其他接口

1. 别名

1.1 描述

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. 任务别名长度限制为 40 字节。（ UTF-8 ）
3. 一个别名最多允许绑定10个clientid。

1.2 应用场景

场景：对自己应用的指定用户名的用户发放抵用券，即可提取用户列表，将通知发给指定别名的用户。使用别名后，方便开发者根据自己账号进行推送。

1.3 对应接口

1.3.1 IAliasResult别名输出结果接口

方法	说明
getResult()	请求结果（true/false）
getErrorMsg()	请求错误信息
getClientIdList()	请求输出cid列表信息
getAlias()	请求输出别名信息

其中getErrorMsg()的输出为：

取值	说明
flow_exceeded	接口流量超限
error	POST参数错误
action_error	action错误
appkey_error	appkey为空
domain_error	appkey访问的域名错误
sign_error	鉴权失败
OtherError	其他未知错误

鉴权请求通过后，getErrorMsg()的输出格式为 `error_code|error_seq|error_msg` (error_seq不用关注)

error_code	error_msg	说明
10000	general_ error	通用 的、 一般 的错误

10001	appid_ error	appId 错误
10002	cid_error	cid 错误
10003	alias_error	alias错误
10004	param_ error	通用 的参数错误
10005	cid_num _over_limit	cid 个数超限
10006	cid_appid_unbind	cid和appid没有绑定关系

1.3.2 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

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

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

代码实例

```
package Push_Message_Demo;

import com.gexin.rp.sdk.base.IAliasResult;
import com.gexin.rp.sdk.http.IGtPush;

public class AliasFunction {
   //采用"Java 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
   static String appId = "nSJtifqVSI7HkPrKHlxhD6";
   static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
   static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
   static String CID = "3e170b169630706f82baf94c8a2b8923";
   static String host = "http://sdk.open.api.igexin.com/apiex.htm";
   static String Alias = "aliastest";

   public static void main(String[] args) throws Exception {
      IGtPush push = new IGtPush(host, appkey, mastersecret);

      IAliasResult bindSCid = push.bindAlias(appId, Alias, CID);
      System.out.println("绑定结果: " + bindSCid.getResult() + "错误码:" + bindSCid.getErrorMsg());

   }
}
```

1.3.3 bindAlias-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

```
IAliasResult bindAlias (appId, targets)
```

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
target	List	是	无	别名绑定用户列表

代码实例

```
import java.util.ArrayList;
import java.util.List;
import com.gexin.rp.sdk.base.IAliasResult;
import com.gexin.rp.sdk.base.impl.Target;
import com.gexin.rp.sdk.http.IGtPush;

public class AliasFunction1 {
   //采用"Java 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
   static String appId = "nSJtifqVSI7HkPrKHlxhD6";
   static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
   static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
   static String host = "http://sdk.open.api.igexin.com/apiex.htm";

   public static void main(String[] args) {
      List<Target> Lcids = new ArrayList<Target>();
      Target target1 = new Target();
      Target target2 = new Target();
      target1.setClientId("3e170b169630706f82baf94c8a2b8923");
      target1.setAlias("个推1");
      target2.setClientId("23170b169630706f82baf94c8a2b8923");
      target2.setAlias("个推2");
      Lcids.add(target1);
      Lcids.add(target2);
      IGtPush push = new IGtPush(host, appkey, mastersecret);
      IAliasResult bindLCid = push.bindAlias(appId, Lcids);
      System.out.println(bindLCid.getResult());
      System.out.println(bindLCid.getErrorMsg());
   }
}
// 注：只要有一个cid绑定成功，getResult返回结果就为true
```

1.3.4 queryClientId-根据别名获取clientid信息

函数说明

```
IAliasResult queryClientId (appId, Alias)
```

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

代码实例

```
import com.gexin.rp.sdk.base.IAliasResult;
import com.gexin.rp.sdk.http.IGtPush;
public class AliasFunction2 {
```

```
   //采用"Java 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
   static String appId = "nSJtifqVSI7HkPrKHlxhD6";
   static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
   static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
    static String Alias = "个推";
    static String host = "http://sdk.open.api.igexin.com/apiex.htm";
    public static void main(String[] args) throws Exception {
        IGtPush push = new IGtPush(host, appkey, mastersecret);
        IAliasResult queryClient = push.queryClientId(appId, Alias);
        System.out.println("根据别名获取的CID: " + queryClient.getClientIdList());
    }
}
```

1.3.5 queryAlias-通过clientid获取别名信息

函数说明

```
IAliasResult queryAlias (appId, cid)
```

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

代码实例

```
import com.gexin.rp.sdk.base.IAliasResult;
import com.gexin.rp.sdk.http.IGtPush;

public class AliasFunction3 {
   //采用"Java 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
   static String appId = "nSJtifqVSI7HkPrKHlxhD6";
   static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
   static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
    static String CID = "548e1e5a08ce0380d31faba6256e9eb7";
    static String host = "http://sdk.open.api.igexin.com/apiex.htm";
    public static void main(String[] args) throws Exception {
        IGtPush push = new IGtPush(host, appkey, mastersecret);
        IAliasResult queryAlias = push.queryAlias(appId, CID);
        System.out.println("根据CID获取的别名: " + queryAlias.getAlias());
    }
}
```

1.3.6 unBindAlias-单个clientid和别名解绑

函数说明

```
IAliasResult unBindAlias (appId, Alias, cid)
```

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
Alias	String	是	无	用户别名

代码实例

```
import com.gexin.rp.sdk.base.IAliasResult;
import com.gexin.rp.sdk.http.IGtPush;

public class AliasFunction4 {
   //采用"Java 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
   static String appId = "nSJtifqVSI7HkPrKHlxhD6";
   static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
   static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
   static String CID = "3e170b169630706f82baf94c8a2b8923";
   static String Alias = "个推";
   static String host = "http://sdk.open.api.igexin.com/apiex.htm";

   public static void AliasUnBind() throws Exception {
                IGtPush push = new IGtPush(host, appkey, mastersecret);
                IAliasResult AliasUnBind = push.unBindAlias(appId, Alias, CID);
                System.out.println("解除绑定结果:" + AliasUnBind.getResult());
        }
    public static void main(String[] args) {
    try {
      AliasUnBind();
    } catch (Exception e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }
}
```

1.3.7 unBindAliasAll-绑定别名的所有clientid解绑

函数说明

```
IAliasResult unBindAliasAll(appId, Alias)
```

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

代码实例

```
import com.gexin.rp.sdk.base.IAliasResult;
import com.gexin.rp.sdk.http.IGtPush;

public class AliasFunction5 {
   //采用"Java 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
   static String appId = "nSJtifqVSI7HkPrKHlxhD6";
   static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
   static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
   static String Alias = "个推";
   static String host = "http://sdk.open.api.igexin.com/apiex.htm";

   public static void main(String[] args) throws Exception {
      IGtPush push = new IGtPush(host, appkey, mastersecret);

      IAliasResult AliasUnBindAll = push.unBindAliasAll(appId, Alias);
      System.out.println("解除绑定结果:" + AliasUnBindAll.getResult());
   }
}
```

2. 标签

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)
```

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	List	是	无	用户tag列表

返回值

IpushResult具体返回值详情查询，请点击IpushResult接口返回值

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

```
import java.util.ArrayList;
import java.util.List;

import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;

public class SetTagDemo {
    //请输入你的appid
    static String appId = "nSJtifqVSI7HkPrKHlxhD6";
    //请输入你的appkey
    static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
    //请输入你的masterSecret
    static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
    static String host = "http://sdk.open.api.igexin.com/apiex.htm";
    static String cid = "";

    public static void main(String[] args) throws Exception {
        setTag();
        System.out.println(cid);

    }
    public static void setTag() {
        IGtPush push = new IGtPush(host, appkey, mastersecret);
        List<String> tagList = new ArrayList<String>();
                tagList.add(String.valueOf("18-20"));
```

```
                tagList.add("杭州");
                tagList.add("美女");
            IQueryResult ret = push.setClientTag(appId, cid, tagList);
            System.out.println(ret.getResponse().toString());
        }
    }
```

2.3.2 getUserTags-获取指定用户的tag属性

函数说明

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

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

返回值

IpushResult具体返回值详情查询，请点击IpushResult接口返回值

代码实例

```
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GetUserTagsDemo {
    //请输入你的appid
    static String appId = "nSJtifqVSI7HkPrKHlxhD6";
    //请输入你的appkey
    static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
    //请输入你的masterSecret
    static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
    static String host = "http://sdk.open.api.igexin.com/apiex.htm";
    static String cid = "9178238acd873a7ed3681e6d33b43655";

    public static void main(String[] args) throws Exception {
        getUserTags();
    }

    public static void getUserTags() {
        IGtPush push = new IGtPush(host, appkey, mastersecret);
        IPushResult result = push.getUserTags(appId, cid);
        System.out.println(result.getResponse().toString());
    }
}
```

3. 停止推送

3.1 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）

3.2 接口函数说明

```
boolean stop(taskid)
```

3.3 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

3.4 返回值

IpushResult具体返回值详情查询，请点击IpushResult接口返回值

3.5 代码实例

```
import com.gexin.rp.sdk.http.IGtPush;

public class StopTaskDemo {
    //请输入你的appid
    static String appId = "nSJtifqVSI7HkPrKHlxhD6";
    //请输入你的appkey
    static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
    //请输入你的masterSecret
    static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
    static String taskid = "taskid-";
    static String host = "http://sdk.open.api.igexin.com/apiex.htm";

    public static void stopTask() {
        IGtPush push = new IGtPush(host, appkey, mastersecret);
        boolean result = push.stop(taskid);
        System.out.println(result);
    }
    public static void main(String[] args) {
        stopTask();
    }
}
```

4. 获取用户状态

4.1 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。

4.2函数说明：

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

4.3 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	目标用户

4.4 返回值

IpushResult具体返回值详情查询，请点击IpushResult接口返回值

4.5 代码实例

```
package Push_Message_Demo;
```

```
import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GetUserStatus {
  //您应用的mastersecret
  private static final String masterSecret = "";
  //您应用的appkey
  private static final String appkey = "";
  //您应用的appId
  private static final String appId="";

  static String CID = "";

  static String host = "http://sdk.open.api.igexin.com/apiex.htm";
  public static void main(String[] args) throws Exception {
    IGtPush push = new IGtPush(host, appkey, masterSecret);
    getUserStatus();
  }

  public static void getUserStatus() {
    IGtPush push = new IGtPush(host, appkey, masterSecret);
    IQueryResult abc = push.getClientIdStatus(appId, CID);
    System.out.println(abc.getResponse());
  }
}
```

5. Badge设置

5.1 描述

Badge即ios用户应用icon上显示的数字，该接口提供了三种设置badge的方式:
1.在原有badge上+N;
2.在原有badge上-N;
3.直接设置badge(数字，会覆盖原有的badge值)

5.2 接口函数说明

```
// 根据 DeviceToken 设置 Badge
IPushResult setBadgeForDeviceToken(badge, appId, deviceTokenList)

// 根据 clientid 设置 Badge
IPushResult setBadgeForCID(badge, appId, cidList)
```

5.3 参数说明

参数名	类型	必需	默认值	参数描述
badge	String	是	无	用户应用icon上显示的数字
appId	String	是	无	用户所属应用id
cidList	List	是	无	目标用户clientid列表
deviceTokenList	List	是	无	iOS用户DeviceToken列表

5.4 返回值

IQueryResult具体返回值详情查询，请点击IpushResult接口返回值

5.5 代码实例-根据 deviceToken 设置 Badge

```
import java.util.ArrayList;
import java.util.List;
```

```
import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;

public class SetBadgeForDeviceTokenDemo {
        // 请填入iOS用户唯一标识
        public static String deviceToken = "5b6f8c6cdc5ab8352a0f7357888308c17fe115d9d162c2a638ad336bc9f50f85";
        // 采用"Java 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
        public static String masterSecret = "5vjiwMEaij5VvYf7VhlGM4";
        public static String appId = "nSJtifqVSI7HkPrKHlxhD6";
        public static String appkey = "WlZGdJlcUB8ds32Y2Thn91";

        public static void setBadgeForDeviceToken() {
                List<String> deviceTokenList = new ArrayList<String>();
                // 用户应用icon上显示的数字
                String badge = "";
                deviceTokenList.add(deviceToken);
                IGtPush push = new IGtPush(appkey, masterSecret);
                // "+1"即在原有badge上加1；具体详情使用请参考该接口描述
                badge = "+1";
                // "-1"即在原有badge上减1；具体详情使用请参考该接口描述
                // badge = "-1";
                // 直接设置badge数值，会覆盖原有数值；具体详情使用请参考该接口描述oooo
                // badge = "1";
                IQueryResult res = push.setBadgeForDeviceToken(badge, appId, deviceTokenList);
                System.out.println(res.getResponse());
        }

        public static void main(String[] args) {
                setBadgeForDeviceToken();
        }
}
```

5.6 代码实例-根据 clientid 设置 Badge

```
import java.util.ArrayList;
import java.util.List;

import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;

public class setBadageForCidDemo {
        // 请填入你的clientid
        public static String cid = "548e1e5a08ce0380d31faba6256e9eb7";
        // 采用"Java 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
        public static String masterSecret = "5vjiwMEaij5VvYf7VhlGM4";
        public static String appId = "nSJtifqVSI7HkPrKHlxhD6";
        public static String appkey = "WlZGdJlcUB8ds32Y2Thn91";

        public static void setBadgeForCid() {
                List<String> cidList = new ArrayList<String>();
                // 用户应用icon上显示的数字
                String badge = "";
                cidList.add(cid);
                IGtPush push = new IGtPush(appkey, masterSecret);
                // "+1"即在原有badge上加1；具体详情使用请参考该接口描述
                badge = "+1";
                // "-1"即在原有badge上减1；具体详情使用请参考该接口描述
                // badge = "-1";
                // 直接设置badge数值，会覆盖原有数值；具体详情使用请参考该接口描述
                // badge = "1";
                IQueryResult res = push.setBadgeForCID(badge, appId, cidList);
                System.out.println(res.getResponse());
        }

        public static void main(String[] args) {
                setBadgeForCid();
        }
}
```

6. 黑名单用户管理

6.1 描述

将指定cid列表中的用户加入/移除黑名单

6.2 接口函数说明

```
// 添加黑名单用户
IPushResult addCidListToBlk(String appId, List<String> cidList)

// 移除黑名单用户
IPushResult restoreCidListFromBlk(String appId, List<String> cidList)
```

6.3 参数说明

参数名	类型	必需	默认值	参数描述
appId	String	是	无	appId
cidList	List	是	无	该appId下的用户cid列表

6.4 返回值

IpushResult具体返回值详情查询，请点击IpushResult接口返回值

6.5 代码实例-添加用户到黑名单

```
import java.util.ArrayList;
import java.util.List;
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;
public class AddBlackCidListDemo {
    //请输入你的cid
    public static String cid = "548e1e5a08ce0380d31faba6256e9eb7";
    //请输入你的appid
    static String appId = "nSJtifqVSI7HkPrKHlxhD6";
    //请输入你的appkey
     static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
    //请输入你的masterSecret
    static String mastersecret = "5vjiwMEaij5VvYf7VhlGM4";
    public static void testBlackCidList() {
        List<String> cidList = new ArrayList<String>();
        cidList.add(cid);
        IGtPush push = new IGtPush(appkey, mastersecret);
        IPushResult pushResult1 = push.addCidListToBlk(appId, cidList);
        System.out.println("黑名单增加结果: " + pushResult1.getResponse()) ;
    }
    public static void main(String[] args) {
        testBlackCidList();
    }
}
```

6.6 代码实例-将用户从黑名单移除

```
import java.util.ArrayList;
import java.util.List;

import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;
public class DelBlackCidListDemo{
    public static String cid = "";
        //采用"Java 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
    public static String masterSecret = "5vjiwMEaij5VvYf7VhlGM4";
    public static String appId = "nSJtifqVSI7HkPrKHlxhD6";
    public static String appkey = "WlZGdJlcUB8ds32Y2Thn91";
```

```
    public static void testRestoreCidList() {
        List<String> cidList = new ArrayList<String>();
        cidList.add(cid);
        IGtPush push = new IGtPush(appkey, masterSecret);
        IPushResult pushResult = push.restoreCidListFromBlk(appId, cidList);
        System.out.println("黑名单删除结果：" + pushResult.getResponse());
    }

    public static void main(String[] args) {
        testRestoreCidList();
    }
}
```

7. 获取推送结果

7.1 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，用户点击数等结果。

7.2 函数说明

```
IPushResult getPushResult(String taskId)
```

7.3 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(格式OSL-yyMM_XXXXXX)

7.4 返回值

```
{
    "result": "",
    "taskId": "",
    "GT": {//个推下发报表
        "sent": "",//成功下发数
        "displayed": "",//展示数
        "clicked": "",//点击
        "feedback": "",//到达
        "result": ""//成功(ok)或错误信息
    },
    "APN": {//ios apn下发
        "sent": "",//下发
                "displayed": "",//apns展示
        "clicked": "",//点击
        "result": ""//成功(ok)或错误信息
    }
}
```

7.5 代码实例

```
package Push_Message_Demo;

import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GetPushMessageResultDemo {
  //您应用的mastersecret
  private static final String MASTERSECRET = "";
  //您应用的appkey
  private static final String APPKEY = "";
  //您要查询的taskid
  private static final String TASKID = "";
```

```
  static String host ="http://sdk.open.api.igexin.com/apiex.htm";
  public static void main(String[] args) {

        IGtPush push = new IGtPush(host,APPKEY, MASTERSECRET);
    Map<String, Object> res = (Map<String, Object>) push.getPushResult(TASKID).getResponse();

    String gt = res.get("GT").toString();
    String apns = res.get("APN").toString();
    System.out.println("个推报表:"+gt+"|apns报表:"+apns);

  }

}
```

8. 获取单日用户数据

8.1 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）

8.2 函数说明

```
IQueryResult queryAppUserDataByDate (String appId,String date)
```

8.3 参数说明

字段	说明
appId	应用ID
date	查询的日期（格式：yyyyMMdd）

8.4 返回值

获取方式 `IQueryResult.getResponse()`

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	累计注册用户数
activeCount	data	活跃用户数
onlineCount	data	在线用户数使用方法

8.5 代码实例

```
package Push_Message_Demo;

import java.util.Map;
```

```
import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GetPushMessageResultDemo {
  //您应用的mastersecret
  private static final String MASTERSECRET = "";
  //您应用的appkey
  private static final String APPKEY = "";
  //要查询的AppId
  private static final String APPID = "";

  static String host ="http://sdk.open.api.igexin.com/apiex.htm";

  public static void main(String[] args) {

    IGtPush push = new IGtPush(host,APPKEY, MASTERSECRET);

    IQueryResult result = push.queryAppUserDataByDate(APPID, "20150525");
    Map<String, Object> data = (Map<String, Object>)result.getResponse().get("data");
    System.out.println(result.getResponse().toString());

      System.out.println("新用户注册总数:"+data.get("newRegistCount"));
      System.out.println("用户注册总数:"+data.get("registTotalCount"));
      System.out.println("活跃用户数:"+data.get("activeCount"));
      System.out.println("在线用户数:"+data.get("onlineCount"));

  }

}
```

9. 获取单日推送数据

9.1 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）

9.2 函数说明

```
IQueryResult queryAppPushDataByDate (String appId,String date)
```

9.3 参数说明

字段	说明
appId	应用ID
date	查询的日期（格式：yyyyMMdd）

9.4 返回值

```
{
    "result": "",
    "data": {
                "appId":"",
                "date":""
        },
        "GT": {//个推下发报表
        "sent": "",//成功下发数
        "displayed": "",//展示数
        "clicked": "",//点击
        "feedback": "",//到达
        "result": ""//成功(ok)或错误信息
    },
    "APN": {//ios apns下发
        "sent": "",//下发
                "displayed": "",//apns展示
        "clicked": "",//点击
        "result": ""//成功(ok)或错误信息
```

```
    }
}
```

9.5 代码实例

```
package Push_Message_Demo;
import java.util.Map;

import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GetPushMessageResultDemo {
  //您应用的mastersecret
  private static final String MASTERSECRET = "";
  //您应用的appkey
  private static final String APPKEY = "";
  //您应用的appid
  private static final String APPID = "";

  static String host ="http://sdk.open.api.igexin.com/apiex.htm";

  public static void main(String[] args) {

    IGtPush push = new IGtPush(host,APPKEY, MASTERSECRET);

    IQueryResult result = push.queryAppPushDataByDate(APPID, "20170125");
        Map<String, Object> res = (Map<String, Object>)result.getResponse();
        String data = res.get("data").toString;
    String gt = res.get("GT").toString();
    String apns = res.get("APN").toString();
        System.out.println(data + "|个推报表:"+gt+"|apns报表:"+apns);
  }

}
```

10. 获取任务组名推送结果

10.1 描述

根据任务组名查询推送结果，返回结果包括百日内联网用户数（活跃用户数）、实际下发数、到达数、展示数、点击数。

10.2 函数说明

```
IQueryResult getPushResultByGroupName(String appId, String groupName)
```

10.3 参数说明

参数名	类型	必需	默认值	参数描述
appId	String	是	无	appId
groupName	String	是	无	推送任务组名

10.4 返回值

```
{
    "result": "",
    "groupName": "",
    "GT": {//个推下发报表
        "sent": "",//成功下发数
        "displayed": "",//展示数
        "clicked": "",//点击
        "feedback": "",//到达
        "result": ""//成功(ok)或错误信息
    },
```

```
    "APN": {//ios apns下发
        "sent": "",//下发
                "displayed": "",//apns展示
        "clicked": "",//点击
        "result": ""//成功(ok)或错误信息
    }
}
```

10.5 代码实例

```
package Push_Message_Demo;
import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;
public class GetPushResultByGroupNameTest {
    //您应用的mastersecret
    public static  String masterSecret = "";
    //您应用的appkey
    public static  String appkey = "";
    //您应用的appid
    public static  String appId = "";
    public static String groupName = "";

    public static void testPushResultByGroupName() {
        final String masterSecret = "";
        IGtPush push = new IGtPush(appkey, masterSecret);
        IQueryResult queryResult = push.getPushResultByGroupName(appId, groupName);

                Map<String, Object> res = (Map<String, Object>)queryResult.getResponse();
        String gt = res.get("GT").toString();
        String apns = res.get("APN").toString();
                System.out.println(res.get("groupName") + "|个推报表:"+gt+"|apns报表:"+apns);
    }

    public static void main(String[] args) {
        testPushResultByGroupName();
    }
}
```

11. 获取24小时在线用户数

11.1 描述

通过接口查询当前时间一天内的在线数（十分钟一个点，一小时六个点）

11.2 接口

11.3 函数说明

```
IQueryResult getLast24HoursOnlineUserStatistics(String appId)
```

11.4 参数说明

参数名	类型	必需	默认值	参数描述
appId	String	是	无	appId

11.5返回值

IQueryResult具体返回值详情查询，请点击IpushResult接口返回值

字段	取值	说明
appId	String	用户appId
onlineStatics	Map<String, long>	24小时用户在线数统计

11.6 代码实例

```
package Push_Message_Demo;
import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;
public class GetOnlineUserTest {
    //您应用的mastersecret
    public static  String masterSecret = "";
    //您应用的appkey
    public static  String appkey = "";
    //您应用的appid
    public static  String appId = "";
    public static String groupName = "";
    public static void testGetOnlineUser() {
        IGtPush push = new IGtPush(appkey, masterSecret);
        IQueryResult queryResult = push.getLast24HoursOnlineUserStatistics(appId);
        System.out.println(queryResult.getResponse().get("appId"));
        System.out.println(queryResult.getResponse().get("onlineStatics"));
    }

    public static void main(String[] args) {
        testGetOnlineUser();
    }
}
```

12. 查询符合条件的用户数

12.1 描述

通过接口查询符合当前查询条件的用户数

12.2 函数说明

```
IQueryResult queryUserCount(String appId, AppConditions conditions)
```

12.3 参数说明

参数名	类型	必需	默认值	参数描述
appId	String	是	无	appId
conditions	AppConditions	是	无	conditions

12.4 返回值

IQueryResult具体返回值详情查询，请点击IpushResult接口返回值

12.5 代码实例

```
package Push_Message_Demo;

import java.util.ArrayList;
import java.util.List;

import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.base.uitls.AppConditions;
import com.gexin.rp.sdk.http.IGtPush;

public class QueryUserCount {

    public static void main(String[] args) {
        String host = "http://sdk.open.api.igexin.com/apiex.htm";
      //使用https的域名
    // String host = "https://api.getui.com/apiex.htm";
```

```
        //您应用的mastersecret
        String masterSecret = "KFDNBNKAVj9bgykwvqgeA5";
        //您应用的appkey
        String appkey = "rAnoicfrNX7915IxPocAL2";
        //您应用的appid
        String appId = "TxzlIyCcfS9KuENjjP4ux1";

        IGtPush gtpush = new IGtPush(host,appkey, masterSecret);
        AppConditions conditions = new AppConditions();
        //新增机型
        List<String> phoneTypes = new ArrayList<String>();
        phoneTypes.add("ANDROID");
        conditions.addCondition(AppConditions.PHONE_TYPE, phoneTypes);

        //新增地区
        List<String> regions = new ArrayList<String>();
        regions.add("浙江省");
        conditions.addCondition(AppConditions.REGION, regions);

        //新增tag
        List<String> tags = new ArrayList<String>();
        tags.add("haha");
        conditions.addCondition(AppConditions.TAG, tags);

        //查询可推送的用户画像
        IQueryResult personaTagResult = gtpush.getPersonaTags(appId);
        System.out.println(personaTagResult.getResponse());

        //新增用户画像
        //工作
        List<String> jobs = new ArrayList<String>();
        jobs.add("0102");
        jobs.add("0110");
        conditions.addCondition("job", jobs);

        //年龄
        List<String> age = new ArrayList<String>();
        age.add("0000");
        conditions.addCondition("age", age);

        //查询
        IQueryResult  result = gtpush.queryUserCount(appId, conditions);
        System.out.println(result.getResponse());
    }

}
```

13. 大数据综合分析用户得到的标签:即用户画像

13.1 描述

通过接口查询当前bi统计的用户画像标签,该接口需要申请后才可正常使用，且主要是让APP查看自己能使用那些标签进行推送

申请用户画像标签联系邮箱：kegf@getui.com

13.2 函数说明

```
IQueryResult getPersonaTags(String appId)
```

13.3 参数说明

参数名	类型	必需	默认值	参数描述
appId	String	是	无	appId

13.4 返回值

IQueryResult具体返回值

字段	取值	说明
result	String	获取tags的结果,success：成功
tags	List	应用的用户画像标签

13.5 代码实例

```
package Push_Message_Demo;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import com.gexin.rp.sdk.base.IQueryResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GetPersonTags {
  List<String> cidList = new ArrayList<String>();

    //采用"Java 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
    //您应用的mastersecret
    public static  String masterSecret = "";
    //您应用的appkey
    public static  String appkey = "";
    //您应用的appid
    public static  String appId = "";

  static String host = "http://sdk.open.api.igexin.com/apiex.htm";
  //使用https的域名
  //static String host = "https://api.getui.com/apiex.htm";

  public static void main(String[] args) throws Exception {
    IGtPush push = new IGtPush(host, appkey, masterSecret);
    getPersonaTags();
  }

  public static void getPersonaTags() {
    IGtPush push = new IGtPush(host, appkey, masterSecret);
    IQueryResult abc = push.getPersonaTags(appId);
    System.out.println(abc.getResponse());
  }
}
```

14. 直播间接口使用手册

14.1 描述

直播间接口是针对直播间类的应用使用个推sdk推送直播间内消息给直播间成员所做的专用接口。原本的tolist接口推送对大量频繁的聊天消息存在效率低、大型直播间需要调多次传cid等问题，而新增的直播间接口对这些都进行了优化。

实际使用中，开发者需要使用新版的服务端sdk，通过createGroup、updateGroup和removeGroup对房间进行创建、修改房间内的cids以及销毁房间，从而维护个推后台的房间和用户的关系。然后直接调用pushMessageToGroup接口进行针对房间内用户的推送即可。

14.2 接口函数说明

```
//创建房间
public IPushResult createGroup(String appId, String groupName)
//更新房间的cid
public IPushResult updateGroup(String appId, String groupId, List<String> addCids, List<String> delCids)
//销毁房间
public IPushResult removeGroup(String appId, String groupId)
//推送指定房间
public IPushResult pushMessageToGroup(String appId, GroupMessage groupMessage)
//查询指定房间的cid状态
public IPushResult queryCidStatusByGroupid(String appId, String groupId, List<String> cids)
//查询指定房间的cid列表
public IPushResult queryCidsByGroupid(String appId, String groupId)
```

14.3 参数说明

参数名	类型	必需	默认值	参数描述
groupName	String	是	无	用户创建的房间名
appId	String	是	无	用户所属应用id
groupId	String	是	无	根据房间名对应生成的id
addCids	List	否	无	增加（进入房间）的cid
delCids	List	否	无	删除（离开房间）的cid
groupMessage	GroupMessage	是	无	消息体

14.4 套餐说明

套餐限制	超限返回值
应用的房间数	GroupNumOverLimit
每个房间的用户量	GroupCidNumOverLimit
每秒每个房间的推送次数	PushTooFrequency
每个房间的消息缓存大小	无返回值，超出会顶掉老的消息
每个房间当天推送消息数	PushTotalOverLimit
每个应用每分钟查询cid状态的次数	QueryTooFrequency
每个应用每分钟查询cid列表的次数	QueryTooFrequency

14.5 返回值

IpushResult具体返回值详情查询，请点击IpushResult接口返回值

14.6 代码实例

- 创建房间

```
import java.io.IOException;
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GroupPush {
//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 = "";
    private static String url = "http://sdk.open.api.igexin.com/apiex.htm";

    public static void main(String[] args) throws IOException {
     String groupName = "test123";//房间名需要唯一，对应唯一的groupid
      IGtPush push = new IGtPush(url, appKey, masterSecret);

      //创建一个房间
      IPushResult createResult = push.createGroup(appId, groupName);
      if(!"Success".equalsIgnoreCase((String) createResult.getResponse().get("result"))){
        return;
      }
      //房间创建成功，获取groupid
      String groupid = (String) createResult.getResponse().get("groupId");
      System.out.println(groupid);
```

```
    }
}
```

- 更新房间的cid

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GroupPush {
//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 = "";
    private static String url = "http://sdk.open.api.igexin.com/apiex.htm";

    public static void main(String[] args) throws IOException {
      IGtPush push = new IGtPush(url, appKey, masterSecret);
      //房间创建成功，获取groupid
      String groupid = "";

      //更新房间中的cid用户列表
      //需要加入房间的cid列表，可为null和空列表
      List<String> addCids = new ArrayList<String>();
      addCids.add("cid1");
      addCids.add("cid2");
      //需要删除房间的cid列表，可为null和空列表
      List<String> delCids = new ArrayList<String>();
      delCids.add("cid3");
      delCids.add("cid4");
      IPushResult updateResult = push.updateGroup(appId, groupid, addCids, delCids);
      if(!"Success".equalsIgnoreCase((String) updateResult.getResponse().get("result"))){
          return;
      }
    }
}
```

- 对房间推送消息

```
import java.io.IOException;
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.base.impl.GroupMessage;
import com.gexin.rp.sdk.http.IGtPush;
import com.gexin.rp.sdk.template.TransmissionTemplate;

public class GroupPush {
//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 = "";
    private static String url = "http://sdk.open.api.igexin.com/apiex.htm";

    public static void main(String[] args) throws IOException {
      IGtPush push = new IGtPush(url, appKey, masterSecret);
      //给该房间（groupid）推送消息
      String groupid = "";
      //requestId: 保证每条消息唯一性，防止相同消息重复下发
      String requestId = "";
      push(push, groupid , requestId);
    }


    //该方法可以自行定义
    public static void push(IGtPush push, String groupid, String requestId){
      // 定义透传通知模板
      TransmissionTemplate template = new TransmissionTemplate ();
      // 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("{\"postContext\":\"测试推送\",\"nickName\":\"测试推送\"}");

      GroupMessage gMessage = new GroupMessage();
      gMessage.setData(template);
      gMessage.setGroupId(groupid);
      gMessage.setRequestId(requestId);

      //将消息推送给房间内的全部cid
      IPushResult ret = push.pushMessageToGroup(appId, gMessage);
      System.out.println(ret.getResponse().toString());
    }
  }
```

- 销毁房间

```
import java.io.IOException;
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GroupPush {
    private static String appId = "";
    private static String appKey = "";
    private static String masterSecret = "";
    private static String url = "http://sdk.open.api.igexin.com/apiex.htm";

    public static void main(String[] args) throws IOException {
      IGtPush push = new IGtPush(url, appKey, masterSecret);
      String groupid = "";
      //当直播结束后进行销毁房间操作
      //如不进行销毁房间操作，该房间将默认在最后一次更新cid后继续存在一天，一天后自动删除
      IPushResult removeResult = push.removeGroup(appId, groupid );
      System.out.println(removeResult.getResponse());
    }
}
```

- 查询cid状态

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GroupPush {
    private static String appId = "";
    private static String appKey = "";
    private static String masterSecret = "";
    private static String url = "http://sdk.open.api.igexin.com/apiex.htm";

    public static void main(String[] args) throws IOException {
        IGtPush push = new IGtPush(url, appKey, masterSecret);

        String groupid = "";
        //根据房间id和cid列表查询cid状态
        List<String> cidList = new ArrayList<String>();
        cidList.add("cid1");
        IPushResult queryResult = push.queryCidStatusByGroupid(appId, groupid, cidList);
        System.out.println(queryResult.getResponse());
    }
}
```

- 查询cid列表

```
import java.io.IOException;
import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GroupPush {
    private static String appId = "";
    private static String appKey = "";
    private static String masterSecret = "";
    private static String url = "http://sdk.open.api.igexin.com/apiex.htm";
```

```
    public static void main(String[] args) throws IOException {
        IGtPush push = new IGtPush(url, appKey, masterSecret);

        String groupid = "";
        //根据房间id查询cid列表
        IPushResult queryResult = push.queryCidsByGroupid(appId, groupid);
        System.out.println(queryResult.getResponse());
    }
}
```

15. 批量获取推送结果

15.1 描述

调用此接口批量查询推送数据，可查询消息有效可下发总数，消息回执总数，用户点击数结果。

15.2 函数说明

```
IPushResult getPushResultByTaskidList(List<String> taskIdList)
```

15.3 参数说明

参数名	类型	必需	默认值	参数描述
taskIdList	List	是	无	任务Id列表

15.4 返回值

```
{
  "result": "",
  "resultList":[
      {
                "taskId":"",
                "GT": {//个推下发报表
                        "sent": "",//成功下发数
                        "displayed": "",//展示数
                        "clicked": "",//点击
                        "feedback": "",//到达
                        "result": ""//成功(ok)或错误信息
                },
                "APN": {//ios apns下发
                        "sent": "",//下发
                        "displayed": "",//apns展示
                        "clicked": "",//点击
                        "result": ""//成功(ok)或错误信息
                }
      }
  ]
}
```

15.5 代码实例

```
package Push_Message_Demo;

import com.gexin.rp.sdk.base.IPushResult;
import com.gexin.rp.sdk.http.IGtPush;

public class GetPushMessageResultDemo {
  //您应用的mastersecret
  private static final String MASTERSECRET = "";
  //您应用的appkey
  private static final String APPKEY = "";

  static String host ="http://sdk.open.api.igexin.com/apiex.htm";
```

```
  public static void main(String[] args) {
    IGtPush push = new IGtPush(host,APPKEY, MASTERSECRET);
    //您要查询的taskIdList
    List<String> taskIdList = new ArrayList<String>();
    taskIdList.add("");

    System.out.println(push.getPushResultByTaskidList(taskIdList).getResponse());
    IPushResult result=push.getPushResultByTaskidList(taskIdList);
    List<Map<String, Object>> resList = (List<Map<String, Object>>) result.getResponse().get("resultList");
    for(Map<String, Object> res : resList){
          String taskId = res.get("taskId").toString();
      String gt = res.get("GT").toString();
      String apns = res.get("APN").toString();
          System.out.println(任务号:+taskId+"|个推报表:"+gt+"|apns报表:"+apns);
    }
  }

}
```