	是	-	通知栏消息布局样式(Style0 系统样式 Style1 个推样式 Style4 背景图样式 Style6 展开式通知样式)，setStyle是新方法，使用了该方法后原来的设置标题、文本等方法就不起效

AbstractNotifyStyle是父类，Style0、Style1、Style4、Style6是其子类

公共成员和方法名	类型	长度	必填	默认值	说明
setIsRing	boolean	—	否	TRUE	收到通知是否响铃：true响铃，false不响铃。默认响铃。
setIsVibrate	boolean	—	否	TRUE	收到通知是否振动：true振动，false不振动。默认振动。
setIsClearable	boolean	—	否	TRUE	通知是否可清除： true可清除，false不可清除。默认可清除。

Style0系统样式和Style1个推样式

成员和方法名	类型	长度	必填	默认值	说明
setTitle	String	40中/英字符	是		通知标题
setText	String	600中/英字符	是		通知内容

setLogo	String	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”
setLogoUrl	String	100中/英字符	否		通知图标URL地址

Style4背景图样式

成员和方法名	类型	长度	必填	默认值	说明
setLogo	String	40中/英字符	否		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”
setBanner_url	String	100中/英字符	是		通过url方式指定动态banner图片作为通知背景图

Style6展开式通知样式,setBigStyle1/setBigStyle2/setBigStyle3 三种方式选一种

成员和方法名	类型	长度	必填	默认值	说明
setTitle	String	40中/英字符	是		通知标题
setText	String	600中/英字符	是		通知内容
setLogo	String	40中/英字符	否		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”
setLogoUrl	String	100中/英字符	否		通知图标URL地址
setBigStyle1(bigImageUrl)	String	100中/英字符	是		通知展示大图样式，参数是大图的URL地址
setBigStyle2(bigText)	String	100中/英字符	是		通知展示文本+长文本样式，参数是长文本
setBigStyle3(bigImageUrl, bannerUrl)	String	100中/英字符	是		通知展示大图+小图样式，参数是大图URL和小图URL

代码实例

```
public static NotificationTemplate notificationTemplateDemo(String appId, String appkey) {
    NotificationTemplate template = new NotificationTemplate();
    // 设置APPID与APPKEY
    template.setAppId(appId);
    template.setAppkey(appkey);
    // 透传消息设置，1为强制启动应用，客户端接收到消息后就会立即启动应用；2为等待应用启动
    template.setTransmissionType(1);
    template.setTransmissionContent("请输入您要透传的内容");
    // 设置定时展示时间
    // template.setDuration("2015-01-16 11:40:00", "2015-01-16 12:24:00");

    Style0 style = new Style0();
    // 设置通知栏标题与内容
    style.setTitle("请输入通知栏标题");
    style.setText("请输入通知栏内容");
    // 配置通知栏图标
    style.setLogo("icon.png");
    // 配置通知栏网络图标
    style.setLogoUrl("");
    // 设置通知是否响铃，震动，或者可清除
    style.setRing(true);
    style.setVibrate(true);
    style.setClearable(true);
    template.setStyle(style);
}
```

```
    return template;
}
```

1.4 效果展示

点击通知栏消息，就可以打开应用并将需要透传的内容发送给应用



新增样式：

style4:



style6bigStyle1:



style6bigStyle2:



style6bigStyle3:



2. 点击通知打开网页模板

2. 描述

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 应用场景

- 场景1：推送广促销活动，用户点击通知栏信息，直接打开到指定的促销活动页面，推送直接到达指定页面，免去了中间过程的用户流失。

2.3 对应接口（LinkTemplate）

参数描述

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

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
setUrl	String		是		点击通知后打开的网页地址

		200 中/ 英 字 符			
setDuration	String		否		收到消息的展示时间
setNotifyStyle	AbstractNotifyStyle	-	否	0	通知栏消息布局样式(Style0 系统样式 Style1 个推样式 Style4 背景图样式 Style6 展开式通知样式), setStyle 是新方法, 使用了该方法后原来的设置标题、文本等方法就不起效, 见详情

LinkTemplate代码实例

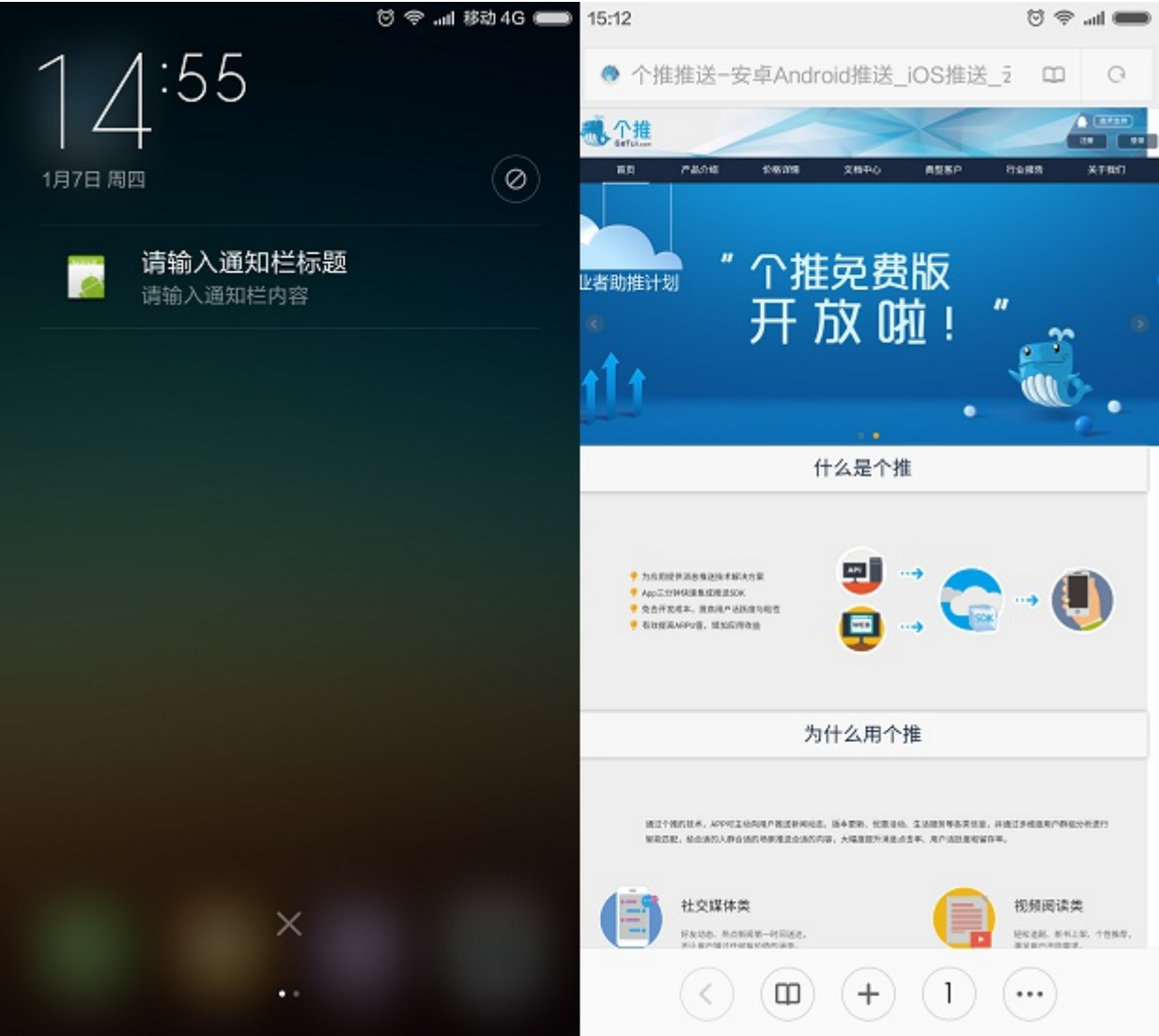
```
public static LinkTemplate linkTemplateDemo(String appId, String appKey) {
    LinkTemplate template = new LinkTemplate();
    // 设置APPID与APPKEY
    template.setAppId(appId);
    template.setAppkey(appKey);

    Style0 style = new Style0();
    // 设置通知栏标题与内容
    style.setTitle("请输入通知栏标题");
    style.setText("请输入通知栏内容");
    // 配置通知栏图标
    style.setLogo("icon.png");
    // 配置通知栏网络图标
    style.setLogoUrl("");
    // 设置通知是否响铃，震动，或者可清除
    style.setRing(true);
    style.setVibrate(true);
    style.setClearable(true);
    template.setStyle(style);

    // 设置打开的网址地址
    template.setUrl("http://www.getui.com");
    // 设置定时展示时间
    // template.setDuration("2015-01-16 11:40:00", "2015-01-16 12:24:00");
    return template;
}
```

2.4 效果展示

点击通知栏的通知消息，就可以自动打开浏览器并且打开代码里面设置的网址



3. 点击通知弹窗下载模板

3.1 描述

消息以弹框的形式展现，点击弹框内容可启动下载任务。

3.2 应用场景

- 场景1：应用有更新，点击推送的更新通知，弹出下载弹窗，点击可启动应用更新下载。

3.3 对应接口(NotyPopLoadTemplate)

参数描述

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。

参数说明

成员和方法名	类型	长度	必填	默认值	说明
setAppId	String	一	是		设定接收的应用

setApkey	String	—	是		用于鉴定身份是否合法
setPopTitle	String	40 中/ 英 字 符	是		弹出框标题
setPopContent	String	600 中/ 英 字 符	是		弹出框内容
setPopImage	String	200 中/ 英 字 符	是		弹出框图标
setPopButton1	String	4 中/ 英 字 符	是		弹出框确定按钮文本显示
setPopButton2	String	4 中/ 英 字 符	是		弹出框取消按钮文本显示
setLoadIcon	String	40 中/ 英 字 符	是		下载图标，如果是本地，则需要加入file://前缀；如果是网络图标，则直接写网络地址
setLoadTitle	String	40 中/ 英 字 符	是		下载标题
setLoadUrl	String	200 中/ 英 字 符	是		下载地址
setAutoInstall	boolean	—	否	false	是否自动安装
setActived	boolean	—	否	false	安装完成后是否自动启动应用程序
setAndroidMark	String	40 中/ 英 字 符	否		设置安卓标识
setSymbianMark	String		否		设置塞班标识

		40 中/ 英 字 符			
setIphoneMark	String	40 中/ 英 字 符	否		设置苹果标识
setDuration	String		否		收到消息的展示时间
setNotifyStyle	AbstractNotifyStyle	-	否	0	通知栏消息布局样式(Style0 系统样式 Style1 个推样式 Style4 背景图样式 Style6 展开式通知样式)，setStyle是新方法，使用了该方法后原来的设置标题、文本等方法就不起效， 见详情

NotyPopLoadTemplate代码实例

```
public static NotyPopLoadTemplate notyPopLoadTemplateDemo(String appId, String appKey) {
    NotyPopLoadTemplate template = new NotyPopLoadTemplate();
    // 设置APPID与APPKEY
    template.setAppId(appId);
    template.setAppkey(appKey);

    Style0 style = new Style0();
    // 设置通知栏标题与内容
    style.setTitle("请输入通知栏标题");
    style.setText("请输入通知栏内容");
    // 配置通知栏图标
    style.setLogo("icon.png");
    // 配置通知栏网络图标
    style.setLogoUrl("");
    // 设置通知是否响铃，震动，或者可清除
    style.setRing(true);
    style.setVibrate(true);
    style.setClearable(true);
    template.setStyle(style);

    // 设置弹框标题与内容
    template.setPopTitle("弹框标题");
    template.setPopContent("弹框内容");
    // 设置弹框显示的图片
    template.setPopImage("");
    template.setPopButton1("下载");
    template.setPopButton2("取消");
    // 设置下载标题
    template.setLoadTitle("下载标题");
    template.setLoadIcon("file://icon.png");
    //设置下载地址
    template.setLoadUrl("http://gdown.baidu.com/data/wisegame/80bab73f82cc29bf/shoujibaidu_16788496.apk");
    // 设置定时展示时间
    // template.setDuration("2015-01-16 11:40:00", "2015-01-16 12:24:00");
    return template;
}
```

3.4 效果展示

点击通知栏消息，弹出下载提示框，点击下载即可自动开始下载



4. 透传消息模版

4.1 描述

透传消息是指消息传递到客户端只有消息内容，展现形式由客户端自行定义。客户端可自定义通知的展现形式，也可自定义通知到达之后的动作，或者不做任何展现。iOS推送也使用该模板。

4.2 应用场景

- 场景1：自定义通知栏样式不想使用默认的通知栏样式，即可使用消息透传的形式，自定义通知栏展现形式，使发送的通知更醒目，更突出。



（红色框内即透传消息，不但样式突出，点击视频可直接播放）

- 场景2：自定义通知到达之后的动作希望用户点击通知后启动应用直接到和通知相关的界面，免去中间跳转的流失。如用户预订更新的某本图书有更新，点击通知直接启动应用到对应图书的页面，免去用户打开应用后的查找，节省中间环节，提高转化。
- 场景3：仅传递信息，不做任何展示推送一串代码给应用，该代码仅此app可以解析。收到透传消息时，界面不作任何展示，用户无感知，应用收到命令后按代码执行操作。

4.3 对应接口

参数描述

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

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
setTransmissionContent	String	2048中/英字符	是		透传内容，不支持转义字符
setTransmissionType	int	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	String		否		收到消息的展示时间
setAPNInfo			否		iOS推送使用该字段

TransmissionTemplate代码实例

```
public static TransmissionTemplate transmissionTemplateDemo() {
    TransmissionTemplate template = new TransmissionTemplate();
    template.setAppId(appId);
    template.setAppkey(appkey);
    // 透传消息设置, 1为强制启动应用, 客户端接收到消息后就会立即启动应用; 2为等待应用启动
    template.setTransmissionType(2);
    template.setTransmissionContent("请输入需要透传的内容");
    // 设置定时展示时间
    // template.setDuration("2015-01-16 11:40:00", "2015-01-16 12:24:00");
    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(ApnPayload 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>

注: setAutoBadge、setAlertMsg、setSound(不设置无声)必须至少设置一项

4.4.3 参数说明:

类名	方法名	说明
APNPayload	setAutoBadge	设置角标，还可以实现显示数字的自动增减，如"+1"、"-1"、"1"等
	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，无声设置为"com.gexin.ios.silence"
	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,Key-Value形式
	addMultiMedia(MultiMedia media)	添加一个多媒体资源，当前最多传入3个资源
	setMultiMedias(List medias)	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	-	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
	setSubtitle	设置子标题,仅支持iOS8.2以上版本
	setSubtitleLocKey	设置当前本地化文件中的子标题字符串的关键字,仅支持iOS8.2以上版本
	addSubtitleLocArgs	设置当前本地化子标题内容中需要置换的变量参数 ,仅支持iOS8.2以上版本

类名	方法名	说明
MultiMedia	setResId(String resId)	设置资源id
	setResUrl(String resUrl)	设置资源url
	setResType(MediaType resType)	设置资源类别 MediaType.pic 图片 MediaType.audio 音频 MediaType.video 视频
	setOnlyWifi(boolean onlyWifi)	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 代码实例：

```
public static TransmissionTemplate getTemplate() {
    TransmissionTemplate template = new TransmissionTemplate();
    template.setAppId(appId);
    template.setAppkey(appkey);
    template.setTransmissionContent("透传内容");
    template.setTransmissionType(2);
    APNPPayload payload = new APNPPayload();
    //在已有数字基础上加1显示, 设置为-1时, 在已有数字上减1显示, 设置为数字时, 显示指定数字
    payload.setAutoBadge("+1");
    payload.setContentAvailable(1);
    payload.setSound("default");
    payload.setCategory("$由客户端定义");

    //简单模式APNPPayload.SimpleMsg
    payload.setAlertMsg(new APNPPayload.SimpleAlertMsg("hello"));

    //字典模式使用APNPPayload.DictionaryAlertMsg
    //payload.setAlertMsg(getDictionaryAlertMsg());

    // 添加多媒体资源
    payload.addMultiMedia(new MultiMedia().setResType(MultiMedia.MediaType.video)
        .setResUrl("http://015mrj259.bkt.clouddn.com/test2.mp4")
        .setOnlyWifi(true));

    template.setAPNInfo(payload);
    return template;
}
private static APNPPayload.DictionaryAlertMsg getDictionaryAlertMsg(){
    APNPPayload.DictionaryAlertMsg alertMsg = new APNPPayload.DictionaryAlertMsg();
    alertMsg.setBody("body");
    alertMsg.setActionLocKey("ActionLockey");
}
```

```
alertMsg.setLockKey("LockKey");
alertMsg.addLocArg("loc-args");
alertMsg.setLaunchImage("launch-image");
// iOS8.2以上版本支持
alertMsg.setTitle("Title");
alertMsg.setTitleLockKey("TitleLockKey");
alertMsg.addTitleLocArg("TitleLocArg");
return alertMsg;
}
```