

# 其他接口

## 1. 别名接口

### 1.1 别名说明

个推使用clientid来标识每个独立的用户，但clientid不等于开发者应用上的用户名，如果希望将消息发给应用上指定用户名的用户，则需要将用户名指定一个用户别名。

为一个或者一批clientid用户定义一个用户别名，通过这个用户别名对一个或一批用户进行推送。目前一个别名最多允许绑定10个clientid。

别名规则说明：

- 1. 有效的别名组成：字母（区分大小写）、数字、下划线、汉字
- 2. 任务备注名长度限制为 40 字节。（ UTF-8 ）
- 3. 一个别名最多允许绑定10个clientid。

### 1.2 对应接口

#### 1.2.1 bindAlias-单个clientid绑定别名

一个clientid只能绑定一个别名，若已绑定过别名的clientid再次绑定新别名，则认为与前一个别名自动解绑，绑定新别名。

函数说明：

```
bindAlias(appId, Alias, cid, callback)
```

参数说明：

参数名	类型	必需	默认值	参数描述
appId	String	是	无	用户所属应用id
Alias	String	是	无	用户别名
cid	String	是	无	用户id
callback	function	否	无	回调函数

代码实例

```
'use strict';
var Getui = require('./GT.push');
//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 = ''; //请填入绑定目标唯一标识号
//绑定的别名, 用户可自行替换
var alias='demo';

// 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';

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

aliasBind();

function aliasBind() {
```

```
gt.bindAlias(APPID, alias, CID, function(err, res) {
  console.log(res);
});
}
```

1.2.2 bindAliasList-多个clientid绑定别名

允许将多个clientid和一个别名绑定，如用户使用多终端，则可将多终端对应的clientid绑定为一个别名，目前一个别名最多支持绑定10个clientid。

函数说明：

bindAlias(APPID, targetList, callback)

参数说明：

参数名	类型	必需	默认值	参数描述
APPID	String	是	无	用户所属应用id
targetList	array	是	无	绑定目标列表
callback	function	否	无	回调函数

代码实例

```
'use strict';
var GeTui = require('./GT.push');
var Target = require('./getui/Target');

//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 = ''; //请填入绑定目标唯一标识号
//绑定的别名, 用户可自行替换
var alias='demo';

// 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';

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

aliasBatch();

function aliasBatch() {
  // var target = new Target()
  //   .setClientId(CID)
  //   .setAlias('_lalala_');
  var target2 = new Target({
    alias: alias,
    clientId: CID
  });
  var targetList = [target2];
  gt.bindAlias(APPID, targetList, function(err, res) {
    console.log(res);
  });
}
// 注: 只要有一个cid绑定成功, 返回结果就为true
```

1.2.3 queryClientId-根据别名获取clientid信息

函数说明：

```
queryClientId(APPID,alias,callback)
```

参数说明：

参数名	类型	必需	默认值	参数描述
APPID	String	是	无	用户所属应用id
alias	String	是	无	别名
callback	function	否	无	回调函数

代码实例

```
'use strict';
var GeTui = require('./GT.push');

//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 = ''; //请填入绑定目标唯一标识号
//绑定的别名, 用户可自行替换
var alias='demo';

// 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';

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

queryCID();
function queryCID() {
    gt.queryClientId(APPID, alias, function(err, res) {
        console.log(res);
    });
}
```

1.2.4 queryAlias-通过clientid获取别名信息

函数说明：

```
queryAlias(APPID,CID, callback)
```

参数说明：

参数名	类型	必需	默认值	参数描述
APPID	String	是	无	用户所属应用id
CID	String	是	无	用户cid
callback	function	否	无	回调函数

代码实例

```
'use strict';
```

```
var GeTui = require('./GT.push');

//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 = ''; //请填入绑定目标唯一标识号

// 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';

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

queryAlias();

function queryAlias() {
    gt.queryAlias(APPID, CID, function(err, res) {
        console.log(res);
    });
}
```

1.2.5 unBindAlias-单个clientid和别名解绑

函数说明：

```
unBindAlias(APPID, alias, CID, callback)
```

参数说明：

参数名	类型	必需	默认值	参数描述
APPID	String	是	无	用户所属应用id
CID	String	是	无	用户cid
alias	String	是	无	用户别名
callback	function	否	无	回调函数

代码实例

```
'use strict';
var GeTui = require('./GT.push');

//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 = ''; //请填入绑定目标唯一标识号
//绑定的别名, 用户可自行替换
var alias='demo';

// 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';

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

aliasUnBind();

function aliasUnBind() {
```

```
gt.unbindAlias(APPID, alias, CID, function(err, res) {
  console.log(res);
});
}
```

1.2.6 unBindAlias-绑定别名的所有clientid解绑

函数说明：

unBindAlias(APPID, alias, callback)

参数说明

参数名	类型	必需	默认值	参数描述
APPID	String	是	无	用户所属应用id
alias	String	是	无	用户别名
callback	function	否	无	回调函数

代码实例：

```
'use strict';
var Getui = require('./GT.push');

//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 = ''; //请填入绑定目标唯一标识号
//绑定的别名, 用户可自行替换
var alias='demo';

// 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';

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

aliasUnBindAll();

function aliasUnBindAll() {
  gt.unbindAlias(APPID, alias, function(err, res) {
    console.log(res);
  });
}
```

2. 标签接口

2.1 描述

tag即为用户标签，个推提供了服务端和客户端接口，允许app针对每个clientid设置标签。用户的喜好、习惯、性别、年龄段等信息，这些信息均可以做为用户分组的标签。

通过标签（tag）方式实现用户分组，将消息发给指定标签用户，更进一步筛选了用户，实现精细化运营。

2.2 应用场景

场景1：一个用户经常看电影，给该用户打一个“movie”标签，当有最新电影更新了，可给tag为movie的这一群用户推送消息。

场景2：音频播放器应用。对不同音乐类型喜好的人群推送不同类别的新音乐通知。

## 2.3 对应接口

### 2.3.1 setClientTag-对指定用户设置tag属性

函数说明

```
setClientTag(appId, cid, tagList,callback)
```

参数说明

参数名	类型	必需	默认值	参数描述
appId	String	是	无	用户所属应用id
cid	String	是	无	目标用户
tagList	Array	是	无	用户tag列表
callback	function	否	无	回调函数

返回值

具体返回值查询，请参考当前页面中的**Result**返回值内容

注：此接口有频次控制，tag的长度、个数、总长度也有限制，申请修改请联系邮箱：kegf@getui.com。

代码实例

```
'use strict';
var Getui = require('./GT.push');

//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 = ''; //请填入绑定目标唯一标识号

// 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';

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

setClientTag();

function setClientTag() {
  gt.setClientTag(APPID, CID, ['tag1','tag2','tag3'], function (err, res) {
    console.log(res);
  })
}
```

### 2.3.2 getUserTags-获取指定用户的tag属性

函数说明

```
getUserTags(appId, cid, callback)
```

参数说明

参数名	类型	必需	默认值	参数描述
appid	String	是	无	用户所属应用id
cid	String	是	无	目标用户
callback	function	否	无	回调函数

返回值

具体返回值详情查询，请参考当前页面中的**Result**返回值内容

代码实例

```
'use strict';
var GeTui = require('./GT.push');

//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 = ''; //请填入绑定目标唯一标识号

// 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';

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

getUserTags();

function getUserTags() {
  gt.getUserTags(APPID, CID, function (err, res) {
    console.log(res);
  })
}
```

### 3. 获取用户状态接口

#### 3.1 描述

调用此接口可获取用户状态，如在线不在线，cid和appid是否对应，appkey是否正确等。

#### 3.2 接口

```
getClientIdStatus(appId, cid, callback)
```

#### 3.3 参数说明

参数名	类型	必需	默认值	参数描述
appid	String	是	无	用户所属应用id
cid	String	是	无	用户id
callback	function	否	无	回调函数

### 3.4 返回值

具体返回值详情查询，请参考当前页面中的**Result**返回值内容

### 3.5 代码实例

```
'use strict';
var GetTui = require('./GT.push');
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 GetTui(HOST, APPKEY, MASTERSECRET);

getClientIdStatus();

function getClientIdStatus() {
  gt.getClientIdStatus(APPID, CID, function (err, res) {
    console.log(res);
  })
}
```

## 4. 停止任务接口

### 4.1 描述

stop-对正处于推送状态，或者未接收的消息停止下发（list或app任务）

### 4.2 接口

```
stop(taskId, callback)
```

### 4.3 参数

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

### 4.4 返回值

具体返回值详情查询，请参考当前页面中的**Result**返回值内容

### 4.5 代码实例

```
'use strict';
var GetTui = require('./GT.push');

//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 APPKEY = '';
var MASTERSECRET = '';

// 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';

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

stoptask();

//推送任务停止
function stoptask() {
    gt.stop('OSA-1125_FBL14mxYjG9eZzVR18edd8', function (err, res) {
        console.log(res);
    });
}
```

## 5. 获取推送结果接口

### 5.1 描述

调用此接口查询推送数据，可查询消息有效可下发总数，消息回执总数等结果。

### 5.2 接口

```
getPushResult(taskId,callback)
```

### 5.3 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
callback	Function	是	无	回调 函数

### 5.4 返回值

具体返回值详情查询，请参考当前页面中的**Result**返回值内容

### 5.5 代码实例

```
'use strict';
var GeTui = require('./GT.push');

//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 = 'W1ZGdJlcUB8ds32Y2Thn91';
var MASTERSECRET = '5vjwMEaij5VvYf7Vh1GM411';

// 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';

//填写taskId, 用户可自行替换
var TASKID = "GT_0214_Ve9YYRyAFb86LW9JETf1A1";

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

getResult();

function getResult() {
    gt.getPushResult(TASKID, function (err, res) {
        console.log(res);
    })
}
```

## 6. 获取单日推送数据接口

### 6.1 描述

调用此接口可以获取某个应用单日的用户数据（用户数据包括：新增用户数，累计注册用户总数，在线峰值，日联网用户数）（目前只支持查询1天前的数据）

### 6.2 接口

```
queryAppUserDataByDate(appId, date, callback)
```

### 6.3 参数说明

字段	说明
appId	应用ID
date	查询的日期（格式：yyyyMMdd）
callback	回调函数

### 6.4 返回值

字段	上级	含义
result	-	成功：ok
data	-	查询数据对象
appId	data	请求的AppId
date	data	查询的日期（格式：yyyyMMdd）
newRegistCount	data	新注册用户数
registTotalCount	data	新注册用户数
activeCount	data	活跃用户数
onlineCount	data	在线用户数

### 6.5 代码实例

```
'use strict';

var GetTui = require('./GT.push');
//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 HOST = 'http://sdk.open.api.igexin.com/apiex.htm';

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

queryAppUserDataByDate();

function queryAppUserDataByDate() {
  gt.queryAppUserDataByDate(APPID, "20150910", function(err, res) {
    console.log(res);
  });
}
```

```
    });  
  }  
}
```

## 7. 获取单日用户数据接口

### 7.1 描述

调用此接口可以获取某个应用单日的推送数据（推送数据包括：发送总数，在线发送数，接收数，展示数，点击数）（目前只支持查询1天前的数据）

### 7.2 接口

```
queryAppPushDataByDate(appId, date, callback)
```

### 7.3 参数说明

字段	说明
appId	应用ID
date	查询的日期（格式：yyyyMMdd）
callback	回调函数

### 7.4 返回值

字段	上级	含义
result	-	成功：ok
data	-	查询数据对象
appId	data	请求的AppId
date	data	查询的日期（格式：yyyyMMdd）
sendCount	data	发送总数
sendOnlineCount	data	在线发送数
receiveCount	data	接收数
showCount	data	展示数
clickCount	data	点击数

### 7.5 代码实例

```
'use strict';  
  
var GetTui = require('./GT.push');  
//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 HOST = 'http://sdk.open.api.igexin.com/apiex.htm';  
var gt = new GetTui(HOST, APPKEY, MASTERSECRET);
```

```
queryAppPushDataByDate();

function queryAppPushDataByDate() {
  gt.queryAppPushDataByDate(APPID, "20150910", function(err, res) {
    console.log(res);
  });
}
```

8. Result返回值

正确返回	返回码	结果说明
	successed_online	用户在线，消息在线下发
	successed_offline	用户离线，消息存入离线系统
	Ok	发送成功
	details	返回用户状态的详细信息
	contentId	任务ID（当result值为ok时，有此字段）
	msgTotal	有效下发数
	msgProcess	消息回执总数

错误返回	返回码	结果说明
	Error	请求信息填写有误
	action_error	未找到对应的action动作
	appkey_error	Appkey填写错误
	domain_error	填写的域名错误或者无法解析
	sign_error	Appkey与ClientId不匹配，鉴权失败
	AppidNoMatchAppKey	appid和鉴权的appkey不匹配
	PushMsgToListOrAppTimesOverLimit	群推次数超过最大值
	PushTotalNumOverLimit	推送个数总数超过最大值
	AppIdNoUsers	该AppId下的用户总数为0
	SendError	消息推送发送错误
	SynSendError	报文发送错误
	flow_exceeded	接口消息推送流量已超限
	TargetListIsNullOrSizeIs0	推送target列表为空
	PushTotalNumOverLimit	推送消息个数总数超限
	TokenMD5NoUsers	target列表没有有效的clientId
	NullMsgCommon	未找到contentId对应的任务

	TaskIdHasBeenCanceled	任务已经被取消
	AppidError	clientId绑定的appid与推送的appid不符
	successed_ignore	无效用户，消息丢弃
	TokenMD5Error	clientId填写有误
	SendError	消息发送错误
	AppidNoAppSecret	appid未找到对应的appSecret
	OtherError	未知错误，无法判定错误类型