

技术文档
PUBLIC (公共)
文档版本： 2.73.0 – 2020-11-18

面向 SAP BTP 的 SAP IoT 服务

© 2021 SAP 股份有限公司或其关联公司版权所有，保留所有权利。

THE BEST RUN SAP

内容

1 面向 SAP BTP 的 SAP IoT 服务. 4

2 发行说明. 5

3 概览. 11

4 开始使用. 13

4.1 启用 物联网 服务. 13

4.2 为 物联网 服务分配角色. 14

4.3 访问 物联网 服务主控室. 15

4.4 部署消息管理服务 (MMS). 16

通过 物联网 服务主控室部署和更新 MMS. 16

通过 API 部署和更新 MMS. 17

为 Java iotmms 应用程序分配 IoT-MMS-User 角色. 18

为 Java 应用程序 iotmms 分配角色 IoT-MMS-DevOps. 19

将消息管理服务绑定到数据源. 20

监控消息管理服务. 20

缩放消息管理服务. 21

5 文档、支持、设置、关于 IoT 服务和注销框. 23

6 物联网 服务主控室. 24

6.1 配置消息类型. 27

创建消息类型. 27

删除消息类型. 28

6.2 配置设备类型. 29

创建设备类型. 29

为设备类型重新生成 OAuth 令牌. 30

删除设备类型. 31

6.3 配置设备. 31

创建设备. 32

编辑自定义属性. 33

为设备重新生成 OAuth 令牌. 34

删除设备. 34

层次结构视图. 35

可视化设备数据. 37

7 远程设备管理服务 API. 38
7.1 API v2 介绍. 38
设备类型. 40
数据类型. 47
消息类型. 48
设备. 51
在层次结构中组织设备. 60
应用程序和账户设置. 63
REST 响应. 64
8 消息管理服务 API. 67
8.1 接口和通信处理. 67
消息格式. 67
样例设备和消息类型. 69
数据发送模式. 71
8.2 配置 API 验证. 72
8.3 数据服务. 74
数据服务 API 验证. 74
HTTP. 75
WebSocket. 80
WebSocket 上的 MQTT. 82
TCP 上的 MQTT. 83
8.4 推送服务. 88
推送服务 API 验证. 91
8.5 消息批次. 92
8.6 消息处理. 93
处理服务 API 验证. 95
数据库存储 (SQL). 96
文档存储 (CMIS). 99
消息转发 (HTTP). 100
智能数据流 (SDS). 102
8.7 消息使用. 104
OData API 验证. 107

1 面向 SAP BTP 的 SAP IoT 服务

物联网 服务可使客户和合作伙伴在云中开发、定制和操作 IoT 业务应用程序。

有关产品发布后的更新，请参阅 物联网 服务的第 5 页上的“发行说明”部分；有关 SAP BTP 的重要更新，请参阅新增内容部分中的最新发行说明。要了解 SAP BTP 上可用服务的完整列表，请参阅 SAP BTP 文档中的功能部分。

相关信息

第 5 页上的“发行说明”
第 11 页上的“概览”
第 24 页上的“物联网 服务主控室”
第 38 页上的“远程设备管理服务 API”
第 67 页上的“消息管理服务 API”

2 发行说明

发行说明

下表提供了自上一版本后新增内容和更改内容的信息。

日期（版本）	描述
2020/11/18 (2.73.0)	• **修复** ○ 已将 HttpDestination 库和 MMS 部署运行时更新为 Java Web Tomcat 8。
2019/10/15 (2.72.1)	• **修复** ○ 已解决服务生命周期管理问题。
2019/9/6 (2.72.0)	• **修复** ○ 解决了用户提供的标识过长时返回的有误错误代码相关的问题。

2019/1/18
(2.62.0)
	• **修复**：
	  ○ 解决了在 Java Web Tomcat 运行时版本 2.115 上运行 MMS 组件时，导致在处理消息的过程中出错的问题。
	  ○ 解决了导致用户为 MMS 创建的配置在重新启动此组件时重置的问题。

2018/6/22
(2.56.0)
	• **修复**：
	  ○ 更新了 MMS 主控室中的 SAP 支持、隐私和法律信息。
	  ○ 解决了在处理确认时，在某些错误情况下导致确认丢失的问题。
	  ○ 解决了 MMS 主控室中导致在使用 Internet Explorer 11 发送 MQTT 测试消息时出现错误的问题。

2018/5/9 (2.55.0)	• **增强**： ○ 自 2018 年 5 月 25 日起，为符合常规数据保护法 (GDPR)，新增部分 作为安全指南的一部分发布。
2018/3/29 (2.54.0)	• **修复** ○ 解决了由于凭据错误导致 MMS 部署失败时造成混乱的错误消息的问题。
2018/2/16 (2.51.0)	• **修复** ○ MMS 部署和更新功能已调整，流程描述已更新。有关详细信息，请参阅第 16 页上的“通过 物联网 服务主控室部署和更新 MMS”部分。
2017/12/22 (2.49.0)	• **修复**： ○ 问题已解决，需要重新启动 MMS 才能使监控参数的更改生效。
2017/6/23 (2.43.0)	• **新功能**： ○ 物联网 服务现支持使用 Google Protocol Buffer (protobuf) 通过 HTTP 数据服务发送数据以及接收通过 HTTP 推送服务发送的数据。有关详细信息，请参阅 第 77 页上的“HTTP” 和第 89 页上的“推送服务”部分。

日期（版本）	描述
2017/6/9 (2.42.0)	• **增强：** ○ MMS 主控室中用于可视化设备数据的基于时间的图表已改进，允许显示更大的图表视图。有关如何可视化设备数据的详细信息，请参阅第 37 页上的“可视化设备数据”部分。
2017/6/9 (2.40.1)	• **补丁：** ○ 解决了导致消息无法从 MMS 正确写入文档服务的 CMIS 处理服务的会话处理问题。

2017/4/28
(2.40.0)
	• **增强：**
	  ○ 现在 物联网 服务文档在新的 SAP Help Portal 上公共可用，提供英语、德语、日语和简体中文版本，可以按第 23 页上的“文档、支持、设置、关于 IoT 服务和注销框”部分中所述从 物联网 服务主控室进行访问。
	• **修复：**
	  ○ 解决了并发调用远程设备管理服务导致数据库约束违规并产生 HTTP 500 返回代码的问题。

2017/4/13
(2.39.0)
	• **增强：**
	  ○ 物联网 服务主控室允许基于标识过滤设备、设备类型和消息类型。
	• **修复：**
	  ○ 解决了 HTTP 处理服务中的一个问题，该问题会导致来自不同设备但是具有相同设备类型和消息类型的消息在某些情况下混合在一起。

2017/3/31
(2.38.0)
	• **增强：**
	  ○ 现在 物联网 服务与新的根证书授权机构 (CA) 集成。旧的根 CA 颁发的证书仍有效。
	  ○ 我们修改了设备类型证书的公用名称信息处理。设备类型标识和租户标识现包含在两个单独的公用名称条目中。

2017/3/3
(2.36.0)
	• **新功能：**
	  ○ 可在 MMS 主控室内将设备数据可视化为基于时间的图表。有关详细信息，请参阅第 37 页上的“可视化设备数据”部分。
	  ○ 物联网 服务主控室引入了附加角色，可通过 API 启用 MMS 的部署和更新，无需用户访问 物联网 服务主控室。
	• **增强：**
	  ○ 我们修改了设备证书的公用名称信息处理。设备标识和租户标识现包含在两个单独的公用名称条目中。

2017/2/17 (2.35.0)	• **新功能：** ○ 您可以在对话框中选择“设置”，为 物联网 服务主控室和 MMS 主控室切换成您的首选语言和主题。有关详细信息，请参阅第 23 页上的“文档、支持、设置、关于 IoT 服务和注销框”部分。
2017/2/3 (2.34.1)	• **补丁：** ○ 解决了阻止客户使用基于证书的验证创建新设备类型的问题。
2017/2/3 (2.34.0)	• **新功能：** ○ 现在支持使用证书用于通过 MQTT/WSS 传输协议发送消息。有关详细信息，请参阅第 82 页上的“WebSocket 上的 MQTT”部分。

日期（版本）	描述

2017/1/20
(2.33.0)
	• **新功能：**
	  ○ 为 物联网 服务主控室增加了简体中文和日语支持。
	• **增强：**
	  ○ 重新构建了“消息管理服务”的部署和更新页面。
	• **修复：**
	  ○ 更正了建模字符串类型的 MessageType 字段时 物联网 服务主控室中的输入验证错误。
	  ○ 更正了远程设备管理服务 API 中验证对象的序列化错误。缺少验证对象时会出现序列化问题。

2016-12-16 (2.32.0)	• **增强：** ○ 扩展 HTTP 处理服务使用户能够自由配置 HTTP 标头。
2016/12/2 (2.31.0)	• **漏洞修复：** ○ 解决了使用 WebSocket 或 WebSocket 上的 MQTT 传输协议发送推送消息时 MMS 推送服务中导致不一致的问题。

2016/11/4
(2.29.0)
	• **新功能：**
	  ○ 扩展了 HTTP 处理服务，支持通过 OAuth 进行验证。此外，还支持配置代理。有关详细信息，请参阅第 100 页上的“消息转发 (HTTP)”部分。
	• **漏洞修复：**
	  ○ HTTP 处理服务现接受全部 HTTP 2xx 响应代码作为请求处理成功的结果。之前版本只接受 HTTP 200 作为成功消息。

2016/10/7
(2.27.0)
	• **新功能：**
	  ○ 物联网 服务现支持基于客户端证书的验证，用于确保 IoT 设备与 物联网 服务之间安全通信。有关详细信息，请参阅第 40 页上的“设备类型”部分和第 51 页上的“设备”部分。
	  ○ 物联网 服务现支持 TCP 上的 MQTT。有关详细信息，请参阅第 83 页上的“TCP 上的 MQTT”部分。
	• **漏洞修复：**
	  ○ 解决了处理服务配置中的问题：存储凭据时产生的问题在注册期间不会以失败告知用户，而是在稍后阶段引发内部错误。
	  ○ 移除了使用远程设备管理 API 进行设备注册时传递验证对象的选项。原因是验证机制由参考的设备类型确定并且在设备注册期间不可更改。

2016/9/9
(2.25.0)
	• **新功能：**
	  ○ 物联网 服务文档现提供德语版本。
	• **增强：**
	  ○ 改进了处理服务配置 UI 的用户输入验证。
	  ○ 改进了 UI，避免 MMS 主控室应用程序数据 UI 中出现截断文本。
	• **漏洞修复：**
	  ○ 解决了通过 MQTT/WebSocket 使用 MMS 推送 API 时造成错误的问题。
	  ○ 解决了用户名元音变音不能正确显示的问题。

日期（版本）	描述

2016/8/26
(2.24.0)
	• **新功能：**
	  ○ 物联网 服务主控室现提供德语版本。
	  ○ IoT RDMS API 已扩展为支持用户定义的标识。
	  ○ MMS OData API 已扩展为支持 $select 查询参数。
	• **增强：**
	  ○ 为提高性能重构了用于处理服务的线程处理。
	• **漏洞修复：**
	  ○ 解决了在 SAP ASE 数据库中处理浮点值时造成的问题。
	  ○ 解决了防止用户以少于三个字符来存储安全属性时造成的问题。
	  ○ 设备类型的 OAuth 客户端的名称现在设置为设备类型名称而不是设备类型标识。

2016/7/29
(2.22.0)
	• **增强：**
	  ○ 针对新的 SAPUI5 版本，在 物联网 服务主控室中进行了兼容性改进和更新。
	  ○ 在 物联网 服务主控室下拉框添加了官方 SAP HANA Cloud Platform 支持页面的链接。有关详细信息，请参阅第 23 页上的“文档、支持、设置、关于 IoT 服务和注销框”部分。
	• **漏洞修复：**
	  ○ 移除了 物联网 服务主控室层次结构视图中的下拉框在 Firefox 和 Chrome 浏览器中被截断的问题。

2016/7/15
(2.21.0)
	• **新功能：**
	  ○ 物联网 服务主控室现支持远程设备管理服务 API v2。
	  ○ 现提供与 SAP HANA Cloud Platform 智能数据流集成的新 SDS 处理服务。有关详细信息，请参阅第 102 页上的“智能数据流 (SDS)”部分。
	• **增强：**
	  ○ 处理服务映射的配置用户界面已经增强，现允许对处理服务属性进行排序。
	• **漏洞修复：**
	  ○ MQTT/WS 数据服务的 OAuth 处理已更正。此错误阻止设备与 MMS 连接并且会产生下列错误消息：“错误用户名和密码”。
	  ○ MQTT/WS 数据服务用户界面已经增强，现允许在建立连接后从相应输入字段中复制 MQTT 端点。

2016/7/1
(2.20.0)
	• **漏洞修复：**
	  ○ 已经修复了 MMS 主控室中 MQTT/WS 样例客户端用户界面的验证问题。
	  ○ 在 CMIS 处理服务中修复了配置属性 `mms.processing.cmis.overwrite_existing_files` 的处理。问题导致通过 CMIS 处理服务的文件更新失败。
	  ○ 修复了配置属性 `mms.appdata.max_rows_number` 的处理。现在用作 MMS OData API 中 'top' 查询参数的缺省值。

2016/6/17 (2.19.0)	• **新功能：**MMS 现在支持 WebSocket 上的 MQTT 作为数据服务。有关详细信息，请参阅第 82 页上的“WebSocket 上的 MQTT”部分。
2016/6/3 (2.18.0)	• **新功能：**RDMS 现提供 API v2.RDMS API v1 被弃用。有关详细信息，请参阅第 38 页上的“远程设备管理服务 API”部分。
2016/5/20 (2.17.0)	• **新功能：**MMS 现在为配置处理服务映射提供用户界面。有关详细信息，请参阅第 93 页上的“消息处理”部分。

日期（版本）	描述
2016/5/6 (2.16.0)	• **新功能**：自定义设备属性可以通过 物联网 服务主控室应用程序进行修改。有关详细信息，请参阅第 33 页上的“编辑自定义属性”部分。
2016/4/22 (2.15.0)	• **新功能**： ○ MMS 现公开了可用于获得和进行应用程序设置的公共配置 API。有关详细信息，请参阅第 72 页上的“配置 API 验证”部分。 ○ 配置用户界面现可用。您可以从 MMS 主控室中相应的磁贴进行访问。 ○ MMS 可配置为自动向 RDMS 添加自定义属性。属性 ‘lastSignal’ 将捕获在特定设备向 MMS 发送数据时的最后一个时间戳。此功能在缺省情况下处于禁用状态，您可以使用 ‘mms.rdms.periodic_last_signal_update_interval_milliseconds’ 配置属性将其激活。
2016/4/8 (2.14.0)	• **新功能**：在 物联网 服务主控室中创建新设备时，您可以添加自定义设备属性。
2016/3/24 (2.13.0)	• **新功能**： ○ MMS 现在支持通过设备类型和消息类型标识注册多个处理服务。 ○ 部署 MMS 时，系统自动创建数据库可用性检查（有关详细信息，请参阅第 20 页上的“监控消息管理服务”部分）。您可以在 SAP HANA Cloud Platform 主控室的“监控”部分中跟踪此检查的状态。此外，显示在 SAP HANA Cloud Platform 主控室中的 MMS 状态图标将根据数据库的可用性更改。
2016/3/11 (2.12.0)	• **增强**： ○ 物联网 服务主控室中针对新的 SAPUI5 版本进行了兼容性改进和更新。 ○ 针对多进程设置中运行的 MMS 进行了主要 ASE 支持改进。 ○ MMS OData 元数据文档已扩展为完全支持 Unicode 模式、表和列名。 • **漏洞修复**： ○ 物联网 服务主控室中进行了语言修正。 ○ 自定义设备属性中的 PATCH 返回正文已从设备对象更改到 RDMS API 中的自定义属性数组。 ○ MMS 主控室布局已调整为使用新的 SAPUI5 版本。
2016/2/26 (2.11.0)	• **新功能**：MMS OData 接口现在支持使用标识属性中提供的 URL 检索单个项目。
2016/2/12 (2.10.0)	• **新功能**：新增了部分，包括常见问题和所提供的解决方案集合。现在 MMS 的推送、OData、和处理 API 也与 OAuth 结合使用。OData 接口现在支持 OData V2 规范中定义的批处理。有关详细信息，请参阅第 91 页上的“推送服务 API 验证”、第 104 页上的“消息使用”和第 93 页上的“消息处理”部分。 • **增强**：简化了 MMS 主控室中的应用数据用户界面中的表选项。
2016/1/15 (2.8.0)	• **新功能**：MMS 现配置为将消息转发到其他 HTTP 端点（请参阅第 100 页上的“消息转发 (HTTP)”部分以获取更多详细信息）。 • **增强**：MMS 的 OData 接口现支持扩展的过滤器选项（neq、lt、le、gt、ge、and、or）（请参阅第 104 页上的“消息使用”部分）。此外，OData 实体属性根据相应数据库表列键入。对于 RDMS，设备现在支持最多 32 个自定义属性，用于存储与设备相关的数据，例如序列号、MAC 或者 IP 地址（请参阅第 51 页上的“设备”部分以获取更多详细信息）。
2015/12/18 (2.7.0)	• **增强**：MMS 现在配置为将二进制消息转发到 SAP HANA Cloud Platform 文档资源库。

日期（版本）	描述
2015/12/3 (2.6.0)	• **增强**：在 物联网 服务主控室中创建新消息类型时，可以通过拖放对消息字段进行排序。此外，对消息类型、设备类型和设备提供完全的 Unicode 支持。不必再通过手动配置步骤为 MMS 调整“验证配置”来启用推送以及 OData API。而且日期消息字段现在可以通过 Unix 时间格式发送，而不仅限于 ISO 8601 格式。
2015/11/19 (2.5.0)	• **增强**：在 MMS 中，消息不仅可以通过 JSON 编码发送，也可以通过查询参数发送。因此，带有预定义和不可修改的 POST 正文有效负载的设备也可以向 MMS 发送消息。现在可以配置目标数据库模式，以供 MMS 在其中创建表。
2015/11/5 (2.4.0)	• **新功能**：现在可以在灵活层次结构中组织设备，例如按组织或位置进行分组。此外，现在还支持二进制消息类型字段。
2015/10/8 (2.3.0)	• **新功能**：您可以使用新的 OData 接口使用 MMS 2.3 处理的消息。还可以使用 API 定义表和列的自定义名称并定制表映射模式。有关详细信息，请参阅消息处理部分。 • **漏洞修复**：SAPUI5 更新到 1.30.6 版本后，MMS 主控室应用数据用户界面（用于检查表）遭到破坏。现已修复此问题。将 MMS 更新到 2.3 版本以应用修复。
2015/7/30 (2.0.0)	• **增强**：消息类型的标识符已从序列 + DeviceTypeId 的组合键值更改到 20 字符的单个十六进制值。为保证完全兼容，消息传递应用程序 (MMS) 必须在使用者账户中更新。现在支持通过 OAuth 自动注册设备。
2015/5/7 (1.0.0)	• 初始版本。

3 概览

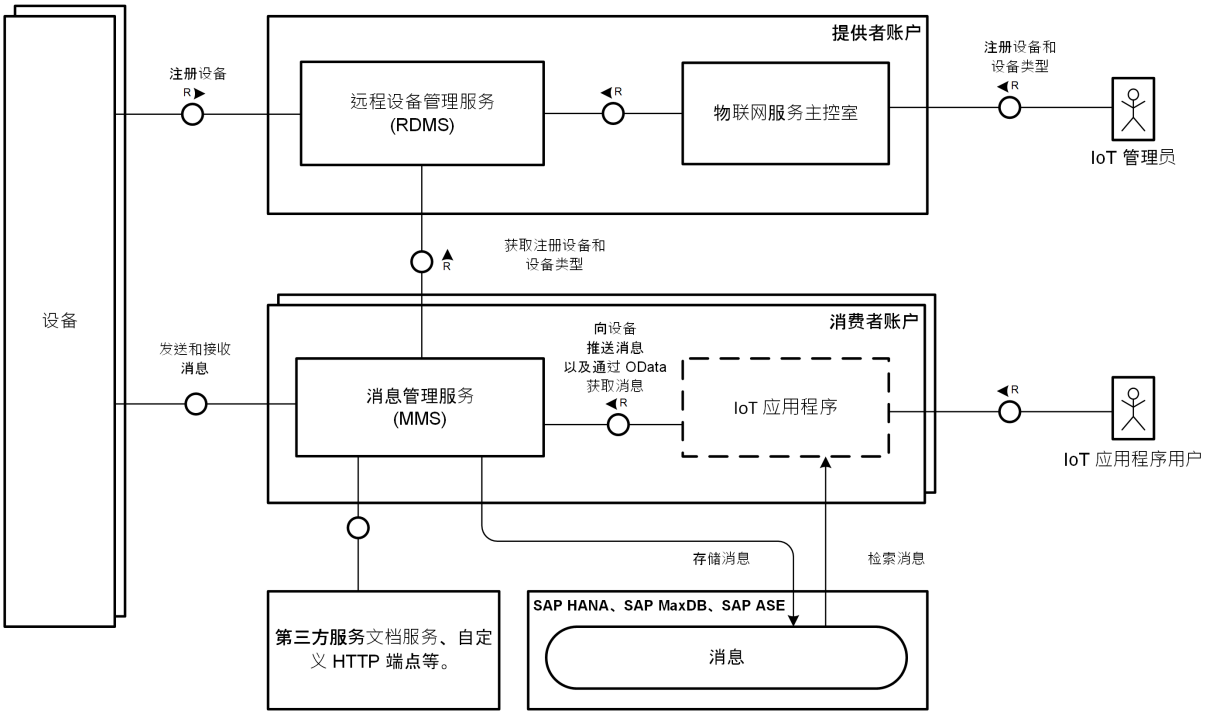
物联网 服务旨在帮助和支持 IoT 应用程序的实施。

该文档描述针对 SAP BTP 账户上可用的 Neo 环境的 物联网 服务的功能。有关自产品发布以来的重要更新的信息，请参阅为 SAP BTP 发布的新增内容部分。

体系结构

物联网 服务提供各种界面，用于注册设备及其特定数据类型，以安全高效的方式向 SAP BTP 上运行的数据库发送数据，以及存储数据和提供对存储数据的访问。对应的服务在两个主要组件间分布：远程设备管理服务 (RDMS) 和消息管理服务 (MMS)。另外，通过名为 物联网 服务主控室的图形用户界面可以访问各种服务。

下图提供了体系结构的概览。



MMS 提供各种 API，可供设备发送数据到 物联网 服务。它负责处理数据并将数据保留在所连接的数据库中。此外可能还有其他用例，要求将数据转发到消息代理、事件流处理器甚至是文档资源库。

物联网 服务主控室是用户与远程设备管理服务 (RDMS) 进行交互的主要界面。可以将其用于注册新的设备，来定义发送和接收消息的模式（消息类型和设备类型），并建立设备所需的必要信任关系来与消息管理服务 (MMS) 进行交互。物联网 服务主控室与 RDMS 作为云服务提供，可以通过租用使用。

相关信息

第 24 页上的“物联网 服务主控室”

4 开始使用

4.1 启用 物联网 服务

先决条件

您拥有 SAP ID 用户并且可访问企业账户中的 SAP BTP 子账户。有关不同账户类型的详细信息，请参阅账户、在主控室中导航到全局账户和子账户部分。有关子账户配额的详细信息，请参阅 SAP BTP 文档中的管理配额部分。

您是子账户的管理员。

在子账户中有可用数据库，可以将其绑定到消息管理服务 (MMS)。有关详细信息，请参阅第 20 页上的“将消息管理服务绑定到数据源”部分。

过程

1. 转到 SAP BTP 主控室子账户中的 “服务”标签。有关详细信息，请参阅 SAP BTP 文档中的 SAP BTP 主控室部分。
2. 选择“物联网”。
3. 选择服务详细信息视图中的“启用”。

   通过执行此步骤，可以订阅远程设备管理服务 (RDMS) 并且获得 物联网 服务主控室的访问权限。

相关信息

SAP BTP 主控室
第 24 页上的“物联网 服务主控室”

4.2 为 物联网 服务分配角色

物联网 服务通过角色受到保护。为授予其他用户访问 物联网 服务主控室和远程设备管理服务 API 的权限，请按照步骤 1-5 所述，为 Java 应用程序 *“iotcockpit”* 和 *“iotrdms”* 分配 *“IoT-User”* 角色。

先决条件

> **i 注意**
>
> 通过 SAP BTP 主控室中的“服务”标签启用 物联网 服务的用户将被自动分配到所需角色。

以 SAP ID 用户身份登录 SAP BTP 主控室。

> **i 注意**
>
> 如果您的子账户配置为使用自定义身份提供者 (IdP)，您需要将 *“IoT-User”* 角色分配给该特定 IdP 已知的用户，而不是使用 SAP ID 用户。有关详细信息，请参阅 SAP BTP 文档中的 Neo 环境中的权限和信任管理部分。

过程

1. 转到子账户的 SAP BTP 主控室中的 “应用程序”标签。选择“订阅”。
2. 选择 *“iotcockpit”* 应用程序。
3. 选择 “角色”标签。
4. 将 *“IoT-User”* 角色分配给您的用户。
5. 同样为 *“iotrdms”* 应用程序执行步骤 1-4。

相关信息

管理角色
第 24 页上的“物联网 服务主控室”

4.3 访问 物联网 服务主控室

物联网 服务主控室是用户与远程设备管理服务 (RDMS) 进行交互的主要界面。可以将其用于注册新的设备，来定义发送和接收消息的模式（消息类型和设备类型），并建立设备所需的必要信任关系来与消息管理服务进行交互 (MMS)。

先决条件

- 以 SAP ID 用户身份登录 SAP BTP 主控室子账户。
- 您已被分配 *“IoT-用户”*角色。

过程

1. 转到 SAP BTP 主控室子账户中的 “服务”标签。
2. 选择“物联网”。
3. 在服务详细信息视图中针对“物联网”订阅选择“转到服务”。
   随即显示 物联网 服务主控室。



有关访问服务的详细信息，请参阅 SAP BTP 文档中的账户部分。

相关信息

第 24 页上的“物联网 服务主控室”

4.4 部署消息管理服务 (MMS)

消息管理服务 (MMS) 是负责从设备接收数据并向设备发送消息的组件。它提供了各种 API 来实现这一目的。

4.4.1 通过 物联网 服务主控室部署和更新 MMS

先决条件

您已在 SAP BTP 主控室中启用了 物联网 服务。因此，如 第 24 页上的“物联网 服务主控室” 部分所述，将自动为子账户创建 物联网 服务主控室和 RDMS 的订阅。

物联网 服务主控室可以用来管理设备类型、消息类型和设备。此外，还允许部署和更新 MMS。

过程

1. 转到 物联网 服务主控室。
2. 选择 “部署消息管理服务”磁贴，从而打开 MMS 部署工具管理。
3. 如果没有预先填写，在 SAP BTP 主控室中填写“账户标识”，然后输入“用户名”和“密码”。

   i 注意

   您可以在子账户的 SAP BTP 主控室的“账户仪表盘”中查看“账户标识”。有关账户仪表盘的详细信息，请参阅 SAP BTP 文档中的云主控室部分。

4. **⚠ 警告**

   在您的子账户中您只负责升级 MMS 版本。

   选择“部署”按钮。如果正在运行 MMS 实例，部署工具管理会首先停止实例。之后，工具管理会检索 MMS 的最新版本，对其进行部署并重新启动 MMS。如果 MMS 实例已停止或者先前从未部署，则工具管理将进行部署，但是不会自动启动。请注意，在启动并运行 MMS 之前，必须将其绑定到数据源。有关详细信息，请参阅

第 20 页上的“将消息管理服务绑定到数据源”部分。成功部署后，将在部署对话框下面显示已部署的 MMS 应用程序链接。

将 MMS 成功部署到子账户后，您需要按照 第 18 页上的“为 Java iotmms 应用程序分配 IoT-MMS-User 角色” 部分所述将用户分配到 *“IoT-MMS-User”* 角色。执行这一操作后，才能访问 MMS 主控室以及使用可用于数据服务和推送服务的样例客户端。

相关信息

第 17 页上的“通过 API 部署和更新 MMS”
第 18 页上的“为 Java iotmms 应用程序分配 IoT-MMS-User 角色”
第 19 页上的“为 Java 应用程序 iotmms 分配角色 IoT-MMS-DevOps”
第 24 页上的“物联网 服务主控室”
第 67 页上的“接口和通信处理”
第 74 页上的“数据服务”
第 88 页上的“推送服务”

4.4.2 通过 API 部署和更新 MMS

先决条件

您已在 SAP BTP 主控室中启用了 物联网 服务。此外，您的用户具有第 19 页上的“为 Java 应用程序 iotmms 分配角色 IoT-MMS-DevOps”部分所述的 *“IoT-MMS-DevOps”* 角色。而且，用户在 SAP BTP 主控室中的子账户必须至少具备“开发人员”角色。有关 SAP BTP 中角色的详细信息，请参阅管理成员权限部分。

> **⚠ 警告**
>
> 您将无法使用 物联网 服务主控室。必须使用类似 REST 客户端的工具发送 HTTP(s) 调用。

过程

1. 转到 REST 客户端并插入以下标头：`Authorization: Basic <base64-encoded username:password>`
2. 执行具有所示 JSON 正文的以下调用：`https://<host>/com.sap.iotservices.cockpit/v1/api/mmsDeploy` 上的 `PUT`

请求正文：

```
{
  "account": <HCP account Id>,
  "user": <user Id>,
  "password": <user password>,
  "host": <host of this HCP landscape>,
  "application": "iotmms",
  "size": "LITE"
}
```

> **i 注意**
>
> 可以通过向具有相同标头的相同 URL 提交 GET 调用来查询几乎所有需要的正文元素。

相关信息

第 16 页上的“部署消息管理服务 (MMS)”
第 16 页上的“通过 物联网 服务主控室部署和更新 MMS”
第 18 页上的“为 Java iotmms 应用程序分配 IoT-MMS-User 角色”

4.4.3 为 Java iotmms 应用程序分配 IoT-MMS-User 角色

要访问消息管理服务 (MMS) 主控室和使用可用于数据服务和推送服务的样例客户端，用户必须分配到 *“IoT-MMS-User”* 角色。

先决条件

以 SAP ID 用户身份登录 SAP BTP 主控室，MMS 已使用 物联网 服务主控室安装完毕。

过程

1. 转到子账户的 SAP BTP 主控室中的 “应用程序”标签。然后选择 *“Java 应用程序”*。
2. 选择 *“iotmms”* 应用程序。
3. 转到应用程序详细信息的 “安全”标签。然后选择“角色”。
4. 将角色 *“IoT-MMS-User”* 分配给您的用户。

相关信息

第 16 页上的“通过 物联网 服务主控室部署和更新 MMS”
第 17 页上的“通过 API 部署和更新 MMS”
第 19 页上的“为 Java 应用程序 iotmms 分配角色 IoT-MMS-DevOps”
第 24 页上的“物联网 服务主控室”

4.4.4 为 Java 应用程序 iotmms 分配角色 IoT-MMS-DevOps

如果用户要在不使用 物联网 服务主控室的情况下更新或部署消息管理服务 (MMS)，必须为其分配 *“IoT-MMS-DevOps”* 角色。

先决条件

以 SAP ID 用户身份登录 SAP BTP 主控室，MMS 已使用 物联网 服务主控室安装完毕。

过程

1. 转到子账户的 SAP BTP 主控室中的 “应用程序”标签。然后选择 *“Java 应用程序”*。
2. 选择 *“iotmms”* 应用程序。
3. 转到应用程序详细信息的 “安全”标签。然后选择“角色”。
4. 将角色 *“IoT-MMS-DevOps”* 分配给您的用户。

相关信息

第 16 页上的“通过 物联网 服务主控室部署和更新 MMS”
第 17 页上的“通过 API 部署和更新 MMS”
第 18 页上的“为 Java iotmms 应用程序分配 IoT-MMS-User 角色”
第 24 页上的“物联网 服务主控室”

4.4.5 将消息管理服务绑定到数据源

操作 MMS 之前，必须将其绑定到数据源。数据源可以是专用数据库，也可以是共享数据库中的模式。

可使用 SAP BTP 主控室创建数据库和模式。有关详细信息，请参阅以下部分：SAP BTP 文档中的使用主控室创建模式成功创建数据源之后，可以通过创建数据源绑定将 MMS 分配到该数据源。有关详细信息，请参阅以下部分：分别为绑定模式或将 SAP HANA 数据库绑定到 Java 应用程序。确保创建 `<DEFAULT>` 数据源绑定。

⚠ 警告

请注意，如果对数据库用户设置 `IOTMMS` 数据库绑定，则经过特定时间后（取决于数据库设置）该用户可能会收到更改密码的提示。建议您使用禁用密码生命周期的技术数据库用户。

示例

名为 `IOTMMS` 的 SAP HANA 技术用户：`CREATE USER IOTMMS PASSWORD <password>; ALTER USER IOTMMS DISABLE PASSWORD LIFETIME;`

有关详细信息，请参阅 SAP HANA 管理指南中的**密码策略配置选项** -> **最大密码生命周期**部分。

相关信息

第 96 页上的“数据库存储 (SQL)”

4.4.6 监控消息管理服务

要长期监控 MMS，您可以创建自定义监控检查。自定义检查以 MMS 提供并定期更新的 JMX 变量为基础。

如果您创建自定义检查，会影响 SAP BTP 主控室所报告的 MMS 状态。正常状态为 *“STARTED”* 或 *“STOPPED”*。附加状态为 *“CRITICAL”* 和 *“WARNING”*。

关于 JMX 检查的详细信息，请参阅 SAP BTP 文档中的配置 JMX 检查以监控应用程序部分。

前提条件

以 SAP ID 用户身份登录 SAP BTP 主控室，MMS 已使用 物联网 服务主控室安装完毕。本地计算机已安装 neo SDK。有关如何执行控制台命令的信息，请参阅 SAP BTP 文档的设置控制台客户端部分。

过程

在本地计算机上使用 neo 控制台命令行，提交如下创建命令：

```
neo create-jmx-check -a yoursubaccountname -b iotmms -u yourusername -n "MMS
Store Size" -O com.sap.iotservices.mms:type=Monitor -A STORE_SIZE_MBYTES -W
"0:500" -C "0:800" -U MB --host neo.ondemand.com -w
```

此示例创建自定义 JMX 检查，可以监控当前内部 MMS 消息存储大小（百万字节 (MB)）。消息存储达到 50 MB 时设置为 *"WARNING"* 状态，达到 800 MB 时设置为 *"CRITICAL"* 状态。

```
neo create-jmx-check -a yoursubaccountname -b iotmms -u yourusername -n "MMS
Database Availability" -O com.sap.iotservices.mms:type=Monitor -A
DB_AVAILABILITY -W "0:1" -C "0:1" --host neo.ondemand.com -w
```

此示例创建检查来指示基础数据库连接的健康情况。使用如下代码（与其他度量相似）：

```
  0 - ok, 1 - warning, 2 - error
```

会发现在 SAP BTP 主控室中创建的 JMX 检查：

1.转到子账户的 SAP BTP 主控室中的"应用程序"标签。在此选择 *"Java* 应用程序"。

2.选择 *"iotmms"*。

3.在"流程"下选择"流程标识"。

4.在"状态摘要" -> "度量"下选择"度量详细信息"查看"当前度量"和"度量历史"。

同样可以创建电子邮件通知。有关详细信息，请参阅 SAP BTP 文档中的 set-alert-recipients 部分。

4.4.7 缩放消息管理服务

要使消息管理服务 (MMS) 适应不同的用例，您可以以水平（向外扩展/向内扩展）和垂直（向上扩展/向下扩展）的方式对其进行缩放，与 SAP BTP 账户中部署的其他 Java 应用程序一样。

先决条件

- MMS 已经通过物联网主控室或通过 API 部署到您的 SAP BTP 账户。有关详细信息，请分别参阅第 16 页上的"通过 物联网 服务主控室部署和更新 MMS"或第 17 页上的"通过 API 部署和更新 MMS"部分。
- 您以 SAP ID 用户身份登录 SAP BTP 主控室，或者已经下载并配置 SAP BTP 控制台客户端。有关详细信息，请参阅 SAP BTP 文档中的设置控制台客户端部分。
- 您的账户有足够的空闲计算单元来运行多个 MMS 进程。有关详细信息，请参阅 SAP BTP 文档中的计算单元和管理子账户配额部分。

背景信息

缺省情况下，MMS 配置为在从 SAP BTP 主控室或者使用 SAP BTP 控制台客户端提交“开始”命令后启动单个应用程序进程。

当 MMS 首次部署到您的 SAP BTP 账户时，它将使用该账户中可用的虚拟硬件资源（计算单元）进行部署，并由分配到 *“IoT-MMS-DevOps”* 角色的账户用户选择。有关详细信息，请参阅第 19 页上的“为 Java 应用程序 iotmms 分配角色 IoT-MMS-DevOps”部分。

水平缩放 MMS

水平缩放 MMS 允许其处理大量的请求，并且在一个应用程序进程崩溃时提供故障转移功能。

您可以通过添加或删除应用程序进程来向外或者向内扩展 MMS。为此，您需要定义应用程序进程的最小和最大数量。请注意，缺省情况下，MMS 配置为仅启动单个进程。

使用 SAP BTP 控制台客户端更改应用程序进程的最小和最大数量的过程在 SAP BTP 文档中的缩放应用程序部分详细介绍。如果您想使用 SAP BTP 主控室来做出这些更改，请参阅 SAP BTP 文档中的启动和停止应用程序。

> **i 注意**
>
> 您不必在每次从物联网主控室或通过 API（重新）部署 MMS 时更改最小值和最大值。在您对账户中的 MMS 实例进行一次此操作之后，MMS 部署程序机制将自动停止并（重新）启动所有正在运行的应用程序进程。

垂直缩放 MMS

垂直缩放 MMS 允许其向应用程序进程授予更多的处理能力 (CPU)、更多的主内存和更多的磁盘空间。

通过选择用于部署 MMS 的计算单元大小来向上或者向下扩展 MMS。有关详细信息，请参阅 SAP BTP 文档中的计算单元和管理子账户配额部分。

5 文档、支持、设置、关于 IoT 服务和注销框

可以访问在 物联网 服务主控室右上角的 物联网 服务的 “文档”。选择用户名旁的下拉框。

如果您遇到了问题并需要关于获取帮助的位置的信息，请选择 “支持”。随即将您重新定向到提供账户类型特定支持的页面。

选择 “设置”以切换成您的首选语言和主题。当前支持语言为英语、德语、日语和中文，支持的主题为蓝水晶（缺省）和高对比黑色。

“关于 *IoT* 服务”对话框提供以下信息：

- 关于 物联网 服务主控室、远程设备管理服务和消息管理服务的版本信息。
- 有关可以建模并实际使用多少设备管理实体的账户配额信息。
- 在何处访问设备管理 REST API 端点。

最后，您可以选择 “注销”。

请参阅 部分以获取任何可能存在的问题的答案。

相关信息

第 5 页上的“发行说明”
第 24 页上的“物联网 服务主控室”

6 物联网 服务主控室

这一部分描述了名为 物联网 服务主控室的图形用户界面，可用于轻松访问各种服务。

物联网 服务主控室是用户与远程设备管理服务 (RDMS) 进行交互的主要界面。可以将其用于注册新的“设备”，来定义发送和接收消息的模式（“消息类型”和“设备类型”），并建立设备与消息管理服务 (MMS) 进行交互所需的必要信任关系。下图提供了 物联网 服务主控室的概览。



用户界面部分与导航路径

物联网 服务主控室由两个主要部分组成：

- “设备管理：”
  - 可以访问远程设备管理服务 (RDMS) 以查看、添加和删除“消息类型”、“设备类型”和“设备”
  - 其中包含了对应三个主要 RDMS 实体的三个磁贴；每个磁贴的右上角显示了可用的实体数：
    - “消息类型”：列出所有消息类型
    - “设备类型”：列出所有设备类型

    - “设备”：列出所有注册的设备
- “消息管理：”
    - “部署消息管理服务”：显示 MMS 的部署页面
    - “发送和查看消息，或执行其他操作”：显示 MMS 的核心服务和数据服务

设备管理

- 磁贴充当三个不同导航路径的入口点。
- 您可以通过选择主页右上角的（刷新）按钮，或主从式架构页面上搜索字段旁的（刷新）按钮，来刷新应用程序。
- 在每个路径中，您可以检查三个 RDMS 实体类型（“设备类型”、“消息类型”或“设备”）之一的相关信息：
    - 左侧的主列表列出了选中的实体类型的所有实例；主列表标题指示显示的实体类型。
    - 主列表左上角的（返回）按钮可以导航回主页。
    - 您可以通过在主列表顶部的搜索字段中输入字符串来在其中进行搜索。当前您只能按名称进行搜索。
    - 右侧的详细信息页面显示了主列表中选择的项目的详细信息。可以下钻至已连接的实体，例如您可以显示某台设备的设备类型详细信息。详细信息页面左上角的（返回）按钮会导航回之前的详细信息页面（请参阅下方的第 25 页上的“下钻导航路径”）。
    - 详细信息页面标题指示当前显示的实体的类型。
    - 当您进行下钻时，主列表会保留在主页上选中的实体类型的上下文。要切换至其他类型，请使用左上角的主列表（返回）按钮来导航回主页，然后选择相应实体类型的磁贴。
    - 在主列表的底部您将会看到用于创建新实体的（创建）按钮，您还可以“删除”所选实体。

      > **i 注意**
      >
      > 请注意，设备类型、消息类型和设备的数量是有限的。

- 可以使用下列**下钻导航路径**：
    - “消息类型”
    - “设备类型”→“消息类型”
    - “设备”→“设备类型”→“消息类型”
- 这些导航路径中的每个步骤都分配了一个唯一的 URL，您可以将其加入到书签，并在浏览器中打开。
- 删除实体：
    - 可以在不受限制的情况下删除设备。
    - 如果此特定设备类型的设备仍然存在，则无法删除该设备类型。
    - 如果存在含有分配到此特定消息类型的消息类型的设备类型，则无法删除该消息类型。
- 故障排除：
    - 加载错误：当您启动应用程序或刷新 物联网 服务主控室时，加载过程可能会由于连接问题而失败。
    - 应用程序加载错误：当您使用包含实体特定的路径（请参阅上面的第 25 页上的“下钻导航路径”）的书签 URL 启动或重新加载应用程序时，如果实体不再可用（即如果该实体已被删除），则无法导航到相应的详细信息页面。
    - 验证：创建实体时会自动验证您的输入，并且只有当您填写了所有必填字段且所有实体均为有效时，才会启用“创建”按钮。如果您的输入无效，则会将相应的输入字段突出显示为红色，并在焦点移动到该输入字段时显示验证消息。

消息管理

- “MMS 部署”支持将 MMS 部署到 SAP BTP 订户账户。选择“部署消息管理服务”，如果没有预先填写，请填写 SAP BTP 账户的所有相关信息（“账户标识”、“用户名”和“密码”）。完成后，选择“部署”。有关详细信息，请参阅第 16 页上的“通过 物联网 服务主控室部署和更新 MMS”部分。

文档、支持、设置、关于 IoT 服务和注销框

在 物联网 服务主控室右上角，可以访问 物联网 服务的“文档”。选择用户名旁的下拉框。

如果您遇到问题并且需要关于获取帮助的信息，请选择“支持”。随即将您重新定向到提供账户类型特定支持的页面。

选择“设置”以切换为您的首选语言和主题。当前支持语言为英语、德语、日语和中文，支持的主题为蓝水晶（缺省）和高对比黑色。

“关于 *IoT* 服务”对话框提供以下信息：

- 关于 物联网 服务主控室、远程设备管理服务和消息管理服务的版本信息。
- 有关可以建模并实际使用多少设备管理实体的账户配额信息。
- 在何处访问设备管理 REST API 端点。

上述选项也可通过 MMS 主控室中的下拉框进行访问。如果您遇到了问题，可以在此处选择访问“文档”并获取“支持”。此外，可以选择“设置”切换成您的首选语言和主题。您可以在“关于”对话框中查找 MMS 版本。最后，您可以选择“注销”。

请参阅 部分以获取任何可能存在的问题的答案。

相关信息

第 16 页上的“部署消息管理服务 (MMS)”
第 16 页上的“通过 物联网 服务主控室部署和更新 MMS”
第 23 页上的“文档、支持、设置、关于 IoT 服务和注销框”
第 27 页上的“配置消息类型”
第 29 页上的“配置设备类型”
第 31 页上的“配置设备”
第 38 页上的“远程设备管理服务 API”
第 67 页上的“消息管理服务 API”

6.1 配置消息类型

前提条件

已为您分配 *"IoT-User"* 角色（请参阅 第 14 页上的"为 物联网 服务分配角色" 部分）并且您具备访问 物联网 服务主控室的权限（请参阅 第 15 页上的"访问 物联网 服务主控室" 部分）。

6.1.1 创建消息类型

背景信息

每个消息类型定义了一个消息格式，其中包含可以由设备发送或接收的消息字段。缺省情况下，创建页面打开时"字段"表包含两个字段：一个"时间戳"字段和一个空字段。如果不需要，可以将"时间戳"字段移除。

> **i 注意**
>
> 可通过将消息字段拖放到期望的位置，更改其顺序。

> **示例**
>
> 设备可以发送其电量消耗。该消息可以包含三个字段，创建消息时的时间戳、安培和电压。在这种情况下，需要创建一个名为"电量消耗"的消息类型，并添加三个字段（一个时间戳字段、两个分别针对安培和电压的双精度字段）。

> **i 注意**
>
> 请注意消息类型的数量是有限的。有关详细信息，请参阅第 23 页上的"文档、支持、设置、关于 IoT 服务和注销框"部分。

过程

1. 在 物联网 服务主控室中选择 "消息类型"磁贴。
2. 在屏幕底部选择 + （创建） 按钮，以添加一个新的消息类型。
3. 输入消息类型的"名称"。

4. 为“字段”表的第一行输入“名称”并选择一个“类型”。

> **i 注意**
>
> 针对“字符串”类型，显示一个可选设置字段，用于设置相应消息字段最大可以设置的字符串长度。最大长度必须为 1 到 5000 之间的数字，缺省值为 255。

5. 可选：选择“字段”表右上角的 “+” 按钮添加附加“字段”。
6. 选择“创建”以继续。

相关信息

第 23 页上的“文档、支持、设置、关于 IoT 服务和注销框”
第 28 页上的“删除消息类型”

6.1.2 删除消息类型

背景信息

可以在不受限制的情况下删除设备和消息类型。如果此特定设备类型的设备仍然存在，则无法删除该设备类型。

> **i 注意**
>
> 只要分配到一个设备类型，消息类型就无法删除。

过程

1. 在 物联网 服务主控室中选择 “消息类型”磁贴。
2. 选择相应消息类型来显示“信息”和“字段”。
3. 在屏幕底部选择“删除”按钮以删除消息类型。
4. 选择“删除”来确认删除操作，如果不想删除消息类型，选择“取消”。

6.2 配置设备类型

前提条件

已为您分配 *"IoT-User"* 角色（请参阅 第 14 页上的"为 物联网 服务分配角色" 部分）并且您具备访问 物联网 服务主控室的权限（请参阅 第 15 页上的"访问 物联网 服务主控室" 部分）。

6.2.1 创建设备类型

背景信息

设备类型指定共享相同规范的一组设备。在 物联网 服务的上下文中，设备类型主要通过其支持的消息类型指定。有关详细信息，请参阅第 27 页上的"创建消息类型"部分。使用消息类型分配将消息类型分配到设备类型中。消息类型分配包含消息类型名称、消息方向和可选分配名称。

> **i 注意**
>
> 请注意设备数量是有限的。有关详细信息，请参阅第 23 页上的"文档、支持、设置、关于 IoT 服务和注销框"部分。

> **i 注意**
>
> 当前，您仅可以在创建设备类型时将消息类型分配到设备类型。因此，您必须在创建设备类型之前创建所有必需的消息类型。如果没有可用的消息类型，创建设备类型页面将不会打开。您必须向新设备类型分配至少一个消息类型。

过程

1. 在 物联网 服务主控室中选择 "设备类型"磁贴。
2. 在屏幕底部选择 + （创建） 按钮，以添加一个新的设备类型。
3. 为设备类型的输入"名称"。
4. 可选：输入具有"更多详细信息"（如自定义配置用户界面或文档页面）的外部应用程序或 Web 页面的 URL。该 URL 在设备类型详细信息视图中将显示为可点击的链接。
5. 如果为 SAP BTP 账户启用了 beta 功能，则"设备注册"部分将可见。选择设备注册期间将使用的所需"验证类型"。有关设备注册验证类型的详细信息（*"OAuth"* 或"客户端证书"），请参阅第 40 页上的"设备类型"部分。

6. 选择消息类型表的 “+” 按钮来分配消息类型。
    a. 可选：输入“分配名称”。如果为空，将显示消息类型名称作为分配名称。
    b. 选择“消息类型”。每个消息类型只能被分配一次到特定的设备类型。
    c. 选择消息“方向”。
7. 选择“创建”以继续。

相关信息

第 23 页上的“文档、支持、设置、关于 IoT 服务和注销框”
第 31 页上的“删除设备类型”
第 27 页上的“创建消息类型”
第 40 页上的“设备类型”

6.2.2 为设备类型重新生成 OAuth 令牌

先决条件

> **⚠ 警告**
>
> 进行以下步骤时，当前的 OAuth 访问令牌将会被撤销，且不可恢复。属于此设备类型的所有设备必须使用新令牌进行后续注册。令牌更新后，将无法恢复旧令牌或用其注册设备。必须确认此步骤才能生成新令牌。

以 SAP ID 用户身份登录到 物联网 服务主控室，且设备类型使用 *“OAuth”* 作为验证类型。

背景信息

特定设备的 OAuth 令牌丢失，或需要更改。

过程

1. 在 物联网 服务主控室中选择“设备类型”磁贴。
2. 选择相应的设备类型。
3. 选择详细信息视图中的“验证”标签。
4. 选择“生成令牌”按钮。
5. 从 *“OAuth 访问令牌”*弹出窗口中复制令牌并将其安装到设备。从设备中移除旧令牌。

6.2.3 删除设备类型

背景信息

可以在不受限制的情况下删除设备和消息类型。

> **i 注意**
>
> 如果此特定设备类型的设备仍然存在，则无法删除该设备类型。删除设备类型后，将自动删除所有相关消息类型。

过程

1. 在 物联网 服务主控室中选择 “设备类型”磁贴。
2. 选择相应的设备类型。
3. 在屏幕底部选择“删除”按钮以删除设备类型。
4. 选择“删除”来确认删除操作。

6.3 配置设备

前提条件

已为您分配 *“IoT-User”* 角色（请参阅 第 14 页上的“为 物联网 服务分配角色” 部分）并且您具备访问 物联网 服务主控室的权限（请参阅 第 15 页上的“访问 物联网 服务主控室” 部分）。您已为想要设置的设备配置设备类型。

6.3.1 创建设备

先决条件

为 物联网 注册设备始终需要一个设备类型。有关如何创建设备类型的详细信息，请参阅第 29 页上的“创建设备类型”部分。

背景信息

设备是可以发送或接收消息的物理对象。

> **i 注意**
>
> 请注意设备数量是有限的。有关详细信息，请参阅第 51 页上的“设备”部分。

过程

1. 在 物联网 服务主控室中选择“设备”磁贴。
2. 在屏幕底部选择 + （创建） 按钮，以添加一个新的设备。
3. 输入设备的“名称”。
4. 在下拉菜单中选择新设备的“设备类型”。
5. 可选：输入具有“更多详细信息”（如自定义配置用户界面或文档页面）的外部应用程序或 Web 页面的 URL。该 URL 在设备详细信息视图中将显示为可点击的链接。

   > **i 注意**
   >
   如果所选的设备类型具有定义的链接，且设备本身的链接为空，则会继承设备类型的链接。有关详细信息，请参阅子部分。
   >
   可选：在“自定义属性”表中可以添加附加属性，每个属性包含一个键和一个值。要添加新属性，选择“添加自定义属性”按钮。如果要在属性表中删除行，选择相应行末尾的垃圾桶图标按钮。有关最大属性数量的详细信息，请参阅第 51 页上的“设备”部分。

6. 选择“创建”以继续。

   > **i 注意**
   >
   根据相应设备类型的设备注册验证类型，设备对验证的通信使用 *“OAuth”* 或“客户端证书”。如果验证类型是 *“OAuth”*，系统会显示具有 OAuth 访问令牌的弹出窗口，其中显示针对新设备生成的令牌标识。如果验证类型是“客户端证书”，设备必须从远程设备管理服务请求证书。有关设备客户端证书的详细信息，请参阅第 51 页上的“设备”部分。

7. 可选：复制生成的设备令牌，在设备上进行安全通信需要该令牌。

相关信息

第 23 页上的“文档、支持、设置、关于 IoT 服务和注销框”
第 29 页上的“创建设备类型”
第 33 页上的“编辑自定义属性”
第 34 页上的“为设备重新生成 OAuth 令牌”
第 51 页上的“设备”

6.3.2 编辑自定义属性

您可以编辑设备的自定义属性。还可以添加附加属性，并且更改或删除现有属性。

背景信息

> **i 注意**
>
> 目前只可以编辑自定义属性。“名称”，“设备类型”，和“更多详细信息”中的条目不可编辑。

过程

1. 在 物联网 服务主控室中选择 “设备”磁贴。
2. 在主列表中选择设备。
3. 在设备详细信息页面屏幕的底部选择“编辑”按钮。
4. 根据需要更改“自定义属性”。有关详细信息，请参阅第 32 页上的“创建设备”部分中的步骤 6。
5. 选择“保存”来保存更改。

相关信息

第 32 页上的“创建设备”

6.3.3 为设备重新生成 OAuth 令牌

先决条件

> **⚠ 警告**
>
> 进行以下步骤时，当前的 OAuth 访问令牌将会被撤销，且不可恢复。必须使用新令牌更新所有受影响的设备。

以 SAP ID 用户身份登录 物联网 服务主控室。

背景信息

特定设备的 OAuth 令牌丢失，或需要更改。

过程

1. 在 物联网 服务主控室中选择 “设备”磁贴。
2. 选择相应的设备。
3. 选择详细信息视图中的“验证”标签。
4. 选择“生成令牌”按钮。
5. 从 “*OAuth* 访问令牌”弹出窗口中复制令牌并将其安装到设备。从设备中移除旧令牌。

6.3.4 删除设备

背景信息

可以在不受限制的情况下删除设备和消息类型。

过程

1. 在 物联网 服务主控室中选择 “设备”磁贴。

2. 选择相应的设备。
3. 在屏幕底部选择“删除”按钮以删除设备。
4. 选择“删除”来确认删除操作，如果不想删除设备，选择“取消”。

6.3.5 层次结构视图

除了扁平列表，RDMS 还允许您在层次结构中安排和查看设备。通过 物联网 服务主控室，您可以构建新的层次结构，将设备链接到层次结构元素，以及删除层次结构元素和层次结构。

浏览层次结构

要进入层次结构视图模式，请选择“设备”磁贴，并选择屏幕底部的（显示）按钮。设备列表随即由树替代。如果没有可用的层次结构，则树显示为空。位于树顶部的下拉框会显示可用的层次结构列表，并允许您在其中切换。树则会显示所选层次结构的层次结构元素。您可以通过选择树节点，或选择层次结构元素详细信息页面上“子项”列表中的项目，来浏览层次结构。随即会自动展开选中的树节点。

详细信息页面顶部提供了位置指示节点导航，允许您在当前显示元素的祖先路径中轻松导航。要查看层次结构自身的详细信息页面，请通过选择选中的树节点或选择层次结构元素的位置指示节点控件中的根链接来取消选中任何所选树节点。选择设备将打开设备详细信息页面。详细信息页面左上角中的箭头按钮会带您返回到当前层次结构的父项元素。

> **i 注意**
>
> 如果您进一步下钻到设备类型，将会离开层次结构模式。

创建和删除层次结构和层次结构元素

要创建新的层次结构，选择主视图页脚中的＋（创建）按钮，然后选择“新建层次结构”。层次结构视图模式中的＋（创建）按钮提供了两个选项：

- “新建层次结构”：打开用于创建新层次结构的页面。
- “新建设备”：打开用于创建新设备的页面。该页面即您在扁平列表视图模式下选择“设备”视图中的＋（创建）按钮时看到的页面。

> **i 注意**
>
> 这将创建一台新设备，不应该与链接设备到层次结构元素相混淆。另请注意，当前版本中，选择此选项时视图模式会切换到扁平视图模式。

在“新建层次结构”页面上，输入新层次结构的“名称”，然后选择“创建”按钮。要删除层次结构，在下拉框中选择层次结构，并确保未选择任何层次结构元素。详细信息页面必须显示层次结构的详细信息。之后，选择主视图页脚中的“删除”按钮。

> **i 注意**
>
> 随即会删除（在层次结构元素情况下）或取消链接（在设备情况下）所有层次结构的子项。

要创建新的层次结构元素，选择父项详细信息页面上的“子项”列表上方的“添加元素”按钮。输入新层次结构元素的名称，然后选择“创建”按钮。要添加或删除新的层次结构元素的子项，选择树或父项的“子项”列表中的元素。这将打开层次结构元素的详细信息页面。之后，按照上述步骤继续操作。要删除层次结构元素，选择树中的元素，并选择主视图页脚中的“删除”按钮。或者选择元素的父项，通过激活其复选框选择“子项”列表中的层次结构元素，然后选择列表上方的“删除”按钮。

> **i 注意**
>
> 您可以选择多个层次结构元素，然后一次将其删除。

链接和取消链接设备

要将设备链接到层次结构元素，选择层次结构元素“子项”列表上方的“链接设备”按钮。随即显示选择对话框，其中显示了尚未链接到此层次结构元素的所有设备。选择要链接的设备，然后选择“确定”。要取消链接一台设备，请通过激活其复选框，选择设备父项的“子项”列表中的设备，然后选择列表上方的“删除”按钮。您可以选择多台设备。

> **i 注意**
>
> 这些设备不会在 RDMS 中删除；它们只是从层次结构元素的子项中取消链接。您依然可以在扁平设备视图中找到这些设备，并可将其重新链接到该层次结构元素或其他层次结构元素。

搜索节点

位于树上方的搜索栏允许您对树进行过滤。搜索不区分大小写，并且仅适用于元素和设备名称。请注意，如果所选节点未包含在搜索结果中，则会自动取消选中所选节点。

相关信息

第 32 页上的“创建设备”

6.3.6 可视化设备数据

背景信息

可在 MMS 主控室内将设备数据可视化为基于时间的图表。

过程

1. 在 物联网 服务主控室中选择 “设备”磁贴。
2. 选择相应的设备。
3. 在屏幕底部选择“打开图表”按钮以导航到在新浏览器标签中打开的图表视图。
4. 选择与 （表视图） 中的相应设备对应的数据表 *“T_IOT_HTTP_PUSH”*。或者，如果想将数据显示在图表中，请选择 （折线图） 用基于时间的方式绘制数据。

7 远程设备管理服务 API

> **⚠ 警告**
>
> 远程设备管理服务 (RDMS) 的 **API v1** 的内容已弃用。请按照第 38 页上的“API v2 介绍”部分中的建议为 RDMS 使用 **API v2**。

> **i 注意**
>
> 请注意，本文档中消息管理服务 (MMS) 的当前 API 版本为 'v1'。有关 MMS API 的详细信息，请参阅第 72 页上的“配置 API 验证”部分。

7.1 API v2 介绍

此部分描述模型设备和设备类型使用的语法，包括支持的消息。其中介绍用于注册的 JSON 模式以及示例与设备发送的可能消息实例。

下表对术语进行了详细描述：

术语	定义
数据类型	每个消息字段必须属于特定数据类型，如整数或字符串。物联网 服务上下文中的数据类型是代表字符、字符串或数值数据的原始数据类型。支持的数据类型预先进行定义，不能修改。
设备	设备是可以在 物联网 服务中注册的物理对象。注册成功后可以发送或接收消息。
设备标识	设备标识是 物联网 服务生成的标识符并且是唯一标识设备的 GUID 。
设备名称	设备名称在注册过程期间分配为人工可读的标识符。由于设备标识是 GUID 并且按平台分配，因此用户可以通过名称识别设备。名称的使用完全没有限制，但必须在所有注册设备之中具有唯一性。
设备类型	设备类型指定具有相同规格的设备组。在 物联网 服务中，设备类型主要通过其支持的消息类型指定。
消息字段	消息字段定义消息类型的一部分。消息字段描述具备特定属性的消息的特性，如字段长度、数据类型、位置和字段名称。
消息类型	消息类型定义设备发送或接收的消息中包含的数据类型。一个消息包括一个或多个消息字段。

技术

当前 API 仅限使用下列技术：

- HTTPS / REST
- JSON 文档
- 支持的 HTTPS 调用 `GET`、`POST`、`DELETE`（特定情况下包括 `PATCH`）

API 版本

通过指定 URL 中 /<api> 之前的 <version> 部分可以获得 API。

```
https://<host>/com.sap.iotservices.dms/<version>/<api>/<entity>
```

本文档指定的当前 API 版本为 'v2'。推荐使用最新版本。

您可以在 物联网 服务主控室通过右上角用户名旁边的"关于"对话来访问 API 端点。有关详细信息，请参阅第 23 页上的"文档、支持、设置、关于 IoT 服务和注销框"部分。

API 定义包含

- JSON 模式和示例
- HTTPS 端点定义
- 响应标准 HTTPS 状态代码

安全性

通过用户验证和 OAuth2 以及客户端证书来保护 API。如没有另外说明，主保护机制为用户验证（基础验证）。部分 API 也可以通过 OAuth2 和客户端证书受到保护，例如启用设备的自动注册。此内容在 API 文档的相关部分中明确提及，以防主要机制不是基本验证。指定基本验证作为部分 HTTP 标头。基本验证的原理遵循 RFC 2617 中的描述。

授权：基本 <base64 编码用户名:密码>

相关信息

第 23 页上的"文档、支持、设置、关于 IoT 服务和注销框"
第 24 页上的"物联网 服务主控室"

7.1.1 设备类型

设备类型指定具有相同规格的的设备组。在 物联网 服务中，此规格被理解为发送、接收和处理特定消息类型的功能。有关如何定义设备类型的消息类型的信息，请参阅第 48 页上的“消息类型”部分。

> **i 注意**
>
> 在 物联网 服务中注册设备类型之前，您必须指定设备类型。

端点

访问所有设备类型：

`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes`

访问特定设备类型：

`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识>`

7.1.1.1 注册新设备类型

定义以下属性来注册新设备类型：

- 设备类型的 `name`
  - 该名称在所有设备类型中必须是唯一的。
  - 最大长度为 255 个字符
- 自动分配设备类型的 `id`
  - 由下列字符组成的哈希值：0-9 a-f
  - 长度为 20 个字符
- `link` 是可以设置的可选属性。
  - 必须是以 http/https 开头的有效 URL。
  - 最大长度为 1,024 个字符。

设备类型注册示例

您可以将设备类型添加到集合，如下所示：

`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes` 上的 `POST`

请求正文：

```
{
  "name": "Device Type 1",
  "link": "http://manufacturer.com/products/dt1"
}
```

响应正文：

```
{
  "id": "a3b7cc7890afb2673eed",
  "name": "Device Type 1",
  "link": "http://manufacturer.com/products/dt1",
  "messageTypes": [],
  "authentication": {
    "type": "oauth",
    "token": "u2jrvvqaathb3bl4dp6dfanw54dl0x"
  }
}
```

JSON 模式验证

```
{
  "$schema":"http://json-schema.org/draft-04/schema#",
  "title":"New Device Type",
  "description":"Registering a new device type",
  "type":"object",
  "properties":{
    "name":{
      "description":"the name of the device type",
      "type":"string",
      "pattern":"^\\S$|^\\S[\\s|\\S]*\\S$",
      "maxLength":255
    },
    "link":{
      "type":"string",
      "maxLength":1024,
      "pattern":"^https?://.*"
    },
    "authentication":{
      "type":"object",
      "properties":{
        "type":{
          "type":"string",
          "enum":[
            "oauth",
            "clientCertificate"
          ]
        }
      }
    }
  },
  "required":[
    "name"
  ]
}
```

7.1.1.2 为使用客户端证书注册新的设备类型 (BETA)

> **⚠ 警告**
>
> 在 物联网 服务中使用客户端证书是一种 beta 功能。要使用此功能，RDMS 的相应账户必须为启用了 beta 的账户。有关详细信息，请参阅 SAP BTP 文档中的在子账户中使用 Beta 功能部分。

定义以下属性来注册新设备类型：

- 设备类型的 `name`
  - 该名称在所有设备类型中必须是唯一的。
  - 最大长度为 255 个字符。
  - 名称属性是可选属性。
- 自动分配设备类型的 `id`。
  - 由下列字符组成的哈希值：0-9 a-f
  - 长度为 20 个字符
- `link` 是可以设置的可选属性。
  - 必须是以 http/https 开头的有效 URL。
  - 最大长度为 1,024 个字符。
- `authentication` 是使用的验证机制。
  - 必须将 `type` 属性设置为 `clientCertificate`。

设备类型注册示例

您可以将设备类型添加到集合，如下所示：

`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes` 上的 `POST`

请求正文：

```
{
  "name": "Device Type 1",
  "link": "http://manufacturer.com/products/dt1",
  "authentication": {
    "type": "clientCertificate"
  }
}
```

响应正文：

```
  {
    "id": "273e5b736a9af59689ba",
    "name": "Device Type 1",
    "link": "http://manufacturer.com/products/dt1",
    "messageTypes": [],
    "authentication": {
      "type": "clientCertificate",
      "p12File": "MIACAQMwgAYJKoZIhvc…",
      "secret": "kdCG2SvwjKc4ftD"
    }
}
```

响应包括使用 JSON 响应中包含的密码进行加密的 base64 编码的 PKCS #12 文件。PKCS #12 文件包含设备类型证书，作为 X.509 证书及其私钥。

7.1.1.3 分配并移除消息类型

创建新设备类型时，系统会为 messageTypes 属性返回空的数组。使用该属性，您可以通过将现有消息类型分配到该设备类型来进一步指定设备类型。

要分配消息类型，需指定三种属性：

- id 是参考的消息类型的标识；这意味着必须存在含有该标识的消息类型
- name
  - 对于相同的设备类型，名称必须是唯一的。
  - 最大长度为 255 个字符。
  - 名称属性是可选属性。
  - 该属性可用于覆盖消息类型名称（例如将“灯”覆盖为“闪光灯”）。
- direction
  - 必须指定方向，并描述消息流的方向。
  - 有效值如下：fromDevice、toDevice、bidirectional。
  - 该属性基本上描述了设备的性质，例如是传感器或传动装置。

分配消息类型的示例

可以使用以下路径，将消息类型分配到设备类型：

https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识>/messageTypes 上的 POST

请求正文：

```
{
  "id": "c763433cee2673eedf34",
  "name": "My message type",
  "direction": "fromDevice"
}
```

响应正文：

```
  {
    "id": "a3b7cc7890afb2673eed",
    ...
    "messageTypes": [
      {
        "id": "c763433cee2673eedf34",
        "name": "My message type",
        "direction": "fromDevice"
      }
    ],
}
```

创建设备类型时分配消息类型的示例

除了将消息类型分配到已存在的设备类型之外，还可以在新建设备类型时分配消息类型。这需要所有消息类型可用；否则尝试新建设备类型将失败。

要在新建设备类型时分配消息类型，必须根据上述示例设置 `messageTypes` 属性。对于每个消息类型，必须指定 `id` 和 `direction`。名称是可选属性。

`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes` 上的 `POST`

请求正文：

```
  {
    "name": "Device Type 1",
    "link": "http://manufacturer.com/products/dt1"
    "messageTypes": [
      {
        "id": "c763433cee2673eedf34",
        "name": "My message type",
        "direction": "fromDevice"
      }
    ]
}
```

移除消息类型的示例

可以使用与 post 相同的路径从设备类型中删除消息类型：

成功之后，`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识>/messageTypes/<标识>` 上的 `DELETE` 会返回 HTTP 响应 `204`。

7.1.1.4 读取设备类型

读取所有设备类型

`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes` 上的 `GET`（正文为空）

响应正文会返回设备类型对象的数组：

```
[
  {
    "id": "a3b7cc7890afb2673eed",
    "name": "Device Type 1",
    ...
  },
  {
    "id": "afb2673eeda3b7cc7890",
    "name": "Device Type 2",
    ...
  }
]
```

读取特定设备类型

Https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识> 上的 GET（正文为空）

响应正文会返回单个设备类型对象：

```
{
  "id": <id,
  "name": "Device Type 1",
  ...
}
```

7.1.1.5 下载特定设备类型的 PKCS #12 文件

https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识>/authentication/p12File 上的 GET（正文为空）。响应是将媒体类型设置为 application/octet-stream 的二进制 PKCS #12。建议文件名为 <device type id>.p12。

7.1.1.6 删除设备类型

成功之后，https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识> 上的 DELETE（正文为空）返回 HTTP 响应 204。

7.1.1.7 修改设备类型验证

成功之后，https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<id>/authentication 上的 DELETE（正文为空）返回 HTTP 响应 204。此调用会删除设备类型的验证对象。

> **i 注意**
>
> 设备注册令牌将失效，无法再用于不使用用户凭据注册设备。

删除验证对象之后，可以使用

https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识>/authentication 上的 POST 来创建新对象

调用必须在消息正文中指定验证类型。响应正文包含设备类型和所有相关的属性。

请求正文：

```
{
  "type": "oauth
}
```

响应正文：

```
   {
     "id": <id>,
     "name": "Device Type 1",
     "link": "http://manufacturer.com/products/dt1",
     "messageTypes": [],
     "authentication": {
        "type": "oauth",
        "token": "u2jrvvqaathb3bl4dp6dfanw54dl0x"
     }
}
```

令牌也可以在单个步骤中使用

`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识>/authentication/token` 上请求正文为空的 `POST` 进行更新。

响应包含更新的验证对象。

```
   {
     "type": "oauth",
     "token": " u2jrvvqaathb3bl4dp6dfanw54dl0x"
}
```

7.1.1.8 修改客户端证书的设备类型验证 (BETA)

> **⚠ 警告**
>
> 在 物联网 服务中使用客户端证书是一种 beta 功能。要使用此功能，RDMS 的相应账户必须为启用了 beta 的账户。有关详细信息，请参阅 SAP BTP 文档中的在子账户中使用 Beta 功能部分。

成功之后，`https://<主机>/com.sap.iotservices.dms/v2/api/deviceTypes/<标识>/authentication` 上的 `DELETE`（正文为空）返回 HTTP 响应 `204`。此调用会删除设备类型的验证对象。

删除验证对象之后，可以使用

在 `https://<host>/com.sap.iotservices.dms/v2/api/deviceTypes/<id>/authentication` 上 `POST`

调用必须在消息正文中指定验证类型。响应正文包含设备类型和所有相关的属性。

请求正文：

```
{
  "type": "clientCertificate"
}
```

响应正文：

```
   {
     "id": <id>,
     "name": "Device Type 1",
     "link": "http://manufacturer.com/products/dt1",
     "messageTypes": [],
     "authentication": {
```

```
        "type": "clientCertificate",
        "p12File": "MIACAQMwgAYJKoZIhvcNAQcBoIAkgASC…",
        "secret": " kD9gbxI3ouTkPGy"
    }
}
```

还可以在单个步骤中使用

在 https://<host>/com.sap.iotservices.dms/v2/api/deviceTypes/<id>/authentication/p12File 上以空的请求正文 POST

响应包含更新的验证对象。

```
  {
    "type": "clientCertificate",
    "p12File": "MIACAQMwgAYJKoZIhvcNAQcBoIAkgASCBAAwgDCABgkqhkiG9w0BBw…",
    "secret": "E5WM3SlvvDAEs8uh"
}
```

7.1.2 数据类型

物联网 服务支持消息类型的自由定义。因此用户可以根据消息类型部分的描述为设备定义其自己的消息格式。数据类型实体中保留受支持数据类型的定义。数据类型包括名称和标识。数据类型到合适数据库数据类型的映射通过名为消息管理服务 (MMS) 的另一服务执行。

有关如何使用数据类型定义消息类型的信息，请参阅第 48 页上的“消息类型”部分。

> **i 注意**
>
> 数据类型预先进行配置，不能更改。

端点

访问数据类型：

https://<host>/com.sap.iotservices.dms/v2/api/dataTypes

读取可用数据类型

https://<主机>/com.sap.iotservices.dms/v2/api/dataTypes 上的 GET（正文为空）

响应正文（示例）：

```
[
  {
    "id": 1,
    "name": "double"
  },
  {
```

```
    "id": 2,
    "name": "integer"
  }
]
```

支持的数据类型与数据库（支持范围等同于对应 Java 数据类型的定义。）：

支持的数据类型	SAP HANA	SAP ASE	SAP MaxDB	更多详细信息
integer	INTEGER	INT	INTEGER	-
long	BIGINT	BIGINT	FIXED(38)	-
float	FLOAT	FLOAT	FLOAT	-
double	DOUBLE	DOUBLE PRECISION	DOUBLE PRECISION	-
boolean	TINYINT	SMALLINT	SMALLINT	-
date	TIMESTAMP	DATETIME	TIMESTAMP	时间戳和日期可以作为 ISO 8601 兼容字符串或 Unix 时间进行发送，自 1970 年 1 月 1 日起表示为秒数。
binary	BLOB	IMAGE	BLOB	二进制内容可作为 Base64 编码字符串发送。因为消息进行批处理然后存储在内存中等待进一步处理，所以我们建议不要发送大于 50 MB 的二进制内容。
string	VARCHAR | CLOB	VARCHAR | TEXT	VARCHAR | CLOB	-

7.1.3 消息类型

在设备可以向 物联网 服务发送消息之前，必须定义设备类型，并且需要指定设备类型支持的消息并将其分配至设备类型。

第 40 页上的“设备类型”和第 51 页上的“设备”部分定义了设备类型定义和设备注册。

端点

访问消息类型：

`https://<主机>/com.sap.iotservices.dms/v2/api/messageTypes`

注册新消息类型

以下是注册新消息类型的必备属性：

- 新消息类型的唯一名称
  - 最大长度为 255 个字符
- 消息类型的字段
  - 必须填写值名称
    - 最大长度为 30 个字符
    - 有效字符 [a-zA-Z0-9_]
    - 不允许使用空格
    - 消息类型的字段名称必须唯一且使用小写字母，例如 'timestamp' 和 'TimeStamp' 是相同的名称
  - 必须填写字段位置
    - 必须为以 1 开头的数字
    - 字段之间不得包含任何间隙（例如，1,2,5 无效）
  - 必须填写数据类型
  - 可以选择性地指定长度属性
  - 最大字段数为 100

消息类型注册示例

`https://<主机>/com.sap.iotservices.dms/v2/api/messageTypes` 上的 `POST`

以下示例显示了包含电压和安培字段的新消息类型电量消耗的注册。

请求正文：

```
{
  "name": "Power Consumption",
  "fields": [
     {
        "position": 1,
        "name": "voltage",
        "type": "double"
     },
     {
        "position": 2,
        "name": "ampere",
        "type": "double"
     }
  ]
}
```

响应正文：

成功之后，返回包含已由平台自动分配的标识的相同定义。

```
{
  "id": "c763433cee2673eedf34",
  "name": "Power Consumption",
  ...
}
```

JSON 模式验证

以下定义介绍了基于 JSON 模式的消息类型注册。此 JSON 模式可用于验证传入消息类型注册。

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "New Message Type",
  "description": "Registering a new message type",
  "type": "object",
  "properties": {
    "name": {
      "description": "the name of the message type",
      "type": "string",
      "maximum": 255
    },
    "fields": {
      "description": "and array of message fields",
      "type": "array",
      "items": {
        "title": "message field",
        "type": "object",
        "properties": {
          "position": {
            "description": "specifies the position of the field in the message",
            "type": "integer",
            "minimum": 1
          },
          "name": {
            "description": "defines the name of the message field",
            "type": "string"
          },
          "type": {
            "description": "defines the data type of the field",
            "type": "string"
          },
          "length": {
            "description": "defines the length of the field",
            "type": "integer",
            "minimum": 1
          }
        },
        "required": [
          "position",
          "name",
          "type"
        ]
      }
    }
  },
  "required": [
    "name",
    "fields"
  ]
}
```

该模式定义目前缺少：

- 有效数据类型的枚举
- 支持字段数量的限制

读取消息类型

读取所有消息类型

https://<主机>/com.sap.iotservices.dms/v2/api/messageTypes 上的 GET（正文为空）

响应正文返回包含所有字段定义的消息类型对象数组：

```
[
  {
     "id": "c763433cee2673eedf34",
     ...
  },
  {
     "id": "2673eedf34c763433cee",
     ...
  }
]
```

读取特定消息类型

https://<主机>/com.sap.iotservices.dms/v2/api/messageTypes/<id> 上的 GET（正文为空）

响应正文会返回单个消息类型对象：

```
{
  "id": "c763433cee2673eedf34",
  ...
}
```

删除消息类型

成功之后，https://<主机>/com.sap.iotservices.dms/v2/api/messageTypes/<标识> 上的 DELETE（正文为空）返回 HTTP 响应 204。

7.1.4 设备

设备是可以发送或接收消息的物理对象。

> **i 注意**
>
> 在注册设备之前，必须定义设备类型。

端点

访问设备实体：

https://<host>/com.sap.iotservices.dms/v2/api/devices

7.1.4.1 注册新设备

为 物联网 服务注册设备始终需要以下条件：

- 按照标识提供的现有 `deviceType` 的参考
- 可读格式的 `name`
  - 最大长度为 255 个字符
  - 对于所有其他设备必须是唯一的
  - 可以参考特定设备名称，例如 'Device 1'
- 可选自定义 `attributes`
  - 每个设备最多可拥有 32 个自定义属性，包括键值和值。
  - 键值限制为字母数字字符和下划线，最大长度为 256 个字符。
  - 值可以为任意字符串，最大长度为 4,096 个字符。
- 可选 `link` 属性
  - 必须为以 http/https 开头的有效 URL
  - 最大长度为 1,024 个字符
  - 如果设备类型具有链接而设备没有
    - 设备将从设备类型中继承链接
    - 继承的链接将使用设备标识进行扩展
    - 链接将类似 `<DeviceTypeLink>?device=<deviceId>`
- 可选用户定义 `id`
  - 如果未设置 `id`，系统会自动生成。
  - 对于所有其他设备必须是唯一的。
  - `id` 限制为字母数字和连字符，最大长度为 36 个字符。

> **i 注意**
>
> `id` 属性无法更改，旨在唯一识别系统中的实体。请勿分配描述性设备标识，因为描述和含义可能会随着时间而更改。我们建议您使用自定义设备属性以了解设备特定详细信息。

可以使用不同的验证方法注册设备。除**基本验证**之外，可以使用 **Oauth** 令牌或**客户端证书**添加设备。

7.1.4.2 使用设备类型令牌注册

设备可以使用设备类型令牌自行注册。必须添加此令牌作为具有以下模式的请求标头：

```
Authorization: Bearer <device registration token>
```

> **i 注意**
>
> 设备注册令牌显示在 物联网 服务主控室中各自设备类型的详细信息页面上。也可以使用设备类型端点进行检索。有关详细信息，请参阅第 40 页上的“设备类型”部分。

该响应包含另一个令牌，此令牌用于将数据从设备发送至 MMS。有关详细信息，请参阅第 74 页上的“数据服务”部分。

Https://<主机>/com.sap.iotservices.dms/v2/api/devices 上的 POST

请求正文：

```
{
  "name": "Device 1",
  "deviceType": "a3b7cc7890afb2673eed",
  "link": "https://manufacturer.com/products/device1",
  "attributes": [
      {
          "key": "SerialNumber",
          "value": "SN 5121982812"
      },
      {
          "key": "IPv4",
          "value": "127.0.0.1"
      }
  ]
}
```

响应正文：

```
{
  "id": "936da01f-9abd-4d9d-80c7-02af85c822a8",
  "deviceType": "a3b7cc7890afb2673eed",
  "name": "Device 1",
  "link": "https://manufacturer.com/products/device1",
  "attributes": ...
  "authentication": {
      "type": "oauth",
      "token": <token>
  }
}
```

7.1.4.3 使用基本验证注册

必须将以下标头添加到注册调用：

Authorization: Basic <base64-encoded username:password>

Https://<主机>/com.sap.iotservices.dms/v2/api/devices 上的 POST

请求正文：

```
{
  "name": "Device 1",
  "deviceType": "a3b7cc7890afb2673eed",
  "link": "https://manufacturer.com/products/device1",
  "attributes": [
      {
          "key": "SerialNumber",
          "value": "SN 5121982812"
      },
      {
          "key": "IPv4",
          "value": "127.0.0.1"
```

```
        }
    ]
}
```

响应正文：

```
{
  "id": "936da01f-9abd-4d9d-80c7-02af85c822a8",
  "deviceType": "a3b7cc7890afb2673eed",
  "name": "Device 1",
  "link": "https://manufacturer.com/products/device1",
  "attributes": ...,
  "authentication": {
      "type": "oauth",
      "token": <token>
  }
}
```

7.1.4.4 使用设备类型证书的设备注册 (BETA)

> **⚠ 警告**
>
> 在 物联网 服务中使用客户端证书是一种 beta 功能。要使用此功能，RDMS 的相应账户必须为启用了 beta 的账户。有关详细信息，请参阅 SAP BTP 文档中的在子账户中使用 Beta 功能部分。

设备可以使用设备类型证书自行注册。此证书必须包含在用于 HTTPS 连接的设备密钥库。之后，会在初始 SSL 握手期间将此证书用作客户端证书。有关详细信息，请参阅 SAP Cloud Platform 物联网入门工具包 。

> **i 注意**
>
> 设备注册证书显示在 物联网 服务主控室中相应设备类型的详细信息页面上。也可以使用设备类型端点进行检索。有关详细信息，请参阅第 40 页上的“设备类型”部分。

要使用设备类型证书创建设备，请使用以下 API 调用：

在 `https://<host>/com.sap.iotservices.dms/v2/api/devices` 上 `POST`

`<host>` 必须包含 `cert` 前缀，表示在加密连接期间使用客户端证书。有关架构主机的详细信息，请参阅 SAP BTP 文档中的地区部分。

请求正文：

```
{
  "name":"Device 1",
  "deviceType":"a3b7cc7890afb2673eed",
  "link":"https://manufacturer.com/products/device1",
  "attributes":[
    {
      "key":"SerialNumber",
      "value":"SN 5121982812"
    },
    {
      "key":"IPv4",
      "value":"127.0.0.1"
    }
  ]
```

```
}
```

响应正文：

```
{
  "id": "936DA01F-9ABD-4D9D-80C7-02AF85C822A8",
  "deviceType": "a3b7cc7890afb2673eed",
  "name": "Device 1",
  "link": "https://manufacturer.com/products/device1",
  "attributes": ...
  "authentication": {
      "type": "clientCertificate"
  }
}
```

JSON 模式验证

```
{
  "$schema":"http://json-schema.org/draft-04/schema#",
  "title":"Device Registration",
  "description":"JSON Message to register a new device",
  "type":"object",
  "properties":{
    "name":{
      "description":"device identifier given by the user",
      "type":"string"
    },
    "deviceType":{
      "description":"id of the device type",
      "type":"string"
    },
    "link":{
      "description":"url to an external resource",
      "type":"string"
    },
    "attributes":{
      "description":"array of custom attributes",
      "type":"array",
      "items":{
        "title":"custom attributes",
        "type":"object",
        "properties":{
          "key":{
            "description":"the key of the custom attribute",
            "type":"string"
          },
          "value":{
            "description":"the value of the custom attribute",
            "type":"string"
          }
        }
      }
    },
    "authentication":{
      "description":"object that describes the authentication",
      "type":"object",
      "properties":{
        "type":{
          "description":"security type",
          "type":"string"
        },
        "token":{
```

```
            "description":"token used to send data for the device",
            "type":"string"
          }
        }
      }
    },
    "required":[
      "name",
      "deviceType"
    ]
}
```

7.1.4.5 将自定义属性添加到设备

Https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/attributes 上的 POST（正文为空）

请求：

```
{
  "key": "SerialNumber",
  "value": "SN 5121982812"
}
```

响应会返回包含新属性的整个设备对象。

7.1.4.6 读取已注册设备

读取所有设备

https://<主机>/com.sap.iotservices.dms/v2/api/dataTypes 上的 GET（正文为空）

响应正文返回设备对象的数组：

```
[
  {
    "id": "936da01f-9abd-4d9d-80c7-02af85c822a8",
    "name": "Device 1",
    "deviceType": "a3b7cc7890afb2673eed",
    "link": ...,
    "attributes": ...
  },
  {
    "id": "2326da01f-2abd-3d3d-69c5-92ff84e228c7",
    "name": "Device 2",
    "deviceType": "afb2673eeda3b7cc7890"
    "attributes": ...
  }
  ...
]
```

读取特定设备

https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识> 上的 GET（正文为空）

响应正文返回设备对象：

```
{
  "id": <id>,
  "name": "Device 1",
  "deviceType": "a3b7cc7890afb2673eed",
  "link": ...,
  "attributes": ...
}
```

读取设备的自定义属性

Https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/attributes 上的 GET（正文为空）

响应正文返回自定义设备属性的数组：

```
[
  {
    "key": "SerialNumber",
    "value": "SN 5121982812"
  },
  {
    "key": "IPv4",
    "value": "127.0.0.1"
  }
]
```

7.1.4.7 删除设备数据

删除整个设备

成功之后，https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识> 上的 DELETE（正文为空）返回 HTTP 响应 204。

删除设备的自定义属性

成功之后，https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/attributes/<键值> 上的 DELETE （正文为空）返回 HTTP 响应 204。

7.1.4.8 更新设备数据

自定义设备属性可以随时进行更改。要更新这些属性，必须使用 HTTP 方法 PATCH。

自定义属性上的 PATCH 具有以下含义

- 自定义属性是键值列表
- 相同键值上的 PATCH 更新值
- 非现有键值上的 PATCH 创建新属性
- 具有空值的现有键值上的 PATCH 删除键值对

因此，您可以使用 PATCH 方法删除和添加新的键值对。PATCH 允许用于：https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/attributes 属性数组上的 PATCH 始终返回修改的数组。

示例：

PATCH 请求正文：

```
  [
    {
      "key": "SerialNumber",
      "value": "SN 5121982818"
    },
    {
      "key": "IPv4",
      "value": null
    }
]
```

说明：

如果已存在 'SerialNumber' 键值并且其值不同，将更新该键值。如果数组中不存在该键值，将会添加该键值。如果存在 'Ipv4' 键值，会将其移除，因为该值已设置为空。其他任何现有属性将不受影响。响应将包含所有属性，包括未修改的属性。

7.1.4.9 修改设备验证

成功之后，https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/authentication 上的 DELETE（正文为空）返回 HTTP 响应 。类似于设备端点，此调用删除用于发送传感器数据的设备的验证对象。

删除验证对象之后，可以使用

https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/authentication 上的 POST 创建新的对象。

调用必须在消息正文中指定验证类型 'oauth'。响应正文包含具有所有相关属性的完整设备。

请求正文：

```
{
  "type": "oauth"
}
```

响应正文：

```
{
  "id": "936da01f-9abd-4d9d-80c7-02af85c822a8",
  "name": "Device 1",
  "deviceType": "a3b7cc7890afb2673eed",
  "link": ...,
  "attributes": ...,
  "authentication": : {
     "type": "oauth",
     "token": "u2jrvvqaathb3bl4dp6dfanw54dl0x"
  }
```

```
}
```

令牌也可以在单个步骤中使用

`Https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/authentication/token` 上的 `POST`（正文为空）进行更新。

响应包含更新的验证对象。

```
{
  "type": "oauth",
  "token": " u2jrvvqaathb3bl4dp6dfanw54dl0x"
}
```

7.1.4.10 使用客户端证书创建设备验证 (BETA)

> **⚠ 警告**
>
> 在 物联网 服务中使用客户端证书是一种 beta 功能。要使用此功能，RDMS 的相应账户必须为启用了 beta 的账户。 有关详细信息，请参阅 SAP BTP 文档中的在子账户中使用 Beta 功能部分。

可以使用

`Https://<主机>/com.sap.iotservices.dms/v2/api/devices/<标识>/authentication` 上的 `POST` 创建对象

其中 `<host>` 必须包含 `cert.` 前缀，表示在加密连接期间使用客户端证书。

调用必须在消息正文中指定验证类型 'clientCertificate'。此外，请求必须包含 base64 编码的证书签名请求 (CSR)，其格式为 PKCS #10。除此 CSR 的公用名称之外，必须提取主体名称的所有其他属性，并且这些属性必须与设备类型证书中的主体名称属性相同。此 CSR 的公用名称必须定义为

`CN=deviceId:<deviceId>, CN=tenantId:<tenantId>`

其中 `<tenantId>` 也必须从设备类型证书中提取。现在应使用两个公用名称。一个公用名称保留设备标识，另一个保留租户标识。此外，该设备类型证书必须包含在用于 HTTPS 连接的设备密钥库。之后，会在初始 SSL 握手期间将此证书用作客户端证书。有关详细信息，请参阅 SAP Cloud Platform 物联网入门工具包 。

响应正文包含具有所有相关属性的完整设备。

请求正文：

```
{
  "type": "clientCertificate",
  "csr": "MIIDAzCCAesCAQAwgb0xCzAJBgNVB…"
}
```

响应正文：

```
{
  "id": "936DA01F-9ABD-4D9D-80C7-02AF85C822A8",
  "name": "Device 1",
  "deviceType": "a3b7cc7890afb2673eed",
  "link": ...,
  "attributes": ...,
```

```
  "authentication": {
     "type": "clientCertificate",
     "x509Certificate": "MIIDfTCCAuagAwIBAgIOE7lkN8E1BOMQAQQHPUcwDQ…"
  }
}
```

响应包含设备的 base64 编码 X.509 客户端证书，用于将数据从设备发送到 MMS。有关详细信息，请参阅第 74 页上的“数据服务”部分。

相关信息

第 40 页上的“设备类型”
第 74 页上的“数据服务”

7.1.5 在层次结构中组织设备

可以在层次结构中组织设备以便对多个设备进行更好的管理。例如，可以按照组织、位置或制造商对设备进行分组。可以将每个设备分配到多个层次结构。层次结构可以自由定义和建模。

维度和层次结构

在 API 表示法中，层次结构被称为“维度”。此术语与 OLAP 中使用的多维度数据建模相关。由于将设备分配到层次结构的目的是为了进行多维度分析，因此我们采纳了这一通用术语。维度的子元素被称为“维度元素”。

读取维度

`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions` 上的 `GET`（正文为空）。

响应正文返回维度数组：

```
[
  {
    "id": {id},
    "name": "Organization",
  },
  {
    "id": {id},
    "name": "Location"
  },
  …
]
```

读取完整维度

`Https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions/{id}` 上的 `GET`（正文为空）

响应正文返回包含所有子维度和元数据的层次结构维度：

```
{
```

```
  "id": {id},
  "name": "Organization",
  "children": [
  "id": {elementId},
  "name": "Company",
  "parent": null,
  "dimension": {id},
  "devices": […],
  "children": […]
  ]
}
```

创建新维度

`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions` 上的 `POST`

请求正文：

```
{
  "name": "Organization"
}
```

响应正文：

```
{
  "id": {id},
  "name": "Organization"
}
```

删除维度

成功之后，`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions/{标识}` 上的 `DELETE`（正文为空）返回 HTTP 响应 `204`。

> **⚠ 警告**
>
> 删除维度也将会删除此特定维度的所有维度元素和设备分配（如果有）。

JSON 模式验证

```
{
  "$schema":"http://json-schema.org/draft-04/schema#",
  "title":"Organizing Devices in Hierarchies",
  "type":"object",
  "properties":{
    "id":{
      "type":"string",
      "pattern":"^[0-9a-f]{20}$"
    },
    "parent":{
      "type":"string",
      "pattern":"^[0-9a-f]{20}$"
    },
    "name":{
      "type":"string",
      "pattern":"^\\S$|^\\S[\\s|\\S]*\\S$",
      "maxLength":255
    }
  }
}
```

维度元素

每个维度可以拥有由嵌套子元素组成的层次结构。此类元素被称为“维度元素”。每个维度元素都被分配到维度或维度元素中，并且可以拥有更多分配的维度元素或设备。

读取特定维度的所有元素（作为列表）

`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions/{dimensionId}/elements` 上的 `GET`

响应正文返回维度的所有维度元素列表。

```
[
  {
    "id": {elementId},
    "name": "Company",
    "parent": {parentElementId},
    "dimension": {dimensionId},
    "devices": […]
  },
  …
]
```

将新的维度元素添加到维度

`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions/{dimensionId}/elements` 上的 `POST`

请求正文：

```
{
  "name": "Company",
  "parent": {parentElementId}
}
```

响应正文：

```
{
  "id": {elementId},
  "name": "Company",
  "parent": {parentElementId},
  "dimension": {dimensionId},
  "devices": […]
}
```

删除维度元素

成功之后，`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions/{dimensionId}/elements/{elementId}` 上的 `DELETE`（正文为空）返回 HTTP 响应 `204`。

> **⚠ 警告**
>
> 删除维度元素也将会删除此特定维度元素的所有子维度元素和设备分配（如果有）。

设备分配

可以将设备分配到维度元素以便在层次结构中对它们进行组织。

添加设备分配

`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions/{dimensionId}/elements/{elementId}/devices` 上的 `POST`

请求正文：

```
{
  "id": {deviceId}
}
```

响应正文：

```
{
  "id": {elementId},
  "name": "Company",
  "parent": {parentElementId},
  "dimension": {dimensionId},
  "devices": [{deviceId},…]
}
```

响应将返回具有新添加的设备分配的相应维度元素。

> **i 注意**
>
> 要简化 API 的用法，可以将整个设备对象发布到各自的路径。在这种情况下，将仅使用设备的标识。并且会忽略所有其他属性。

删除设备分配

成功之后，`https://<服务器>/com.sap.iotservices.dms/v2/api/dimensions/{dimensionId}/elements/{elementId}/devices{deviceId}` 上的 `DELETE`（正文为空）返回 HTTP 响应 `204`。

相关信息

第 35 页上的“层次结构视图”

7.1.6 应用程序和账户设置

每个用户都可以访问应用程序和账户设置以检查某些属性，例如针对特定账户支持的最大设备数。用户只能在只读模式下访问该设置，且不允许对其进行修改。

读取设置

对具有空正文的 `https://<host>/com.sap.iotservices.dms/v2/api/settings` 执行 `GET` 操作

相应正文包含设置对象：

```
{
    "applicationVersion": "2.49.2"
    "apiVersion": "2",
    "dbSchemaVersion": "3.1.0",
    "tenantId": "342d8e46-ae05-4ea1-bb14-ab6e67deb251",
    "devicesLimit": 50,
    "deviceTypesLimit": 10,
    "messageTypesLimit": 100,
    "messageFieldsLimit": 100,
    "supportedDataTypes": [
        "boolean",
        "integer",
        "long",
        "float",
        "double",
        "string",
        "date",
        "binary"
    ]
}
```

7.1.7 REST 响应

请求消息正文可能包含与本文档中定义的 JSON 模式冲突的无效 JSON 参数或其可能违反系统中的约束。例如，用户尝试通过非唯一设备类型名称来创建设备类型。

本文档中所述用于指定 JSON 模式的 API 调用由系统进行验证。如果发生违约情况，您将收到相应的 HTTPS 响应代码和消息。此功能支持用户与 RDMS 互动。

为了统一 MMS 与 RDMS 之间响应的格式，已商定一般消息结构。此结构在如下模式中指定。

JSON 模式

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Device Registration",
  "description": "JSON Message created for an error",
  "type": "object",
  "properties": {
    "code": {
      "description": "status code for ",
      "type": "integer"
    },
    "message": {
      "description": "general error message",
      "type": "string"
    },
    "errors": {
      "description": "multiple specific error message",
      "type": "array",
      "items": {
        "title": "error messages",
        "type": "object",
        "properties": {
```

```
            "parameter": {
              "description": "name of the invalid parameter",
              "type": "string"
            },
            "invalid value": {
              "description": " value of the invalid parameter ",
              "type": "string"
            },
            "description": {
              "description": "describes which constraint is broken",
              "type": "string"
            },
            "required": [
              "description"
            ]
          }
        }
      }
    }
  }
```

当前 RDMS 不使用“代码”和“消息”属性。目前为止其仅用于 MMS 上下文中。

响应

基于 W3C 定义 返回 HTTPS 响应代码。例如，未通过验证的 REST 调用响应为 400 –“请求无效”。表中显示所有可能的响应代码，按 HTTPS 方法分组。

HTTPS 方法	请求内容类型	响应内容类型	响应代码	状态	示例路径
POST	application/json	application/json	200	成功	/dataTypes /deviceTypes /devices /dimensions /messageTypes
			400	消息主体不符合文档中描述的 JSON 模式定义	
			409	通过标识或名称引用的实体无法由系统辨认	
			409	提供的项目名称不唯一	
GET（项目列表）	-	application/json	200	成功	/dataTypes /deviceTypes /devices /dimensions /messageTypes /settings

HTTPS 方法	请求内容类型	响应内容类型	响应代码	状态	示例路径
GET（单一项目）	-	application/json	200	成功	/deviceTypes/<id> /devices/<id> /dimensions/<id> /messageTypes/<id>
			404	指定的标识不存在	
DELETE	-	application/json	204	成功	
			404	指定的标识不存在	

i 注意

请注意，仅当发送消息主体时 REST-API 服务返回内容类型为 'application/json' 的消息，否则不发送响应的内容类型。如果客户端试图访问未在本文档中描述的 URL，响应代码包含 `404`。如果 REST 方法在 URL 中调用但尚未实现，服务器返回响应代码 `405`。

8 消息管理服务 API

此部分介绍如何使用消息管理服务 (MMS)。MMS 直接在消费者账户中部署并维护（不同于 RDMS 由 SAP 在提供者账户中托管并操作）。

> **⚠ 警告**
>
> 请注意，本文档中 MMS 的当前 API 版本为 'v1'。根据第 38 页上的“远程设备管理服务 API”部分所述，API 'v2' 版本仅对 RDMS 可用。

相关信息

第 16 页上的“通过 物联网 服务主控室部署和更新 MMS”
第 31 页上的“配置设备”

8.1 接口和通信处理

消息管理服务 (MMS) 提供设备用于发送消息的接口。当前 MMS 支持 HTTP(S)、WebSocket、WebSocket 上的 MQTT 和 TCP 上的 MQTT。以下部分说明了这些通信接口使用的消息格式。

> **i 注意**
>
> 第 74 页上的“数据服务”和第 88 页上的“推送服务”部分更详细地说明了 API 本身。

8.1.1 消息格式

设备在 RDMS 中成功注册，并分配给某个设备类型后，设备可以开始发送符合为相应设备类型定义的消息类型的消息。

所有数据服务 API 的消息格式相同。无论是通过 HTTP(S)、WebSocket、WebSocket 上的 MQTT 还是 TCP 上的 MQTT 发送消息，有效负载的格式不会发生更改。

有效的 JSON 有效负载模式如下所示：

```
{
  "title": "MMS json message schema",
  "type": "object",
  "properties": {
```

```
    "messageType": {
      "id": "messageType",
      "type": "string",
      "required":true,
      "description": "id of the message type the messages conform to"
    },
    "messages": {
      "id": "messages",
      "type": "array",
      "required":true,
      "items": {
        "type": "object",
        "description": "list of messages that conform to the message type
definition with the given id"
      }
    },
    "sequence": {
      "id": "sequence",
      "type": "string",
      "required":false,
      "description": "sequence id used to correlate requests and
acknowledgments, only required for async-ack data sending mode"
    },
    "mode": {
      "id": "mode",
      "type": "string",
      "required":false,
      "description": "data sending mode (sync, async or async-ack)"
    }
  }
}
```

重要的是请注意，设备标识不是消息本身的一部分，而是定义为数据服务端点的一部分。在 HTTP API 和 WebSocket API 的情况下，设备标识是 URL 的一部分。有关详细信息，请参阅第 74 页上的“数据服务”部分。

MMS 的有效请求如下所示：

```
{
  "messageType":"m0t0y0p0e1",
  "messages":[
    {
      "sensor":"sensor1",
      "value":22,
      "timestamp":1413191650
    }
  ]
}
```

请求发送符合标识为 **m0t0y0p0e1** 的消息类型的消息。消息包含三个字段**传感器**、**值**和**时间戳**。这些字段是相应消息类型定义的一部分。消息类型定义列在第 69 页上的“样例设备和消息类型”部分中。

> **i 注意**
>
> 不仅消息中的字段集要与相应消息类型定义对应，值也要与其对应。例如，将消息类型 **m0t0y0p0e1** 的**时间戳**字段定义为**日期**类型。定义为**日期**的字段有效值为字符串值，符合 ISO 8601 标准或表示 Unix 时间的长整型值（从 1970 年 1 月 1 日起为秒数）。二进制字段值必须是 base64 编码。

还可以在符合同一消息类型的单个请求中发送多个消息：

```
{
  "messageType":"m0t0y0p0e1",
  "messages":[
    {
```

```
      "sensor":"sensor1",
      "value":22,
      "timestamp":1413191650
    },
    {
      "sensor":"sensor2",
      "value":5,
      "timestamp":1413191654
    }
  ]
}
```

因为消息类型定义中的每个消息字段定义了所在位置（请参阅上述消息类型示例），因此还可以通过忽略字段名称来压缩消息数组。

```
{
  "messageType":"m0t0y0p0e1",
  "messages":[
    ["sensor1", 22, 1413191650],
    ["sensor2", 5, 1413191654],
  ]
}
```

此外，可选字段**模式**可用于确定数据发送模式。有关详细信息，请参阅第 71 页上的“数据发送模式”。

```
{
  "mode":"sync",
  "messageType":"m0t0y0p0e1",
  "messages":[
    {
      "sensor":"sensor1",
      "value":22,
      "timestamp":1413191650
    }
  ]
}
```

如果使用 `async-ack` 模式，则需要**序列**属性。此处指定的标识可用于将设备从 MMS 接收的确认和过去发送的请求相关联。确认通知设备是否请求已成功处理并且所有消息都已存储在数据库中，或者是否出现了错误。

```
{
  "mode":"async-ack",
  "sequence":"11223344",
  "messageType":"m0t0y0p0e1",
  "messages":[
    {
      "sensor":"sensor1",
      "value":22,
      "timestamp":1413191650
    }
  ]
}
```

8.1.2 样例设备和消息类型

缺省情况下，MMS 部署创建可用于测试数据服务的三个样例设备。

标识为 **d000-e000-v000-i000-c000-e001** 和 **d000-e000-v000-i000-c000-e002** 的样例设备支持三个消息类型：**m0t0y0p0e1**、**m0t0y0p0e2** 和 **m0t0y0p0e3**。

```
   {
     "id":"m0t0y0p0e1",
     "direction":"fromDevice",
     "fields":[
       {
         "name":"sensor",
         "type":"string"
       },
       {
         "name":"value",
         "type":"string"
       },
       {
         "name":"timestamp",
         "type":"date"
       }
     ]
   }
   {
     "id":"m0t0y0p0e2",
     "direction":"fromDevice",
     "fields":[
       {
         "name":"rotation",
         "type":"long"
       },
       {
         "name":"angle",
         "type":"long"
       }
     ]
   }
   {
     "id":"m0t0y0p0e3",
     "direction":"toDevice",
     "fields":[
       {
         "name":"opcode",
         "type":"string"
       },
       {
         "name":"operand",
         "type":"string"
       }
     ]
}
```

样例设备 **d000-e000-v000-i000-c000-e003** 支持两个消息类型 **m0t0y0p0e4** 和 **m0t0y0p0e5**：

```
   {
     "id":"m0t0y0p0e4",
     "direction":"fromDevice",
     "fields":[
       {
         "name":"machine",
         "type":"string"
       },
       {
         "name":"state",
         "type":"string"
       },
       {
         "name":"timestamp",
```

```
          "type":"date"
        }
      ]
    }
    {
      "id":"m0t0y0p0e5",
      "direction":"fromDevice",
      "fields":[
        {
          "name":"filename",
          "type":"string"
        },
        {
          "name":"file",
          "type":"binary"
        }
      ]
}
```

有关设备、设备类型和消息类型的数据模型的详细信息，请参阅第 38 页上的“远程设备管理服务 API”部分。

8.1.3 数据发送模式

为了满足不同用例中关于服务数据处理的不同要求，MMS 提供三个不同的数据发送模式：**sync**、**async** 和 **async-ack**。

Sync – 同步发送

> **i 注意**
>
> 请注意以 **sync** 模式发送消息可能导致显著的响应时间，因为对消息进行批量存储。有关详细信息，请参阅第 92 页上的“消息批次”部分。

以同步方式发送数据。这意味着处理完发送到 MMS 的所有消息后，才会解除对设备请求的阻止。通常这意味着消息已经存储在数据库中。如果设备收到正面响应，则数据成功被持久化。此模式只可提供给缺省情况下不是异步的数据服务协议。在当前 MMS 实施中，此模式只可用于第 75 页上的“HTTP” API。

Async – 异步发送

以异步方式发送数据。这意味着 MMS 接受此请求后就返回设备请求。这并不是说消息已经存储在数据库中。只是说消息已经成功添加到暂时 MMS 消息队列。设备没有收到任何信息是否已成功插入到数据库的确认。

Async-ack – 异步发送与确认

以异步方式发送数据。这意味着 MMS 接受此请求并将消息添加到其内部暂时消息队列后就返回设备请求。不过，如果插入成功则设备接收确认。根据使用的数据服务协议，确认作为其他服务 (HTTP API) 来提供或作为从 MMS 到客户端的响应消息来实施 (WebSocket)。为了能够关联收到的确认和设备发出的请求，设备需要将序列标识添加到请求第 67 页上的“消息格式”。

相关信息

第 74 页上的“数据服务”

第 88 页上的“推送服务”
第 92 页上的“消息批次”
第 93 页上的“消息处理”

8.2 配置 API 验证

MMS 提供了配置 API，可用于检索和设置整个应用程序的属性。通过**基本验证**和 **OAuth** 方法来保护配置 API。

基本验证

> **i 注意**
>
> 如果使用基本验证，请确保更改密码后更新凭据。如果验证请求失败太多，SAP BTP 将自动屏蔽对应用户一段时间。

相应的用户需分配到 **IoT-MMS-User** 角色。有关详细信息，请参阅第 18 页上的“为 Java iotmms 应用程序分配 IoT-MMS-User 角色”部分。令牌需作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Basic <base64-encoded username:password>
```

OAuth

为使用 OAuth 进行验证，需执行几个步骤来创建 OAuth 客户端，定义 MMS 所需范围并且通过 **config** 范围请求 OAuth 令牌。

过程

1.使用 SAP BTP 主控室创建“授权许可”类型为“客户端凭据”的 OAuth 客户端。有关详细信息，请参阅 OAuth 2.0 配置 -> **注册 OAuth 客户端**部分。

2.使用 SAP BTP 主控室为 'iotmms' 部署定义 OAuth 范围。有关详细信息，请参阅 OAuth 2.0 配置 -> **定义 OAuth 范围**部分。范围列表取决于想要启用 OAuth 访问的 API。

- **Config** 用于访问 `https://<host>/com.sap.iotservices.mms/v1/api/http/config`

3.通过 REST 客户端请求 OAuth 令牌。有关详细信息，请参阅启用 OAuth 2.0 客户端凭据授予部分。为此，请为您的 SAP BTP 账户列出的令牌端点 URL 执行 HTTP POST。确保您已定义客户端应授权使用的范围集。如果客户端只能访问配置 API，则只需要 **config** 范围。样例请求如下所示：

```
POST https://<token endpoint URL>
```

```
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <base64-encoded oauth_client_id:oauth_client_secret>
grant_type=client_credentials&
scope=<space separated list of scopes that should be enabled for the client>
```

作为 HTTP 响应的一部分返回的令牌需要作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Bearer <oauth token>
```

为了获得全部属性，提交 HTTP 请求到：

```
GET https://<host>/com.sap.iotservices.mms/v1/api/http/config
```

通过如下 HTTP 请求可设置属性：

```
PUT https://<host>/com.sap.iotservices.mms/v1/api/http/config
Content-Type: application/json;charset=utf-8
{
   "config":[
        {"key”:"property1", "value”:”value1"},
        {"key”:"property2", “value”:"value2"},
        …
   ]
}
```

此外，MMS 提供的用户界面可以在设置配置属性值的时候使用。还可以导入或导出属性。另外，用户界面提供了全部可用属性的综合描述。您可以通过 MMS 主控室中的磁贴来访问此用户界面。

相关信息

第 92 页上的“消息批次”

8.3 数据服务

以下部分描述数据服务 API - HTTP API、WebSocket API、WebSocket 上的 MQTT 和 TCP 上的 MQTT API - 并且为通过以上协议发送数据提供样例请求。

8.3.1 数据服务 API 验证

数据服务 API 通过 *"OAuth"* 和"客户端证书" (BETA) 的方式进行保护。

OAuth

可以使用 物联网 服务主控室检索 **OAuth** 访问令牌。有关详细信息，请参阅第 34 页上的"为设备重新生成 OAuth 令牌"部分。另外，设备可检索令牌，方法是通过使用 RDMS API 调用注册新设备。有关详细信息，请参阅第 51 页上的"设备"部分。

对于 HTTP 和 WebSocket API，需要将令牌作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Bearer <oauth token>
```

对于 WebSocket 上的 MQTT 和 TCP API 上的 MQTT，连接到 MQTT 客户端时，分别将设备标识和 Oauth 令牌用作用户名和密码。

客户端证书 (BETA)

> **⚠ 警告**
>
> 在 物联网 服务中使用客户端证书是一种 beta 功能。要使用此功能，MMS 的相应账户必须为启用了 beta 的账户。有关详细信息，请参阅 SAP BTP 文档中的在子账户中使用 Beta 功能部分。

设备可以检索客户端证书，方法是通过使用 RDMS API 调用注册新设备。有关详细信息，请参阅第 51 页上的"设备"部分。

该证书必须包含在用于 HTTPS/WSS 连接的设备密钥库中。之后，会在初始 SSL 握手期间将此证书用作客户端证书。如果提供附加客户端证书的 OAuth 令牌，则 OAuth 将强制作为所有数据服务 API 的验证机制，WebSocket 上的 MQTT 除外。有关详细信息，请参阅 SAP Cloud Platform 物联网入门工具包 。

请注意，*"OAuth"* 和"客户端证书"的端点不同。对于证书，数据服务端点的 <主机> 部分必须包含 `cert` 前缀，表示在加密连接期间使用客户端证书。有关架构主机的详细信息，请参阅 SAP BTP 文档中的地区部分。

> **⚠ 警告**
>
> 必须启用 MMS 应用程序以便使用客户端证书平台。有关详细信息，请参阅 SAP BTP 文档中的 add-platform-domain 部分。

相关信息

第 34 页上的“为设备重新生成 OAuth 令牌”
第 51 页上的“设备”
第 75 页上的“HTTP”
第 80 页上的“WebSocket”
第 82 页上的“WebSocket 上的 MQTT”
第 83 页上的“TCP 上的 MQTT”

8.3.2 HTTP

要使用 HTTP API 将数据发送到端点，设备需要将 HTTP POST 请求发送到 URL `https://<主机>/com.sap.iotservices.mms/<版本>/api/http/data/<设备标识>`，其中<版本>在此文档中是 MMS 的当前 API 版本 'v1' 且<设备标识>是发送数据的设备标识符。有关 MMS 的 API 版本的详细信息，请参阅第 72 页上的“配置 API 验证”部分。

SAP BTP 的协议是 HTTPS。请求正文应为格式符合特定消息类型的 JSON 消息。有关详细信息，请参阅第 67 页上的“接口和通信处理”部分。

使用可选字段“模式”，可以根据字段值控制数据发送模式：

- `sync` - 同步消息传递，成功处理消息后才会解除对请求的阻止。

  > **i 注意**
  >
  > 请注意以 *“sync”* 模式发送消息可能导致显著的响应时间，因为对消息进行批量存储。有关详细信息，请参阅第 92 页上的“消息批次”部分。

  处理消息时，服务器使用代码“200 - 正常”进行响应。
  如果处理时间异常长（当前该值配置为 60,000 毫秒），服务器将传递模式切换为 `async-ack`，使用代码“202 - 已接受”和含确认服务链接的消息进行响应，以查询请求的处理状态。此链接包含请求的标识。如果客户端想要指定此标识，请求必须包含可选字段序列，具有设备独有的标识符。否则，标识由服务器生成。
- `async` - 异步消息传递，无确认。如果已经成功验证请求且处理继续，则服务器使用代码“202 - 已接受”进行响应。
- `async-ack` - 异步消息传递，存储确认直到通过确认服务对其进行主动检索。使用此模式时，字段**序号**为必填并且必须包含设备唯一的序号。此序列号可用于使用确认查询请求的处理状态。如果已经成功验证请求且处理继续，则服务器使用代码“202 - 已接受”进行响应。

> **i 注意**
>
> 如果未设置字段，则 `async` 模式将在缺省情况下使用。

具有“设备标识 *d000-e000-v000-i000-c000-e001*”和模式 `sync` 的样例设备的有效示例请求为：

```
POST https://<host>/com.sap.iotservices.mms/v1/api/http/data/d000-e000-v000-i000-
c000-e001
Content-Type: application/json;charset=utf-8
{
```

```
  "mode": "sync",
  "messageType": "m0t0y0p0e1",
  "messages": [
    {
      "sensor": "sensor1",
      "value": "20",
      "timestamp": 1413191650
    },
    {
      "sensor": "sensor2",
      "value": "25",
      "timestamp": 1413191999
    }
  ]
}
```

如果处理时间异常长，则服务器发送的响应消息如下所示：

```
 {"msg":"<https://<host>/com.sap.iotservices.mms/v1/api/http/ack/d000-e000-v000-i000-c000-e001/550e8400-e29b-11d4-a716-446655440000"}
```

此处是具有“设备标识 *d000-e000-v000-i000-c000-e001*”、“消息类型标识 *m0t0y0p0e2*” 和模式 `async-ack` 的样例设备的有效示例请求：

```
POST https://<host>/com.sap.iotservices.mms/v1/api/http/data/d000-e000-v000-i000-c000-e001
Content-Type: application/json;charset=utf-8
{
  "mode": "async-ack",
  "sequence": "4711",
  "messageType": "m0t0y0p0e2",
  "messages": [
    {
      "rotation": 10,
      "angle": 65
    }
  ]
}
```

下面提供了“设备标识 *d000-e000-v000-i000-c000-e003*” 的一台设备的有效示例请求：

```
POST https://<host>/com.sap.iotservices.mms/v1/api/http/data/d000-e000-v000-i000-c000-e003
Content-Type: application/json;charset=utf-8
{
  "messageType": "m0t0y0p0e4",
  "messages": [
    {
      "machine": "m1",
      "state": "running",
      "timestamp": 1413191650
    }
  ]
}
```

还可以发送 GZIP 压缩的有效负载。端点和消息格式保持一致。在发送之前压缩请求正文和使用 HTTP 内容类型 *“application/gzip”* 是开发人员的任务。

通过 URL 参数发送消息

如果无法如上所述将消息作为 HTTP 正文中的 JSON 有效负载发送，还可以通过 URL 参数发送消息。

> **i 注意**
>
> 每个请求只能发送一条消息。由于客户端或服务器中的 URL 长度限制，可以限制消息长度。

要发送消息，设备需要将 HTTP POST 请求发送到 URL

`https://<主机>/com.sap.iotservices.mms/v1/api/http/data/<设备标识>/<消息类型标识>/<模式>/<序列标识>?<消息>`。

参数的语义与以缺省消息格式引入的相应属性相同：

- <设备标识>
- <消息类型标识>
- <序列标识>是可选项，可用于指定序列标识
- <消息>是必需项，其中包含作为查询参数的消息

将请求本身限制为此模式，指定的值必须按此顺序出现。例如，在不指定模式的情况下无法指定序列标识。

具有“设备标识 *d000-e000-v000-i000-c000-e001*”、“消息类型标识 *m0t0y0p0e1*” 和模式 sync 的设备的有效样例消息为：

```
POST https://<host>/com.sap.iotservices.mms/v1/api/http/data/d000-e000-v000-i000-c000-e001/m0t0y0p0e1/sync?sensor=sensor1&value=20&timestamp=1413191650
```

您还可以发送混合请求，在此通过 URL 参数和正文本身发送消息。要使用此方法，您必须通过 URL 参数忽略消息的消息类型字段。然后必须在 HTTP 请求的正文中发送缺少的属性。处理期间，首先处理 URL 中的消息类型字段。如果缺少某个消息类型字段，则将 HTTP 正文映射到这个缺少的字段。

具有**设备标识 d000-e000-v000-i000-c000-e001** 的设备和作为 HTTP POST 正文一部分发送的时间戳值的有效样例消息为：

```
POST https://<host>/com.sap.iotservices.mms/v1/api/http/data/d000-e000-v000-i000-c000-e001/m0t0y0p0e1?sensor=sensor1&value=20
1413191650
```

针对时戳消息类型字段自动映射 HTTP 正文内容 '1413191650'。

通过 Google Protocol Buffer (protobuf) 发送消息

协议缓冲区 是用于对结构化数据进行序列化的一种独立于语言和平台的可扩展机制。

要将协议缓冲区与 MMS 结合使用，您可以向 HTTP 数据服务发送协议缓冲区编码的二进制有效负载。HTTP 内容类型必须设置为 `application/x-protobuf`。

已生成协议缓冲区有效负载的模式必须与消息格式部分描述的典型 MMS 消息格式匹配。这意味着协议缓冲区模式与 JSON MMS 模式相同。它包含 `Request` 部分，该部分拥有消息类型标识、发送模式和 `Message` 数组。各个消息必须与现有消息类型定义相对应。

以下是协议缓冲区中的有效示例定义：

```
  syntax = "proto3";
  // default is ASYNC mode
  enum Mode {
      ASYNC = 0;
      SYNC = 1;
      ASYNC_ACK = 2;
  }
  // MMS request schema
  message Request {
    string messageType = 1;
    Mode mode = 2;
    string sequence = 3;
    repeated Message messages = 4;
  }
  // Message schema that reflects a corresponding message type definition
  message Message {
    string sensor = 1;
    string value = 2;
    int64 timestamp = 3;
}
```

下表显示了 IoT 数据类型与协议缓冲区数据类型的映射：

数据类型	协议缓冲区标量值类型
boolean	bool
integer	int32
double	double
float	float
binary	bytes
long	int64
string	string
date	int64 (Unix time)
binary	google.protobuf.Any

为现有消息类型创建协议缓冲区模式时，需注意以下几点：

- 协议缓冲区不支持以数字值开头的字段名称。确保为消息类型字段进行相应命名。
- 协议缓冲区字段名称、类型和位置必须与消息类型匹配。
- 类型为 `google.protobuf.Any` 的字段存储为二进制字段。我们建议您使用扁平数据结构。
- 未设置显式值的字段将由 MMS 设置为缺省值（例如，bool=false， int32=0 等）。在此情况下，您不会看到任何错误消息。

确认服务

必须由设备使用单独服务进行显式请求确认。要请求对给定设备和序列的确认，必须将正文为空的 HTTP POST 发送至：

`https://<主机>/com.sap.iotservices.mms/v1/api/http/ack/<设备标识>/<序列标识>`

HTTP 响应包含序列的处理状态作为 HTTP 响应代码，JSON 正文包含信息消息。状态代码具有以下语义：

- `200` - 成功处理请求。最终状态。
- `202` - 已接受请求，信息消息告知您是否仍然需要验证请求（例如，如果与 RDMS 同步必须提前进行）或请求是否已经验证以及是否正在处理队列中。中间状态。
- `400` - 请求验证失败，信息消息指定请求失败的类型。最终状态。
- `413` - 请求过大。最终状态。
- `404` - 给定设备或序列未知。
- `500` - 处理失败，信息消息指定服务器错误的类型。最终状态。
- `503` - 可能由于暂时超载，当前无法处理请求。最终状态。

代码 `200`、`400`、`413`、`500` 和 `503` 表示最终状态。其中某个状态的序列信息仅传递一次，然后从服务器删除。尝试再次查询相同序列的 `ack` 服务导致代码为 `404`。

要请求对设备的所有确认，必须将正文为空的 HTTP POST 发送至：`https://<主机>/com.sap.iotservices.mms/v1/api/http/ack/<设备标识>`

JSON 响应包含所有未决确认，包括序号及格式如下的相应响应代码：

```
[
  {"code":200, "sequences":[seqNo1,...,seqNoN]},
  {"code":400, "sequences":[seqNoX,...,seqNoY]},
  {"code":413, "sequences":[seqNoS,...,seqNoT]},
  {"code":500, "sequences":[seqNoA,...,seqNoZ]},
  {"code":503, "sequences":[seqNoF,...,seqNoK]}
]
```

请注意只有在最终状态的序列这样传递。不包括仍在处理中的序列，因为这会产生很大的数据传输量。

样例客户端

MMS 使用 HTTP API 为发送数据提供样例客户端。可以使用磁贴“通过 *HTTP* 进行消息传递”，或使用 URL 从 MMS 主页访问该客户端：

`https://<主机>/com.sap.iotservices.mms/#/dataservice/http`

样例客户端的用户界面非常简单明了。使用此客户端，可以向端点发送消息、从 MMS 查看回复，还可以接收确认。

要使用 HTTP API 向端点发送数据：

- 填写“数据端点”URL 文本框。端点 URL 的格式如上所述。
- 填写要发送的“消息”的文本框。您提供的消息必须符合您在之前步骤中指定的设备支持的消息类型。
- 选择“发送”。

您会在名为“从服务器回复”的表中看到 MMS 发送的响应。如果请求成功，“应用程序数据”视图中会提供您刚发送的数据。要访问此视图，请在 MMS 主页上选择“显示已存储的消息”磁贴并选择与 （表视图）内各自设备对应的数据表。或者，如果想以图表形式显示数据，请选择 （折线图）按基于时间的方式绘制数据。

相关信息

第 74 页上的“数据服务”
第 74 页上的“数据服务 API 验证”
第 80 页上的“WebSocket”
第 82 页上的“WebSocket 上的 MQTT”
第 83 页上的“TCP 上的 MQTT”
第 92 页上的“消息批次”

8.3.3 WebSocket

要使用 WebSocket 将数据发送到 MMS，必须将 WebSocket 客户端连接到：

```
wss://<主机>/com.sap.iotservices.mms/v1/api/ws/data/<device id>
```

用于此目的的协议为 WSS (WebSocket Secure)。连接之后，设备可以发送格式与使用的协议相对应的 JSON 消息。使用可选字段模式，可以根据以下字段值控制数据传递模式：

- `async-ack` - 异步消息传递，服务器通过 WebSocket 向客户端发送确认。使用此模式时，必须提供字段序列并且序列中必须包含对设备而言独一无二的序列号。此序列号可用于查询使用确认服务的请求的处理状态。
- `async` - 异步消息传递，无确认

如果未设置 `mode` 字段，将使用缺省模式 `async`。

下面提供了“设备标识 *d000-e000-v000-i000-c000-e001*” 的一台设备的有效示例请求：

```
  {
    "mode":"async-ack",
    "sequence":"4811",
    "messageType":"m0t0y0p0e1",
    "messages":[
      {
        "sensor":"sensor1",
        "value":"20",
        "timestamp":1413191650
      }
    ]
}
```

下面提供了“设备标识 *d000-e000-v000-i000-c000-e003*” 的一台设备的有效示例请求：

```
  {
    "messageType":"m0t0y0p0e4",
    "messages":[
      {
```

```
          "machine":"m1",
          "state":"running",
          "timestamp":1413191650
        }
      ]
}
```

还可以发送 GZIP 压缩的有效负载。端点和消息格式保持一致。在发送有效负载之前将其压缩是开发人员的任务。WebSocket 客户端需要将压缩的有效负载作为二进制消息而非文本消息发送。有关详细信息，请参阅 WebSocket 规范 中的**二进制框**。用来发送二进制消息的相应方法可能根据 WebSocket 客户端实施不同而不同。

确认服务

要接收关于成功将消息插入数据库的确认，必须指定模式属性和序列属性。通过用于将数据从设备发送到 MMS 的同一 WebSocket 连接，将确认从 MMS 发送至设备。使用 `async-ack` 模式接收确认的有效示例如下所示：

```
  {
    "mode":"async-ack",
    "sequence":"0815",
    "messageType":"m0t0y0p0e1",
    "messages":[
      {
        "sensor":"sensor1",
        "value":"20",
        "timestamp":1413191650
      }
    ]
}
```

样例客户端

类似于 HTTP API，MMS 提供使用 WebSocket API 发送数据的样例客户端。可以通过使用磁贴“通过 *WebSocket* 进行消息传送”选项，或使用以下 URL 来从 MMS 主页访问该客户端：

`https://<主机>/com.sap.iotservices.mms/#/dataservice/ws`

样例客户端的用户界面非常简单明了。使用此客户端，可以向端点发送消息并从 MMS 查看回复。

要使用 WebSocket API 向端点发送数据，请执行以下步骤：

- 填写“端点”URL 文本框。端点 URL 的格式如上所述。
- 填写要发送的“消息”的文本框。您提供的消息必须符合您在之前步骤中指定的设备支持的消息类型。
- 选择“发送”。

您会在名为“从服务器回复”的表中看到 MMS 发送的响应。如果请求成功，“应用程序数据”视图中会提供您刚发送的数据。要访问此视图，请在 MMS 主页上选择“显示已存储的消息”磁贴并选择与 （表视图） 内各自设备对应的数据表。或者，如果想将数据显示在图表中，请选择 （折线图） 用基于时间的方式绘制数据。

相关信息

第 74 页上的“数据服务”
第 74 页上的“数据服务 API 验证”
第 75 页上的“HTTP”
第 82 页上的“WebSocket 上的 MQTT”
第 83 页上的“TCP 上的 MQTT”

8.3.4 WebSocket 上的 MQTT

MMS 支持使用 WebSocket 作为基础网络传输方法的 MQTT 3.1.1。有关详细信息，请参阅 MQTT 规范中的 6 使用 WebSocket 作为网络传输 部分。要通过 MQTT 向 MMS 发送数据，需要将设备连接到 MMS MQTT WebSocket 端点，然后通过该连接发送与 MQTT 3.1.1 兼容的消息。

可将启用了 WebSocket 的 MQTT 直接连接到：

```
wss://<主机>/com.sap.iotservices.mms/v1/api/ws/mqtt
```

> **i 注意**
>
> 要连接到 MMS，必须确保除纯 TCP 连接之外，您的 MQTT 客户端库还支持 WebSocket 连接。

MQTT 客户端使用的客户端标识是分配到 RDMS 中相应设备的设备标识。MQTT 客户端可以使用 OAuth 或客户端证书进行验证。当使用 OAuth 时，MQTT 客户端必须将“用户名”和“密码”分别设置为“设备标识”和设备 OAuth 令牌。当使用客户端证书时，客户端必须在 WSS 连接的设备密钥库中包括此证书。然后在初始 SSL 握手期间将使用该证书。

设置连接后，客户端便可通过将标准 物联网 服务 JSON 有效负载发布到主题 `iot/data/<设备标识>`，向 MMS 发送消息。

此处是具有“设备标识 *d000-e000-v000-i000-c000-e001*”的设备的有效示例：

```
PUBLISH iot/data/d000-e000-v000-i000-c000-e001
{
  "mode":"async "
  "messageType":"m0t0y0p0e1",
  "messages": [
    {
      "sensor":"sensor1",
      "value":"20",
      "timestamp":1413191650
    }
  ]
}
```

要从 MMS 接收成功消息和错误消息，MQTT 可以订阅特定设备主题 `iot/log/<设备标识>`。想要使用 `async-ack` 模式发送消息，从而接收确认的客户端需要订阅主题 `iot/ack/<设备标识>`。此外，MMS 还提供了基于 WebSocket 上的 MQTT 的推送服务。这意味着您可以使用 MMS 的推送 API 向通过 MQTT 连接的客户端发送消息。接收推送消息的客户端需要订阅主题 `iot/push/<设备标识>`。如果没有此类订阅，消息将不会推送至客户端。

> **i 注意**
>
> MMS 提供了设备和通过 MQTT 的 MMS 之间的直接链接。因此，将应用与允许的主题名称和授权发布者以及订户相关的严格规则。设备可以专门发布和订阅上面指定的主题，这些主题通过相应设备标识识别。出于安全原因，禁止使用主题通配符。MMS 支持 MQTT 3.1.1 规范中的 4.3.1 和 4.3.2 两部分定义的服务质量 (QoS) 级别 0 和 1。不接受最后的消息。通过断开连接关闭会话。这意味着客户端始终以干净的会话开始，因此需要刷新所有订阅。

样例客户端

MMS 提供了一个样例客户端，用于使用 WebSocket 上的 MQTT API 方法来发送数据。可以通过使用磁贴“通过 *WS 上的 MQTT 进行消息传送*”选项，或使用以下 URL 来从 MMS 主页访问该客户端：

```
https://<主机>/com.sap.iotservices.mms/#/dataservice/ws-mqtt
```

要使用 WebSocket 上的 MQTT API 将数据发送到端点，请执行以下操作：

- 填写“数据端点” URL 文本框。端点 URL 的格式如上所述。
- 填写“设备标识”文本框。
- 选择“连接”。
- 填写要发送的“消息”的文本框。您提供的消息必须符合您在之前步骤中指定的设备支持的消息类型。
- 选择“发送”。

您会在名为“从服务器回复”的表中看到 MMS 发送的响应。如果请求成功，“应用程序数据”视图中会提供您刚发送的数据。要访问此视图，请在 MMS 主页上选择“显示已存储的消息”磁贴并选择与 （表视图） 内各自设备对应的数据表。或者，如果想将数据显示在图表中，请选择 （折线图） 用基于时间的方式绘制数据。

相关信息

第 74 页上的“数据服务”
第 74 页上的“数据服务 API 验证”
第 75 页上的“HTTP”
第 80 页上的“WebSocket”
第 83 页上的“TCP 上的 MQTT”

8.3.5 TCP 上的 MQTT

消息管理服务 (MMS) 支持 TCP 上的本地 MQTT 作为网络传输方法。有关详细信息，请参考 MQTT 规范 。要通过 MQTT 将数据发送到 MMS，设备需要连接到 MQTT 代理，然后通过此连接发送与 MQTT 兼容的消息。

> **i 注意**
>
> 与 WebSocket API 上的 MQTT（MQTT 客户端直接连接并发布到 MMS）不同，TCP API 上的 MQTT 允许客户端连接并发布到“外部”代理。在这种情况下，MMS 将充当下方介绍的特殊主题的订户。

对于本地 MQTT API，每台注册的设备通过发布到特定于该设备的主题来将其数据发送到代理。发布数据的主题模式为

`iot/data/iotmms<账户标识>/v1/<设备标识>.`

对于将数据发送到部署在 SAP BTP 账户“a00000001”上的 MMS 的标识为 d000-e000-v000-i000-c000-e001 的设备，其主题应为

`iot/data/iotmmsa00000001/v1/d000-e000-v000-i000-c000-e001。`

设置连接后，客户端便可通过将标准 物联网 服务 JSON 有效负载发布到主题，向 MMS 发送消息。

此处是具有“设备标识 *d000-e000-v000-i000-c000-e001*” 的设备的有效示例：

```
PUBLISH iot/data/iotmmsa00000001/v1/d000-e000-v000-i000-c000-e001
{
  "mode":"async"
  "messageType":"m0t0y0p0e1",
  "messages": [
    {
      "sensor":"sensor1",
      "value":"20",
      "timestamp":1413191650
    }
  ]
}
```

MMS 订阅了通配符主题 `iot/data/iotmms<account id>/v1/#` 来接收通过所有注册的设备发送的数据。与 HTTP 和 WebSocket API 相似，MQTT API 支持两种数据服务模式：`async` 和 `async-ack`。`async` 模式中保留传入消息，但不会向设备传回确认。`async-ack` 模式会将确认传回到发送消息的设备。对于 `async-ack` 模式，必须按照 第 76 页上的“HTTP” 部分中 `async-ack` 模式的示例数据中所述指定参数 `sequence`。

使用特定于设备的不同主题模式来发送确认。发送确认的主题模式为

`iot/ack/iotmms<account id>/v1/<device id>。`

对于标识为 d000-e000-v000-i000-c000-e001，账户标识为 a00000001 的设备，其主题应为

`iot/ack/iotmmsa00000001/v1/d000-e000-v000-i000-c000-e001。`

设备必须订阅该主题来接收向其发送的确认。

可以使用以下主题检索消息处理期间的日志信息：

`iot/log/iotmms<account id>/v1/<device id>。`

对于标识为 d000-e000-v000-i000-c000-e001，账户标识为 a00000001 的设备，其主题应为

`iot/log/iotmmsa00000001/v1/d000-e000-v000-i000-c000-e001。`

此外，MMS 还提供了基于 MQTT 的推送服务。这意味着您可以使用 MMS 的推送 API 向通过 MQTT 连接的客户端发送消息。接收推送消息的客户端需要订阅主题

`iot/push/iotmms<account id>/v1/<device id>。`

如果没有此类订阅，消息将不会推送至客户端。

> **i 注意**
>
> 重要一点是了解 MQTT 代理不是 MMS 自身的一部分，而是外部组件。就像注册的设备，MMS 本身也只是一个客户端。通过订阅相关主题，它会接收设备发布的数据，并通过发布到相关主题传回确认。

MQTT 客户端使用的客户端标识即是分配到远程设备管理服务 (RDMS) 中相应设备的设备标识。此外，MQTT 客户端需要分别将用户名和密码设置为设备标识和设备 OAuth 令牌。

代理设置

建议您使用外部 MQTT 代理，例如 Eclipse Mosquito 或 Apache ActiveMQ 。有关安装和配置流程的信息，请参阅所用产品的文档。

> **⚠ 警告**
>
> 您负责设置代理，使设备只能发布和订阅其设备特定的主题，并且 MMS 可以发布和订阅所有设备特定的主题。您选择的代理为此提供的功能可能会有所不同。

在下方可以找到基于访问控制列表 (ACL) 的 Eclipse Mosquito 安全设置。

"1.创建用户并将其添加到密码文件"

创建 `passwd` 文件和 `mms` 的用户：

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd mms
```

为您的设备添加用户：

```
sudo mosquitto_passwd /etc/mosquitto/passwd d000-e000-v000-i000-c000-e001
```

这些命令会提示用户输入密码。所有设备在注册时都必须使用其设备标识作为用户名，使用 Oauth 令牌作为密码。可以根据需要选择 `mms` 用户凭据。

"2.配置访问规则"

将下列行添加到 `/etc/mosquitto/mosquitto.conf`

```
allow_anonymous false
password_file /etc/mosquitto/passwd
acl_file /etc/mosquitto/mqtt.acl
```

"3.示例 ACL 配置"

文件位置：`/etc/mosquitto/mqtt.acl`

```
#anonymus access (always as first entry required)
topic read $SYS/#
#this is the user for the communication from mms to the broker
#mms is allowed to subscribe and publish to all iot topics
```

```
user mms
topic write iot/#
topic read iot/#
#these patterns limit the devices communication to their given topics
pattern write iot/data/iotmms<account id>v1/%u
pattern read iot/log/iotmms<account id>/v1/%u
pattern read iot/ack/iotmms<account id>/v1/%u
pattern read iot/push/iotmms<account id>/v1/%u
```

模式对所有用户有效。必须将 `<account id>` 替换为部署 MMS 的账户标识。

"4.允许流程使用配置"

```
mosquitto -c /etc/mosquitto/mosquitto.conf
```

或通过以下命令重新加载配置

```
kill -HUP <processid>
```

根据这些示例，用户可以设置其选择的代理以满足他们的要求。

MMS 设置

MMS 提供了配置 API，可用于检索和设置数据服务的属性。有关详细信息，请参阅第 72 页上的"配置 API 验证"部分。您可以通过使用 API 调用，或使用提供的用户界面来进行此设置。您可以通过 MMS 主控室中的磁贴来访问此用户界面。TCP 上的 MQTT 的相关设置为：

`MQTT connection enabled (ID: mms.mqtt.enable)`

此属性用于打开或关闭完整的 TCP 上的 MQTT 数据服务。

`MQTT broker endpoint (ID: mms.mqtt.broker_url)`

此属性指定 MQTT 代理端点。对于未加密的 TCP 连接，使用模式 `tcp://<hostname>:<port>`。在这种情况下，您通常会使用端口 `1883`。建议您通过使用以下模式来使用加密通信：`ssl://< hostname >:<port>`。通常情况下，端口为 `8883`。

`MQTT user name (ID: mms.mqtt.user)`

连接到代理的用户名。

`MQTT password (ID: mms.mqtt.pwd)`

连接到代理的密码。

`MQTT quality of service level (ID: mms.mqtt.qos)`

在与代理进行通信期间，处理服务必须应用服务质量 (QoS) 级别。缺省级别为 `1`，这意味着消息至少传递一次。其他有效的设置有 `2` –"只传递一次"或 `0` –"即发即弃"。

`MQTT reconnect interval (ID: mms.mqtt.reconnect_interval_milliseconds)`

如果与 MQTT 代理的连接丢失或无法建立，则此间隔会指定重复尝试重新连接的频率。缺省值为 60000，即 1 分钟。

`MQTT action timeout (ID: mms.mqtt.action_timeout)`

指定必须完成 MQTT 操作（如连接、订阅、取消订阅、断开连接）的毫秒数。缺省值为 10000。

样例客户端

MMS 提供了一个样例客户端，用于使用 TCP 上的 MQTT 方法来发送数据。可以通过使用磁贴“通过 *TCP* 上的 *MQTT* 进行消息传送”选项，或使用以下 URL 来访问该客户端：

`https://<主机>/com.sap.iotservices.mms/#/dataservice/mqtt`

您可以使用此客户端来测试 TCP 上的 MQTT。它会建立到代理服务的 WebSocket 连接，该代理服务连接到在 MMS 中配置的 MQTT 代理端点。连接后，客户端可以将消息发布到设备特定的数据主题。此外，客户端还可以订阅设备特定的日志，确认和推送主题以接收来自服务器的响应或使用 MMS 的推送 API 发送的消息。

要使用 TCP API 上的 MQTT 将数据发送到端点：

- 填写“设备标识”文本框。
- 选择“连接”。
- 填写要发送的“消息”的文本框。您提供的消息必须符合您在之前步骤中指定的设备支持的消息类型。
- 选择“发送”。

您会在名为“从服务器回复”的表中看到 MMS 发送的响应。如果请求成功，“应用程序数据”视图中会提供您刚发送的数据。要访问此视图，请在 MMS 主页上选择“显示已存储的消息”磁贴并选择与 （表视图） 内各自设备对应的数据表。或者，如果想将数据显示在图表中，请选择 （折线图） 用基于时间的方式绘制数据。

相关信息

第 74 页上的“数据服务”
第 72 页上的“配置 API 验证”
第 74 页上的“数据服务”
第 75 页上的“HTTP”
第 80 页上的“WebSocket”
第 82 页上的“WebSocket 上的 MQTT”

8.4 推送服务

推动服务可用于向设备推送消息。任何 IoT 应用程序甚至其他设备都可以使用推送服务 API。

MMS 提供各种推送服务方法。每种方法对应一个特定的传输协议。当前，支持通过 HTTP、WebSocket、WebSocket 上的 MQTT 和 TCP 上的 MQTT 推送消息。

推送服务接受以下格式的消息：

```
{
  "method":"http|ws|mqtt|ws-mqtt",
  "sender":"IoT application",
  "messageType":"m0t0y0p0e4",
  "messages":[{…}, {…}]
}
```

根据指定的方法，可以通过 WebSocket 协议直接向设备推送消息，或者通过 HTTP 将消息提供给轮询设备。

> **i 注意**
>
> 必须使用 `toDevice` 或 `bidirectional` 方向定义消息类型。使用 `fromDevice` 方向定义的消息会导致错误并且不会被推送到客户端。

推送到客户端的消息不包含 `method` 属性，但包含所有其他属性。

```
[
  {
    "sender":"IoT application",
    "messageType":"m0t0y0p0e4",
    "messages":[{…}, {…}]
  }
]
```

通过 WebSocket 向样例设备 *"d000-e000-v000-i000-c000-e001"* 发送消息的有效样例请求如下所示：

```
POST https://<host>/com.sap.iotservices.mms/v1/api/http/push/d000-e000-v000-i000-
c000-e001
Content-Type: text/plain;charset=utf-8
{
  "method":"ws",
  "sender":"My IoT application",
  "messageType":"m0t0y0p0e3",
  "messages":[
    {
      "opcode":"switch on",
      "operand":"led 1"
    },
    {
      "opcode":"set temp",
      "operand":"30"
    }
  ]
}
```

样例客户端

MMS 提供了使用推送服务发送数据的样例客户端。可以使用“向设备推送消息”磁贴或使用以下 URL 从 MMS 主页中访问此客户端：

```
Https://<主机>/com.sap.iotservices.mms/#/pushservice
```

样例客户端的用户界面非常简单明了。使用此客户端，可以向端点发送消息并从 MMS 查看回复。

要使用推送服务向端点发送数据，请执行以下步骤：

- 填写“设备标识”。
- 选择要使用的推送“方法”，*“http”*、*“mqtt”*、*“ws”* 或 *“ws-mqtt”*。
- 填写要推送的“消息”的文本框。您提供的消息必须符合您在之前步骤中指定的设备支持的消息类型。
- 选择“推送”。

您会在名为“从服务器回复”的表中看到 MMS 发送的响应。如果请求成功，“应用程序数据”视图中会提供您使用 *“http”* 方法刚发送的数据。要访问此视图，请在 MMS 主页上选择“显示已存储的消息”磁贴并选择与 （表视图）内相应设备对应的数据表 *“T_IOT_HTTP_PUSH”*。或者，如果想以图表形式显示数据，请选择 （折线图）按基于时间的方式绘制数据。

接收从 MMS 发送到设备的消息

您可以通过以下传输协议从 MMS 接收推送消息：HTTP、WebSocket、WebSocket 上的 MQTT 和 TCP 上的 MQTT。

使用 HTTP 接收推送消息

鉴于 WebSocket 是经过设计的双向协议，HTTP 推送实施基于 MMS 中的消息存储。推送消息存储在 MMS 中，必须通过设备主动检索。要接收已使用推送服务推送到特定设备并且已由 MMS 缓存的消息，您可以将 HTTP GET 请求发送到：

```
https://<主机>/com.sap.iotservices.mms/v1/api/http/data/<设备标识>
```

根据指定的 HTTP 接受标头，您可以接收作为 JSON 编码有效负载（此为缺省值 `application/json`）或作为 Google Protocol Buffer 编码二进制有效负载（`application/x-protobuf`）的消息。

接收作为 Google Protocol Buffer 的推送消息

如果 HTTP 接受标头设置为 `application/x-protobuf`，MMS 会将可用消息编码为 Google Protocol Buffer 消息。要解码二进制消息，必须定义合适的协议缓冲区模式。

以下是协议缓冲区中的有效示例定义：

```
  syntax = "proto3";
  // MMS response array schema
  message Array {
     repeated Response array = 1;
  }
  // MMS response schema
  message Response {
    string messageType = 1;
    string sender = 2;
    repeated Message messages = 3;
  }
  // Message schema that reflects a corresponding message type definition
  message Message {
    string opcode = 1;
    string operand = 2;
}
```

有关数据类型转换规则和定义规则，请参阅第 77 页上的“通过 Google Protocol Buffer (protobuf) 发送消息”部分。

使用 WebSocket 接收推送消息

要通过 WebSocket 从 MMS 接收设备推送消息，设备需要建立到 MMS 的 WebSocket 连接，如下所示：

`wss://<主机>/com.sap.iotservices.mms/v1/api/ws/data/<设备标识>`

还可以使用 MMS 数据服务示例客户端通过 WebSocket 接收推送消息。在 MMS 主页中，可以通过选择“通过 *WebSocket* 进行消息传送”磁贴选项或者打开以下 URL 访问此客户端：

`https://<主机>/com.sap.iotservices.mms/#/dataservice/ws`

使用 WebSocket 上的 MQTT 接收推送消息

要通过 WebSocket 上的 MQTT 从 MMS 接收设备推送消息，设备必须通过 MQTT 客户端连接到 MMS 并建立连接，如下所示：

`wss://<主机>/com.sap.iotservices.mms/v1/api/ws/mqtt`

MQTT 客户端使用的客户端标识是分配到 RDMS 中相应设备的设备标识。此外，MQTT 客户端必须将“用户名”和“密码”分别设置为“设备标识”和“设备 *OAuth*” 令牌。

最终，设备必须订阅推送主题：`iot/push/<device id>`

还可以使用 MMS 数据服务示例客户端通过 WebSocket 上的 MQTT 接收推送消息。在 MMS 主页中，可以通过选择“通过 *WS* 上的 *MQTT* 进行消息传送”磁贴选项或者打开以下 URL 访问此客户端：

`https://<主机>/com.sap.iotservices.mms/#/dataservice/ws-mqtt`

使用 TCP 上的 MQTT 接收推送消息

如果要使用 TCP 上的 MQTT，必须先设置 MQTT 代理，并按照 第 85 页上的“TCP 上的 MQTT” 小节所述进行配置。

配置 MQTT 代理后，当设备要通过 TCP 上的 MQTT 从 MMS 接收推送消息时，设备必须连接到您的 MQTT 代理，如下所示：

```
tcp://<MQTT broker host>
```

MQTT 客户端使用的客户端标识是分配到 RDMS 中相应设备的设备标识。此外，MQTT 客户端必须按照 MQTT 代理配置设置“用户名”和“密码”。

最终，设备必须订阅以下推送主题：`iot/push/iotmms<account id>/v1/<device id>`

还可以使用 MMS 数据服务示例客户端通过 TCP 上的 MQTT 接收推送消息。在 MMS 主页中，可以通过选择“通过 *TCP* 上的 *MQTT* 进行消息传送”磁贴选项或者打开以下 URL 访问此客户端：

```
https://<主机>/com.sap.iotservices.mms/#/dataservice/mqtt
```

相关信息

第 40 页上的“设备类型”
第 67 页上的“接口和通信处理”
第 74 页上的“数据服务”
第 75 页上的“HTTP”
第 80 页上的“WebSocket”
第 82 页上的“WebSocket 上的 MQTT”
第 83 页上的“TCP 上的 MQTT”
第 91 页上的“推送服务 API 验证”

8.4.1 推送服务 API 验证

通过“基本验证”和 *“OAuth”* 保护推送服务 API。

基本验证

> **i 注意**
>
> 如果使用基本验证，请确保更改密码后更新凭据。如果验证请求失败太多，SAP BTP 将自动屏蔽对应用户一段时间。

相应的用户需分配到 *"IoT-MMS-User"* 角色。有关详细信息，请参阅第 18 页上的"为 Java iotmms 应用程序分配 IoT-MMS-User 角色"部分。令牌需作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Basic <base64-encoded username:password>
```

OAuth

为使用 OAuth 进行验证，需执行几个步骤来创建 OAuth 客户端，定义 MMS 所需范围并且通过 *"push"* 范围请求 OAuth 令牌。

过程

1.使用 SAP BTP 主控室创建"授权许可"类型为"客户端凭据"的 OAuth 客户端。有关详细信息，请参阅配置 OAuth 2.0 -> **注册 OAuth 客户端**部分。

2.使用 SAP BTP 主控室为 'iotmms' 部署定义 OAuth 范围。有关详细信息，请参阅配置 OAuth 2.0 -> **定义 OAuth 范围**。范围列表取决于想要启用 OAuth 访问的 API。

- **推送**用于访问 `https://<主机>/com.sap.iotservices.mms/v1/api/http/push`

3.通过 REST 客户端请求 OAuth 令牌。有关详细信息，请参阅启用 OAuth 2.0 客户端凭据授予部分。为此，请为您的 SAP BTP 账户列出的令牌端点 URL 执行 HTTP POST。确保您已定义客户端应授权使用的范围集。如果客户端只能访问推送 API，则只需要**推送**范围。样例请求如下所示：

```
POST https://<token endpoint URL>
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <base64-encoded oauth_client_id:oauth_client_secret>
grant_type=client_credentials&
scope=<space separated list of scopes that should be enabled for the client>
```

作为 HTTP 响应的一部分返回的令牌需要作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Bearer <oauth token>
```

相关信息

第 88 页上的"推送服务"

8.5 消息批次

MMS 可以批量处理收到的消息，以避免对性能产生不利影响。为实现这一目的，会将消息缓存在内存中，直到达到定义的消息数量（缺省消息数量为 5,000）。此外，计时器会在预定义的时间间隔（缺省值为 5 秒）内触发处

理，以防在该期限内未达到批量大小限制。因此，请求仅发送少量的消息。为此，在收到 MMS 的响应之前，可能会阻止未以同步模式达到批量限制的消息一段时间。

配置批次

MMS 允许您调整缺省批次行为。可以使用配置 API 设置相应属性。有关详细信息，请参阅第 72 页上的“配置 API 验证”部分。

以下属性影响批处理。

属性	说明
`mms.processing.force_timeout_milliseconds`	如果没有完整的消息批次可用，最长经过多少毫秒可以处理消息。期望值为整数，部署的缺省值为 5,000。
`mms.processing.check_period_for_commit_timeout_milliseconds`	最长经过多少毫秒将检查处理服务超时。该数值应小于实际处理超时。期望值为整数，部署的缺省值为 1,000。
`mms.processing.batch_size`	处理服务在单个批次中处理的最大消息数量。接收的消息在内存中维护并在达到大小之后以单个批次进行处理。期望值为整数，部署的缺省值为 5,000。

相关信息

第 67 页上的“接口和通信处理”
第 72 页上的“配置 API 验证”
第 31 页上的“配置设备”

8.6 消息处理

MMS 为处理传入消息提供不同方法。基于实际用例，消息需要保留在数据库中或转发到消息代理、事件流处理器、文档资源库或任何其他系统。为保证此程度的灵活性，MMS 提供处理服务。*“sql”* 是缺省处理服务，意味着所有传入消息都存储在数据库中。除此之外，MMS 提供 *“cmis”* 处理服务，此服务可以将传入二进制消息转发到文档资源库。您也可以使用 *“noop”* 处理服务来暂时禁用消息处理。在此案例中，所有消息将被放弃。关于可用处理服务的详细信息，请参阅“相关信息”部分。

处理服务根据设备类型标识和消息类型标识对注册。

处理服务用户界面

处理服务可通过使用处理服务用户界面管理，此界面可通过 MMS 主控室或使用处理服务 API 访问。

处理服务 API

现有映射配置可以通过提交 HTTP GET 请求来检索：

`https://<host>/com.sap.iotservices.mms/v1/api/http/processing/<设备类型标识>/<消息类型标识>`

设备类型标识和消息类型标识对的现有映射可以通过提交 HTTP DELETE 请求来删除：

`https://<host>/com.sap.iotservices.mms/v1/api/http/processing/<设备类型标识>/<消息类型标识>`

> **i 注意**
>
> 正如推送 API，处理服务 API 通过基本验证和 OAuth 受到保护（使用**处理**范围）。有关详细信息，请参阅第 95 页上的“处理服务 API 验证”部分。

处理服务 API 可使您为每对设备类型标识和消息类型标识定义多重处理服务。要使用此功能，请添加所需服务到服务注册请求：

```
PUT https://<host>/com.sap.iotservices.mms/v1/api/http/processing
Content-Type: application/json;charset=utf-8
{
  "deviceType":"<device type id>",
  "messageType":"<message type id>",
  "processingServices":[
    {
      "name":"sql"
      ...
    },
    {
      "name":"http"
      ...
    }
  ]
}
```

如果注册多重服务，只要所有注册服务成功处理请求，MMS 就会以成功代码 `20x` 响应。如果不成功，MMS 返回错误代码 `500` 以及表明服务失败的消息列表。

有关服务特定配置的更多详细信息，请参阅处理服务特定的部分。

相关信息

第 95 页上的“处理服务 API 验证”
第 96 页上的“数据库存储 (SQL)”

第 99 页上的“文档存储 (CMIS)”
第 100 页上的“消息转发 (HTTP)”
第 102 页上的“智能数据流 (SDS)”

8.6.1 处理服务 API 验证

通过**基本验证**和 **OAuth** 保护处理服务 API。

基本验证

> **i 注意**
>
> 如果使用基本验证，请确保更改密码后更新凭据。如果验证请求失败太多，SAP BTP 将自动屏蔽对应用户一段时间。

相应的用户需分配到 *“IoT-MMS-User”* 角色。有关详细信息，请参阅第 18 页上的“为 Java iotmms 应用程序分配 IoT-MMS-User 角色”部分。令牌需作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Basic <base64-encoded username:password>
```

OAuth

为使用 OAuth 进行验证，需执行几个步骤来创建 OAuth 客户端，定义 MMS 所需范围并且通过**处理**范围请求 OAuth 令牌。

过程

1.使用 SAP BTP 主控室创建“授权许可”类型为“客户端凭据”的 OAuth 客户端。有关详细信息，请参阅配置 OAuth 2.0 -> **注册 OAuth 客户端**部分。

2.使用 SAP BTP 主控室为 `iotmms` 部署定义 OAuth 范围。有关详细信息，请参阅配置 OAuth 2.0 -> **定义 OAuth 范围**。范围列表取决于想要启用 OAuth 访问的 API。

- **处理**以访问 `https://<host>/com.sap.iotservices.mms/v1/api/http/processing`

3.通过 REST 客户端请求 OAuth 令牌。有关详细信息，请参阅 OAuth 2.0 客户端凭据授予部分。为此，请为您的 SAP BTP 账户列出的令牌端点 URL 执行 HTTP POST。确保您已定义客户端应授权使用的范围集。如果客户端只能访问处理服务 API，则只需要**处理**范围。样例请求如下所示：

```
POST https://<token endpoint URL>
Content-Type: application/x-www-form-urlencoded
```

```
Authorization: Basic <base64-encoded oauth_client_id:oauth_client_secret>
grant_type=client_credentials&
scope=<space separated list of scopes that should be enabled for the client>
```

作为 HTTP 响应的一部分返回的令牌需要作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Bearer <oauth token>
```

相关信息

第 93 页上的“消息处理”
第 96 页上的“数据库存储 (SQL)”
第 99 页上的“文档存储 (CMIS)”
第 100 页上的“消息转发 (HTTP)”
第 102 页上的“智能数据流 (SDS)”

8.6.2 数据库存储 (SQL)

缺省情况下，MMS 将 SQL 处理服务应用到传入消息中。因此，会将消息存储到相关的数据库中。

数据库和模式绑定

部署在 SAP BTP 上的每一个应用程序都可以绑定一个或多个数据源。数据源是专用数据库。

可以通过 SAP BTP 主控室创建数据库和模式。有关详细信息，请参考 SAP BTP 文档中的创建数据库或使用主控室创建模式部分。成功创建数据源之后，可以通过创建数据源绑定将数据源分配到 MMS。有关详细信息，请分别参阅绑定模式或将 SAP HANA 数据库绑定到 Java 应用程序部分。确保创建 <DEFAULT> 数据源绑定。

> **i 注意**
>
> 如果未显式定义数据源绑定，MMS 会在启动时尝试自动在共享数据库中创建模式并绑定到该模式。但是，此操作只能在提供共享数据库的架构中执行。

数据库表和表映射

缺省情况下，MMS 将传入消息存储到相关数据库。MMS 会基于设备类型和消息类型定义自动创建各自的消息表。如果具有相应名称和列的表已存在，MMS 会将消息存储在该表中，而不是创建新的表。

> **i 注意**
>
> 一个设备只能分配到一种设备类型。设备类型是消息类型的集合。消息类型由一系列消息类型字段构成。每个字段包含一个名称、位置和数据类型。

消息表的类型从消息类型标识派生。MMS 根据命名模式 `T_IOT_[Message Type ID]` 为每个消息类型创建一个表。例如，如果创建了支持 `m0t0y0p0e1` 和 `m0t0y0p0e2` 的设备类型，MMS 会生成两个数据库表，名称为 `T_IOT_M0T0Y0P0E1` 的第一个表和名称为 `T_IOT_M0T0Y0P0E2` 的第二个表。类型为 `m0t0y0p0e1` 的消息将存储在第一个表中，类型为 `m0t0y0p0e2` 的消息将存储在第二个表中。消息源通过 `G_DEVICE` 列标识，该列存储了发送消息的设备的标识符。此外，会自动创建名称为 `G_CREATED` 的列用来存储已插入消息的时间戳。

消息表模式的其余列直接从消息类型定义派生。将每个消息字段映射到以前缀 `C_` 开头的列名称。这对于遵守多个数据库中列名称的命名规则十分必要，例如使用禁止的特定关键字。列类型从消息字段类型和基础数据库类型之间的静态映射派生。

定制表映射

如上所述，缺省情况下，将 MMS 配置为将 SOL 处理服务应用到传入消息。这意味着遵循列出的表创建和表命名方法将消息存储到数据库。

如果用例要求定制的表或列名称而不是自动创建的表或列名称，例如，因为目标表已存在，可以通过为设备类型对和相应消息类型配置 SOL 处理服务实现这一要求。可以通过将 HTTP PUT 请求提交到以下地址来调整 SQL 处理服务配置：

```
https://<主机>/com.sap.iotservices.mms/v1/api/http/processing
```

> **i 注意**
>
> 通过**基本验证**和 **OAuth** 保护处理服务 API（使用**处理**范围）。有关详细信息，请参阅第 95 页上的“处理服务 API 验证”部分。

为符合**设备类型标识 d0t0y0p0e1** 的**消息类型标识 m0t0y0p0e1** 的消息定制映射的样例请求如下所示：

```
PUT https://<host>/com.sap.iotservices.mms/v1/api/http/processing
Content-Type: application/json;charset=utf-8
{
  "deviceType": "d0t0y0p0e1",
  "messageType": "m0t0y0p0e1",
  "processingServices": [
    {
      "name": "sql",
      "properties": {
        "mms.processing.sql.mappings": {
          "table_name": "my_table",
          "column_names": [
            {
              "field": "sensor",
              "column": "my_sensor"
            },
            {
              "field": "timestamp",
              "column": "my_timestamp"
            }
          ],
```

```
            "add_default_columns": false
          },
          "mms.processing.sql.schema_name": "my_schema",
          "mms.processing.sql.hana.table_type": "column"
        }
      }
    ]
}
```

在此示例中，将表名称设置为 **my_ table** 而不是 `T_IOT_M0T0Y0P0E1`。

此外，定制列名称。处理服务将为消息类型字段 **sensor** 使用列名称 **my_sensor** 而不是 `C_SENSOR`，并且为消息类型字段 **timestamp** 使用 **my_timestamp** 而不是 `C_TIMESTAMP`。请注意，会根据缺省列名称模式为未显式映射到自定义列名称的消息类型字段分配自动生成的名称。另外还需注意，通过禁用缺省列可以禁用 MMS 根据缺省自动生成的所有 `G_` 列。

此外，将模式名称设置为 **my_schema** 而不是分配给数据库用户的缺省模式。

此外，定制表类型。根据映射，会将表创建为**列**存储表而不是**行**存储表。此属性仅影响 SAP HANA 数据库表。如果 MMS 绑定到其他数据库，将会忽略此属性。

> **i 注意**
>
> MMS 假设目标模式已存在并且相应用户拥有在模式中创建表所需的所有权限。

> **i 注意**
>
> 向 MMS 发送特定消息类型的消息并因此创建目标表之后，便无法再更改列名称映射。对于已创建的表，MMS 随后无法进行更改。因此，您可以直接在数据库中手动变更表模式或更改表类型（例如，从行存储到列存储），或者必须删除该表以允许 MMS 创建符合新的列名称的新表。

您还可以使用 MMS 主控室的处理服务配置用户界面配置 SQL 处理服务。

相关信息

管理数据库模式
使用主控室创建模式
将 SAP HANA 数据库绑定到 Java 应用程序
第 93 页上的“消息处理”
第 95 页上的“处理服务 API 验证”
第 99 页上的“文档存储 (CMIS)”
第 100 页上的“消息转发 (HTTP)”
第 102 页上的“智能数据流 (SDS)”

8.6.3 文档存储 (CMIS)

除 SQL 处理服务之外，MMS 还提供了 CMIS 处理服务。可用于将具有二进制消息类型字段的传入消息转发到 SAP BTP 账户内的文档资源库。

文档资源库

文档资源库可使用 SAP BTP 主控室管理。有关详细信息，请参阅 SAP BTP 文档中的在主控室中管理资源库部分。

作为使用 CMIS 处理服务的前提条件，文档资源库必须已就位。MMS 本身不会创建任何资源库。请确保您已记下想要在其中存储二进制消息的资源库的**资源库名称**和**资源库键值**。

文档映射

CMIS 处理服务仅适用于包含**二进制**消息类型字段的消息类型。此字段中发送的内容将作为文档存储在资源库中。此外，需要类型为**字符串**的消息类型字段。该字段用于定义存储文档的名称。

有关消息类型字段的详细信息，请参阅第 47 页上的“数据类型”部分。

有效的消息类型为部分介绍的消息类型 **m0t0y0p0e5**。此消息类型具有一个称为 **file** 的**二进制**类型字段和一个称为 **filename** 的**字符串**类型字段。

服务配置

可以通过将 HTTP PUT 请求提交到以下地址来配置 CMIS 处理服务：

```
https://<host>/com.sap.iotservices.mms/v1/api/http/processing
```

> **i 注意**
>
> 通过**基本验证**和 **OAuth** 保护处理服务 API（使用**处理**范围）。有关详细信息，请参阅第 95 页上的“处理服务 API 验证”部分。

为符合**设备类型标识 d0t0y0p0e1** 的**消息类型标识 m0t0y0p0e1** 的消息注册 CMIS 处理服务的样例请求如下所示：

```
PUT https://<host>/com.sap.iotservices.mms/v1/api/http/processing
Content-Type: application/json;charset=utf-8
{
  "deviceType": "d0t0y0p0e2",
  "messageType": "m0t0y0p0e5",
  "processingServices": [
    {
      "name": "cmis",
      "properties": {
        "mms.processing.cmis.repo_name": "my_document_repo",
        "mms.processing.cmis.repo_key": "my_document_repo_key",
```

```
            "mms.processing.cmis.binary_content_field": "file",
            "mms.processing.cmis.string_name_field": "filename",
            "mms.processing.cmis.target_folder_path": "my_folder",
            "mms.processing.cmis.overwrite_existing_files": true
          }
        }
      ]
}
```

在此示例中，将配置 CMIS 处理服务以使用具有键值 **my_document_repo_key** 的文档资源库 **my_document_repo**。

此外，将通过二进制消息类型字段的名称将其绑定到文档内容。对于在此示例中使用的消息类型，字段称为 **file**。另外，在此示例中，还将字符串消息类型字段绑定到文档名称 **filename**。

最后两个属性为可选属性。**Target_folder_path** 属性可用来决定存储文档的文件夹。**overwrite_existing_files** 决定如果发送了具有相同名称的新文档，是否应覆盖已存在的文档。

您还可以使用 MMS 主控室的处理服务配置用户界面配置 CMIS 处理服务。

相关信息

第 93 页上的“消息处理”
第 95 页上的“处理服务 API 验证”
第 96 页上的“数据库存储 (SQL)”
第 100 页上的“消息转发 (HTTP)”
第 102 页上的“智能数据流 (SDS)”

8.6.4 消息转发 (HTTP)

HTTP 处理服务可用于将传入消息转发到其他 web 服务。根据用例，服务可以转发关于传入消息的实际消息或通知。无论哪种方式，HTTP POST 请求都会发送到配置的 URL。

服务配置

可以通过将 HTTP PUT 请求提交到以下地址来配置 HTTP 处理服务：

```
https://<host>/com.sap.iotservices.mms/v1/api/http/processing
```

> **i 注意**
>
> 通过**基本验证**和 **OAuth** 保护处理服务 API（使用**处理**范围）。有关详细信息，请参阅第 95 页上的“处理服务 API 验证”部分。

对于符合**设备类型标识 d0t0y0p0e1** 的**消息类型标识 m0t0y0p0e1**，为消息注册 HTTP 处理服务的样例请求如下：

```
   PUT https://<host>/com.sap.iotservices.mms/v1/api/http/processing
   Content-Type: application/json;charset=utf-8
   {
     "deviceType":"d0t0y0p0e1",
     "messageType":"m0t0y0p0e1",
     "processingServices":[
       {
         "name":"http",
         "properties":{
           "mms.processing.http.url":"https://my_service_endpoint",
           "mms.processing.http.send_mode":"messages",
           "mms.processing.http.auth_mode":"basicauth",
           "mms.processing.http.basicauth.username":"<username>",
           "mms.processing.http.basicauth.password":"<password>"
           "mms.processing.http.proxy_enabled":<true|false>,
           "mms.processing.http.proxy_host":"<proxy host>",
           "mms.processing.http.proxy_port":<proxy port>
         }
       }
     ]
}
```

在此示例中，配置 HTTP 处理服务，从而将 HTTP POST 请求发送至 URL `https://my_service_endpoint`。

示例中使用的 **send_mode** 是消息。这意味着将所有传入**消息**转发至相应 URL。这种情况下，请求有效负载如下所示：

```
   {
     "device":"d000-e000-v000-i000-c000-e001",
     "messageType":"m0t0y0p0e1",
     "messages":
[{"sensor":"sensor1","value":"20","timestamp":"2014-10-13T11:14:10.000+02:00"}]
}
```

服务支持的第二个 **send_mode** 是**通知**。这意味着只将关于传入消息的通知转发至相应 URL。这种情况下，请求有效负载如下所示：

```
   {
     "device":"d000-e000-v000-i000-c000-e001",
     "messageType":"m0t0y0p0e1",
     "messageCount":1
}
```

HTTP 请求将按照设备和不同消息类型发送。

此外，可使用不同的验证模式配置服务。您当前可以配置以下模式：**none**、**basicauth** 或 **oauth**。上述示例中使用的模式为需要**用户名**和**密码**的 **basicauth**。

要使用验证模式 **oauth**，您需要更改验证模式并提供**令牌**而不是用户名和密码。以下示例显示了服务配置。

```
"processingServices": [
  {
    "name":"http",
    "properties":{
      "mms.processing.http.url":"https://my_service_endpoint",
      "mms.processing.http.send_mode":"messages",
      "mms.processing.http.auth_mode":"oauth",
      "mms.processing.http.oauth.token":"<token>"
```

```
    }
  }
]
```

此外，您可以通过添加可选属性自由配置 HTTP 标头。

```
"mms.processing.http.headers":
[
  {
    "name":"http_header_name1",
    "value":"http_header_value1"
  },...,
  {
    "name":"http_header_nameN",
    "value":"http_header_valueN"
  }
  ]
```

> **i 注意**
>
> 请注意，您可能需要为 Internet 连接性配置代理。要验证当前代理配置，请检查相应 JVM 系统属性。

系统属性	hana.ondemand.com
`http.proxyHost` 和 `https.proxyHost`	proxy
`http.proxyPort` 和 `https.proxyPort`	8080

您还可以使用 MMS 主控室的处理服务配置用户界面配置 HTTP 处理服务。

相关信息

第 93 页上的“消息处理”
第 95 页上的“处理服务 API 验证”
第 96 页上的“数据库存储 (SQL)”
第 102 页上的“智能数据流 (SDS)”

8.6.5 智能数据流 (SDS)

SDS 处理服务可用于将传入消息转发到 SAP BTP 流式分析。

SDS 允许您构建实时处理传入事件数据流的应用程序，并收集和处理传入信息。有关 SDS 的常规信息以及配置说明的详细信息，请参阅 SAP 流式分析文档。

SDS 服务启用

SDS 处理服务使用 SDS 提供的流处理 Web 服务 (SWS) 接口来将消息转发到流引擎。有关如何启用 SDS 以及设置 SWS 的详细信息，请参阅相应的 SAP 流式分析文档。

SDS 项目设置

SDS 处理服务会将特定消息类型的所有消息转发到相应的 SDS 输入流。因此，消息类型的模式和输入流的模式必须共享同一属性。消息类型字段到输入流字段的映射与 SQL 处理服务将消息类型字段映射到表列的方式非常相似。第 96 页上的“数据库存储 (SQL)”部分中的**数据库表和表映射**子部分中说明了该映射方法。在此基础上，第 67 页上的“接口和通信处理”部分中**样例设备和消息类型**子部分中定义的样例消息类型 **m0t0y0p0e1** 对应的用于创建 SDS 输入流的连续计算语言 (CCL) 文件如下所示：

```
   CREATE INPUT STREAM T_IOT_ M0T0Y0P0E1 SCHEMA (
         g_device string,
         g_created msdate,
         c_sensor string,
         c_value string,
c_timestamp msdate);
```

在上述示例中，您可以看到自动生成的对应每个前缀为 `c_` 的消息类型字段的一个输入流字段和两个附加输入流字段，其中包含消息插入的设备标识 (`g_device`) 和时间戳 (`g_created`)。

在您选择的 SDS 项目和工作区中创建输入流后，可以自由根据用例需要的输入流建模其他适配器和查询。

服务配置

可以通过将 HTTP PUT 请求提交到以下网址来配置 SDS 处理服务：

```
https://<host>/com.sap.iotservices.mms/v1/api/http/processing
```

> **i 注意**
>
> 通过**基本验证**和 **OAuth** 保护处理服务 API（使用**处理**范围）。有关详细信息，请参阅第 95 页上的“处理服务 API 验证”部分。

对于符合**设备类型标识 d0t0y0p0e1** 的**消息类型标识 m0t0y0p0e1** 的消息，注册 SDS 处理服务的样例请求如下所示：

```
PUT https://<host>/com.sap.iotservices.mms/v1/api/http/processing
Content-Type: application/json;charset=utf-8
{
    "deviceType":"d0t0y0p0e1",
    "messageType":"m0t0y0p0e1",
    "processingServices":[
        {
            "name":"sds",
            "properties":{
                "mms.processing.sds.protocol":"wss|https",
```

```
                "mms.processing.sds.host":"<streaming web service host>",
                "mms.processing.sds.port":<streaming web service port>,
                "mms.processing.sds.user":"<username>",
                "mms.processing.sds.pwd":"<password>",
                "mms.processing.sds.workspace":"<workspace name>",
                "mms.processing.sds.project":"<project name>",
                "mms.processing.sds.input_stream":"<input stream name>",
                "mms.processing.sds.proxy_enabled":<true|false>,
                "mms.processing.sds.proxy_host":"<proxy host>",
                "mms.processing.sds.proxy_port":<proxy port>
            }
        }
    ]
}
```

在服务配置请求中，您可以定义应使用何种**协议**来将数据转发到定义的流处理 Web 服务端点（**主机**和**端口**）。目前支持 HTTP 和 WebSocket。您还需提供**用户**和**密码**来连接到端点。

此外，SDS 处理服务配置必须包含目标 SDS **工作区**、**项目**和 **input_stream**。

> **i 注意**
>
> 请注意，您可能需要为 Internet 连接性配置代理。要验证当前代理配置，请检查相应 JVM 系统属性。

系统属性	hana.ondemand.com
`http.proxyHost` 和 `https.proxyHost`	proxy
`http.proxyPort` 和 `https.proxyPort`	8080

您还可以使用 MMS 主控室的处理服务配置用户界面配置 SDS 处理服务。

相关信息

SAP 流式分析
第 93 页上的“消息处理”
第 95 页上的“处理服务 API 验证”
第 96 页上的“数据库存储 (SQL)”
第 99 页上的“文档存储 (CMIS)”
第 100 页上的“消息转发 (HTTP)”

8.7 消息使用

缺省情况下，MMS 将接收的消息存储到附加到 MMS 实例的数据库中，该实例在使用者账户中运行。

> **i 注意**
>
> 请注意，本部分仅适用于已使用 SQL 处理服务处理的消息。使用 CMIS 处理服务存储为文档的消息需要通过 SAP BTP 提供的“文档服务”进行使用。

您可以在 SAP BTP 主控室的“数据库和模式”部分配置数据库模式绑定。有关详细信息，请参阅 SAP BTP 文档中的管理数据库模式部分。第 93 页上的“消息处理”部分详细介绍了 MMS 用于存储消息的表名称和表模式。

MMS 在使用者账户中运行，因此开发人员拥有基础数据库的完全访问权限。因此，IoT 业务应用程序可以通过直接访问生成的表，利用基础数据库提供的完整性能和功能使用消息。

SAP BTP 提供了两个使用数据库中数据的选项：

- 使用 Java 应用程序和 SAP HANA / SAP ASE 服务（JDBC 或 JPA）
- 使用本机 SAP HANA XS 应用程序（SAP HANA 扩展应用程序服务）

此外，MMS 还提供了 OData API，可用于访问表并使用 MMS 维护的消息。该 API 主要用于需要快速建模 IoT 业务应用程序的用例。对于生产用例，我们建议您使用单独组件直接从数据库使用消息。

通过 Java 和 SAP HANA / SAP ASE 服务进行使用

SAP HANA / SAP ASE 服务为托管在 SAP BTP 上的应用程序提供内存和关系数据库存储设备。使用 SAP HANA / SAP ASE 服务的应用程序可以通过纯 JDBC 和 JPA 构建。要使用 SAP HANA / SAP ASE 服务访问存储的消息，需要将使用的 Java 应用程序绑定到由 MMS 用来存储消息的相同数据库模式。

有关 SAP HANA / SAP ASE 服务的详细信息，请参阅 SAP BTP 文档中的在 Neo 环境中使用服务部分。

通过 SAP HANA XS 进行使用

使用 SAP HANA XS，SAP BTP 提供了运行时以在云中构建和执行本机 XS 应用程序。应用程序可以使用 JavaScript 直接访问数据库，或者可以自动生成 OData Web 服务。要使用 XS 访问 MMS 消息，必须将 MMS 绑定到专用的 SAP HANA XS 实例。要创建这样的实例并将数据库模式绑定到 MMS，您可以使用 SAP BTP 主控室。

有关整体 XS 功能的详细信息，请参阅 SAP HANA Platform 文档中的 SAP HANA 开发人员指南。

SAP HANA 平台提供了两种访问数据库的方法：生成稍后可以通过 HTTP 使用的 **OData Web 服务**或使用 **SAP HANA XS 数据库访问 API**。这两种方法在 SAP HANA Platform 文档的 SAP HANA 开发人员指南的创建 OData 服务定义和编写服务器端 JavaScript 代码部分中有详细描述。

使用内置 OData API 使用

MMS 提供了 OData 2.0 版兼容 API，可用于浏览所有已创建的表并使用 MMS 存储的消息。Odata 服务文档可通过以下端点进行检索：

`https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc`

> **i 注意**
>
> 通过**基本验证**和 **OAuth** 保护 OData API（使用**应用**范围）。有关详细信息，请参阅第 107 页上的“OData API 验证”部分。

响应包含 MMS 维护的表的集合。表的更多详细描述及其列可通过使用以下端点访问元数据文档来进行检索：`https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/$metadata`

来自特定表的消息可通过以下端点访问：`https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/<table name>`

响应包含存储在相应表中的所有消息。缺省情况下，会将响应格式化为 AtomPub。但是，通过添加查询参数 $format=json，可以将该格式更改为 JSON。

此外，支持以下查询参数：

- 从表中获取消息计数 (**$count**)：`https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/[table name]/$count`
- 获取消息的限制数量 (**$top**)：`<https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/[table name]?$top=2`
- 获取含偏移的消息 (**$skip**)：`<https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/[table name]?$skip=2`
- 获取按消息字段排序的消息 (**$orderby**)：`<https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/[table name]?$orderby=G_CREATED [asc|desc]`
- 获取按消息字段值过滤的消息（**$filter** - 支持的操作符为：eq、neq、lt、le、gt、ge 和 or）：`https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/[table name]]?$filter=G_DEVICE eq 'd000-e000-v000-i000-c000-e001'`
- 获取具有消息字段子集的消息 (**$select**)：`<https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/[table name]?$select=G_DEVICE`

还可以组合查询参数以创建更复杂的查询。具有排序和 JSON 格式化的页码示例为：`https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/[table name]?$skip=0&$top=50&$orderby=G_CREATED desc&$format=json`

API 还支持使用 OData 2.0 版说明中定义的 **$batch** 参数进行批处理：`https://<host>/com.sap.iotservices.mms/v1/api/http/app.svc/$batch`

8.7.1 OData API 验证

通过“基本验证”和 *“OAuth”* 方法来保护 API。

基本验证

> **i 注意**
>
> 如果使用基本验证，请确保更改密码后更新凭据。如果验证请求失败太多，SAP BTP 将自动屏蔽对应用户一段时间。

相应的用户需分配到 *“IoT-MMS-User”* 角色。有关详细信息，请参阅第 18 页上的“为 Java iotmms 应用程序分配 IoT-MMS-User 角色”部分。令牌需作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Basic <base64-encoded username:password>
```

OAuth

为使用 OAuth 进行验证，需执行几个步骤来创建 OAuth 客户端，定义 MMS 所需范围并且通过 **app** 范围请求 OAuth 令牌。

过程

1.使用 SAP BTP 主控室创建“授权许可”类型为“客户端凭据”的 OAuth 客户端。有关详细信息，请参阅配置 OAuth 2.0 -> **注册 OAuth 客户端**部分。

2.使用 SAP BTP 主控室为 *“iotmms”* 部署定义 OAuth 范围。有关详细信息，请参阅配置 OAuth 2.0 -> **定义 OAuth 范围**。范围列表取决于想要启用 OAuth 访问的 API。

- **应用**用于访问 `https://<主机>/com.sap.iotservices.mms/v1/api/http/app[.svc]`

3.通过 REST 客户端请求 OAuth 令牌。有关详细信息，请参阅启用 OAuth 2.0 客户端凭据授予部分。为此，请为您的 SAP BTP 账户列出的令牌端点 URL 执行 HTTP POST。确保您已定义客户端应授权使用的范围集。如果客户端应只能访问 OData API，则只需要**应用**范围。样例请求如下所示：

```
POST https://<token endpoint URL>
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <base64-encoded oauth_client_id:oauth_client_secret>
grant_type=client_credentials&
scope=<space separated list of scopes that should be enabled for the client>
```

作为 HTTP 响应的一部分返回的令牌需要作为请求标头添加到所有请求。标头具有以下模式：

```
Authorization: Bearer <oauth token>
```

相关信息

第 104 页上的“消息使用”

法律角度的重要免责声明

超链接

部分链接按图标和/或鼠标悬停文本分类。这些链接能够提供更多信息。
关于图标：

- 带有图标  的链接：您正访问不是由 SAP 托管的网站。使用这些链接，即视为您同意（除非您与 SAP 的协议中另有明确规定）以下内容：
  - 链接的网站的内容不属于 SAP 文档。您不得基于这些信息对 SAP 提出任何产品索赔。
  - SAP 不对链接的网站的内容提出同意或反对，也不保证其可用性和正确性。SAP 不对使用此类内容造成的损害承担任何责任，除非损害是由于 SAP 的重大过失或故意的不当行为造成的。
- 带有图标  的链接：您将离开该特定 SAP 产品或服务的文档，访问 SAP 托管的网站。使用这些链接，即视为您同意（除非您与 SAP 的协议中另有明确规定）您不得基于这些信息对 SAP 提出任何产品索赔。

外部平台上承载的视频

一些视频可能指向第三方视频承载平台。SAP 无法保证第三方平台上存储的视频将来可供观看。此外，这些平台上承载的任何广告或其他内容（例如，建议观看的视频或导航到同一站点上承载的其他视频）均与 SAP 无关，也不受 SAP 控制。

测试功能和其他实验功能

实验功能不一定是 SAP 未来版本官方交付范围的一部分。这意味着，SAP 可随时出于任何原因对实验功能做出更改，且不再另行通知。实验功能不得用于生产用途。您不得在上线运营环境中或者使用尚未充分备份的数据来演示、测试、评估或以其他方式使用实验功能。
实验功能旨在尽早获得反馈，从而支持客户和合作伙伴相应地影响未来产品。您（在 SAP 社区中）提供反馈，即视为您接受文稿或衍生作品的知识产权应属于 SAP 的专有财产。

示例代码

任何软件编码和/或代码片段均为示例，不得用于生产用途。示例代码仅仅是为了更好地解释和呈现某些代码的语法和编写规则。SAP 不保证代码的准确性和完整性。SAP 不对使用示例代码造成的错误或损害承担任何责任，除非损害是由于 SAP 的重大过失或故意的不当行为造成的。

无偏见语言

SAP 支持多元和包容文化。在可能的情况下，我们在文档中会使用无偏见语言来指代所有文化、族裔、性别和能力的人。

www.sap.com/contactsap

© 2021 SAP 股份有限公司或其关联公司版权所有，保留所有权利。

未经 SAP 股份有限公司或其关联公司明确许可，任何人不得以任何形式或为任何目的复制或传播本文件的任何内容。本文件包含的信息可能会更改，且不再另行事先通知。

由 SAP 股份有限公司及其分销商营销的部分软件产品包含其它软件供应商的专有软件组件。各国的产品规格可能不同。

本资料由 SAP 股份有限公司或其关联公司提供，仅供参考，不构成任何形式的陈述或保证，其中如若存在任何错误或疏漏，SAP 或其关联公司概不负责。与 SAP 或其关联公司产品和服务相关的保证仅限于该等产品和服务随附的保证声明（若有）中明确提出的保证。本文件中的任何信息均不构成额外保证。

SAP 和本文件中提及的其它 SAP 产品和服务及其各自标识均为 SAP 股份有限公司（或其关联公司）在德国和其他国家的商标或注册商标。本文件中提及的所有其它产品和服务名称分别是其各自公司的商标。

如欲了解更多商标信息和声明，请访问：https://www.sap.com/about/legal/trademark.html。

THE BEST RUN SAP