

消息通知服务

用户指南

文档版本 01
发布日期 2022-04-12



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 产品介绍	1
1.1 什么是消息通知服务	1
1.2 产品优势	1
1.3 应用场景	2
1.4 权限管理	2
1.5 基本概念	4
1.6 区域和可用区	5
2 快速入门	7
2.1 如何发布主题消息	7
3 主题管理	11
3.1 创建主题	11
3.2 修改显示名	12
3.3 设置主题策略	12
3.3.1 基本模式	13
3.4 订阅主题	14
3.5 发布主题消息	15
3.5.1 发布主题消息简介	15
3.5.2 向主题发布文本消息	15
3.5.3 向主题发布 JSON 消息	16
3.5.4 向主题发布模板消息	17
3.6 删除主题	19
4 订阅管理	20
4.1 添加订阅	20
4.2 请求订阅	21
4.3 取消订阅	22
4.4 删除订阅	22
5 消息模板管理	24
6 记录消息通知服务	26
6.1 记录消息通知服务简介	26
6.2 支持审计的关键操作列表	26
6.3 查看审计日志	27

7 消息传输日志	29
8 权限管理	32
8.1 创建用户并授权使用 SMN	32
9 关于配额	34
10 常见问题	35
10.1 消息通知服务有哪些优势?	35
10.2 消息通知服务支持的传输协议有哪些?	35
10.3 消息通知服务的主题名称有何格式要求?	35
10.4 每个用户可以创建多少个主题?	35
10.5 每个主题最多支持多少个订阅者?	36
10.6 可以向主题发布多少条消息?	36
10.7 每个用户可以创建多少个消息模板?	36
10.8 单个主题是否支持通过多种传输协议进行订阅?	36
10.9 主题创建者是否可以控制所创建的主题中允许的传输协议?	36
10.10 订阅终端地址可以修改吗?	36
10.11 是否可删除已发布的消息?	36
10.12 是否会保证将消息传送到订阅的终端节点?	36
10.13 每条消息会让订阅者收到多少次?	37
10.14 向主题推送消息后, 订阅者为什么没有收到消息?	37
10.15 为什么用户接收到的邮件中图片无法显示?	37
10.16 如何获取用户的账号 ID?	39
A 附录	40
A.1 JSON 格式消息说明	40
A.2 模板格式消息说明	41
A.3 不同协议消息说明	42
A.4 订阅确认流控说明	42
A.5 授权操作与 SMN 接口的映射关系	42
A.6 HTTP(S)终端节点使用样例	43
A.6.1 HTTP(S)终端节点使用样例简介	43
A.6.2 HTTP(S)消息格式	44
A.6.3 校验消息签名	47
A.6.4 示例代码	49
B 修订记录	54

1 产品介绍

1.1 什么是消息通知服务

消息通知服务（Simple Message Notification, SMN）是可靠的、可扩展的、海量的消息处理服务。它可以依据用户的需求主动推送通知消息，最终用户可以通过电子邮件等方式接收。

SMN的主题订阅模型，旨在提供一对多的消息订阅以及通知功能，能够实现一站式集成多种推送通知方式。在SMN中有两种类型的客户端，即发布者和订阅者。发布者通过主题发布消息，由SMN将消息推送给主题订阅者，订阅者可以为邮箱地址、手机号码及URL地址。

主题作为消息的集合，是一个逻辑访问点和通信渠道，拥有唯一的主题名称。主题创建者可以设置主题策略，授权其他云服务或者用户操作该主题，例如查询主题订阅者列表、发布消息等。

1.2 产品优势

对比其他传统消息中间件组件，消息通知服务的核心优势如表1-1所示。

表 1-1 产品优势

核心优势	消息通知服务	传统消息中间件
服务简便	直接使用消息通知服务，只需要使用三个简单的API (创建Topic、订阅Topic、发送消息)，就能够快速高效的发送消息，使用门槛极低，快速融入业务使用。	自建消息通知服务成本高，接口使用复杂，学习曲线高，融于业务时间周期长。
稳定可靠	消息在多数据中心冗余，Topic支持透明迁移。消息推送失败，可以设置消息推送到SMN进行持久化。服务单节点故障，请求会自动迁移到可用节点。	关键业务使用对消息通知服务的稳定性和可靠性要求很高，需要解决消息不丢问题，并能提供多种措施保障业务的连续性。

核心优势	消息通知服务	传统消息中间件
多协议通知	使用消息通知服务，只需要通过一次发布请求，就能向各种协议的订阅者推送消息。	业务需要发送电子邮件，或者进行HTTP推送，开发多种协议的消息收发系统，周期长。
安全	消息通知服务数据安全是基于Topic进行安全隔离，用户未经授权不能访问队列消息，有效保护用户业务安全。	业务数据访问需要有安全保护措施，没有认证授权的系统随意获取消息会导致严重的数据安全和隐私风险。

1.3 应用场景

- **系统告警**

系统告警是由预定义阈值触发的通知，通过邮件、HTTP和HTTPS等多种通知方式发送给特定用户。举例来说，很多云服务都使用SMN，从而可使您在事件发生（如云审计服务检测到云服务资源发生关键操作）时能立即接收到通知。

- **与云服务的集成**

将SMN作为消息连接不同的云服务，可降低系统复杂度，提升服务使用效率。例如将消息从云服务（如CES）通知到其他服务（如OBS），实现服务解耦。即便一个服务出现故障，也不会影响到其他服务。

- **错峰流控**

上下游系统处理能力有差异时，可以使用SMN转储系统间的通信数据，提供消息堆积缓冲能力，减少下游系统的压力，可减少系统崩溃等问题，提高系统可用性，降低系统实现的复杂性。

1.4 权限管理

如果您需要对云服务平台上创建或购买的SMN资源，为企业中的员工设置不同的访问权限，以达到不同员工之间的权限隔离，您可以使用统一身份认证服务（Identity and Access Management，简称IAM）进行精细的权限管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制云服务资源的访问。

通过IAM，您可以在账号中给员工创建IAM用户，并授权控制他们对云服务资源的访问范围。例如您的员工中有负责软件开发的人员，您希望他们拥有SMN的使用权限，但是不希望他们拥有删除SMN等高危操作的权限，那么您可以使用IAM为开发人员创建用户，通过授予仅能使用SMN，但是不允许删除SMN的权限，控制他们对SMN资源的使用范围。

如果账号已经能满足您的要求，不需要创建独立的IAM用户进行权限管理，您可以跳过本章节，不影响您使用SMN的其它功能。

IAM是云服务平台提供权限管理的基础服务，无需付费即可使用，您只需要为您账号中的资源进行付费。

关于IAM的详细介绍，请参见[IAM产品简介](#)。

SMN 权限

默认情况下，账号管理员创建的IAM用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使得用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对云服务进行操作。

SMN部署时通过物理区域划分，为项目级服务。授权时，“作用范围”需要选择“区域级项目”，然后在指定区域对应的项目中设置相关权限，并且该权限仅对此项目生效；如果在“所有项目”中设置权限，则该权限在所有区域项目中都生效。访问SMN时，需要先切换至授权区域。

如表1-2所示，包括了SMN的所有系统角色。

表 1-2 SMN 系统角色

系统角色	描述	类别	依赖关系
SMN Administrator	消息通知服务的管理员权限，拥有该权限的用户拥有消息通知服务所有执行权限。	系统角色	依赖Tenant Guest角色，需要在同项目中勾选依赖的角色。

表1-3列出了SMN常用操作与系统权限的授权关系，您可以参照该表选择合适的系统权限。

表 1-3 常用操作与系统权限的关系

操作	SMN Administrator
创建主题	√
更新主题	√
删除主题	√
查询主题列表	√
订阅主题	√
设置主题标签	√
设置主题策略	√
发布主题消息	√
添加订阅	√
请求订阅	√
取消订阅	√
查询订阅者列表	√
创建消息模板	√
更新消息模板	√

操作	SMN Administrator
删除消息模板	√
查询消息模板	√

1.5 基本概念

项目

项目用于将OpenStack的资源（计算资源、存储资源和网络资源）进行分组和隔离。项目可以是一个部门或者一个项目组。一个账户中可以创建多个项目。

协议

SMN支持发送的消息类型，包括邮件、HTTP(S)。

发布者

消息发布者，向主题发送消息的对象。

订阅者

消息订阅者，从主题接收消息的订阅终端。

添加订阅时，用户可以根据需要选择不同的传输协议，每种协议对应不同类型的订阅者，具体对应关系如下所示：

- 邮件传输协议，订阅者为邮箱地址。
- HTTP/HTTPS传输协议，订阅者为URL。

主题

主题是消息发布或客户端订阅通知的特定事件类型。它作为发送消息和订阅通知的信道，是消息隔离基本单元，为发布者和订阅者提供一个可以相互交流的通道。

URN

URN是SMN用来标识资源的字符串，目前存在以下几种类型。

- 主题URN
创建主题后，SMN将为该主题分配的唯一资源标识，其中包括服务名称（SMN）、区域名称、项目ID和主题名称。形如：
urn:smn:region:cffe4fc4c9a54219b60dbaf7b586e132:Mytopic。主题URN将作为创建主题时使用的API调用的一部分返回。无论何时发布者或者订阅者需要对该主题执行任何操作，都应引用该唯一主题URN。
- 订阅URN
成功订阅主题后，由系统生成的订阅者唯一资源标识，其中包括服务名称（SMN）、区域名称、项目ID、主题名称、订阅ID等。形如：
urn:smn:region:cffe4fc4c9a54219b60dbaf7b586e132:Mytopic:5293b436967f45

0abc51e0c36347b27a。在您确认订阅和取消订阅时，页面会显示订阅URN，方便您确认是否操作正确。

消息模板

消息模板指消息内容的固定格式。发布消息时可以使用已创建的消息模板生成模板消息。如果消息模板中存在模板变量，生成模板消息时系统会将模板变量自动替换为用户设置的参数值。

模板变量

创建消息模板时，在消息模板里设置的变量。在使用消息模板发送消息时，可以使用参数值来替换此变量。

如：“The Arts and Crafts Exposition will be held from {startdate} through {enddate}. We sincerely invite you to join us.” 其中，“{startdate}”和“{enddate}”即为模板变量。

1.6 区域和可用区

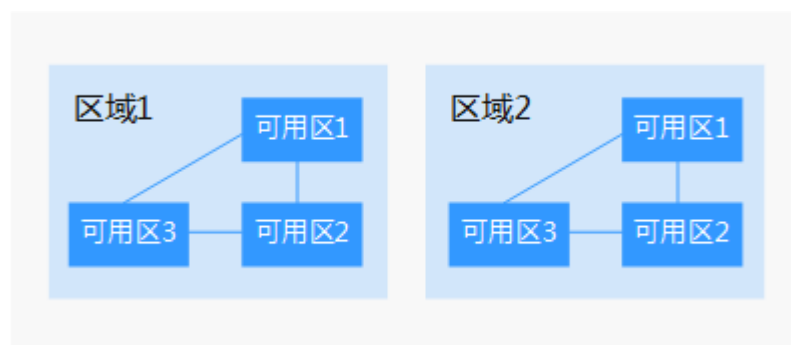
什么是区域、可用区？

区域和可用区用来描述数据中心的位置，您可以在特定的区域、可用区创建资源。

- 区域（Region）指物理的数据中心。每个区域完全独立，这样可以实现最大程度的容错能力和稳定性。资源创建成功后不能更换区域。
- 可用区（AZ, Availability Zone）是同一区域内，电力和网络互相隔离的物理区域，一个可用区不受其他可用区故障的影响。一个区域内可以有多个可用区，不同可用区之间物理隔离，但内网互通，既保障了可用区的独立性，又提供了低价、低时延的网络连接。

图1-1阐明了区域和可用区之间的关系。

图 1-1 区域和可用区



如何选择区域？

建议就近选择靠近您或者您的目标用户的区域，这样可以减少网络时延，提高访问速度。

如何选择可用区？

是否将资源放在同一可用区内，主要取决于您对容灾能力和网络时延的要求。

- 如果您的应用需要较高的容灾能力，建议您将资源部署在同一区域的不同可用区内。
- 如果您的应用要求实例之间的网络延时较低，则建议您将资源创建在同一可用区内。

区域和终端节点

当您通过API使用资源时，您必须指定其区域终端节点。有关的区域和终端节点的更多信息，请参阅[地区和终端节点](#)。

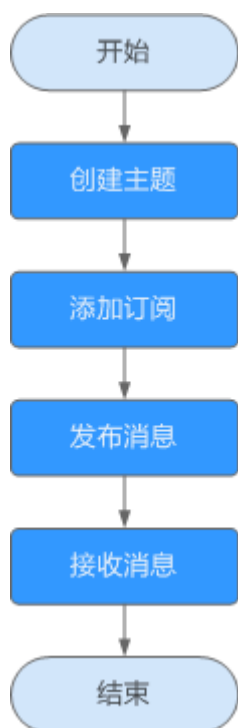
2 快速入门

2.1 如何发布主题消息

在熟悉了SMN的基本概念之后，您可以通过消息通知服务的管理控制台进行基本主题订阅操作。

通过主题发布消息的流程图如图2-1所示。

图 2-1 发布主题消息流程



操作场景

假设某公司定期举行展览会，需要邀请相关人员，此时消息内容不变，仅需更改日期。这种场景的特性是：用户发送的通知消息中，大部分内容是固定的，只有少数内容随着每次事件指标变化而变化。针对这种需求，用户可以创建一个消息模板，在模

板中，用模板变量代替变化的内容，每次通知时，选择消息模板并设置模板变量值便可完成特定的消息通知。

步骤 1 创建主题


1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“主题”。
进入主题页面。
5. 在主题页面，单击“创建主题”。
6. 在“主题名称”框中，输入主题名称，在“显示名”框中输入相关描述，如表2-1所示。

表 2-1 创建主题参数说明

参数	说明
主题名称	创建的主题名称，用户可自定义名称，规范如下： <ul style="list-style-type: none">• 只能包含字母，数字，短横线(-)和下划线(_)，且必须由大写字母、小写字母或数字开头。• 名称长度限制在1-255字符之间。• 主题名称为主题的唯一标识，一旦创建后不能再修改主题名称。
显示名	显示名，长度限制在192字节或64个中文字。 说明 推送邮件消息时，如果未设置主题的显示名，发件人呈现为“username@example.com”，如果已设置主题的显示名，发件人则呈现为“显示名<username@example.com>”。
企业项目	企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。
标签	标签由键值对组成，用于标识云资源，可对云资源进行分类和搜索。 <ul style="list-style-type: none">• 键的长度最大36字符，值的长度最大43字符，只能包含数字、字母、“-”、“_”。

7. 单击“确定”，主题创建成功。
主题创建成功后，系统会自动生成主题URN，主题URN是主题的唯一资源标识，不可修改。新创建的主题将显示在主题列表中。
8. 单击主题名称，可查看主题详情（包含主题URN、显示名等）、主题订阅总数。

步骤 2 添加订阅

1. 登录管理控制台。


2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“订阅”。
5. 在订阅页面。单击“添加订阅”，开始添加消息订阅。
此时将显示“添加订阅”页面。
6. 向指定主题添加订阅。
 - a. 单击“主题名称”输入框右侧的“选择主题”，在弹出的选择主题对话框中选择主题。
 - b. 在“协议”下拉框中选择订阅终端支持的协议，在“订阅终端”输入框中输入对应的订阅终端，填写说明如表2-2所示。

表 2-2 添加订阅参数说明

参数	说明
主题名称	在弹出的对话框中选择的主题的名称。
协议	在下拉框中选择需要订阅终端支持的协议，提供的选项为“邮件”、“HTTP”、“HTTPS”。
订阅终端	<p>订阅的终端地址，邮件、HTTP以及HTTPS终端支持批量输入，批量添加时，每个终端地址占一行。最多可输入10个终端。</p> <ul style="list-style-type: none"> ● 选择“邮件”终端，请输入有效的邮箱地址。 例如： username@example.com username2@example.com ● 选择“HTTP”终端，请输入公网网址。 例如： http://example.com/notification/action ● 选择“HTTPS”终端，请输入公网网址。 例如： https://example.com/notification/action

7. 单击“确定”。
新增订阅将显示在页面下方的订阅列表中。

说明

- 为了防止恶意攻击订阅终端，云平台会对订阅终端进行流控保护，触发流控规则后，消息通知服务不再发送订阅确认消息，具体的保护规则请参见《消息通知服务用户指南》的附录“订阅确认流控说明”部分。
- 消息通知服务不会校验订阅终端是否真实存在。
- 添加订阅后，消息通知服务会向订阅终端发送订阅确认信息，信息中包含订阅确认的链接。订阅确认的链接在48小时内有效，用户需要及时在手机端、邮箱或其他协议终端确认订阅。
- 发送请求订阅消息也会作为发送条数进行计费。

3 主题管理

3.1 创建主题

操作场景

主题是消息发布或客户端订阅通知的特定事件类型。它作为发送消息和订阅通知的信道，为发布者和订阅者提供一个可以相互交流的通道。

创建主题


1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“主题”。
进入主题页面。
5. 在主题页面，单击“创建主题”。
6. 在“主题名称”框中，输入主题名称，在“显示名”框中输入相关描述，如表3-1所示。

表 3-1 创建主题参数说明

参数	说明
主题名称	创建的主题名称，用户可自定义名称，规范如下： <ul style="list-style-type: none">• 只能包含字母，数字，短横线(-)和下划线(_)，且必须由大写字母、小写字母或数字开头。• 名称长度限制在1-255字符之间。• 主题名称为主题的唯一标识，一旦创建后不能再修改主题名称。

参数	说明
显示名	显示名，长度限制在192字节或64个中文字。 说明 推送邮件消息时，如果未设置主题的显示名，发件人呈现为“username@example.com”，如果已设置主题的显示名，发件人则呈现为“显示名<username@example.com>”。
企业项目	企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。
标签	标签由键值对组成，用于标识云资源，可对云资源进行分类和搜索。 <ul style="list-style-type: none">键的长度最大36字符，值的长度最大43字符，只能包含数字、字母、“-”、“_”。

- 单击“确定”，主题创建成功。
主题创建成功后，系统会自动生成主题URN，主题URN是主题的唯一资源标识，不可修改。新创建的主题将显示在主题列表中。
- 单击主题名称，可查看主题详情（包含主题URN、显示名等）、主题订阅总数。

3.2 修改显示名

操作场景

您已经创建了一个主题，您现在可以修改显示名。

修改显示名

- 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
- 在左侧导航栏，选择“主题”。
进入主题页面。
- 在主题列表中，选择一个主题，单击右侧“操作”栏的“更多”，在下拉列表中选择“修改显示名”。在弹出的对话框中输入新的显示名。

说明

推送邮件消息时，如果未设置主题的显示名，发件人呈现为“username@example.com”，如果已设置主题的显示名，发件人则呈现为“显示名<username@example.com>”。

- 单击“确定”，修改成功。

3.3 设置主题策略

3.3.1 基本模式

只有和主题创建者处于同一个账号下的全部用户拥有权限通过此主题发送消息。通过主题策略的设置，您可授权或拒绝云服务或其他用户对主题资源进行特定类型操作，如查询主题详情、发布消息等。授权其他用户后，主题创建者依然可以操作该主题。

设置主题策略

1. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
2. 在左侧导航栏，选择“主题”。
进入主题页面。
3. 在主题列表中，选择一个主题，单击右侧“操作”栏的“更多”，在下拉列表中选择“设置主题策略”。
4. 在弹出的对话框中，设置访问策略为基本模式的主题策略。
基本模式只可对用户和服务设置发布消息的操作权限，详细说明请参见表3-2。

表 3-2 基本模式说明

策略对象	参数设置	说明
可发布消息的用户	仅自己（主题创建者）	仅和主题创建者处于同一个账号下的全部用户拥有权限通过此主题发送消息。
	所有人	所有用户均有权向该主题发布消息。
	仅如下用户	<p>设置可向该主题发布消息的用户。格式为 <code>urn:csp:iam::domainId:root</code>。其中，<code>domainId</code>即为用户的账号ID。</p> <p>您只需填写用户的“账号ID”，其余内容由系统自动补全，输入完成后，单击“确定”。可输入多个ID，数量没有限制，但生成的策略大小不超过30KB。</p> <p>“账号ID”的获取方式：登录消息通知服务控制台，单击右上角用户名下的“我的凭证”，可查看“账号ID”。</p>
可发布消息的服务	例如“OBS”。不同区域支持的“可发布消息的服务”不同，具体以管理控制台显示为准。	<p>勾选服务名称表示授权该云服务可对该主题进行操作。</p> <p>说明 默认情况云监控和Anti-DDoS拥有向所有用户的主题发布消息的权限。已获取授权的云服务如何使用SMN的详细信息请参见相应服务的用户指南。</p>

3.4 订阅主题

操作场景

要接收发布至主题的消息，您必须向该主题添加订阅者。

添加订阅

1. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
2. 在左侧导航栏，选择“主题”。
进入主题页面。
3. 在主题列表中，选择您要向其添加订阅者的主题，在右侧“操作”栏单击“添加订阅”。
此时将显示“添加订阅”对话框。
4. 在“协议”下拉框中选择您需要的协议，在“订阅终端”输入框中输入对应的订阅终端，填写说明如表3-3所示。

表 3-3 添加订阅参数说明

参数	说明
主题名称	在弹出的对话框中选择的主题的名称。
协议	在下拉框中选择需要订阅终端支持的协议，提供的选项为“邮件”、“HTTP”、“HTTPS”。
订阅终端	<p>订阅的终端地址，邮件、HTTP以及HTTPS终端支持批量输入，批量添加时，每个终端地址占一行。最多可输入10个终端。</p> <ul style="list-style-type: none"> • 选择“邮件”终端，请输入有效的邮箱地址。 例如： username@example.com username2@example.com • 选择“HTTP”终端，请输入公网网址。 例如： http://example.com/notification/action • 选择“HTTPS”终端，请输入公网网址。 例如： https://example.com/notification/action

5. 单击“确定”。
新增订阅将显示在页面下方的订阅列表中。

📖 说明

- 为了防止恶意攻击订阅终端，云平台会对订阅终端进行流控保护，触发流控规则后，消息通知服务不再发送订阅确认消息，具体的保护规则请参见《消息通知服务用户指南》的附录“订阅确认流控说明”部分。
- 消息通知服务不会校验订阅终端是否真实存在。
- 添加订阅后，消息通知服务会向订阅终端发送订阅确认信息，信息中包含订阅确认的链接。订阅确认的链接在48小时内有效，用户需要及时在手机端、邮箱或其他协议终端确认订阅。
- 发送请求订阅消息也会作为发送条数进行计费。

3.5 发布主题消息

3.5.1 发布主题消息简介

消息通知服务提供了以下三种通用的方式。

- 发布文本格式的消息。
- 发布JSON格式的消息。
- 发布模板格式的消息。

一旦一条新的消息已经发布，消息通知服务试图将消息发布至每个已经向主题确认订阅的终端节点。

如果使用“HTTP”或“HTTPS”协议接收消息，用户的HTTP(S)要开通防火墙策略，允许SMN访问，SMN通过公网发送消息到HTTP(S)终端节点。消息通知服务会自动组装消息，用户接收到整条消息由消息头和消息体组成，具体参数含义请参考[HTTP\(S\)消息格式](#)。

3.5.2 向主题发布文本消息

操作场景

文本消息是指将消息内容以文本的形式发送到订阅终端。

前提条件

所有的订阅终端必须确认订阅才能收到推送消息，未确认的订阅终端不会收到消息。

操作步骤

1. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
2. 在左侧导航栏，选择“主题”。
进入主题页面。
3. 在主题列表中，选择您要向其发布消息的主题，在右侧“操作”栏单击“发布消息”。
4. 在弹出的发布消息对话框中，主题名称为已选择的需要发布消息的主题名称，不可修改。
其它参数根据界面提示填写，填写说明如[表3-4](#)所示。

表 3-4 发布消息参数说明

参数	说明
消息标题	消息的标题，长度必须少于512字节。可不填写。
消息类型	<p>消息格式有三种，分别为“文本消息”、“JSON消息”和“模板消息”。每次发布消息只可选取其中一种。</p> <ul style="list-style-type: none"> “文本消息”：按文本格式发送的消息。 “JSON消息”：按JSON格式发送的消息。 “模板消息”：按模板格式发送的消息。模板定义请参考消息模板管理。
消息内容	消息的内容。消息内容不能为空且大小不超过256KB。

- 单击“确定”。

消息将推送至不同的订阅终端。各个终端接收到的消息内容说明请参考[不同协议消息说明](#)。

3.5.3 向主题发布 JSON 消息

操作场景

如果您需要向不同的订阅协议发送不同的内容，可以向主题发布JSON消息，它可以定义分别向邮件和HTTP(S)订阅协议发送不同内容。

前提条件

所有的订阅终端必须确认订阅才能收到推送消息，未确认的订阅终端不会收到消息。

操作步骤

- 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
- 在左侧导航栏，选择“主题”。
进入主题页面。
- 在主题列表中，选择您要向其发布消息的主题，在右侧“操作”栏单击“发布消息”。
- 在弹出的发布消息对话框中，主题名称为已选择的需要发布消息的主题名称，不可修改。其它参数根据界面提示填写。填写说明请参考[表3-5](#)。

表 3-5 发布消息参数说明

参数	说明
消息标题	消息的标题，长度必须少于512字节。可不填写。

参数	说明
消息类型	消息格式有三种，分别为“文本消息”、“JSON消息”和“模板消息”。每次发布消息只可选取其中一种。 <ul style="list-style-type: none">“文本消息”：按文本格式发送的消息。“JSON消息”：按JSON格式发送的消息。“模板消息”：按模板格式发送的消息。模板定义请参考消息模板管理。
消息内容	消息的内容。消息内容不能为空且大小不超过256KB。

此处“消息类型”选择“JSON消息”，“消息内容”可以手动输入JSON格式的消息，也可以通过工具自动生成JSON格式的消息。

- 手动输入JSON格式的消息，JSON格式的规范请参考[JSON格式消息说明](#)。
- 通过工具自动生成JSON格式的消息，请参考步骤5~8。

5. 单击“生成JSON消息”。
6. 在弹出的对话框中，在“消息”框中输入内容“This is a default message.”，在“协议”处勾选所有协议。

选择不同的协议，生成的JSON格式消息的大小不同。此处输入消息内容后，系统会自动计算已输入内容的字节数、生成的JSON格式消息的字节数以及剩余可输入的字节数。生成的JSON格式消息的字节数包含消息框中显示的花括号、双引号、空格、回车、协议名称和消息内容累计字节数。JSON格式消息的计算方法，请参见[JSON格式消息说明](#)的[JSON消息计算](#)部分。

7. 单击“确定”，系统自动生成JSON消息并显示在消息内容框中。
8. 您可以在生成的各个协议的消息内容中手动修改消息，这样就可以实现向不同协议发送不同的消息内容。系统生成JSON格式的消息时默认生成“default”，当匹配不到消息协议时，按“default”中的内容发送。关于JSON格式的消息描述，请参考[JSON格式消息说明](#)。
9. 单击“确定”。

消息将推送至不同的订阅终端。各个终端接收到的消息内容说明请参考[不同协议消息说明](#)。

3.5.4 向主题发布模板消息

操作场景

消息模板指消息的固定格式。如果您要发布的消息内容丰富，并且这条消息需要多次发布，每次发布时需要变更的内容极少，此时您可以套用已创建的消息模板向订阅者发送消息。

在相同的消息模板名称下面，可创建一个或多个不同协议类型的消息模板来适配不同协议的订阅者，但必须包含一个Default协议类型的模板。不同协议类型的消息模板可以设置不同的模板内容，使用{tag}占位符代替变量。如果没有预置的Default协议的模板，将不允许发送该模板。

使用模板格式发送消息时，需要选择模板名称，名称选定后，系统会按照Default、Email、HTTP、HTTPS的协议顺序，将该模板名称下不同协议的所有模板内容中的{tag}标签列出，重复的{tag}只列出一，并在右侧显示拥有该{tag}的协议名称。设


置{tag}值之后，消息通知服务会自动替换模板的{tag}标签为对应的参数值。如果{tag}没有设置参数值，模板内容不做替换。消息通知服务会按照模板名称和订阅者协议类型获取模板内容，如果对应订阅协议的模板内容不存在，将使用Default模板内容发送。

在这一部分中，您可以按照模板格式进行消息推送至您设定的订阅终端地址。在发送模板消息之前，您需要先创建好消息模板，关于消息模板的详细信息请参见[消息模板管理](#)。


前提条件

所有的订阅终端必须确认订阅才能收到推送消息，未确认的订阅终端不会收到消息。

创建模板


1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“消息模板”。
5. 请参见[消息模板管理](#)的[创建模板](#)步骤创建消息模板。
例如使用以下信息创建模板
 - 模板名称：“tem_001”
 - 协议：“Default”
 - 模板内容：“The Arts and Crafts Exposition will be held from {startdate} through {enddate}. We sincerely invite you to join us.”

向主题发布模板消息

1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“主题”。
进入主题页面。
5. 在主题列表中，选择您要向其发布消息的主题，在右侧“操作”栏单击“发布消息”。
6. 在弹出的发布消息对话框中，主题名称为已选择的需要发布消息的主题名称，不可修改。其它参数根据界面提示填写。
此处“消息类型”选择“模板消息”，“消息内容”可以手动输入模板消息，也可以通过工具自动生成模板格式的消息。消息内容不能为空且大小不超过256KB。
 - 手动输入模板格式的消息，模板格式规范请参见《消息通知服务用户指南》的附录“模板格式消息说明”部分。
 - 通过工具自动生成模板格式的消息，请参见步骤7~10。
7. 单击“生成模板消息”。

8. 在弹出的对话框中选择模板“tem_001”并在“变量”页签中设置“值”。
系统自动将设置的值替换，其他内容则与模板内容保持一致。此处选择的模板名称下只有Default协议的模板，则该主题下所有已确认的订阅终端地址均可收到该模板消息。
9. 选择“预览”页签，单击“消息预览”，预览模板消息。
该条模板消息的消息内容为“The Arts and Crafts Exposition will be held from February 10 through February 21. We sincerely invite you to join us.”，
10. 单击“确定”。
生成的消息内容显示模板名称和所有{tag}标签。
11. 单击“确定”。
消息将推送至不同的订阅终端。各个终端接收到的消息内容说明请参考《消息通知服务用户指南》的附录“不同协议消息说明”部分。

3.6 删除主题

1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“主题”。
进入主题页面。
5. 在主题列表中，选择一个主题，单击右侧“操作”栏的“更多”，在下拉列表中选择“删除”。
6. 单击“确定”，完成删除。

说明

删除主题时，将会删除该主题的所有订阅。

4 订阅管理

4.1 添加订阅

操作场景

订阅是您将订阅者注册到主题的操作。要接收发布至主题的消息，您必须订阅一个订阅终端到该主题。终端节点可以是邮箱地址或HTTP(S)终端。为终端节点订阅主题且确认订阅后，终端节点能够接收到向该主题发布的所有消息。

您可以拥有多个主题，每个主题有多个订阅者。当您需要向创建的主题添加订阅者操作，或者向已获取授权的主题添加订阅者时，可参考本章节。

添加订阅


1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“订阅”。
5. 在订阅页面。单击“添加订阅”，开始添加消息订阅。
此时将显示“添加订阅”页面。
6. 向指定主题添加订阅。
 - a. 单击“主题名称”输入框右侧的“选择主题”，在弹出的选择主题对话框中选择主题。
 - b. 在“协议”下拉框中选择订阅终端支持的协议，在“订阅终端”输入框中输入对应的订阅终端，填写说明如表4-1所示。

表 4-1 添加订阅参数说明

参数	说明
主题名称	在弹出的对话框中选择的主题的名称。

参数	说明
协议	在下拉框中选择需要订阅终端支持的协议，提供的选项为“邮件”、“HTTP”、“HTTPS”。
订阅终端	订阅的终端地址，邮件、HTTP以及HTTPS终端支持批量输入，批量添加时，每个终端地址占一行。最多可输入10个终端。 <ul style="list-style-type: none">选择“邮件”终端，请输入有效的邮箱地址。 例如： username@example.com username2@example.com选择“HTTP”终端，请输入公网网址。 例如： http://example.com/notification/action选择“HTTPS”终端，请输入公网网址。 例如： https://example.com/notification/action

7. 单击“确定”。

新增订阅将显示在页面下方的订阅列表中。

说明


- 为了防止恶意攻击订阅终端，云平台会对订阅终端进行流控保护，触发流控规则后，消息通知服务不再发送订阅确认消息，具体的保护规则请参见《消息通知服务用户指南》的附录“订阅确认流控说明”部分。
- 消息通知服务不会校验订阅终端是否真实存在。
- 添加订阅后，消息通知服务会向订阅终端发送订阅确认信息，信息中包含订阅确认的链接。订阅确认的链接在48小时内有效，用户需要及时在手机端、邮箱或其他协议终端确认订阅。
- 发送请求订阅消息也会作为发送条数进行计费。

4.2 请求订阅

操作场景

如果订阅者未收到订阅请求消息，用户可以再次请求订阅，请求订阅的流控规则请见[订阅确认流控说明](#)。可一次向一个或多个订阅终端发送确认请求。

请求订阅

- 登录管理控制台。
- 在管理控制台左上角单击  图标，选择区域和项目。
- 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
- 在左侧导航栏，选择“订阅”。

5. 在订阅列表中，选择一个或多个“状态”为“未确认”的订阅。
6. 单击订阅列表上方的“请求订阅”，确定发送该订阅的确认请求。
7. 获取请求订阅消息，确认订阅。

说明

- 为了防止恶意攻击订阅终端，云平台会对订阅终端进行流控保护，触发流控规则后，消息通知服务不再发送订阅确认消息，具体的保护规则请参见[订阅确认流控说明](#)。
- 消息通知服务不会校验订阅终端是否真实存在。
- 添加订阅后，消息通知服务会向订阅终端发送订阅确认信息，信息中包含订阅确认的链接。订阅确认的链接在48小时内有效，用户需要及时在手机端、邮箱或其他协议终端确认订阅。
- 发送请求订阅消息也会作为发送条数进行计费。

4.3 取消订阅

操作场景

订阅者订阅了主题之后，会收到系统发出的确认请求消息。只有订阅者确认订阅才能收到消息。当订阅者不需要收到消息时，可取消订阅。

注意

目前SMN服务的订阅管理能力对终端订阅者开放，终端订阅者必须保护自己的订阅链接，避免被其他人窃取后无法收到或收到非本人期望得到的通知信息。

取消订阅

不同的订阅协议对应不同的取消订阅方式：


- 邮件取消订阅：消息通知服务向指定邮箱地址发送电子邮件消息，其中包含取消订阅的链接，订阅者单击进入取消订阅的页面。如果取消成功，页面会显示取消订阅成功，稍后消息通知服务会向这个订阅者发送一封确认取消订阅的邮件(48小时内有效)。如果误操作了取消订阅，可以通过该消息重新订阅。
- HTTP(S)取消订阅，消息通知服务向指定的URL发送消息。消息体中包含取消订阅的链接，订阅者需要访问该链接，如果取消订阅成功，将获取HTTPS协议的200返回码，稍后消息通知服务会向这个订阅者发送一次取消订阅的消息(48小时内有效)。如果误操作了取消订阅，可以通过访问重新订阅的链接重新订阅。HTTP(S)消息头和消息体参数含义请参见[HTTP\(S\)终端节点使用样例简介](#)。

4.4 删除订阅


操作场景

当消息订阅者无需接收某主题推送的消息时，您可删除该订阅终端。支持批量删除。

从主题详情页面删除订阅

1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“主题”。
进入主题页面。
5. 在主题列表中，选择一个主题，单击主题名称，进入主题详情页面。
6. 在主题详情页面下方区域的“订阅总数”，查看该主题的订阅列表。
7. 选择一个或者多个订阅终端，单击订阅列表上方的“删除”。
8. 在弹出框中单击“确定”，完成订阅终端的删除。

从订阅页面删除订阅

1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“订阅”。
进入订阅页面。
5. 在订阅列表中，选择一个或多个订阅，单击订阅列表上方的“删除”。
6. 在弹出框中单击“确定”，完成订阅终端的删除。

5 消息模板管理

操作场景

消息模板指消息的固定格式，发布消息时可以使用已创建的消息模板向订阅者发送消息。使用模板发送消息，发送时会自动替换模板变量为对应的参数值。

消息模板通过消息名称进行分组，消息名称下面可以根据不同的协议创建不同的模板。每个模板名称下面都必须创建一个Default模板。当按照模板格式推送消息时，不同协议订阅者优先会选择模板名称下面对应的协议模板，如果对应的协议模板不存在，则采用默认Default的模板。如果没有预置的Default协议的模板，将不允许发送该模板。

当您需要发送模板格式的消息时，可参考本章节管理消息模板。

创建模板


1. 登录管理控制台。
2. 在管理控制台左上角单击  图标，选择区域和项目。
3. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。
4. 在左侧导航栏，选择“消息模板”。
5. 在消息模板页面。单击“创建消息模板”，开始创建消息模板。
此时将显示“创建消息模板”页面。
6. 在“模板名称”框中，输入模板名称，选择所使用的“协议”，在“模板内容”框中输入模板内容，参数说明如表5-1所示。

表 5-1 创建消息模板参数说明

参数	说明
模板名称	创建的模板名称，用户可自定义名称，规范如下： <ul style="list-style-type: none">• 只能包含字母，数字，短横线(-)和下划线(_)，只能以字母或数字开头。• 名称长度限制在1-64字节之间。• 一旦创建后不能再修改。

参数	说明
协议	设置选择该模板推送消息时的接收终端类型，模板一旦选定不能再修改。 取值范围为：Default、HTTP、HTTPS和邮件。 协议默认为“Default”。如果用户不选择协议，则使用默认值。
模板内容	用户可自定义模板内容。 请以{xxx}作为占位符创建模板。在使用该模板发送消息时，{xxx}可以被替换成具体内容。{xxx}中的xxx只能包含大写字母、小写字母、数字、-、.和_，且必须由大写字母、小写字母或数字开头，不能超过21个字符。 模板内容具体规则如下： <ul style="list-style-type: none">● 模板内容仅支持纯文本格式。● 模板内容不能为空。● 模板内容大小至多为256KB。● 模板内容中允许最多256个（计算重复的）标签，去重后不超过90个。● 发送消息时，每个标签允许的值大小为1KB。

例如使用以下信息创建模板

- 模板名称：“tem_001”
- 协议：“Default”
- 模板内容：“The Arts and Crafts Exposition will be held from {startdate} through {enddate}. We sincerely invite you to join us.”

7. 单击“确定”。

新创建的模板会显示在页面下方模板列表中。

修改模板

1. 在消息模板页面模板列表中，选择需要修改的消息模板名称。
2. 单击右侧“操作”栏的“修改”，修改“模板内容”。

删除模板

1. 在消息模板页面模板列表中，选择需要删除的消息模板名称。
2. 单击右侧“操作”栏的“删除”，确认删除该消息模板。

6 记录消息通知服务

6.1 记录消息通知服务简介

通过云审计服务，您可以记录与消息通知服务相关的操作事件（例如发出了何种请求、发出请求的源IP地址、何人发出的请求以及发出请求的时间），便于日后的查询、审计和回溯。

云审计记录的操作类型有三种，通过云平台账户登录管理控制台执行的操作，通过云服务支持的API执行的操作，以及系统内部触发的操作。

6.2 支持审计的关键操作列表

在您的账户中启用云审计服务后，当SMN服务的API被调用时，日志文件将跟踪记录该操作，并依据时间和数据的变化将日志文件转储到对象存储桶。

如果您在没有登录的情况下调用SMN的取消订阅（Unsubscribe）操作，云审计则不会记录这些操作。例如，当您需要取消订阅时，在电子邮件通知中单击取消订阅的链接，此时会在未登录的情况下调用Unsubscribe，这种情况下您的取消订阅操作则不会被云审计记录。

云审计支持的消息通知服务操作列表如表6-1所示。

表 6-1 云审计支持的 SMN 操作列表

操作名称	资源类型	事件名称
创建主题	topic	createTopic
删除主题	topic	deleteTopic
更新主题	topic	updateTopic
更新主题策略	topic	updateTopicAttribute
删除所有主题策略	topic	deleteTopicAttributes
删除指定名称的主题策略	topic	deleteTopicAttributeByName

操作名称	资源类型	事件名称
添加订阅	subscription	subscribe
删除订阅	subscription	delsubscribe
创建消息模板	message_template	createMessageTemplate
批量创建消息模板	message_template	batchCreateMessageTemplate
更新消息模板	message_template	updateMessageTemplate
删除消息模板	message_template	deleteMessageTemplate


6.3 查看审计日志

操作场景

在您开启了云审计服务后，系统开始记录云服务资源的操作。云审计服务管理控制台保存最近7天的操作记录。

本节介绍如何在云审计服务管理控制台查看最近7天的操作记录。

操作步骤

1. 单击“服务列表”，选择“管理与部署 > 云审计服务”，进入云审计服务信息页面。
2. 单击左侧导航树的“事件列表”，进入事件列表信息页面。
3. 事件列表支持通过筛选来查询对应的操作事件。当前事件列表支持四个维度的组合查询，详细信息如下：
 - 事件类型、事件来源、资源类型和筛选类型。
在下拉框中选择查询条件。
其中筛选类型按资源ID筛选时，还需手动输入某个具体的资源ID。
 - 操作用户：在下拉框中选择一个或多个具体的操作用户。
 - 事件级别：可选项为“所有事件级别”、“Normal”、“Warning”、“Incident”，只可选择其中一项。
 - 时间范围：可选择查询最近七天内任意时间段的操作事件。
4. 在需要查看的记录左侧，单击  展开该记录的详细信息。
5. 在需要查看的记录右侧，单击“查看事件”，弹出的窗口显示该操作事件结构的详细信息。

日志记录条目

云审计中每个日志条目都由一个JSON格式的事件组成。一个日志条目表示一条SMN接口请求，内容主要包括所请求的操作、操作的时间和日期、所操作的参数以及生成该请求的用户信息，其中用户信息来自统一身份认证服务。

以下示例显示了CreateTopic、DeleteTopic和UpdateTopic操作的CloudTrace记录条目。


```
{
  "time": "2017-02-15 14:21:50 GMT+08:00",
  "user": "xxx",
  "request": "xxx",
  "response": "xxx",
  "code": 200,
  "service_type": "SMN",
  "resource_type": "topic",
  "resource_id": "topicUrn instance",
  "source_ip": "127.0.0.1",
  "trace_name": "createTopic",
  "trace_rating": "normal",
  "trace_type": "ApiCall",
  "api_version": "2.0",
  "project_id": "tenantId instance",
  "record_time": "2017-02-15 14:21:50 GMT+08:00",
  "trace_id": "xxx"
}

{
  "time": "2017-02-15 14:12:15 GMT+08:00",
  "user": "xxx",
  "response": "xxx",
  "code": 200,
  "service_type": "SMN",
  "resource_type": "topic",
  "resource_id": "topicUrn instance",
  "source_ip": "127.0.0.1",
  "trace_name": "deleteTopic",
  "trace_rating": "normal",
  "trace_type": "ApiCall",
  "api_version": "2.0",
  "project_id": "tenantId instance",
  "record_time": "2017-02-15 14:12:15 GMT+08:00",
  "trace_id": "xxx"
}


{
  "time": "2017-02-13 15:38:30 GMT+08:00",
  "user": "xxx",
  "request": "xxx",
  "response": "xxx",
  "code": 200,
  "service_type": "SMN",
  "resource_type": "topic",
  "resource_id": "topicUrn instance",
  "source_ip": "127.0.0.1",
  "trace_name": "updateTopic",
  "trace_rating": "normal",
  "trace_type": "ApiCall",
  "api_version": "2.0",
  "project_id": "tenantId instance",
  "record_time": "2017-02-13 15:38:30 GMT+08:00",
  "trace_id": "xxx"
}
```

7 消息传输日志

操作场景


如果您想了解您发布至主题的消息的发送情况，您可以使用消息传输日志功能。该功能支持查看邮件、HTTP、HTTPS等协议的消息发送情况。配置消息传输日志时需要您对接云日志服务，并且已经创建需要关联的云日志组和日志流。

配置云日志

1. 在“云日志服务”页面创建日志组。
 - a. 登录管理控制台。
 - b. 在管理控制台左上角单击  图标，选择区域和项目。
 - c. 选择“管理与监管” > “云日志服务”。
进入云日志服务页面。
 - d. 在左侧导航栏，选择“日志管理”。
进入日志管理页面。
 - e. 单击“创建日志组”，在弹出框内，输入日志组名称。
 - f. 单击“确定”，创建完成。
2. 在“云日志服务”界面创建日志流。
 - a. 选择已创建的日志组名称，进入该日志组页面。
 - b. 单击“创建日志流”，在弹出框内，输入日志流名称。
 - c. 单击“确定”，创建完成。

配置消息传输日志

在“消息通知服务”页面配置消息传输日志。

1. 在创建主题页面配置。
 - a. 登录管理控制台。
 - b. 在管理控制台左上角单击  图标，选择区域和项目。
 - c. 选择“应用服务” > “消息通知服务”。
进入消息通知服务页面。

- d. 在左侧导航栏，选择“主题管理” > “主题”。
进入主题页面。
- e. 在主题页面，单击需要配置的主题名称。
- f. 在该主题详情页面的“消息传输日志”页签，单击“配置访问日志”。
- g. 单击“启动日志记录”，选择您在“云日志服务”中创建的云日志组和日志流。
- h. 单击“确定”，配置完成。

查看访问日志

当您配置了消息传输日志，可以查看消息传输日志的详细信息。

日志显示格式如下：

```
{"message_id":"$message_id","project_id":"$project_id","topic_urn":"$topic_urn","subscriber_urn":"$subscriber_urn","protocol_name":"$protocol_name","endpoint":"$endpoint","status":"$status","http_code":$http_code,"create_time":"$create_time","send_time":"$send_time"}
```

不支持修改日志格式。日志字段说明如表1所示。

表 7-1 字段说明

参数	参数类型	描述
message_id	String	消息ID。
project_id	String	项目ID。
topic_urn	String	Topic的唯一的资源标识
subscriber_urn	String	订阅者的唯一资源标识。
protocol_name	String	不同协议对应不同的endpoint（接受消息的接入点）。目前支持的协议包括： “email”：邮件传输协议，endpoint为邮箱地址。 “sms”：短信传输协议，endpoint为手机号码。 “functiongraph” FunctionGraph（函数）传输协议，endpoint为一个函数 “functionstage”： FunctionStage（工作流）传输协议，endpoint为一个函数工作流 “http”、“https”：HTTP/HTTPS传输协议，endpoint为URL。
endpoint	String	接受消息的接入点。

参数	参数类型	描述
status	String	消息状态。目前包括以下状态： “DELIVERED”：已送达。 “FAIL_DELIVERED”：发送失败。 “REJECTS”：已拒绝。触发流控制机制。
http_code	Integer	HTTP返回码，仅支持HTTP/HTTPS协议消息。
create_time	String	消息创建时间。时间格式为UTC时间，YYYY-MM-DDTHH:MM:SSZ。
send_time	String	消息发送时间。时间格式为UTC时间，YYYY-MM-DDTHH:MM:SSZ。

日志示例

```
{ "message_id": "1ae49922602a42fc83acb9689a2eb5f4", "project_id": "5a9f32e4f1ec4bbe9695ff9da51c2925", "topic_urn": "urn:smn:regionid:5a9f32e4f1ec4bbe9695ff9da51c2925:demo", "subscriber_urn": "urn:smn:regionid:5a9f32e4f1ec4bbe9695ff9da51c2925:demo:b55c3c6fa7cd471b9f24818d530a8740", "protocol_name": "https", "endpoint": "https://127.0.0.1:443/https", "status": "DELIVERED", "http_code": 200, "create_time": "2022-11-01T00:00:00Z", "send_time": "2022-11-01T00:00:10Z" }
```

以上日志示例对应的字段如下：

表 7-2 日志示例对应的字段

参数	示例
message_id	1ae49922602a42fc83acb9689a2eb5f4
project_id	5a9f32e4f1ec4bbe9695ff9da51c2925
topic_urn	urn:smn:regionid:5a9f32e4f1ec4bbe9695ff9da51c2925:demo
subscriber_urn	urn:smn:regionid:5a9f32e4f1ec4bbe9695ff9da51c2925:demo:b55c3c6fa7cd471b9f24818d530a8740
protocol_name	https
endpoint	https://127.0.0.1:443/https
status	DELIVERED
http_code	200
create_time	2022-11-01T00:00:00Z
send_time	2022-11-01T00:00:10Z

8 权限管理

8.1 创建用户并授权使用 SMN

如果您需要对您所拥有的SMN进行精细的权限管理，您可以使用[统一身份认证服务](#)（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用SMN资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将SMN资源委托给更专业、高效的其他或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用SMN服务的其它功能。

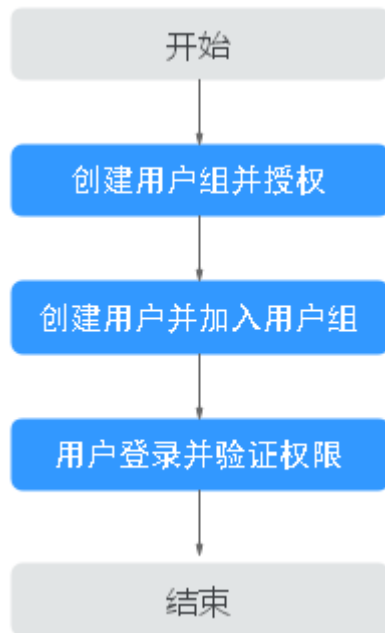
本章节为您介绍对用户授权的方法，操作流程如[图8-1](#)所示。

前提条件

给用户组授权之前，请您了解用户组可以添加的SMN权限，并结合实际需求进行选择，SMN支持的系统权限，请参见：[。](#)如果您需要对除SMN之外的其它服务授权，IAM支持服务的所有权限请参见[系统权限](#)。

示例流程

图 8-1 给用户授予 SMN 权限流程



1. **创建用户组并授权**

在IAM控制台创建用户组，并授予消息通知服务权限“SMN Administrator”。

2. **创建用户并加入用户组**

在IAM控制台创建用户，并将其加入1中创建的用户组。

3. **用户登录**并验证权限

新创建的用户登录控制台，切换至授权区域，验证权限：

- 在“服务列表”中选择消息通知服务，进入SMN的“总览>主题”页面，单击右上角“创建主题”，尝试创建主题，如果创建主题成功，表示“SMN Administrator”已生效。
- 在“服务列表”中选择除消息通知服务外（假设当前权限仅包含SMN Administrator）的任一服务，如果提示权限不足，表示“SMN Administrator”已生效。



9 关于配额

什么是配额？

为防止资源滥用，平台限定了各服务资源的配额，对用户的资源数量和容量做了限制。如您最多可以创建多少台弹性云服务器、多少块云硬盘。

如果当前资源配额限制无法满足使用需要，您可以申请扩大配额。

怎样查看我的配额？

1. 登录管理控制台。
2. 单击管理控制台左上角的 ，选择区域和项目。
3. 单击页面右上角的“My Quota”图标 。
系统进入“服务配额”页面。
4. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。
如果当前配额不能满足业务要求，请参考后续操作，申请扩大配额。

如何申请扩大配额？

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。
系统进入“服务配额”页面。
3. 在页面右上角，单击“申请扩大配额”。
4. 在“新建工单”页面，根据您的需求，填写相关参数。
其中，“问题描述”项请填写需要调整的内容和申请原因。
5. 填写完毕后，勾选协议并单击“提交”。

10 常见问题

10.1 消息通知服务有哪些优势？

消息通知服务的突出优势有以下三点：

- 消息通知体系需要大量投入资源进行开发以及维护，使用SMN服务可以大幅降低成本。
- 无需考虑消息通知体系的可靠性以及扩容问题。
- 消息通知服务接入简单、学习成本低。

10.2 消息通知服务支持的传输协议有哪些？

消息通知服务支持的协议有“邮件”、“HTTP”和“HTTPS”等。

- “邮件”：消息以电子邮件形式发送到订阅的邮箱地址。
- “HTTP”、“HTTPS”：消息以HTTP/HTTPS请求的形式发送到订阅的URL，目前只支持公网域名或者公网IP地址。

10.3 消息通知服务的主题名称有何格式要求？

- 主题名称不能超过255个字符，允许使用字母数字字符以及连字符（-）和下划线（_）。
- 一个账户中的主题名称必须唯一，删除主题后，可以重新使用该主题名称。
- 创建主题后，主题名称为主题的唯一标识，一旦创建后不能再修改主题名称。SMN将为该主题分配唯一的URN，其中包括服务名称（SMN）、区域名称、项目ID和主题名称。URN将作为创建主题时使用的API调用的一部分返回。无论何时发布者或者订阅者需要对该主题执行任何操作，都应引用该唯一主题URN。

10.4 每个用户可以创建多少个主题？

单用户默认可创建3000个主题。

10.5 每个主题最多支持多少个订阅者？

单主题默认可添加10000个订阅者。

10.6 可以向主题发布多少条消息？

向主题发布消息没有限量。

10.7 每个用户可以创建多少个消息模板？

单用户默认可创建100个消息模板。

10.8 单个主题是否支持通过多种传输协议进行订阅？

单个主题支持多种传输协议定义，目前支持HTTP/HTTPS、邮件传输协议。

10.9 主题创建者是否可以控制所创建的主题中允许的传输协议？

暂不支持主题传输协议的设置。

10.10 订阅终端地址可以修改吗？

订阅终端地址不允许修改。如果需要更换某个主题下的订阅终端地址，可删除原有的订阅终端，重新在该主题下选择订阅协议并输入对应的订阅终端。

10.11 是否可删除已发布的消息？

已发布的消息不能删除。

10.12 是否会保证将消息传送到订阅的终端节点？

SMN向终端节点推送消息时，使用异步方式送达，不保证消息送达的时效性。如果业务要求消息准实时送达，请慎重考虑是否使用SMN服务。

如果订阅终端节点可以送达，消息通知服务会确保消息传送到订阅终端节点。

如果订阅终端节点无法送达，消息通知服务会将发送失败的消息保存到失败的消息队列尾部，等待下次被发送。如果发送失败次数超过6次，消息将直接丢弃，不再保存到失败的消息队列，且不会给用户发送“消息发送失败”的相关信息。

失败的消息队列中的消息处理间隔时间是不确定的，它取决于失败的消息队列的长度。一般会在数小时之内处理完成，如果失败消息队列中的内容太多，则会在一天之内处理完成。

10.13 每条消息会让订阅者收到多少次？

每条消息只会让订阅者收到1次，如遇网络问题或设备故障可能会有重发的情况。

10.14 向主题推送消息后，订阅者为什么没有收到消息？

在排除终端设备本身的问题，比如信号不好。最可能原因是**您未完成确认订阅或信息被拦截**。

优先排查信息是否被拦截，查看您的邮件是否被处理为垃圾信息。

如果仍然不能解决您的问题，请联系技术支持。

邮件协议

优先排查垃圾邮件，查看您的邮件是否被处理为垃圾邮件。

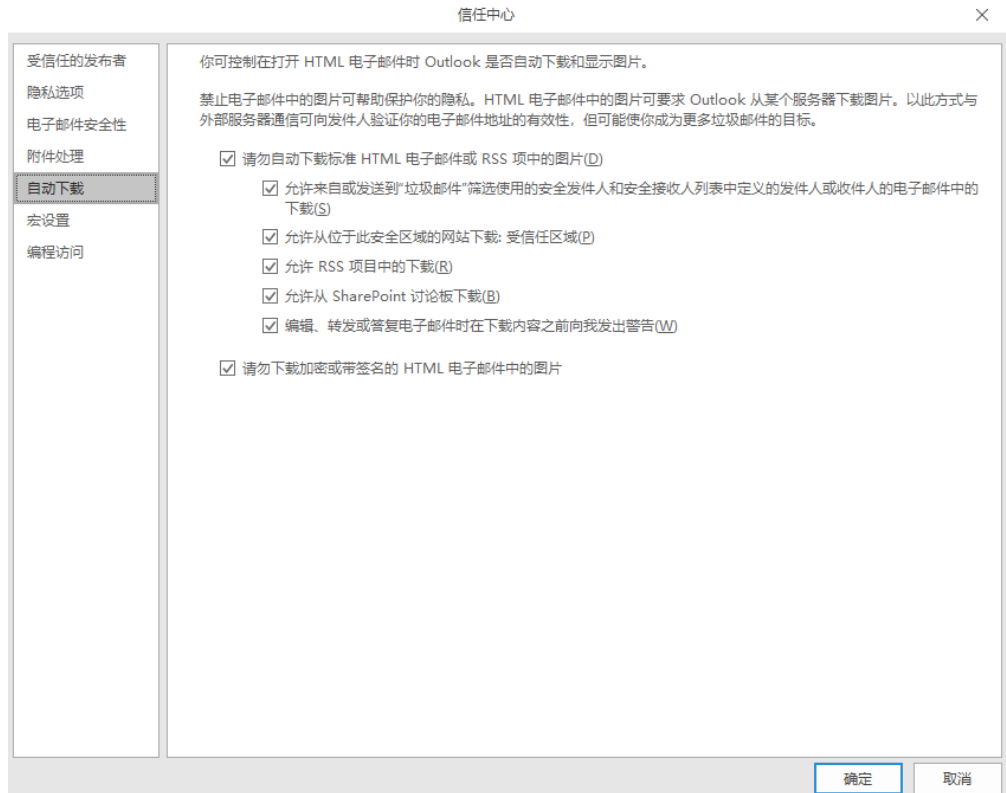
因容灾，流量调整等原因，邮箱发送地址动态变动，无法提供具体邮箱发送地址。

10.15 为什么用户接收到的邮件中图片无法显示？

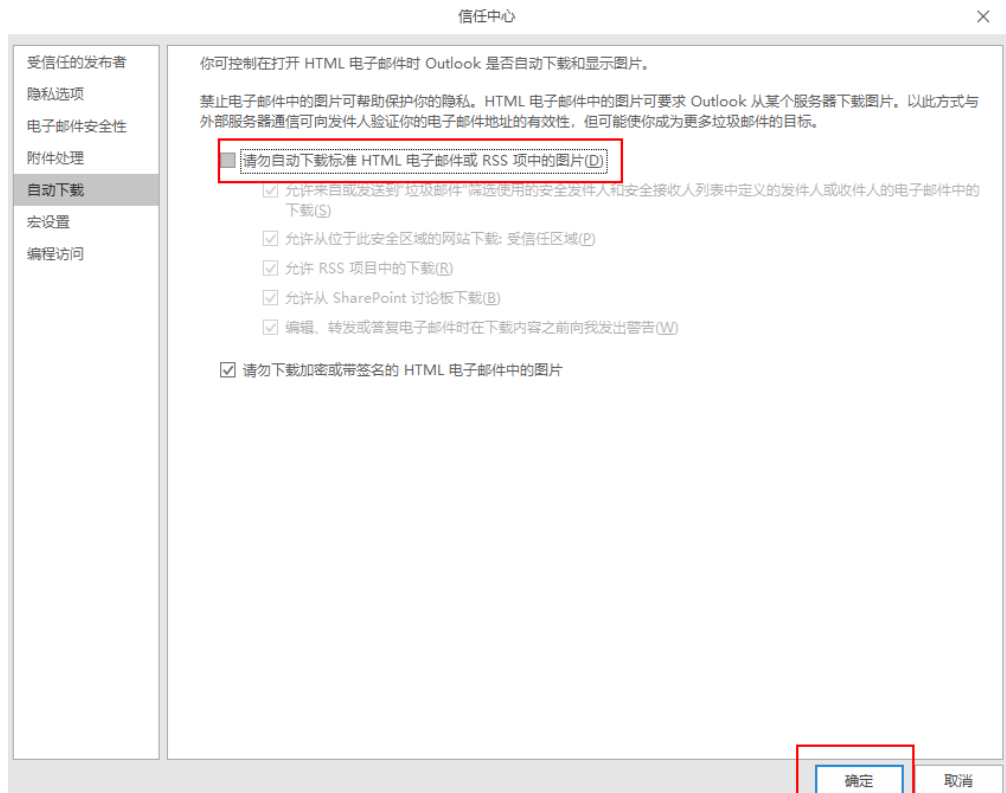
部分邮箱为帮助保护您的隐私，默认禁止自动从Internet下载图片，导致邮件中图片无法显示，您需要在您的邮箱中设置允许自动下载图片即可正常显示图片。

以邮箱Outlook2019版本举例如何设置允许自动下载图片功能。具体步骤如下：

1. 打开Outlook，单击左上角的“文件”。
2. 找到“选项”，单击“选项”，打开选项界面。
3. 找到“信任中心”，单击“信任中心设置”。
进入“信任中心”界面。



4. 在“自动下载”页面，取消勾选“请勿自动下载标准HTML电子邮件或RSS项中的图片”，单击“确定”。



 说明

其它邮箱请自行在浏览器中搜索“如何设置允许自动下载图片功能”。

10.16 如何获取用户的账号 ID?

1. 登录管理控制台。
2. 单击右上角用户名下的“我的凭证”。
3. 在用户信息中获取用户的“账号ID”

A 附录

A.1 JSON 格式消息说明

JSON 格式体说明

JSON格式主要用来处理用户想对不同的订阅者类型发送不同的消息，目前支持的协议Default、HTTP、HTTPS、email，其中Default是必须的，Email的内容将发给邮件类型的订阅者，其它协议依次如下所示：

```
{
  "default": "Dear Sir or Madam, this is a default message.",
  "email": "Dear Sir or Madam, this is an email message.",
  "http": "{message:'Dear Sir or Madam, this is an HTTP message.'}",
  "https": "{message:'Dear Sir or Madam, this is an HTTPS message.'}",
}
```

推荐将Default设置为通用的消息内容，对于特殊的订阅者类型再设置相应消息内容。

JSON 格式约束

- 满足JSON格式要求。
- JSON格式中必须有Default协议。
- 整个JSON消息的大小不超过256KB。

JSON 消息计算

选择不同的协议，生成的JSON格式消息的大小不同。JSON格式消息的字节数包含消息框中显示的花括号、双引号、空格、回车、协议名称和消息内容累计字节数。具体字节计算方式以输入消息“This is a default message.”为例。

消息：“This is a default message.”，消息本身包含26个字节。

输入消息，选择协议，生成JSON消息时，系统默认生成Default协议的消息。

```
{
  "default": "This is a default message.",
  "protocol1": "This is a default message.",
  "protocol2": "This is a default message.",
}
```

```
.....  
}
```

1个Default协议加上用户选择的协议个数，记为N。

消息体包含的固定字节数：

- 每种协议的消息包含3个空格，共N个协议： $3 \times N = 3N$
- 每种协议的消息包含4个双引号，共N个协议： $4 \times N = 4N$
- 每种协议的消息包含1个冒号，共N个协议： $1 \times N = N$
- 每种协议的消息包含1条消息内容This is a default message.，共N个协议： $26 \times N = 26N$
- 逗号 (N - 1) 个： $1 \times (N - 1) = (N - 1)$
- 回车 (N + 1) 个： $1 \times (N + 1) = (N + 1)$
- 花括号2个： $1 \times 2 = 2$
- 协议名称 default 1个： $7 \times 1 = 7$

选择的协议字节数：

- 协议名称 HTTP 1个： $4 \times 1 = 4$
- 协议名称 HTTPS 1个： $5 \times 1 = 5$
- 协议名称 email 1个： $5 \times 1 = 5$

合计： $36N + 9 + \text{选择的协议字节数}$

例如，您选择了“HTTP”、“HTTPS”和“email”3个协议，生成的消息如下：

```
{  
  "default": "This is a default message.",  
  "email": "This is a default message.",  
  "http": "This is a default message.",  
  "https": "This is a default message."  
}
```

加上Default，则N的值为4，生成的JSON消息字节数计算如下：

- 固定字节数： $36 \times 4 + 9 = 153$
- 协议名称 HTTP 1个： $4 \times 1 = 4$
- 协议名称 HTTPS 1个： $5 \times 1 = 5$
- 协议名称 email 1个： $5 \times 1 = 5$

合计： $153 + 4 + 5 + 5 = 167$

A.2 模板格式消息说明

模板格式消息指固定格式的消息，可以通过tag的方式来处理变量的部分。

通过模板发送消息，建议消息大小不超过256KB，模板格式消息如下所示：

```
{  
  "message_template_name": "confirm_message",  
  "tags": {  
    "topic_urn": "urn:smn:regionId:xxxx:SMN_01"  
  }  
}
```

表 A-1 模板格式消息参数说明

参数	说明
message_template_name	模板名称，不能为空。模板名称可以在消息模板列表中获取，模板名称下面必须有一个Default协议的模板，没有匹配上的模板采用Default模板。
tags	模板中出的占位符标签，模板名称下面有不同的模板，不同的模板可能tags不一样，用户在此处需要把所有的tag标签都填在tags内，他是一个jsonmap对象

A.3 不同协议消息说明

不同的订阅协议接收到的消息包含的内容有所不同。

- 邮件和HTTP(S)终端收到的消息包含：消息主题、消息内容和取消订阅的链接。

A.4 订阅确认流控说明

为了控制用户对订阅者进行恶意的攻击，云平台对添加订阅时发送订阅确认消息进行了流控。流控策略既适用于从控制台发送订阅确认消息也适用于通过调用API发送订阅确认消息。根据不同的协议有不同的流控策略：

- 对于一个邮件订阅者，1个小时最多允许发送20次订阅确认消息，2天最多允许发送40次订阅确认，如果超过这个阈值，将7天之内不对这个邮件订阅者发送订阅确认邮件。如果这个邮件订阅者对这个用户的确认信息进行了确认，之前的流控规则将清空置零。
- 对于一个HTTP(S)订阅者，10分钟最多允许发送200次订阅确认消息。

A.5 授权操作与 SMN 接口的映射关系

表 A-2 授权操作与 SMN 接口的映射关系

授权操作	接口名称	接口功能
SMN:UpdateTopic	UpdateTopic	修改主题的属性，目前仅支持修改topic的display_name字段。
SMN>DeleteTopic	DeleteTopic	删除一个主题以及它的所有订阅者。删除Topic操作可能会导致未推送的消息无法再推送给该Topic的订阅者。
SMN:QueryTopicDetail	QueryTopicDetail	查询Topic的详细信息。
SMN:ListTopicAttributes	ListTopicAttributes	查询Topic的属性信息。

授权操作	接口名称	接口功能
SMN:UpdateTopicAttribute	UpdateTopicAttribute	更新Topic的属性。
SMN>DeleteTopicAttributes	DeleteTopicAttributes	删除所有Topic属性。
SMN>DeleteTopicAttributeByName	DeleteTopicAttributeByName	删除指定名称的Topic属性。
SMN:ListSubscriptionsByTopic	ListSubscriptionsByTopic	分页获取特定主题的订阅列表，订阅列表按照订阅创建时间进行升序排列。分页查询可以指定offset以及limit。如果指定主题不存在订阅者，返回空列表。
SMN:Subscribe	Subscribe	为指定Topic添加一个订阅者，并向订阅者发送一个确认的消息。待订阅者进行ConfirmSubscription确认后，该订阅者才能收到Topic发布的消息。
SMN:Unsubscribe	Unsubscribe	删除一个订阅，订阅需要鉴权才能删除，仅订阅者或者这个主题的拥有者可以删除，删除时会对身份进行确认。
SMN:Publish	Publish	将消息发送给Topic的所有订阅端点。当返回消息ID时，该消息已被保存并开始尝试将其推送给主题的订阅者。消息格式，取决于该主题每一个订阅者的通知协议。

A.6 HTTP(S)终端节点使用样例

A.6.1 HTTP(S)终端节点使用样例简介

向该订阅终端发布消息时，消息通知服务自动组装消息，用户可以接收到的由消息通知服务发送至HTTP(S)终端的消息共分为两种，管理类消息和业务类消息。管理类消息即确认订阅消息和取消订阅消息，业务类消息即推送消息。用户接收到的整条消息包含两部分内容，分别是消息头和消息体。消息体和消息头具体的参数配置请参考本章节。

A.6.2 HTTP(S)消息格式

操作场景

注意

接收SMN服务发送的HTTP(S)消息时，终端证书的CN名称请参考业界规范，部分特殊字符可能导致https消息发送失败。

描述SMN服务向HTTP(S)终端发送消息的格式，您可以通过消息头部信息中的消息类型来区分消息格式。消息格式包括：HTTP(S)订阅确认消息的格式、HTTP(S)推送消息的格式和HTTP(S)取消订阅消息的格式三种消息格式，HTTP/HTTPS协议均为POST方式。

当SMN服务向HTTP(S)终端发送消息时，头部包括请求X-SMN-MESSAGE-TYPE, X-SMN-MESSAGE-ID, X-SMN-TOPIC-URN, X-SMN-SUBSCRIPTION-URN，如[表A-3](#)所示。

表 A-3 消息 HTTP 头部参数说明

参数	说明
X-SMN-MESSAGE-TYPE	消息类型，消息类型分别有： <ul style="list-style-type: none"> • SubscriptionConfirmation • Notification • UnsubscribeConfirmation
X-SMN-MESSAGE-ID	消息唯一标识。
X-SMN-TOPIC-URN	主题的唯一标识，表示消息归属的主题。
X-SMN-SUBSCRIPTION-URN	订阅终端的唯一标识。 只在HTTP(S)推送消息和HTTP(S)取消订阅消息时，设置此字段。

说明

关于HTTP头部字段的说明：

- RFC 7230 章节3.2中规定，HTTP头部的字段名称不区分大小写。
- RFC 7540 章节8.1.2中规定，对于支持HTTP/2的服务端，头部字段信息需要统一改为全小写。
- 对于用户自定义的HTTP头部字段，SMN也将采用上述的处理策略。
- 用户如果从HTTP头部中获取相关字段，建议采用不区分大小写的方式。

HTTP(S)订阅确认消息的格式

添加HTTP(S)订阅终端以后，SMN会向订阅终端推送一条订阅确认的消息，下面将描述消息的HTTP Content，它是JSON格式的字符串，其中包括subscribe_url值，订阅确认需要GET请求subscribe_url地址进行确认，具体JSON字段说明如[表A-4](#)所示。

表 A-4 HTTP(S)订阅确认消息的格式参数说明

参数	说明
type	消息类型，SubscriptionConfirmation。
signature	签名信息。 签名字段包括message，message_id，subscribe_url，timestamp，topic_urn和type，具体签名信息校验，请参见 校验消息签名 。
topic_urn	主题的唯一标识，表示消息归属的主题。
message_id	消息唯一标识。
signature_version	签名的版本信息，目前是V1。
message	订阅确认消息的描述。
subscribe_url	订阅确认需要访问的URL。
signing_cert_url	用于消息签名的证书URL，不需要鉴权，可直接访问。
timestamp	消息第一次发送的时间戳。

HTTP(S)订阅确认消息样例如下：

```
{
  "signature": "ViE96uGbBkl
+S8eWqgebi5KdmRht2U8+Rs88yuyMHq1k4h3jUkcDZ6HCqTqdpJ8nrLcdqETyyEiOQyTszJdU05z
+LhfE8jerCCdSbL4zeInVkydHh0kcCRWmORye0/EuQ/gLC1UIXwwUsqbUCPnBRhNFXOeXMOPPCzk
+d04xjy4QHd1H/bHxgsY3AlTe0gCFT068Zru7OK6w9aQaY44mXnN3OWGmBmoHyFab5TRXLSQNZ/9u/
Vj646cQMMal0PPQ30QzGYD0MtzgDZi12m8jMTHAnMkTEcbLaEgaqmaoEnATSpEcspFKNXv2skwk7rsVakMOI
SpMH3+qC6RzhETA2A==",
  "topic_urn": "urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic",
  "message_id": "d86c201092574e71a3ca85826652c58b",
  "signature_version": "v1",
  "type": "SubscriptionConfirmation",
  "message": "You are invited to subscribe to topic:
urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic. To confirm this
subscription, please visit the subscribe_url included in this message. The subscribe_url is valid only within 48
hours.",
  "subscribe_url": "https://console.xxx.com/smn/subscription/unsubscribe?
subscription_urn=urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic:653e212a
43884f7188ca656c537e31ce",
  "signing_cert_url": "https://console.xxx.com/smn/
SMN_region01_b3974c411807498bb532b3cd6cd65d91.pem",
  "timestamp": "2019-08-12T22:40:56Z"
}
```

HTTP(S)推送消息的格式

HTTP(S)订阅终端订阅确认后，终端将收到SMN推送的消息，下面将描述HTTP Content信息，它是JSON串，具体字段说明如[表A-5](#)所示。

表 A-5 HTTP(S)推送消息的格式参数说明

参数	说明
type	消息类型，Notification。

参数	说明
signature	签名信息。 签名字段包括message, message_id, subject, timestamp, topic_urn和type,具体签名信息校验, 如果subject不存在, 则不参与校验, 请参见 校验消息签名 。
subject	消息标题。
topic_urn	主题的唯一标识, 表示消息归属的主题。
message_id	消息唯一标识。
signature_version	签名的版本信息, 目前是V1。
message	订阅推送消息的描述。
unsubscribe_url	通过访问该链接可以取消终端订阅。
signing_cert_url	用于消息签名的证书URL。
timestamp	消息第一次发送的时间戳。

HTTP(S)推送消息样例如下:

```
{
  "signature": "ViE96uGbBkl
+58eWqgebi5KdmRht2U8+Rs88yuyMHq1k4h3jUkcDZ6HCqTqdpJ8nrLcdqETyyEiOQyTszJdU05z
+LhfE8jerCCdSbL4zeInVkydHh0kcCRWmORye0/EuQ/gLC1UIXwvUsqbUCPnBRhNFXOeXMOPPCzK
+d04xjy4QHd1H/bHxgsY3AlTe0gCFT068Zru7OK6w9aQaY44mXnN3OWGmBmoHyFab5TRXLSQNz/9u/
Vj646cQMMai0PPQ30QzGYD0MtzgDZi12m8jMTHAnMkTEcbLaEgaqmaoEnATSpEcspFKNXv2skwk7rsVakMOI
SpMH3+qC6RzhETA2A==",
  "topic_urn": "urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic",
  "message_id": "d86c201092574e71a3ca85826652c58b",
  "signature_version": "v1",
  "type": "Notification",
  "message": "{\"enterpriseProjectId\": \"0\", \"eventTime\": \"2019-08-12 22:40:55.040632\",
\\\"chargingMode\\\": \\\"postPaid\\\", \\\"cloudserviceType\\\": \\\"xxx.service.type.bandwidth\\\", \\\"eventType\\\": 1,
\\\"regionId\\\": \\\"region01\\\", \\\"tenantId\\\": \\\"057eefe55400d2742f8cc0017870ceef\\\", \\\"resourceType\\\":
\\\"xxx.resource.type.bandwidth\\\", \\\"resourceSpecCode\\\": \\\"19_bgp\\\", \\\"resourceSize\\\": 10, \\\"resourceId\\\":
\\\"e091f1b1-08ef-4e2b-a27e-f85e4c19026a\\\", \\\"resourceSizeMeasureId\\\": 15, \\\"resourceName\\\":
\\\"elbauto_2019_08_13_06_40_46\\\"}",
  "unsubscribe_url": "https://console.xxx.com/smn/subscription/unsubscribe?
subscription_urn=urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic:653e212a
43884f7188ca656c537e31ce",
  "signing_cert_url": "https://console.xxx.com/smn/
SMN_region01_b3974c411807498bb532b3cd6cd65d91.pem",
  "timestamp": "2019-08-12T22:40:56Z"
}
```

HTTP(S)取消订阅消息的格式

HTTP(S)订阅终端取消订阅确认后, 终端将收到SMN推送的取消订阅确认消息, 下面将描述HTTP Content信息, 它是JSON串, 具体字段说明如[表A-6](#)所示。

表 A-6 HTTP(S)取消订阅消息的格式参数说明

参数	说明
type	消息类型，UnsubscribeConfirmation。
signature	签名信息。 签名字段包括message, message_id, subscribe_url、timestamp, topic_urn和type,具体签名信息校验,请参见 校验消息签名 。
topic_urn	主题的唯一标识,表示消息归属的主题。
message_id	消息唯一标识。
signature_version	签名的版本信息,目前是V1。
message	取消订阅确认消息的描述。
subscribe_url	再次订阅确认需要访问的URL。
signing_cert_url	用于消息签名的证书URL。
timestamp	消息第一次发送的时间戳。

HTTP(S)取消订阅消息样例如下:

```
{
  "signature": "ViE96uGbBkl
+S8eWqgebi5KdmRht2U8+Rs88yuyMHq1k4h3jUkcDZ6HCqTqdpJ8nrLcdqETyyEiOQyTszJdU05z
+LhfE8jerCCdSbL4zeInVkydHh0kcCRWmORye0/EuQ/gLC1UIXwwUsqbUCPnBRhNFXOeXMOPPCzk
+d04xjy4QHd1H/bHxgsY3AlTe0gCFT068Zru7OK6w9aQaY44mXnN3OWGmBmoHyFab5TRXLSQNz/9u/
Vj646cQMMal0PPQ30QzGYD0MtgDZi12m8jMTHAnMkTEcbLaEgaqmaoEnATSpEcspFKNXv2skwk7rsVakMOI
SpMH3+qC6RzhETA2A==",
  "topic_urn": "urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic",
  "message_id": "d86c201092574e71a3ca85826652c58b",
  "signature_version": "v1",
  "type": "UnsubscribeConfirmation",
  "message": "You are unsubscribed from topic:
urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic. To subscribe to this topic
again, please visit the subscribe_url included in this message. The subscribe_url is valid only within 48
hours.",
  "subscribe_url": "https://console.xxx.com/smn/subscription/unsubscribe?
subscription_urn=urn:smn:region01:0553db98c800d5192f9bc01232b89622:vpc_status_report_topic:653e212a
43884f7188ca656c537e31ce",
  "signing_cert_url": "https://console.xxx.com/smn/
SMN_region01_b3974c411807498bb532b3cd6cd65d91.pem",
  "timestamp": "2019-08-12T22:40:56Z"
}
```

A.6.3 校验消息签名

操作场景

为了确保安全,提供订阅确认、取消订阅和消息的签名认证,在消息的接收方需要对SMN发送的消息进行校验,消息类型包括订阅确认消息、通知消息和取消订阅通知消息。使用SMN消息中的信息,您可以通过签名串验证消息的合法性。

操作步骤

验证SMN发送的HTTP(S)请求的流程如下：

1. SMN将根据您终端节点接收的HTTP消息，使用其中关键的键-值信息进行签名验证，具体的验证签名键信息请参考[特定消息类型的签名键值信息](#)。
2. 通过HTTP消息中的signing_cert_url获取证书内容的地址，通过该地址获取证书内容，并且生成X509证书。

说明

下载证书的链接始终是HTTPS请求，需要验证下载链接服务器的证书的真伪。

3. 从X509证书中提取公钥，用于验证收到消息的可靠性和完整性。
4. 确定消息类型。消息类型为键type对应的值，不同的消息类型对应不同的签名证书验证方式。
5. 创建签名字符串。将消息中的键值对取出来，使其键值按字典序排序。每一个键占一行，每一个值占一行，键所对应的值在该键所占行的下一行。

特定消息类型的签名键值信息

1. 消息通知

- 消息通知必须包括以下键，如果subject为空，则不参与签名：

```
message
message_id
subject
timestamp
topic_urn
type
```

- 一个消息通知使用的签名键值信息示例如下：

```
message
My test message
message_id
88c726942175432bac921eafd0036163
subject
demo
timestamp
2016-08-15T07:29:16Z
topic_urn
urn:smn:regionId:74dc9e44d0cc4573adfce91cdfdd3ba9:xxxx
type
Notification
```

说明

每一个键占一行，每一个值占一行，键所对应的值在该键所占行的下一行。

2. 订阅确认和取消订阅

- 订阅确认和取消订阅的消息中必须包含以下键：

```
message
message_id
subscribe_url
timestamp
topic_urn
type
```

- 一个订阅确认的消息简单示例如下所示：

```
message
You are invited to subscribe to topic:
urn:smn:regionId:d91989905b8449b896f3a4f0ad57222d:demo. To confirm this subscription,
please visit the following SubscribeURL in this message.
message_id
def5c309cbff44d5a870787ed937edf8
```

```
subscribe_url
https://ip地址/smn/subscription/confirm?region_id&token&topic_urn:demo
timestamp
2016-08-15T07:29:16Z
topic_urn
urn:smn:regionId:d91989905b8449b896f3a4f0ad57222d:demo
type
SubscriptionConfirmation
```

📖 说明

每一个键占一行，每一个值占一行，键所对应的值在该键所占行的下一行。

A.6.4 示例代码

Java 语言

验证消息是否有效，其中signing_cert_url、signature是从[HTTP\(S\)消息格式](#)描述获取的值，message为特定消息的签名键值。以下为示例代码，仅供参考。

```
private static void isMessageValid(String signing_cert_url,
    String signature, Map<String, String> message) {
    InputStream in = null;
    try {
        URL url = new URL(signing_cert_url);
        in = url.openStream();
        CertificateFactory cf = CertificateFactory.getInstance("X.509");
        X509Certificate cert = (X509Certificate) cf.generateCertificate(in);
        Signature sig = Signature.getInstance(cert.getSigAlgName());
        sig.initVerify(cert.getPublicKey());
        sig.update(buildSignMessage(message).getBytes("UTF-8"));
        byte[] sigByte = Base64.getDecoder().decode(signature);
        if (sig.verify(sigByte)) {
            System.out.println("Verify success");
        } else {
            System.out.println("Verify failed");
        }
    } catch (Exception e) {
        throw new SecurityException("Verify method failed.", e);
    } finally {
        if (in != null) {
            try {
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

📖 说明

使用Java 8 版本以下的用户，可以使用第三方jar包commons-codec.jar进行Base64解码。
并将上述样例代码中的"byte[] sigByte = Base64.getDecoder().decode(signature);"调整为
"byte[] sigByte = Base64.decodeBase64(signature);"。

构建校验签名的示例代码

```
private static String buildSignMessage(Map<String,String> msg) {
    String type = msg.get("type");
    String message = null;
    if ("Notification".equals(type)){
        message = buildNotificationMessage(msg);
    } else if ("SubscriptionConfirmation".equals(type) ||
        "UnsubscribeConfirmation".equals(type)){
        message = buildSubscriptionMessage(msg);
    }
}
```

```
    }
    return message;
}

private static String buildSubscriptionMessage(Map<String, String> msg) {
    String stringMessage = "message\n";
    stringMessage += msg.get("message") + "\n";
    stringMessage += "message_id\n";
    stringMessage += msg.get("message_id") + "\n";
    stringMessage += "subscribe_url\n";
    stringMessage += msg.get("subscribe_url") + "\n";
    stringMessage += "timestamp\n";
    stringMessage += msg.get("timestamp") + "\n";
    stringMessage += "topic_urn\n";
    stringMessage += msg.get("topic_urn") + "\n";
    stringMessage += "type\n";
    stringMessage += msg.get("type") + "\n";
    return stringMessage;
}

private static String buildNotificationMessage(Map<String, String> msg)
{
    String stringMessage = "message\n";
    stringMessage += msg.get("message").toString() + "\n";
    stringMessage += "message_id\n";
    stringMessage += msg.get("message_id").toString() + "\n";
    if (msg.get("subject") != null){
        stringMessage += "subject\n";
        stringMessage += msg.get("subject").toString() + "\n";
    }
    stringMessage += "timestamp\n";
    stringMessage += msg.get("timestamp").toString() + "\n";
    stringMessage += "topic_urn\n";
    stringMessage += msg.get("topic_urn").toString() + "\n";
    stringMessage += "type\n";
    stringMessage += msg.get("type").toString() + "\n";
    return stringMessage;
}
```

Node.js

```
const fs = require('fs');
const crypto = require('crypto');
const jsrsasig = require('jsrsasign');

/**
 * 校验消息签名
 * @param pemFile 签名文件存储路径(下载到本地证书文件路径)
 * @param signature 待验证的签名值
 * @param message 待验证的消息内容
 * @returns {boolean} true: 签名值验证通过; false: 签名值校验不通过
 */
function verifyMessage(pemFile, signature, message) {
    const pubPem = fs.readFileSync(pemFile);
    const verify = crypto.createVerify(signatureAlgorithm(pubPem));
    verify.update(buildSignMessage(message));
    const verifyResult = verify.verify(pubPem, signature, 'base64');
    if (verifyResult) {
        console.log("verify success");
        return true;
    } else {
        console.log('verify failed, result: ' + verifyResult);
        return false;
    }
}

/**
 * 从证书中获取签名算法
 */
```

```
function signatureAlgorithm(pubPem) {
  const certObject = new jsrsag.X509();
  certObject.readCertPEM(pubPem.toString());
  let algorithm = certObject.getSignatureAlgorithmField();
  if (algorithm.split('with').length > 1) {
    algorithm = algorithm.split('with')[1] + '-' + algorithm.split('with')[0];
  }
  return algorithm;
}

function buildSignMessage(msg) {
  const type = msg.type;
  let message = "";
  if (type === 'Notification') {
    message = buildNotificationMessage(msg);
  } else if (type === 'SubscriptionConfirmation') {
    message = buildSubscriptionMessage(msg);
  }
  return message;
}

function buildNotificationMessage(msg) {
  let signMessage = 'message\n' + msg.message + '\n';
  signMessage += 'message_id\n' + msg.message_id + '\n';
  if (msg.subject) {
    signMessage += 'subject\n' + msg.subject + '\n';
  }
  signMessage += 'timestamp\n' + msg.timestamp + '\n';
  signMessage += 'topic_urn\n' + msg.topic_urn + '\n';
  signMessage += 'type\n' + msg.type + '\n';
  return signMessage;
}

function buildSubscriptionMessage(msg) {
  let signMessage = 'message\n' + msg.message + '\n';
  signMessage += 'message_id\n' + msg.message_id + '\n';
  signMessage += 'subscribe_url\n' + msg.subscribe_url + '\n';
  signMessage += 'timestamp\n' + msg.timestamp + '\n';
  signMessage += 'topic_urn\n' + msg.topic_urn + '\n';
  signMessage += 'type\n' + msg.type + '\n';
  return signMessage;
}
```

📖 说明

该示例代码已在Nodejs v14.17.5版本上测试通过。

Go 语言

```
package demo

import (
  "bytes"
  "crypto"
  "crypto/rsa"
  "crypto/x509"
  "encoding/base64"
  "encoding/json"
  "encoding/pem"
  "fmt"
  "io/ioutil"
)

type Message struct {
  Signature      string `json:"signature"`
  Subject        *string `json:"subject"`
  TopicUrn       string `json:"topic_urn"`
  MessageId      string `json:"message_id"`
  SignatureVersion string `json:"signature_version"`
}
```



```

Type      string `json:"type"`
Message   string `json:"message"`
SubscribeUrl string `json:"subscribe_url"`
UnsubscribeUrl string `json:"unsubscribe_url"`
SigningCertUrl string `json:"signing_cert_url"`
Timestamp string `json:"timestamp"`
}

func VerifyMessage(pemFile string, message string) bool {
    msg := Message{}
    err := json.Unmarshal([]byte(message), &msg)
    if err != nil {
        fmt.Println("Convert json to struct failed")
        return false
    }
    pemContent, err := ioutil.ReadFile(pemFile)
    if err != nil {
        fmt.Println("Read pem file failed")
        return false
    }
    certDerblock, _ := pem.Decode(pemContent)
    if certDerblock == nil {
        fmt.Println("Decode pem file failed")
        return false
    }
    cert, err := x509.ParseCertificate(certDerblock.Bytes)
    if err != nil {
        fmt.Println("Parse cert failed")
        return false
    }

    msgString := buildMessage(&msg)
    msgHash := crypto.SHA256.New()
    msgHash.Write([]byte(msgString))
    msgHashSum := msgHash.Sum(nil)

    decodeSign, _ := base64.StdEncoding.DecodeString(msg.Signature)

    publicKey := cert.PublicKey.(*rsa.PublicKey)
    err = rsa.VerifyPKCS1v15(publicKey, crypto.SHA256, msgHashSum, decodeSign)
    if err != nil {
        fmt.Println("Verify failed")
        return false
    } else {
        fmt.Println("Verify success")
        return true
    }
}

func buildMessage(msg *Message) string {
    if msg.Type == "Notification" {
        return buildNotificationMessage(msg)
    } else if msg.Type == "SubscriptionConfirmation" || msg.Type == "UnsubscribeConfirmation" {
        return buildSubscriptionMessage(msg)
    }
    return ""
}

func buildNotificationMessage(msg *Message) string {
    buf := bytes.Buffer{}
    buf.WriteString("message\n" + msg.Message + "\n")
    buf.WriteString("message_id\n" + msg.MessageId + "\n")
    // msg中存在Subject字段不存在的场景，需要特殊处理
    if msg.Subject != nil {
        buf.WriteString("subject\n" + *msg.Subject + "\n")
    }
    buf.WriteString("timestamp\n" + msg.Timestamp + "\n")
    buf.WriteString("topic_urn\n" + msg.TopicUrn + "\n")
    buf.WriteString("type\n" + msg.Type + "\n")
}

```

```
    return buf.String()
}

func buildSubscriptionMessage(msg *Message) string {
    buf := bytes.Buffer{}
    buf.WriteString("message\n" + msg.Message + "\n")
    buf.WriteString("message_id\n" + msg.MessageId + "\n")
    buf.WriteString("subscribe_url\n" + msg.SubscribeUrl + "\n")
    buf.WriteString("timestamp\n" + msg.Timestamp + "\n")
    buf.WriteString("topic_urn\n" + msg.TopicUrn + "\n")
    buf.WriteString("type\n" + msg.Type + "\n")
    return buf.String()
}
```

 **说明**

该示例代码已在go 1.15版本上测试通过。

B 修订记录

版本日期	变更说明
2022-04-12	第一次正式发布。