消息推送方式

本章介绍了Node.js的消息推送方法，如何推送可详细阅读本章。

1. 对单个用户推送消息

1.1 描述

向单个clientid或别名用户推送消息。

注：个推使用clientid来标识每个独立的用户，每一台终端上每一个app拥有一个独立的clientid。

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 pushMessageToSingle(message, target,requestId, callback)

1.2 应用场景

- 场景1：某用户发生了一笔交易，银行及时下发一条推送消息给该用户。
- 场景2：用户定制了某本书的预订更新，当本书有更新时，需要向该用户及时下发一条更新提醒信息。

这些需要向指定某个用户推送消息的场景，即需要使用对单个用户推送消息的接口。

1.3 对应接口

```
pushMessageToSingle(message, target, callback)
```

1.4 参数说明

参数名	类型	必需	默认值	参数描述
Message	object	是	无	消息体
target	object	是	无	推送目标
callback	function	否	无	回调函数

callback(err,res)

参数名	必需	默认值	参数描述
err	是	无	错误
res	是	无	返回信息

1.5 返回值

推送返回值详情查询，请点击Result返回值

1.6 代码实例

```
'use strict';
var GeTui = require('./GT.push');
var Target = require('./getui/Target');
var SingleMessage = require('./getui/message/SingleMessage');
var TransmissionTemplate = require('./getui/template/TransmissionTemplate');
//采用"NodeJs SDK 快速入门", "第二步 获取访问凭证 "中获得的应用配置，用户可以自行替换
var APPID = 'JroCkPGgpF6LzFQqqoWlhA';
var APPKEY = 'Mjv706pTKt5cTcjtqaToz8';
var MASTERSECRET = 'uIBtmad7RK706cy5MKdfp3';
```

```
var CID = 'e560b884d8d9bf5bc5a0f9da545a11f3';
//别名推送方式
//var ALIAS = '';
var HOST = 'http://sdk.open.api.igexin.com/apiex.htm';

var gt = new GeTui(HOST, APPKEY, MASTERSECRET);

gt.connect(function () {
    pushMessageToSingle();
});
function pushMessageToSingle() {
    var template = TransmissionTemplateDemo();
    //单推消息体
    var message = new SingleMessage({
        isOffline: true,                        //是否离线
        offlineExpireTime: 3600 * 12 * 1000,    //离线时间
        data: template                          //设置推送消息类型
    });
    //接收方
    var target = new Target({
        appId: APPID,
        clientId: CID
                //alias:ALIAS
    });

    target.setAppId(APPID).setClientId(CID);
    //target.setAppId(APPID).setAlias(ALIAS);
    gt.pushMessageToSingle(message, target, function(err, res){
        if(err != null && err.exception != null && err.exception instanceof  RequestError){
            var requestId = err.exception.requestId;
            console.log(err.exception.requestId);
            //发送异常重传
            gt.pushMessageToSingle(message,target,requestId,function(err, res){
                console.log(err);
                console.log(res);
            });
        }
    });
}
function TransmissionTemplateDemo() {
    var template =  new TransmissionTemplate({
        appId: APPID,
        appKey: APPKEY,
        transmissionType: 1,
        transmissionContent: '测试离线'
    });
    //iOS推送需要设置的pushInfo字段
    //var payload = new APNPayload();
    //var alertMsg = new SimpleAlertMsg();
    //alertMsg.alertMsg="AlertMsg";
    //payload.alertMsg = alertMsg;
    //payload.badge=5;
    //payload.contentAvailable =1;
    //payload.category="ACTIONABLE";
    //payload.sound="test1.wav";
    //payload.customMsg.payload1="payload";
    //template.setApnInfo(payload);
    return template;
}
```

2. 对指定列表用户推送消息

2.1 描述

上传clientid或别名列表，对列表中所有clientid或别名用户进行消息推送，如果仅对单个用户推送务必使用单推接口，否则会严重影响推送性能，如果对少量甚至几个用户推送同样的消息，建议使用单推实现，性能会更高

2.2 应用场景

- 场景1：对于抽奖活动的应用，需要对已知的某些用户推送中奖消息，就可以通过clientid列表方式推送消息。
- 场景2：向新客用户发放抵用券，提升新客的转化率，就可以事先提取新客列表，将消息指定发送给这部分指定clientid用户。

2.3 对应接口

2.3.1 pushMessageToList-对指定用户列表推送消息

接口说明

```
pushMessageToList(message, targets, callback)
```

参数说明

参数名	类型	必需	默认值	参数描述
Message	object	是	无	消息体
Targets	object	是	无	推送目标列表
callback	function	否	无	回调函数

返回值

推送具体返回值详情查询，请点击Result返回值

注：此接口有频次控制，申请修改请联系邮箱：kegf@getui.com 。

代码实例

```
'use strict';

var GeTui = require('./GT.push');
var Target = require('./getui/Target');
var BaseTemplate = require('./getui/template/BaseTemplate');
var TransmissionTemplate = require('./getui/template/TransmissionTemplate');
var ListMessage = require('./getui/message/ListMessage');

// http的域名
var HOST = 'http://sdk.open.api.igexin.com/apiex.htm';

//https的域名
//var HOST = 'https://api.getui.com/apiex.htm';

//采用"NodeJs SDK 快速入门", "第二步 获取访问凭证 "中获得的应用配置
var APPID = '';
var APPKEY = '';
var MASTERSECRET = '';
var CID1 = '';                          //请填入cid1
var CID2 = '';                          //请填入cid2
//var ALIAS1 = '';                      //请填入ALIAS1
//var ALIAS2 = '';                      //请填入ALIAS2


var gt = new GeTui(HOST, APPKEY, MASTERSECRET);
pushMessageToList();

function pushMessageToList() {
    //process.env.gexin_pushList_needDetails = true;
    //process.env.gexin_pushList_needAsync=true;
    //process.env.=true;
    // var taskGroupName = 'test';
    var taskGroupName = "toList任务组名";
    var template = TransmissionTemplateDemo();

    //定义"ListMessage"类型消息对象，设置消息内容模板、是否支持离线发送、以及离线消息有效期(单位毫秒)
    var message = new ListMessage({
        isOffline: true,
        offlineExpireTime: 3600 * 12 * 1000,
        data: template
    });
```

```
    gt.getContentId(message, taskGroupName, function (err, res) {
        var contentId = res;
        //接收方1
        var target1 = new Target({
            appId: APPID,
            clientId: CID1
//          alias:ALIAS1
        });
                //接收方2
        var target2 = new Target({
            appId: APPID,
            clientId: CID2
//          alias:ALIAS2
        });

        var targetList = [target1,target2];

        console.log("getContentId", res);
        gt.pushMessageToList(contentId, targetList, function (err, res) {
            console.log(res);
        });
    });
}

function TransmissionTemplateDemo() {
    var template =  new TransmissionTemplate({
        appId: APPID,
        appKey: APPKEY,
        transmissionType: 1,
        transmissionContent: '测试离线'
    });

    return template;
}
```

2.3.2 getContentId-获取taskid接口

函数说明

```
getContentId(message,taskGroupName,callback)
```

参数说明

参数名	类型	必需	默认值	参数描述
Message	object	是	无	消息体
taskGroupName	string	否	无	任务别名（详见[任务组名推送](./push_method/#_15)），可为空
callback	function	否	否	回调函数

注：此接口有频次控制，申请修改请联系邮箱：kegf@getui.com 。

代码实例

```
gt.getContentId(message, taskGroupName, function (err, res) {
        var contentId = res;
        //接收方1
        var target1 = new Target({
            appId: APPID,
            clientId: CID
//            alias:'_lalala_'
        });
```

具体代码详见pushMessageToList代码实例

2.3.3 CancelContentId-取消taskid接口

taskid被标识为无效

函数说明

```
CancelContentId (taskId, callback)
```

参数说明

参数名	类型	必需	默认值	参数描述
Taskid	String	是	无	任务id(格式OSL-yyMM_XXXXXX)
callback	function	否	无	回调函数

代码实例

```
'use strict';
var GeTui = require('./GT.push');
var Target = require('./getui/Target');
var HOST = 'http://sdk.open.api.igexin.com/apiex.htm';
//采用"NodeJs SDK 快速入门", "第二步 获取访问凭证 "中获得的应用配置，用户可自行替换
var APPID = 'b03c5cfef65ed30108f0a3fd82c3f6b411';
var APPKEY = '110000';
var MASTERSECRET = 'a02a76119b20d4e31620d7597a3b4f35';
var CID = '289cf3a73ee6e37e456faa4582e96b2e';

var gt = new GeTui(HOST, APPKEY, MASTERSECRET);
gt.connect(function () {
gt.cancelContentId('11', function(err, res) {
 console.log(res);
})
});
```

3. 对指定应用群推消息

3.1 描述

对单个或多个指定应用的所有用户群发推送消息。注：个推使用AppID来标识每个独立的应用。

3.2 应用场景

- 场景1，某app周年庆，群发消息给该app的所有用户，提醒用户参加周年庆活动。

3.3 接口

```
pushMessageToApp(message, taskidAlias, callback)
```

3.4 参数

参数名	类型	必需	默认值	参数描述
Message	Object	是	无	消息体
taskidAlias	String	否	无	任务别名（详见任务组名推送）,可为空

callback	function	否	无	回调函数

3.5 返回值

result具体返回值详情查询，请点击Result返回值

注：此接口有频次控制，申请修改请联系邮箱：kegf@getui.com 。

3.6 代码实例

```
'use strict';

var GeTui = require('./GT.push');
var Target = require('./getui/Target');
var BaseTemplate = require('./getui/template/BaseTemplate');
var TransmissionTemplate = require('./getui/template/TransmissionTemplate');
var AppMessage = require('./getui/message/AppMessage');

// http的域名
var HOST = 'http://sdk.open.api.igexin.com/apiex.htm';

//https的域名
//var HOST = 'https://api.getui.com/apiex.htm';

//采用"NodeJs SDK 快速入门"， "第二步 获取访问凭证 "中获得的应用配置
var APPID = '';
var APPKEY = '';
var MASTERSECRET = '';
var CID = '';                           //请填入cid

var gt = new GeTui(HOST, APPKEY, MASTERSECRET);

pushMessageToApp();

function pushMessageToApp() {
    // var taskGroupName = 'test';
    var taskGroupName = null;
    var template = TransmissionTemplateDemo();

    // 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)
    var message = new AppMessage({
        isOffline: false,
        offlineExpireTime: 3600 * 12 * 1000,
        data: template,
        appIdList: [APPID],
        phoneTypeList: ['ANDRIOD','IOS'],
        provinceList: ['浙江','上海','北京'],
        tagList: ['开心'],
        speed: 10000
    });

    gt.pushMessageToApp(message, taskGroupName, function (err, res) {
        console.log(res);
    });
}
function TransmissionTemplateDemo() {
    var template =  new TransmissionTemplate({
        appId: APPID,
        appKey: APPKEY,
        transmissionType: 1,
        transmissionContent: '测试离线'
    });
    //APN简单推送
    //var payload = new APNPayload();
    ////var alertMsg = new SimpleAlertMsg();
    ////alertMsg.alertMsg="";
    ////payload.alertMsg = alertMsg;
    //payload.badge=5;
    //payload.contentAvailable =1;
    //payload.category="";
    //payload.sound="";
    ////payload.customMsg.payload1="";
    //template.setApnInfo(payload);
```

```
    //APN高级推送
    //var payload = new APNPayload();
    //var alertMsg = new DictionaryAlertMsg();
    //alertMsg.body = "body";
    //alertMsg.actionLocKey = "actionLocKey";
    //alertMsg.locKey = "locKey";
    //alertMsg.locArgs = Array("locArgs");
    //alertMsg.launchImage = "launchImage";
    ////ios8.2以上版本支持
    //alertMsg.title = "title";
    //alertMsg.titleLocKey = "titleLocKey";
    //alertMsg.titleLocArgs = Array("titleLocArgs");
    //
    //payload.alertMsg=alertMsg;
    //payload.badge=5;
//    payload.contentAvailable =1;
//    payload.category="";
//    payload.sound="";
//    payload.customMsg.payload1="payload";
//    template.setApnInfo(payload);
    return template;
}
```

4. WiFi推送

4.1 描述

仅在wifi条件下才展示通知内容，2G/3G/4G等非wifi环境下，不显示推送内容，充分帮用户节省流量。

4.2 应用场景

- 主要用于富媒体、视频、应用下载等推送，仅在wifi环境下展现推送消息，用较精美的富文本内容展示通知，非wifi环境下不显示通知，或者显示另外一条普通文本的通知信息，充分帮用户节省流量。

4.3 对应接口

在message中设置setPushNetWorkType为1，推送时只有通过wifi登录在线的用户才收到消息，手机网络登录用户的消息进离线，等该用户wifi登录后才获取该条离线消息。

Tosingle（对单个用户推送消息）、tolist（对指定用户列表推送消息）、toapp（对指定应用群推消息）三个接口都支持wifi推送。

```
//1：wifi ，0：不限，默认不限
message.setPushNetWorkType(1);
```

4.4 代码实例片段

```
'use strict';
var SingleMessage = require('./getui/message/SingleMessage');
var message = new SingleMessage();
message.setPushNetWorkType(1);
//或 var message = new SingleMessage({pushNetWorkType: 1});
```

具体推送详情代码请参考pushMessageToSingle代码实例

5. 定速推送

5.1 描述

定速推送旨在解决个推群推系统在全量推送时速度过快，导致部分客户服务器连接压力过大的问题。提供接口设置让用户按自身情况控制推送速度。

5.2 应用场景

- 全量推送时希望能控制推送速度不要太快，缓减服务器连接压力，可设置定速推送。如果未设置则按默认推送速度发送。

5.3 接口

在message中设置setSpeed为100，则全量送时个推控制下发速度在100条/秒左右。

只有toapp（对指定应用群推消息）支持定速推送。

message.setSpeed(100);

5.4 代码实例片段

```
'use strict';
var AppMessage= require('./getui/message/AppMessage‘);
var message = new AppMessage ();
message. setSpeed(100);
//或 var message = new AppMessage ({speed: 100});
```

具体推送详情代码请参考pushMessageToApp代码实例

6. 任务组名推送

6.1 描述

一个应用同时下发了n个推送任务，为了更好地跟踪这n个任务的推送效果，可以把他们组成一个任务组，在查询数据时，只需要输入该任务组名即可同时查到n个任务的数据结果。

6.2 应用场景

- 场景：做AB test，分别下发A组、B组推送任务，将A、B任务建成“任务组1”，查数据时，仅需要查找任务组1，即可以一起看到A、B两组测试的结果，可以更直观地对比数据。

6.3 接口

命名同一个应用的不同taskid为同一个任务组名，任务组名由第三方填写。tolist（对指定用户列表推送消息）、toapp（对指定应用群推消息）接口支持该功能。

6.3.1 Tolist接口

gt.getContentId(message, taskid别名, callback);

代码实例片段

```
gt.getContentId(message, taskGroupName, function (err, res) {
        var contentId = res;
        //接收方1
        var target1 = new Target({
            appId: APPID,
            clientId: CID
//            alias:'_lalala_'
        });
```

具体推送代码详见pushMessageToList代码实例

6.3.2 Toapp接口

gt.pushMessageToApp(message,taskid别名,callback); **具体推送详情代码请参考pushMessageToApp代码实例**