

阿里云 微消息队列MQTT

管控 API 参考

文档版本：20190913

法律声明

阿里云提醒您 在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
<code>[]</code> 或者 <code>[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 管控 API 版本说明.....	1
2 管控 API 接入指南.....	2
3 错误码列表.....	4
4 设备管理接口.....	5
4.1 查询设备连接信息.....	5
4.2 查询 Group 下在线设备的数量.....	7
4.3 根据 Topic 查询在线设备的数量.....	9
4.4 查询历史在线设备统计.....	11
5 Group 管理接口.....	14
5.1 创建 Group ID.....	14
5.2 查询 Group ID 列表.....	15
5.3 删除 Group ID.....	17
6 资源报表接口.....	19
6.1 消息收发统计.....	19

1 管控 API 版本说明

本文介绍微消息队列 MQTT 的 API 版本发布信息。

通过本文，您可了解管控 API 的每个版本的功能。详情请见下表。

表 1-1: 管控 API 版本说明

API 版本	发布日期	说明
3.1.0	2019 年 2 月 15 日	<ul style="list-style-type: none">· 不兼容 2.x.x 版本，和线上控制台功能一致· 提供实例化所有功能· consumerId 和 producerId 概念变更为 groupId· 去除 OnsRegionId 参数，简化调用方式
2.1.4	2019 年 1 月 25 日	<ul style="list-style-type: none">· 兼容 2.x.x 版本· OnsTopicList 返回实例化信息字段
2.0.1	2018 年 4 月 17 日	新增创建 MQTT GroupId 的接口
2.0.0	2017 年 9 月 19 日	<ul style="list-style-type: none">· 新增消息轨迹查询接口· 写操作接口新增访问权限控制
1.3.3	2017 年 8 月 10 日	<ul style="list-style-type: none">· 新增 Topic 禁读禁写接口· 新增 ConsumerId 禁读接口

2 管控 API 接入指南

本文介绍微消息队列 for IoT 的 API 的接入方法和注意事项，包含 SDK 的获取以及初始化参数的设置。

1. SDK 获取

直接填写以下 *pom.xml* 的配置，依赖 API 的 SDK 即可。

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <optional>true</optional>
    <version>4.3.3</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-ons</artifactId>
    <version>3.1.0</version>
  </dependency>
</dependencies>
```

2. 公共参数设置

使用 API 前需启动 API 的客户端，而启动客户端需设置接入点、accessKey、secretKey 等参数信息，具体示例如下：

```
/**
 *API 的接入点，设置为目标 Region
 */
String regionId = "XXXXXX";
/**
 *鉴权使用的 AccessKeyId，由阿里云官网管理控制台获取
 */
String accessKey = "XXXXXXXXXXXXXXXXXXXX";
/**
 *鉴权使用的 AccessKeySecret，由阿里云官网管理控制台获取
 */
String secretKey = "XXXXXXXXXXXXXXXXXXXX";
IClientProfile profile= DefaultProfile.getProfile(regionId,
accessKey, secretKey);
IAcsClient iAcsClient= new DefaultAcsClient(profile);
```

参数说明

启动客户端需设置的参数及其描述如下表所示。

表 2-1: 参数说明

参数名称	描述
regionId	API 的网关所在地域 (Region)
accessKey	您在阿里云服务器管理控制台上获取的 AccessKeyId
secretKey	您在阿里云服务器管理控制台上获得的 AccessKeySecret

地域列表

微消息队列 for IoT 的 API 所支持的地域如下表所示。

表 2-2: 地域列表

地域名称	RegionId	备注
公共云华北2	cn-beijing	使用公共云华北2 地域的用户使用此接入点
公网	mq-internet-access	使用公网地域的用户使用此接入点
公共云华东1	cn-hangzhou	使用公共云华东1 地域的用户使用此接入点
公共云华东2	cn-shanghai	使用公共云华东2 地域的用户使用此接入点
公共云华南1	cn-shenzhen	使用公共云华南1 地域的用户使用此接入点
公共云新加坡	ap-southeast-1	使用新加坡地域的用户使用此接入点
金融云华东1	cn-hangzhou-finance	使用金融云华东1 地域的用户使用此接入点
金融云华南1	cn-shenzhen-finance-1	使用金融云华南1 地域的用户使用此接入点
金融云华东2	cn-shanghai-finance-1	使用金融云华东2 地域的用户使用此接入点

3 错误码列表

如果成功调用微消息队列 MQTT 的 API，则会返回给客户端 ResponseCode=200；如果调用失败，会返回对应的异常错误码以及描述信息。您可以根据以下异常信息对照表查找对应的解决方法。

表 3-1: 错误码列表

错误码	现象&原因	解决方法
ONS_SYSTEM_ERROR	消息队列 MQMQ 后端异常	通过阿里云工单联系技术支持人员
ONS_SERVICE_UNSUPPORTED	当前调用在对应的 Region 区域不支持	请联系技术支持人员确认该接口的开通情况
ONS_INVOKE_ERROR	Open API 接口调用失败	联系技术支持人员处理
BIZ_FIELD_CHECK_INVALID	参数检验失败	请参考 API 手册检查各个参数传入是否合法
BIZ_TOPIC_NOT_FOUND	Topic 没有找到	请检查 Topic 输入是否合法，或者是否创建过
BIZ_SUBSCRIPTION_NOT_FOUND	目标订阅关系 GID 找不到	请检查 GID 是否创建过，或者查询条件错误
BIZ_PUBLISHER_EXISTED	指定 GID 已经存在	更换 GID 的名称重试请求
BIZ_SUBSCRIPTION_EXISTED	指定 GID 已经存在	更换 GID 的名称重试请求
BIZ_CONSUMER_NOT_ONLINE	指定 GID 的客户端不在线	请确保消费端在线，然后重试请求
BIZ_NO_MESSAGE	当前查询条件没有匹配消息	请检查查询条件，并确认查询范围内是否发过消息
BIZ_REGION_NOT_FOUND	请求的 Region 找不到	确认请求的 Region 参数是否合法
BIZ_TOPIC_EXISTED	指定 Topic 已经存在	更改 Topic 名称重试请求
BIZ_PUBLISH_INFO_NOT_FOUND	请求的 GID 没有找到	确认 GID 是否存在，或者请求条件错误
EMPOWER_EXIST_ERROR	当前授权关系已经存在	核实请求参数，重试或者先查询
EMPOWER_OWNER_CHECK_ERROR	当前用户不是授权 Topic 的 Owner	确认资源所属关系

4 设备管理接口

4.1 查询设备连接信息

OnsMqttQueryClientByClientId

根据 Client ID 查询指定设备的连接信息，包含在线信息、订阅关系等。

描述

使用场景

OnsMqttQueryClientByClientId 接口一般用于线上追踪单个设备的运行状态及排查问题。输入 Client ID 即可查到对应设备是否在线、设备地址、当前的订阅关系等信息。

关于 Client ID 的详细信息，请参见[名词解释](#)。

使用限制

由于 OpenAPI 面向的场景是用户自定义管控开发，服务端会对过快的调用进行限流（每分钟 30 次），因此不要在业务的主流程中使用本接口。如需判断设备是否在线，请使用设备上下线通知功能。

请求参数列表

名称	类型	是否必需	描述
OnsPlatform	String	否	请求来源，默认来源是 POP 平台
PreventCache	Long	是	用于 CSRF 校验，设置为系统当前时间即可，单位毫秒（ms）
ClientId	String	是	需要查询的目标 Client ID
InstanceId	String	是	需查询的设备所对应的 MQTT 实例 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数，每个请求的 ID 都是唯一的，可用于排查定位问题。

名称	类型	描述
HelpUrl	String	帮助链接
Data	MqttClientInfoDo	设备在线信息数据结构

MqttClientInfoDo 数据结构列表

名称	类型	描述
Online	Boolean	设备是否在线，取值说明如下： · true：在线 · false：不在线
ClientId	String	设备的 Client ID
SocketChannel	String	设备连接的 IP 地址
LastTouch	Long	最后更新时间
SubScriptonData	List(SubscriptionDo)	该设备当前的订阅关系集合

SubscriptionDo 数据结构列表

名称	类型	描述
ParentTopic	String	MQTT 的一级父 Topic
SubTopic	String	MQTT 的多级子 Topic，如果没有则为 NULL。
Qos	Integer	订阅关系的 QoS 级别，取值说明如下： · 0：最多分发一次 · 1：至少达到一次 · 2：仅分发一次 关于 QoS 的详细信息，请参见 名词解释 。

使用示例

本示例仅提供一个参考，从杭州接入点接入，查询指定 Client ID 所对应的设备的连接数据。

```
public static void main(String[] args) {
    String regionId = "cn-hangzhou";
    String accessKey = "XXXXXXXXXXXXXXXXXXXX";
    String secretKey = "XXXXXXXXXXXXXXXXXXXX";
    IClientProfile profile= DefaultProfile.getProfile(regionId,
accessKey,secretKey);
```

```

        IAcsClient iAcsClient= new DefaultAcsClient(profile);
        OnsMqttQueryClientByClientIdRequest request = new OnsMqttQueryClientByClientIdRequest();
        request.setPreventCache(System.currentTimeMillis());
        request.setAcceptFormat(FormatType.JSON);
        request.setClientId("GID_XXX@@@XXXXXX");
        request.setInstanceId("aaaaa");
        try {
            OnsMqttQueryClientByClientIdResponse response =
            iAcsClient.getAcsResponse(request);
            OnsMqttQueryClientByClientIdResponse.MqttClientInfoDo
            clientInfoDo = response.getMqttClientInfoDo();
            System.out.println(clientInfoDo.getOnline() + " " +
                clientInfoDo.getClientId() + " " +
                clientInfoDo.getLastTouch() + " " +
                clientInfoDo.getSocketChannel());
            for (OnsMqttQueryClientByClientIdResponse.MqttClient
            InfoDo.SubscriptionDo subscriptionDo : clientInfoDo.getSubScriptonData
            ()) {
                System.out.println(subscriptionDo.getParentTopic()
                + " " + subscriptionDo.getSubTopic() + " " + subscriptionDo.getQos
                ());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}

```

4.2 查询 Group 下在线设备的数量

OnsMqttQueryClientByGroupId

根据 Group ID (GID) 统计属于该 Group 下的在线设备的数量。

描述

使用场景

OnsMqttQueryClientByGroupId 接口一般用于业务分析，统计一个 GID 下终端设备的活跃程度。

使用限制

由于 OpenAPI 面向的场景是用户自定义管控开发，服务端会对过快的调用进行限流（每分钟 30 次），因此不要在业务的主流程中使用本接口。

请求参数列表

名称	类型	是否必需	描述
OnsPlatform	String	否	请求来源，默认来源是 POP 平台

名称	类型	是否必需	描述
PreventCache	Long	是	用于 CSRF 校验, 设置为系统当前时间即可, 单位毫秒 (ms)
GroupId	String	是	需查询的 GID
InstanceId	String	是	需查询的 GID 所属的 MQTT 实例 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数, 每个请求的 ID 都是唯一的, 可用于排查定位问题
HelpUrl	String	帮助链接
MqttClientSetDo	MqttClientSetDo	Group 下的设备在线信息数据结构

MqttClientSetDo 数据结构列表

名称	类型	描述
OnlineCount	Long	Group 所有在线设备数量

使用示例

本示例仅提供一个参考, 从杭州接入点接入, 查询指定 GID 的在线设备数量信息。

```

public static void main(String[] args) {
    String regionId = "cn-hangzhou";
    String accessKey = "XXXXXXXXXXXXXXXXXXXX";
    String secretKey = "XXXXXXXXXXXXXXXXXXXX";
    IClientProfile profile= DefaultProfile.getProfile(regionId,
accessKey,secretKey);
    IAcsClient iAcsClient= new DefaultAcsClient(profile);
    OnsMqttQueryClientByGroupIdRequest request = new OnsMqttQueryClientByGroupIdRequest();
    request.setPreventCache(System.currentTimeMillis());
    request.setAcceptFormat(FormatType.JSON);
    request.setGroupId("GID_XXXXX");
    request.setInstanceId("aaaaa");
    try {
        OnsMqttQueryClientByGroupIdResponse response =
iAcsClient.getAcsResponse(request);
        OnsMqttQueryClientByGroupIdResponse.MqttClientSetDo
clientSetDo = response.getMqttClientSetDo();
        System.out.println(clientSetDo.getOnlineCount() + " "
+

```

```

        clientSetDo.getPersistCount() + " ");
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}

```

4.3 根据 Topic 查询在线设备的数量

OnsMqttQueryClientByTopic

根据 MQTT 的 Topic 统计订阅该 Topic 的在线设备的数量。

描述

使用场景

OnsMqttQueryClientByTopic 接口一般用于业务分析。业务上订阅某个特定 Topic 的设备一般是参与某个特殊活动的设备，可以根据 Topic 查询参与该活动的设备数量。

使用限制

由于 OpenAPI 面向的场景是用户自定义管控开发，服务端会对过快的调用进行限流（每分钟 30 次），因此不要在业务的主流程中使用本接口。如需判断设备是否在线，请使用设备上下线通知功能。

请求参数列表

名称	类型	是否必需	描述
OnsPlatform	String	否	请求来源，默认来源是 POP 平台
PreventCache	Long	是	用于 CSRF 校验，设置为系统当前时间即可，单位毫秒 (ms)
ParentTopic	String	是	需查询的目标父 Topic
SubTopic	String	否	需查询的目标子 Topic，如果没有则不填
InstanceId	String	是	需查询的 Topic 所属的 MQTT 实例 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数，每个请求的 ID 都是唯一的，可用于排查定位问题
HelpUrl	String	帮助链接
MqttClientSetDo	MqttClientSetDo	设备在线信息数据结构

MqttClientSetDo 数据结构列表

名称	类型	描述
OnlineCount	Long	订阅指定 Topic 的所有在线设备数量

使用示例

本示例仅提供一个参考，从杭州接入点接入，查询订阅指定 Topic 的在线设备的数量。

```

public static void main(String[] args) {
    String regionId = "cn-hangzhou";
    String accessKey = "XXXXXXXXXXXXXXXXXXXX";
    String secretKey = "XXXXXXXXXXXXXXXXXXXX";
    IClientProfile profile= DefaultProfile.getProfile(regionId,
accessKey,secretKey);
    IAcsClient iAcsClient= new DefaultAcsClient(profile);
    OnsMqttQueryClientByTopicRequest request = new OnsMqttQueryClientByTopicRequest();
    request.setInstanceId("aaaaa");
    request.setPreventCache(System.currentTimeMillis());
    request.setAcceptFormat(FormatType.JSON);
    request.setParentTopic("XXXX");
    request.setSubTopic("/secTopic/0/1");//此处如果没有子 Topic 则不填；如果有，一定是完整的信息
    try {
        OnsMqttQueryClientByTopicResponse response = iAcsClient.getAcsResponse(request);
        OnsMqttQueryClientByTopicResponse.MqttClientSetDo clientSetDo = response.getMqttClientSetDo();
        System.out.println(clientSetDo.getOnlineCount() + " "
+
        clientSetDo.getPersistCount() + " ");
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}

```

4.4 查询历史在线设备统计

OnsMqttQueryHistoryOnline

根据 Group ID (GID) 和给定的时间段查询历史在线设备的数量曲线。

描述

使用场景

OnsMqttQueryHistoryOnline 接口一般用于生成业务报表。

使用限制

由于 OpenAPI 面向的场景是用户自定义管控开发，服务端会对过快的调用进行限流（每分钟 30 次），因此不要在业务的主流程中使用本接口。如需判断设备是否在线，请使用设备上下线通知功能。

请求参数列表

名称	类型	是否必需	描述
OnsRegionId	String	是	当前操作的 MQTT 实例所在地域 (Region) 关于地域的详细信息，请参见 管控 API 接入指南 。
OnsPlatform	String	否	请求来源，默认来源是 POP 平台
PreventCache	Long	是	用于 CSRF 校验，设置为系统当前时间即可，单位毫秒 (ms)
GroupId	String	是	需查询的目标 GID
BeginTime	Long	是	查询的起始时间
EndTime	Long	是	查询的终止时间，起止时间范围建议尽可能在当天，否则后端会自动截断
InstanceId	String	是	需查询的 GID 所属的 MQTT 实例 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数，每个请求的 ID 都是唯一的
HelpUrl	String	帮助链接
data	Data	数据集合

Data 数据集列表

名称	类型	描述
Title	String	Table 的名称
Records	List(StatsDataDo)	采集点信息

StatsDataDo 数据集列表

名称	类型	描述
X	Long	横轴，毫秒时间戳
Y	Float	纵轴，数量

使用示例

本示例仅提供一个参考，从杭州接入点接入，查询指定 GID 下在过去一小时在线设备的数量信息。

```

public static void main(String[] args) {
    String regionId = "cn-hangzhou";
    String accessKey = "XXXXXXXXXXXXXXXXXXXX";
    String secretKey = "XXXXXXXXXXXXXXXXXXXX";
    IClientProfile profile= DefaultProfile.getProfile(regionId,
accessKey,secretKey);
    IAcsClient iAcsClient= new DefaultAcsClient(profile);
    OnsMqttQueryHistoryOnlineRequest request = new OnsMqttQue
ryHistoryOnlineRequest();
    request.setPreventCache(System.currentTimeMillis());
    request.setAcceptFormat(FormatType.JSON);
    request.setGroupId("XXX");
    request.setBeginTime(System.currentTimeMillis()-1000*3600);
    request.setEndTime(System.currentTimeMillis());
    request.setInstanceId("aaaaa");
    try {
        OnsMqttQueryHistoryOnlineResponse response = iAcsClient.
getAcsResponse(request);
        OnsMqttQueryHistoryOnlineResponse.Data data =response.
getData();
        System.out.println(data.getTitle()+"\n"+
data.getRecords());
    } catch (ServerException e) {

```



```
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
```

5 Group 管理接口

5.1 创建 Group ID

OnsMqttGroupIdCreate

在服务器上创建微消息队列 MQTT 的分组，即 Group ID (GID)。

描述

本接口限企业铂金版用户专用，请前往[企业铂金版购买页](#)查看详情。

使用场景

新应用上线时，在创建 Topic 资源后，如需使用 MQTT 客户端收发消息，首先需在控制台或者调用本接口创建 GID。

请求参数列表

名称	类型	是否必需	描述
OnsPlatform	String	否	该请求来源，默认来源是 POP 平台
PreventCache	Long	是	用于 CSRF 校验，设置为系统当前时间即可，单位毫秒 (ms)
GroupId	String	是	创建的 MQTT 分组的 ID
InstanceId	String	是	需要创建 GID 的 MQTT 实例的 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数，每个请求的 ID 都是唯一的
HelpUrl	String	帮助链接

使用示例

本示例在 cn-hangzhou 地域 (Region) 创建名为 GID_Test 的 Group。

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

```
String regionId = "cn-hangzhou";
String accessKey = "XXXXXXXXXXXXXXXXXXXX";
String secretKey = "XXXXXXXXXXXXXXXXXXXX";
IClientProfile profile= DefaultProfile.getProfile(regionId
, accessKey, secretKey);
IAcsClient iAcsClient= new DefaultAcsClient(profile);
OnsMqttGroupIdCreateRequest request = new OnsMqttGro
upIdCreateRequest();
request.setPreventCache(System.currentTimeMillis());
request.setAcceptFormat(FormatType.JSON);
request.setGroupId("GID_Test");
request.setInstanceId("aaaaa");
try {
    OnsMqttGroupIdCreateResponse response=iAcsClient.
getAcsResponse(request);
    System.out.println(response.getRequestId());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

5.2 查询 Group ID 列表

OnsMqttGroupIdList

查询目标地域 (Region) 里您所有的 Group ID (GID) 。

描述

OnsMqttGroupIdList 接口一般用于管理您所有的 GID 的场景。您可先获取 GID 列表，然后查询单个 GID 的相关数据。

请求参数列表

名称	类型	是否必需	描述
OnsPlatform	String	否	请求来源，默认来源是 POP 平台
PreventCache	Long	是	用于 CSRF 校验，设置为系统当前时间即可，单位毫秒 (ms)
InstanceId	String	是	需查询的 GID 的 MQTT 实例 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数，每个请求的 ID 都是唯一的，可用于排查定位问题
HelpUrl	String	帮助链接
Data	MqttGroupIdDo	GID 信息数据结构

MqttGroupIdDo 数据结构列表

名称	类型	描述
Id	Long	数据编号
ChannelId	Integer	访问的途径编号，阿里云环境为“0”
Owner	String	GID 所属账号
GroupId	String	GID
Topic	String	该 GID 关联的父 Topic
Status	Integer	当前状态，取值说明如下： · 0：服务中 · 1：冻结 · 2：暂停
CreateTime	Long	创建时间，单位是毫秒时间戳
UpdateTime	Long	最后更新时间，单位是毫秒时间戳
InstanceId	String	当前 GID 所属的实例 ID
IndependentNaming	Boolean	当前 GID 所属实例是否有命名空间，取值说明如下： · true：拥有独立命名空间，资源命名确保实例内唯一，跨实例之间可重名 · false：无独立命名空间，实例内或者跨实例之间，资源命名必须全局唯一

使用示例

```
public static void main(String[] args) { String regionId = "cn-
hangzhou"; String accessKey = "XXXXXXXXXXXXXXXXXXXX"; String secretKey
= "XXXXXXXXXXXXXXXXXXXX"; IClientProfile profile= DefaultProfile.
getProfile(regionId,accessKey,secretKey); IAcsClient iAcsClient=
new DefaultAcsClient(profile); OnsMqttGroupIdListRequest request
= new OnsMqttGroupIdListRequest(); request.setPreventCache(System
.currentTimeMillis()); request.setAcceptFormat(FormatType.JSON);
request.setInstanceId("aaaaa"); try { OnsMqttGroupIdListResponse
response=iAcsClient.getAcsResponse(request); List groupIdList=
response.getData(); for(OnsMqttGroupIdListResponse.MqttGroupIdDo
groupDo:groupIdList){ System.out.println(groupDo.getId()+" "+ groupDo.
getChannelId()+" "+ groupDo.getOwner()+" "+ groupDo.getGroupId()+" "+
groupDo.getTopic()+" "+ groupDo.getStatus()+" "+ groupDo.getCreateT
ime()+" "+ groupDo.getUpdateTime()); } } catch (ServerException e) {
e.printStackTrace(); } catch (ClientException e) { e.printStackTrace
(); } }
```

5.3 删除 Group ID

OnsMqttGroupIdDelete

在服务器上删除已经存在的微消息队列 MQTT 的分组，即 Group ID (GID)。

描述

本接口限企业铂金版用户专用，请前往[企业铂金版购买页](#)查看详情。

使用场景

新应用迭代过程中，如果部分 GID 不再使用，可以在控制台或者调用本接口删除 GID。

请求参数列表

名称	类型	是否必需	描述
OnsPlatform	String	否	该请求来源，默认来源是 POP 平台
PreventCache	Long	是	用于 CSRF 校验，设置为系统当前时间即可，单位毫秒 (ms)
GroupId	String	是	需删除的 GID
InstanceId	String	是	需删除 GID 的 MQTT 实例的 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数，每个请求的 ID 都是唯一的
HelpUrl	String	帮助链接

使用示例

本示例在 cn-hangzhou 地域 (Region) 删除名为 GID_MingduanTest 的 GID。

```

public static void main(String []args) {
    String regionId = "cn-hangzhou";
    String accessKey = "XXXXXXXXXXXXXXXXXXXX";
    String secretKey = "XXXXXXXXXXXXXXXXXXXX";
    IClientProfile profile= DefaultProfile.getProfile(regionId
, accessKey, secretKey);
    IAcsClient iAcsClient= new DefaultAcsClient(profile);
    OnsMqttGroupIdDeleteRequest request = new OnsMqttGro
upIdDeleteRequest();
    request.setPreventCache(System.currentTimeMillis());
    request.setAcceptFormat(FormatType.JSON);
    request.setGroupId("GID_MingduanTest");
    request.setInstanceId("aaaaa");
    try {
        OnsMqttGroupIdDeleteResponse response=iAcsClient.
getAcsResponse(request);
        System.out.println(response.getRequestId());
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
    
```

6 资源报表接口

6.1 消息收发统计

OnsMqttQueryMsgTransTrend

根据 Topic 等信息查询历史消息收发量。

描述

使用场景

OnsMqttQueryMsgTransTrend 接口一般用于生成数据报表、统计业务规模等场景。

使用限制

由于 OpenAPI 面向的场景是用户自定义管控开发，服务端会对过快的调用进行限流（每分钟 30 次），因此不要在业务的主流程中使用本接口。如需判断设备是否在线，请使用设备上下线通知功能。

请求参数列表

名称	类型	是否必需	描述
OnsPlatform	String	否	请求来源，默认来源是 POP 平台
PreventCache	Long	是	用于 CSRF 校验，设置为系统当前时间即可，单位毫秒 (ms)
ParentTopic	String	是	需查询的父 Topic
SubTopic	String	否	需查询的子 Topic，如果没有子 Topic 或者需查询父 Topic 的所有信息，可为空
MsgType	String	否	消息类别，取值说明如下： <ul style="list-style-type: none"> · P2P：点对点 · SUB：发布/订阅 如果不填则默认查所有消息，其他值无效。

名称	类型	是否必需	描述
TpsType	String	是	查询类别，取值说明如下： <ul style="list-style-type: none"> · TPS：消息生产 TPS · SUM：查询消息生产总量 其他值无效。
TransType	String	是	收发类别，取值说明如下： <ul style="list-style-type: none"> · PUB：查询生产端消息生产总量 · SUB：查询消费端消息消费总量 其他值无效。
Qos	Integer	否	查询的 QoS 级别，取值说明如下： <ul style="list-style-type: none"> · 0：最多分发一次 · 1：至少达到一次 · 2：仅分发一次 其他值或者不填都默认查所有消息。关于 QoS 的详细信息，请参见 #unique_8 。
BeginTime	Long	是	查询的起始时间
EndTime	Long	是	查询的终止时间，起止时间范围建议尽可能在当天，否则后端会自动截断。
InstanceId	String	是	需查询的消息所属的 MQTT 实例 ID

返回参数列表

名称	类型	描述
RequestId	String	为公共参数，每个请求的 ID 都是唯一的
HelpUrl	String	帮助链接

名称	类型	描述
data	Data	数据集合

Data 数据集列表

名称	类型	描述
Title	String	Table 的名称
Records	List(StatsDataDo)	采集点信息

StatsDataDo 数据集列表

名称	类型	描述
X	Long	横轴, 毫秒时间戳
Y	Float	纵轴, 数据 (TPS 或者总量)

使用示例

本示例仅提供一个参考, 从杭州接入点接入, 查询指定 Topic 过去一小时的消息发送数量信息。

```

public static void main(String[] args) {
    String regionId = "cn-hangzhou";
    String accessKey = "XXXXXXXXXXXXXXXXXXXX";
    String secretKey = "XXXXXXXXXXXXXXXXXXXX";
    IClientProfile profile= DefaultProfile.getProfile(regionId,
accessKey,secretKey);
    IAcsClient iAcsClient= new DefaultAcsClient(profile);
    OnsMqttQueryMsgTransTrendRequest request = new OnsMqttQue
ryMsgTransTrendRequest();
    request.setInstanceId("aaaaa");
    request.setPreventCache(System.currentTimeMillis());
    request.setAcceptFormat(FormatType.JSON);
    request.setParentTopic("MQTT_TOPIC_ONSMONITOR_BJ");
    //request.setMsgType("sub");
    request.setTpsType("SUM");
    request.setTransType("PUB");
    //request.setQos(1);
    request.setBeginTime(System.currentTimeMillis()-1000*3600);
    request.setEndTime(System.currentTimeMillis());
    try {
        OnsMqttQueryMsgTransTrendResponse response = iAcsClient.
getAcsResponse(request);
        OnsMqttQueryMsgTransTrendResponse.Data data =response.
getData();
        System.out.println(data.getTitle()+"\n"+
data.getRecords());
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}

```

