设备发放

最佳实践

文档版本	01
发布日期	2021-08-23

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 设备厂商多实例接入方案

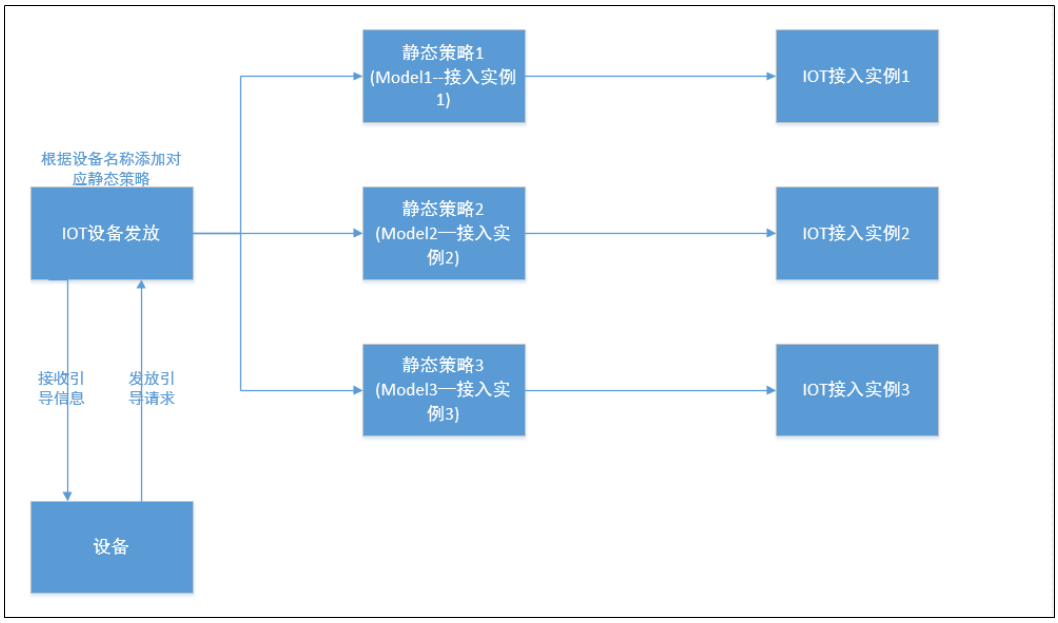
场景说明

随着厂商IoT设备的不断增加，或者厂商设备本来就面向多个销售中心（不同的销售中心有着自己的IoT接入实例），面对这样的场景，厂商在设备出厂的时候不知道设备最终需要连接到哪个IoT接入实例，设备出厂后不希望再对设备进行二次烧录。为此，华为IoT解决方案推出设备发放服务，设备出厂统一烧录设备发放的地址，通过在设备发放服务预置不同的发放策略，设备上电后自动发放到对应的IoT接入实例，并把对应的IoT接入实例地址下发给设备，设备就可以和目的接入实例实现通信。

前提条件

- 已注册华为云官方账号**设备发放服务**。未注册可单击**注册页面**完成注册。
- 已完成实名制认证。未完成可在华为云上单击**实名认证**完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问**设备接入服务**，单击“立即使用”后开通该服务。开通设备接入服务，进入到**设备发放服务**页面，单击“授权开通”对服务进行授权。
- 对应的设备接入服务已经上传了产品。

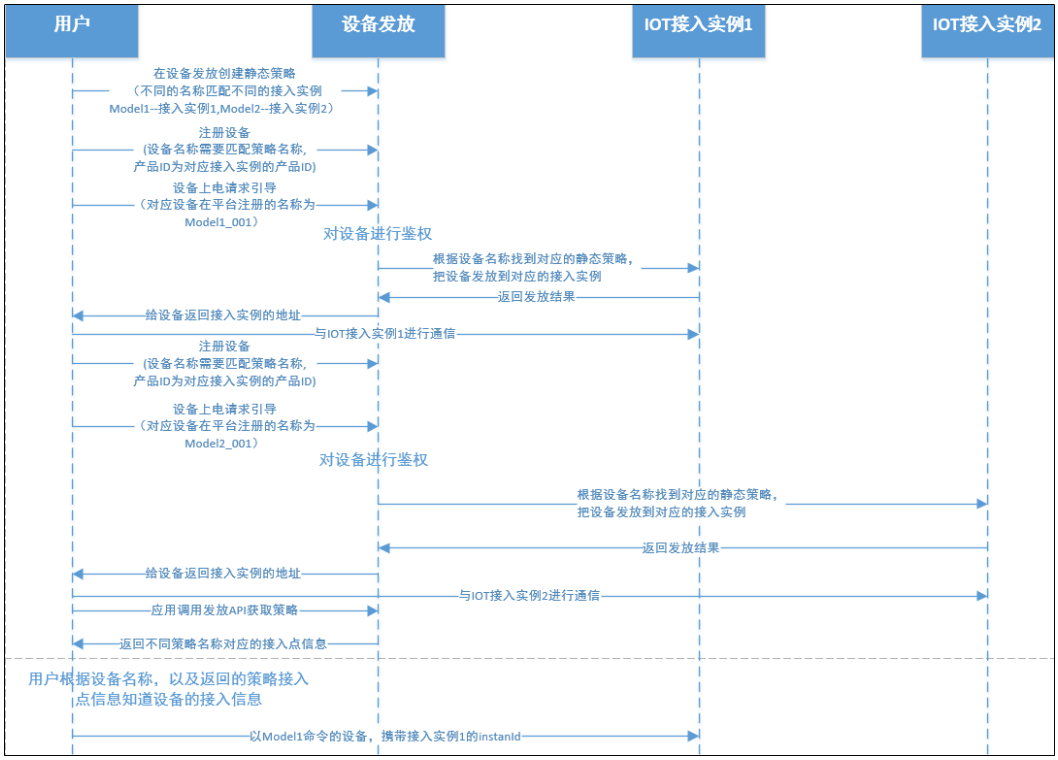
整体方案



基于MQTT协议的上行请求引导和下行接受引导信息的业务定义如下：

业务场景	通信Topic	报文Payload
设备请求引导消息	$oc/devices/{device_id}/sys/bootstrap/up	/
服务接收引导信息	$oc/devices/{device_id}/sys/bootstrap/down	{ "address": "10.0.0.1:8883" }

业务流程



1. 创建策略。其中以静态策略为例。
2. 注册设备。注册MQTT协议的设备，体验发放业务。可以注册单个，也可以批量注册设备。
3. 设备请求引导。
4. 设备接收信息，解析出IoT接入实例地址，与IoT接入实例1进行通信。
5. 动态扩容设备接入实例，创建对应实例的资源空间，在对应资源空间下上传产品。
6. 注册设备。选择对应实例的产品创建设备。
7. 新增策略。添加新实例的静态策略，以不同的关键字命名静态策略。
8. 设备请求引导。
9. 设备接收信息，解析出IoT接入实例地址，与IoT接入实例2进行通信。
10. 应用主动**设备发放控制台**与设备侧通信。

创建策略

有两种使用场景：**设备发放控制台**

1. 用户预先开通了多个实例，通过设备的不同销售数据，根据对应的策略，设备上电自动发放到对应的实例。
2. 用户预先只开通一个实例，预先创建对应的策略把设备发放到当前实例。当对应实例达到上限，动态开通新的接入实例，用户删除当前策略，添加新的策略，把新上电的设备发放到新开通的实例。

本示例讲述的是第二种使用场景。

登录**设备发放控制台**，进入到策略页面的静态策略页面，单击添加实例，根据关键字发放到指定的IoTDA实例。

图 1-1 创建静态策略

图 1-2 创建静态策略详情

注册设备

本次以注册单个MQTT密钥设备为示例，登录**设备发放控制台**，进入到“设备”页面，单击“注册”，选择对应接入点的产品进行设备注册，设备名称为“Model1_001”，填写设备的密钥，发放策略选择“静态策略”，创建设备。

说明

实际使用中，用户可以使用批量注册功能注册大量设备。下载批量注册模板，按照模板说明正确填写后上传批量注册文件以批量注册设备。建议每个设备名称的以策略的关键字开头（比如策略关键字为Model1，设备名称为Model1_XXX001）。

图 1-3 设备列表

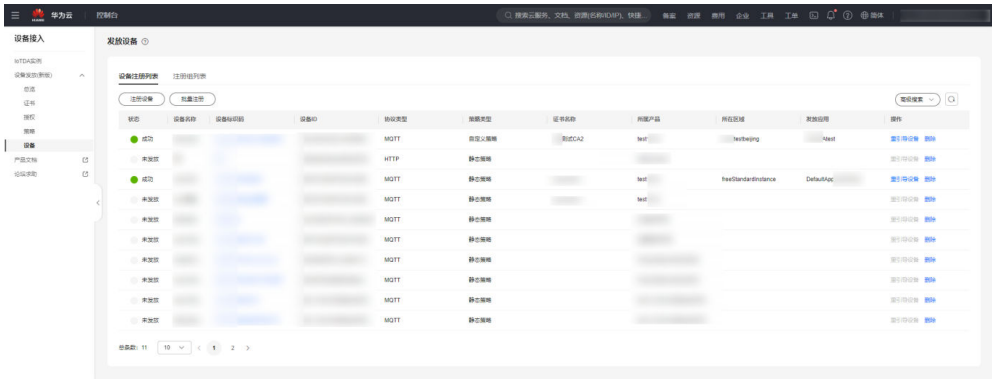


图 1-4 注册设备产品列表

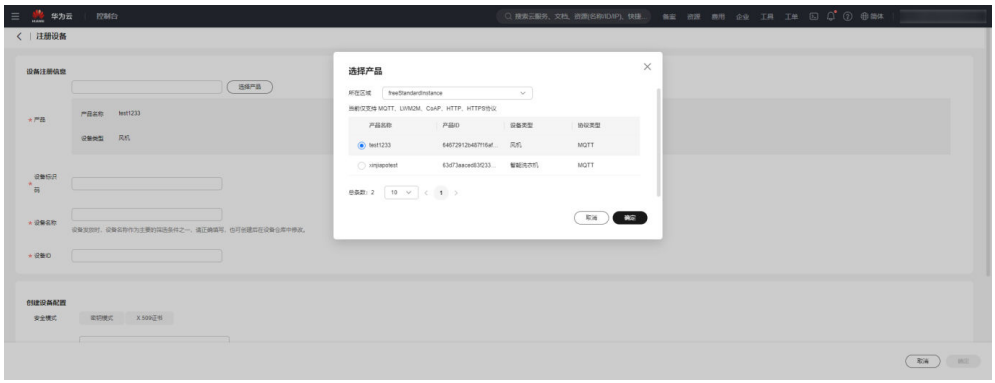
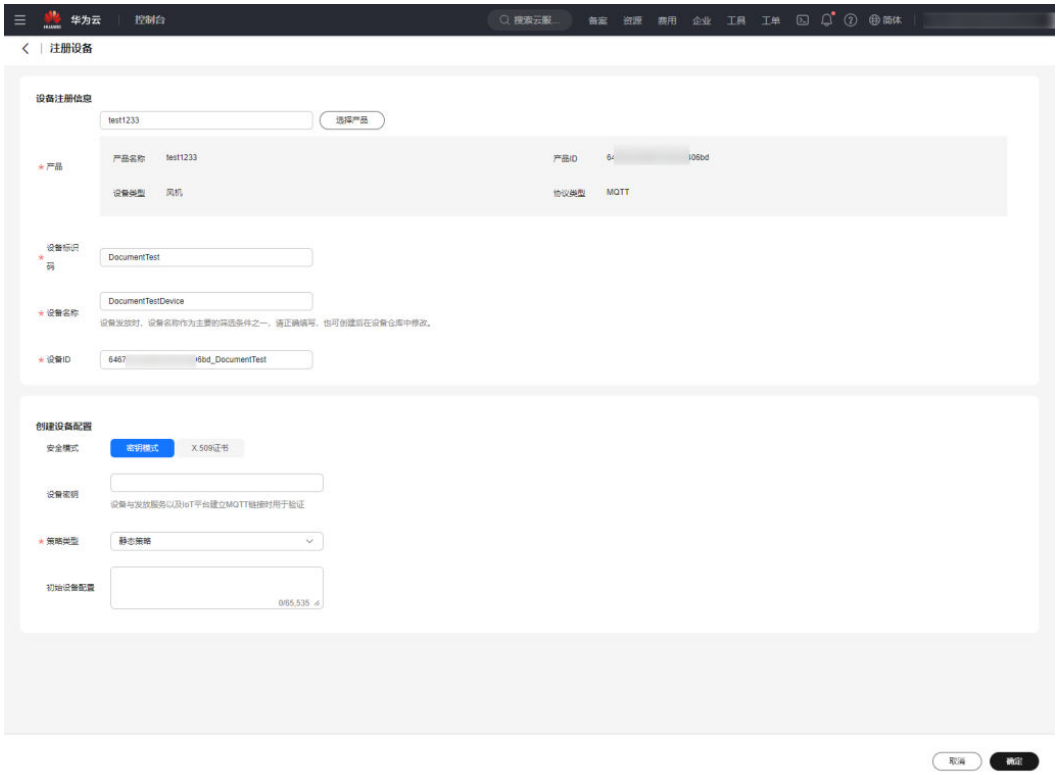


图 1-5 创建密钥模式静态策略设备



设备引导请求

终端节点

区域名称	区域	终端节点（Endpoint）	端口	协议
华北-北京四	cn-north-4	iot-bs.cn-north-4.myhuaweicloud.com	8883	MQTTS

下载并修改华为SDK示例代码进行设备引导（这里以**java sdk代码**为示例）。

说明

用IDEA/Eclipse打开SDK代码工程，修改iot-device-demo目录下的DEMO示例“BootstrapSample”中的参数，其中deviceId和secret替换为**注册设备**中生成的设备ID和密钥即可，bootstrapUri为上述**终端节点**。

```
package com.huaweicloud.sdk.iot.device.demo;

import com.huaweicloud.sdk.iot.device.bootstrap.BootstrapClient;


/**
 * 演示设备启动时，通过引导服务获取真实的服务器地址
 */
public class BootstrapSample {

    public static void main(String[] args) {

        String deviceId = "60136b0682401c03e0b1ccf5_aaa_device2";
        String secret = "73ce5656f5d4a9919532";
        String bootstrapUri = "ssl://100.95.158.64:8883";

        //创建引导客户端，发起引导
        BootstrapClient bootstrapClient = new BootstrapClient(bootstrapUri, deviceId, secret);
        DefaultBootstrapActionListener defaultBootstrapActionListener = new DefaultBootstrapActionListener(deviceId,
            secret, bootstrapClient);
        bootstrapClient.bootstrap(defaultBootstrapActionListener);

    }
}
```

运行DEMO程序，看到如下日志，代表设备发放成功，并且已经收到设备发放下发的设备接入地址。如果程序运行没报错，在对应的设备接入平台可以看到设备，并已在线。

```
2021-01-29 14:46:47  INFO BootstrapClient:50 - create BootstrapClient: 60136b0682401c03e0b1ccf5_aaa_device2
2021-01-29 14:46:48  INFO MqttConnection:167 - try to connect to ssl://100.95.158.64:8883
2021-01-29 14:46:48  INFO MqttConnection:200 - connect success ssl://100.95.158.64:8883
2021-01-29 14:46:48  INFO MqttConnection:105 - Mqtt client connected. address :ssl://100.95.158.64:8883
2021-01-29 14:46:48  INFO MqttConnection:240 - publish message topic =  $oc/devices/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/up, msg =
2021-01-29 14:46:48  INFO DefaultBootstrapActionListener:30 - bootstrap success:$oc/devices/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/down
2021-01-29 14:46:49  INFO DefaultBootstrapActionListener:30 - bootstrap success:null
2021-01-29 14:46:49  INFO MqttConnection:85 - messageArrived topic =  $oc/devices/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/down, msg = {"address":"100.95.174.182:1883","initConfig":null}
2021-01-29 14:46:49  INFO BootstrapClient:102 - bootstrap ok address:100.95.174.182:1883
2021-01-29 14:46:49  INFO DefaultBootstrapActionListener:30 - bootstrap success:100.95.174.182:1883
```

收到设备发放下发的设备接入地址后，需要关闭设备侧的设备发放的连接，用的新的URL地址与设备接入通信，进行相关业务。

```
public class DefaultBootstrapActionListener implements ActionListener {

    private static final Logger log = LogManager.getLogger(DefaultBootstrapActionListener.class);

    private String deviceId;

    private String secret;

    private BootstrapClient bootstrapClient;

    public DefaultBootstrapActionListener(String deviceId, String secret, BootstrapClient bootstrapClient) {
        this.deviceId = deviceId;
        this.secret = secret;
        this.bootstrapClient = bootstrapClient;
    }

    @Override
    public void onSuccess(Object context) {
        String address = (String) context;
        log.info( s: "bootstrap success:" + address);

        //引导成功后关闭客户端
        bootstrapClient.close();
        IoTDevice device = new IoTDevice( serverUri: "ssl://" + address, deviceId, secret);
        if (device.init() != 0) {
            return;

        }

        //上报消息
        device.getClient().reportDeviceMessage(new DeviceMessage("hello"),  listener: null);
    }

    @Override
    public void onFailure(Object context, Throwable var2) {
        log.error( s: "bootstrap WFH failed: {}", var2.getMessage());
    }
}
```

新增 IoT 接入实例

用户新增IoT接入实例，创建对应实例的资源空间，在对应资源空间下上传产品。

新增策略

当新的实例扩容后，添加新实例的静态策略，通不过不同的关键字命名关联新的实例。

图 1-6 创建静态策略详情

然后按照上面的操作以新策略的名称注册设备，请求引导即可。

应用主动与设备通信

多实例场景下，当应用侧需要主动与设备侧通信时，比如主动下发命令，需要应用侧知道设备是在哪个实例上，应用要规划好每批命令设备发放哪个实例，在设备发放通

过查询静态策略，找到对应设备名称的设备的实例ID，调用设备接入北向API下发命令需要指定实例ID。

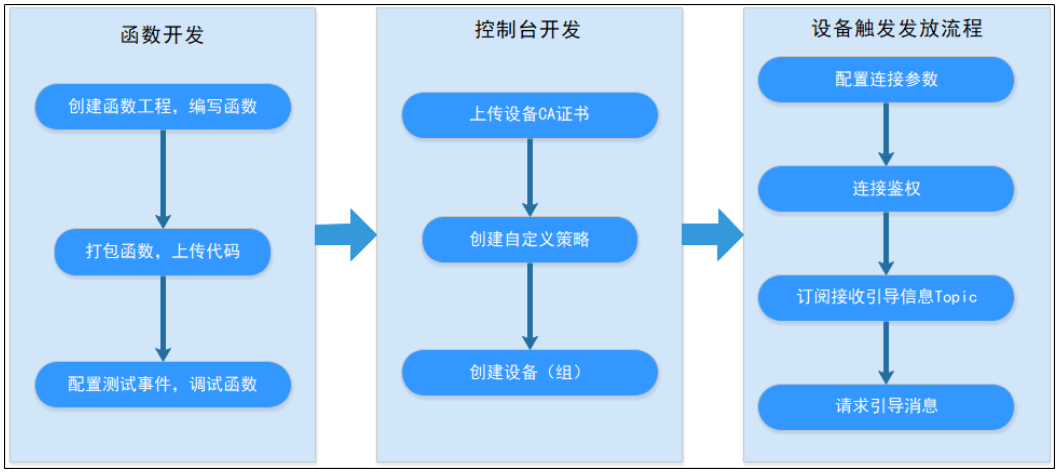
2 结合函数服务通过自定义策略发放证书认证的设备

场景说明

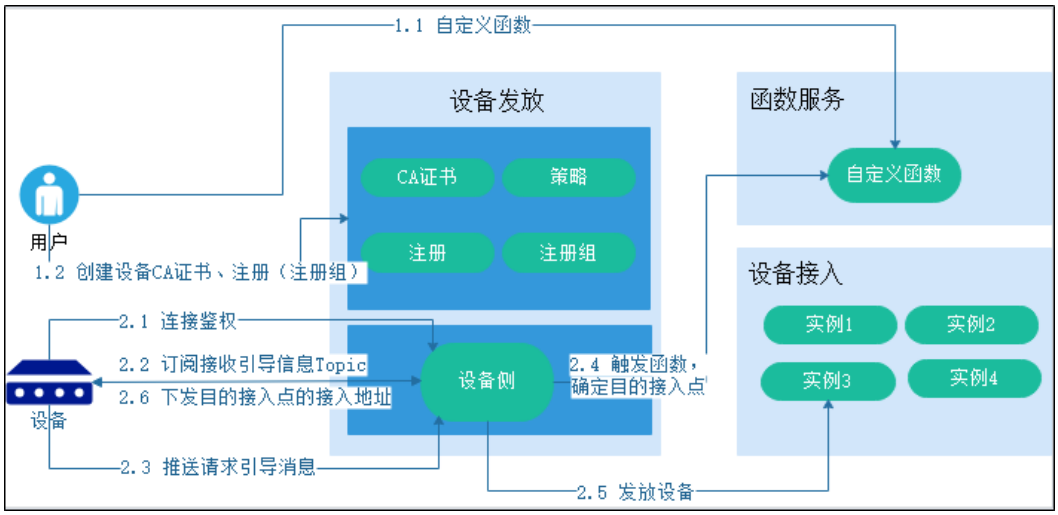
本文以MQTT.fx的设备模拟器替代真实的设备，结合**函数工作流服务（FunctionGraph）**，带您快速体验结合函数服务使用设备发放服务，通过自定义策略将设备发放到指定的物联网平台（设备接入实例）上。

整体流程

使用证书认证的设备采用自定义策略发放设备操作流程如下图所示。



通过自定义策略发放设备功能流程图如下图所示。

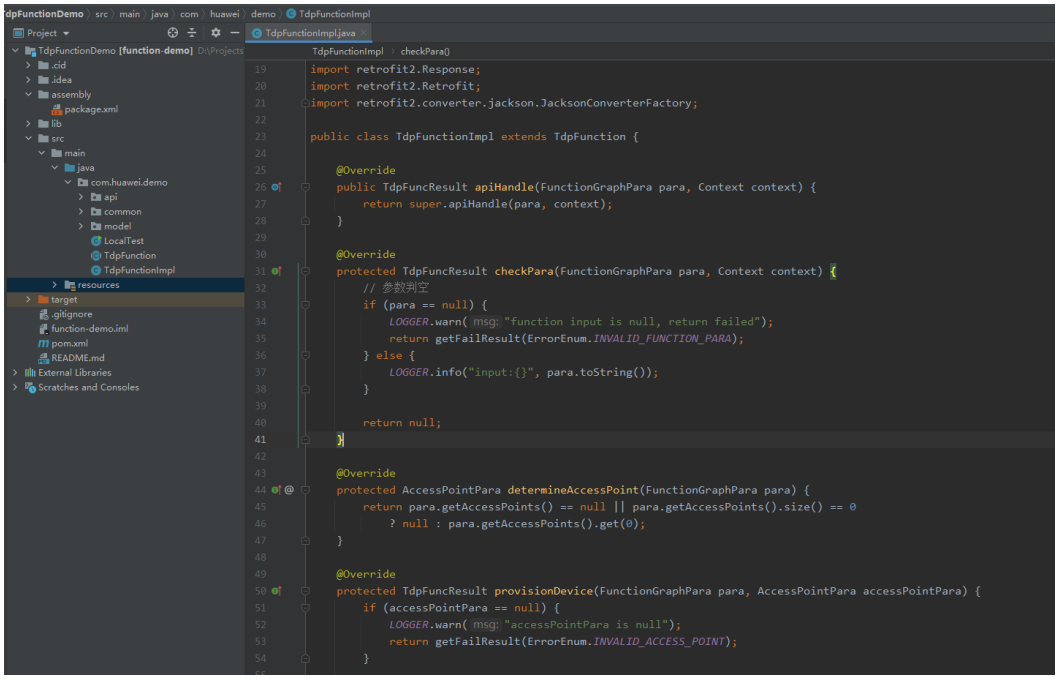


函数开发

自定义策略的函数代码编写指南参见**创建自定义策略函数**。

此处以Java语言为例，如何编写Java语言的函数，参见**函数工作流 FunctionGraph > 开发指南> 如何开发函数> Java函数开发指南**。

步骤1 创建函数工程，编写函数。



关键流程代码参考如下：

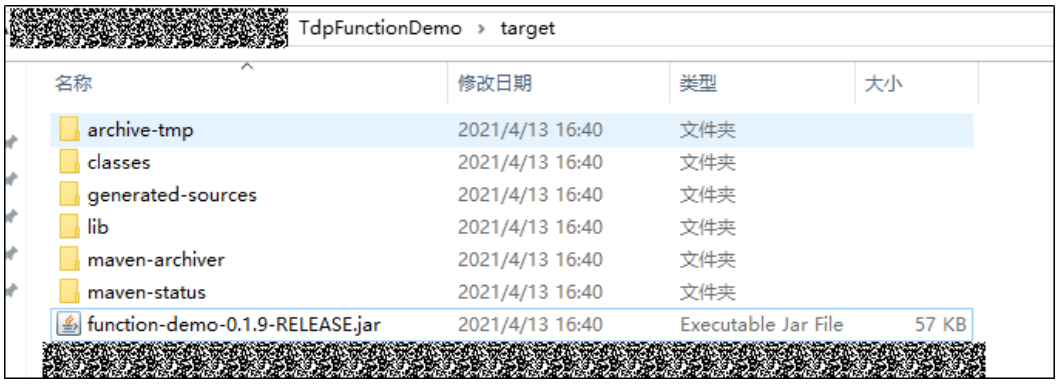
```
package com.huawei.demo;

import com.huawei.demo.common.logger.DefaultLogger;
import com.huawei.demo.model.AccessPointPara;
import com.huawei.demo.model.FunctionGraphPara;
import com.huawei.demo.model.TdpFuncResult;
import com.huawei.services.runtime.Context;
```



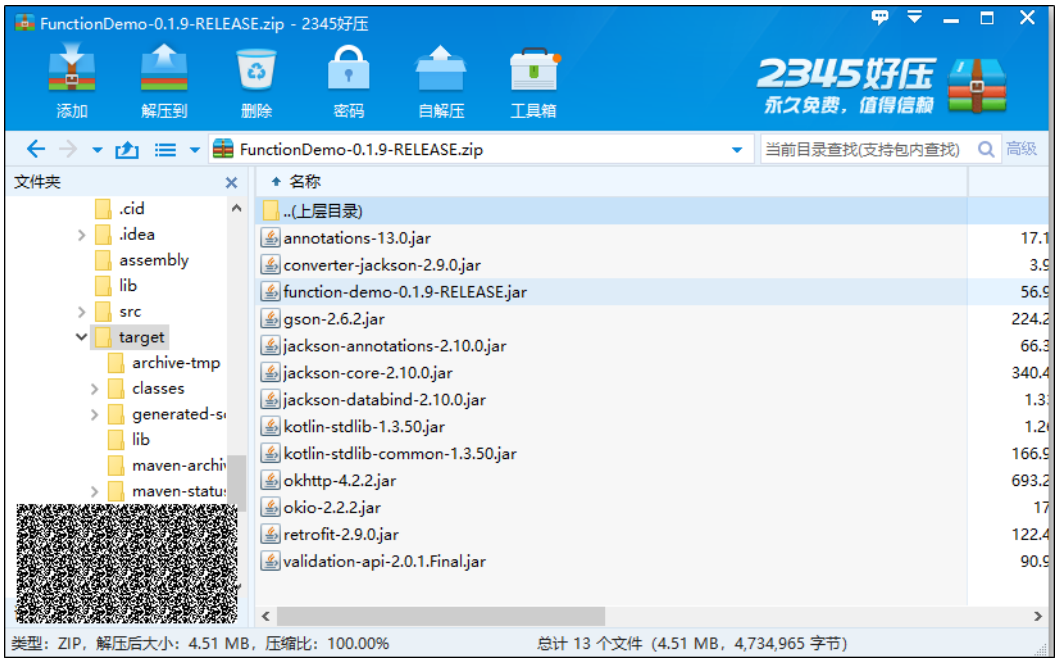
```
import java.util.Optional;

/**
 * 实现该抽象类。
 * WARNING：{@link #apiHandle(FunctionGraphPara, Context)} 该方法必须在子类中定义，否则函数无法触发该函数接口！！！
 * <pre>
 * {@code
 *     @Override
 *     public TdpFuncResult apiHandle(FunctionGraphPara para, Context context) {
 *         return super.apiHandle(para, context);
 *     }
 * }
 * </pre>
 */
public abstract class TdpFunction {
    protected static final DefaultLogger LOGGER = new DefaultLogger(TdpFunction.class);

    /**
     * 函数定义
     */
    public TdpFuncResult apiHandle(FunctionGraphPara para, Context context) {
        // 获取日志
        DefaultLogger.init(Optional.ofNullable(context)
            .map(Context::getLogger)
            .orElse(null));

        // 校验入参
        TdpFuncResult result = checkPara(para, context);
        if (result != null) {
            return result;
        }

        // 确定接入点
        AccessPointPara accessPointPara = determineAccessPoint(para);

        // 发放设备
        result = provisionDevice(para, accessPointPara);

        LOGGER.info("result:{}", result);
        return result;
    }

    /**
     * 校验入参
     */
    protected abstract TdpFuncResult checkPara(FunctionGraphPara para, Context context);

    /**
     * 从备选的接入点中选择合适的接入点
     */
    protected abstract AccessPointPara determineAccessPoint(FunctionGraphPara para);

    /**
     * 调用设备发放的发放接口发放设备
     */
    protected abstract TdpFuncResult provisionDevice(FunctionGraphPara para, AccessPointPara accessPointPara);
}
```

步骤2 打包函数，上传代码。

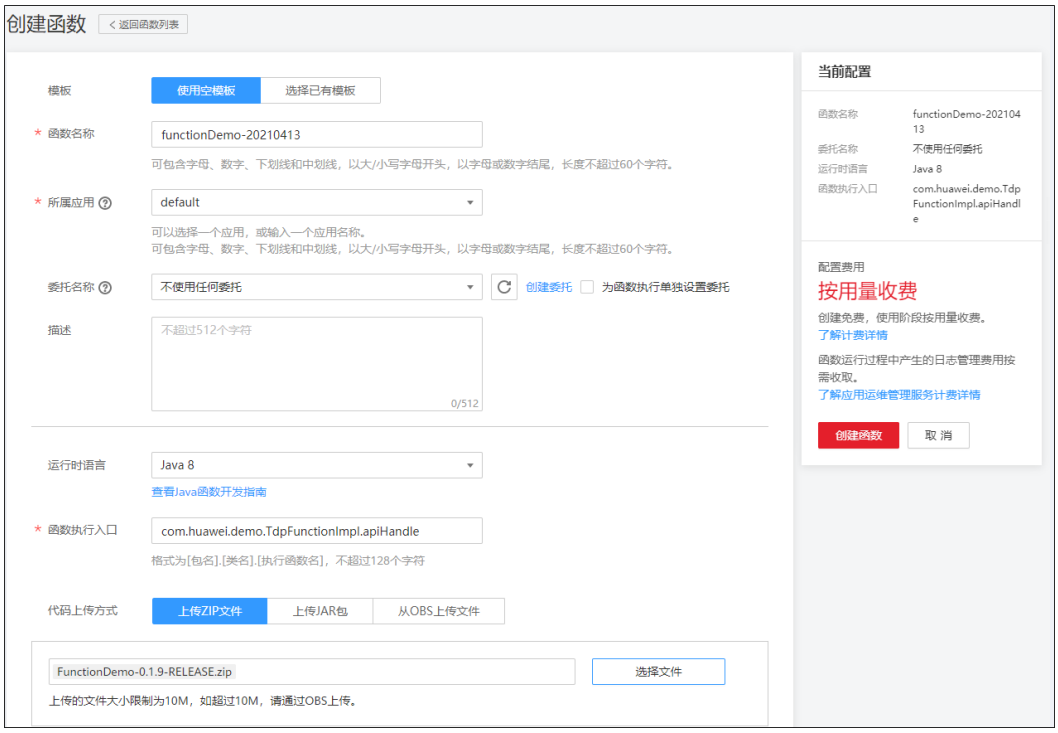
如不依赖第三方Jar包，可将工程打包成一个单独的Jar包，如下图所示。



如工程依赖第三方Jar包，则需将工程Jar包和所有第三方Jar包打包成一个zip包，如下图所示。

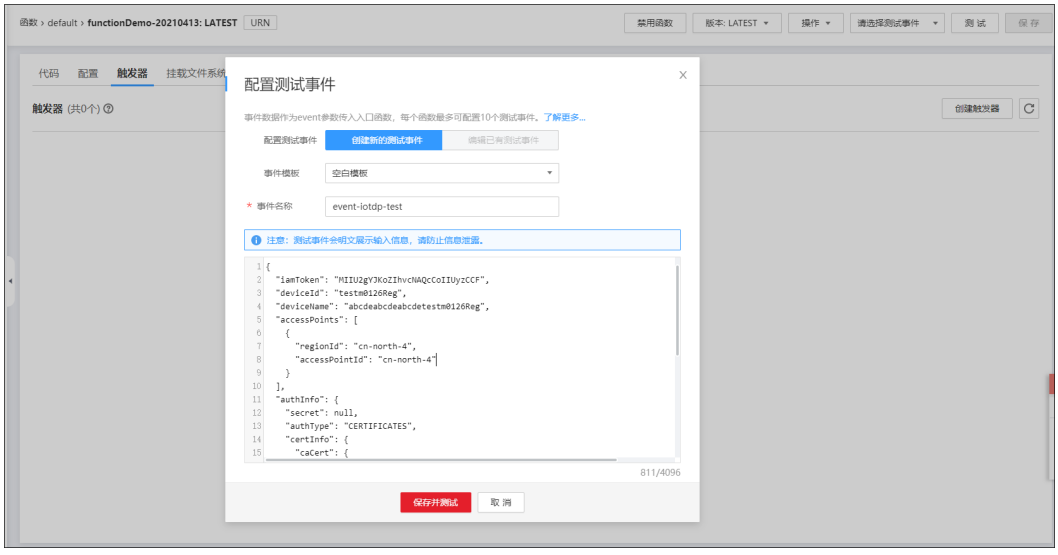


进入函数工作流服务控制台，创建函数。函数执行入口以[包名].[类名].[执行函数名]格式填写。



步骤3 配置测试事件，调试函数。

创建并配置测试事件，调试函数，如下图所示。



----结束

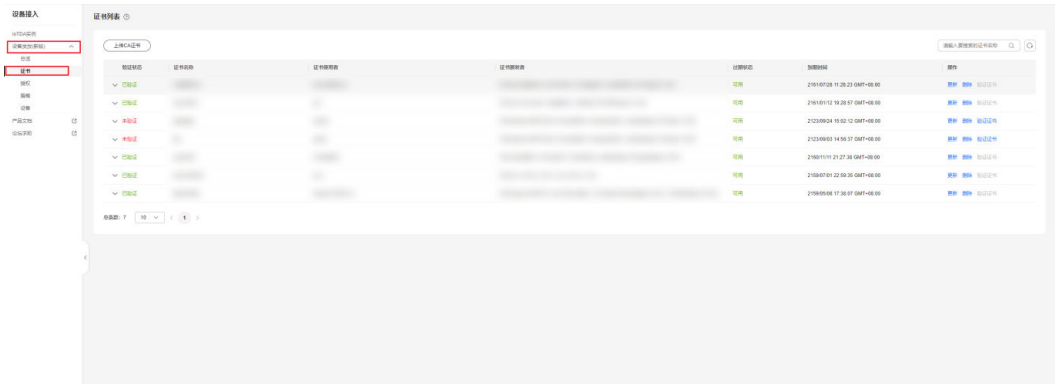
控制台开发

登录**设备发放控制台**。

步骤1 进入“**证书**”界面，上传并验证设备CA证书。

如何上传并验证证书参见**证书** 。

图 2-1 证书列表



步骤2 进入“**策略**”界面，添加自定义策略。

如何添加自定义策略实例参见**添加自定义策略实例**。

图 2-2 添加自定义策略

步骤3 进入“**设备**”界面，单击左上角“注册设备”，进行注册设备。

如何注册设备，参见**设备** 。

图 2-3 注册设备

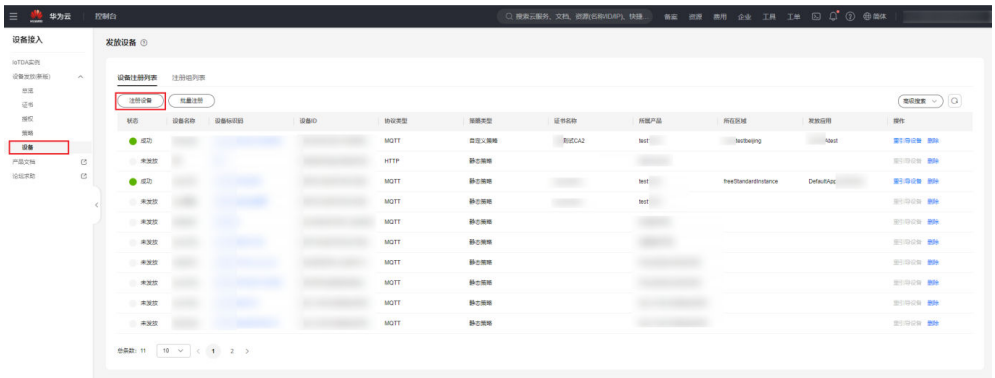
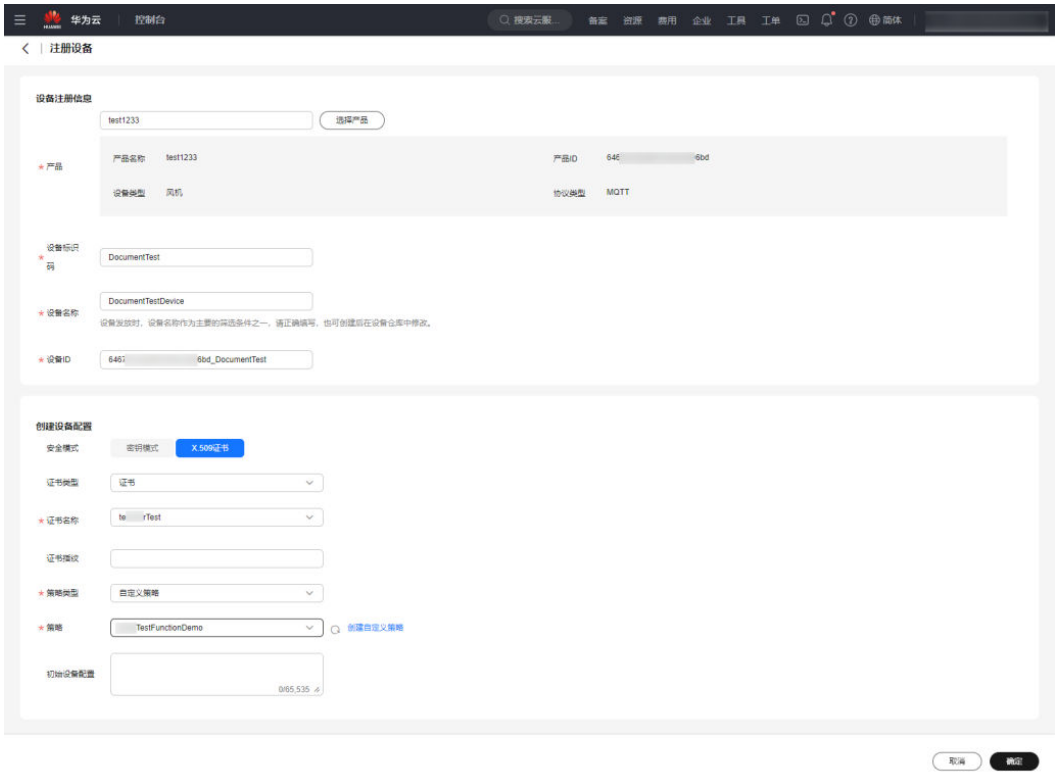


图 2-4 创建证书模式自定义策略设备



----结束

设备触发放流程

步骤1 配置连接参数。

证书认证设备如何使用MQTT.fx接入设备发放请参见**MQTT X509证书认证接入示例**。

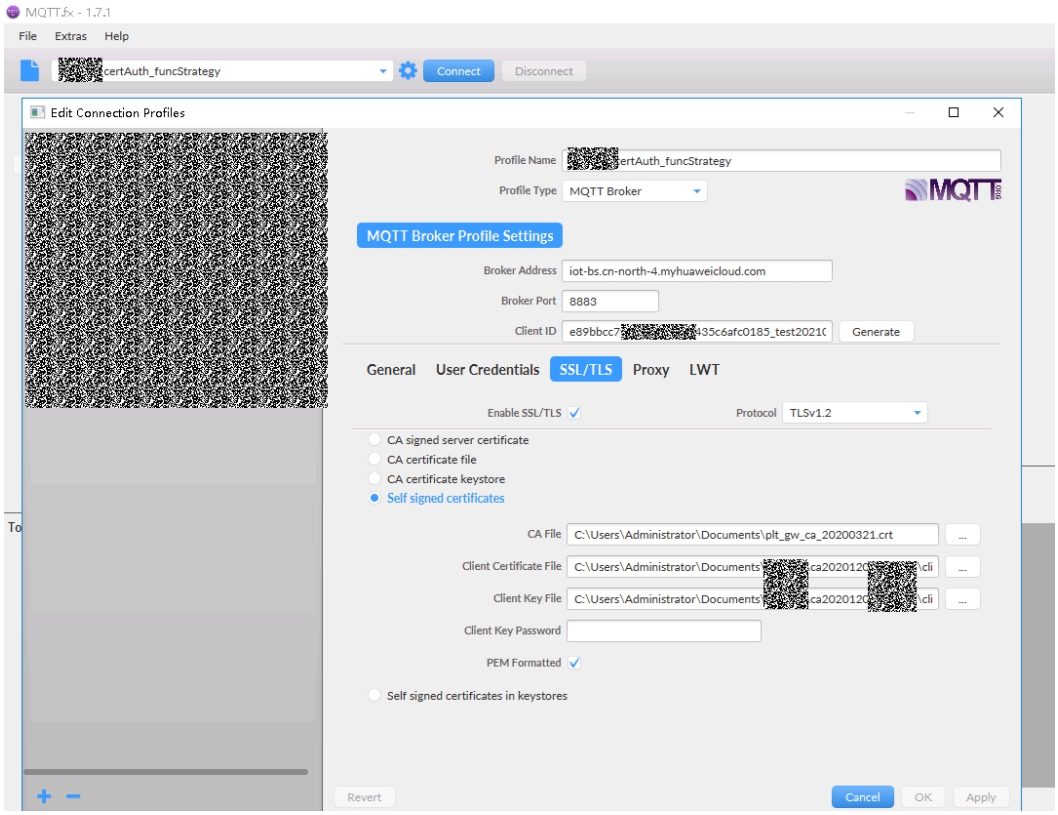
MQTT Broker Profile Settings

Broker Address填写设备发放设备侧接入地址，Broker Port填写设备发放设备侧接入端口，Client ID按规则（[设备ID]_0_0_[10位日期时间]）填写，具体规则参见**MQTT CONNECT连接鉴权**。

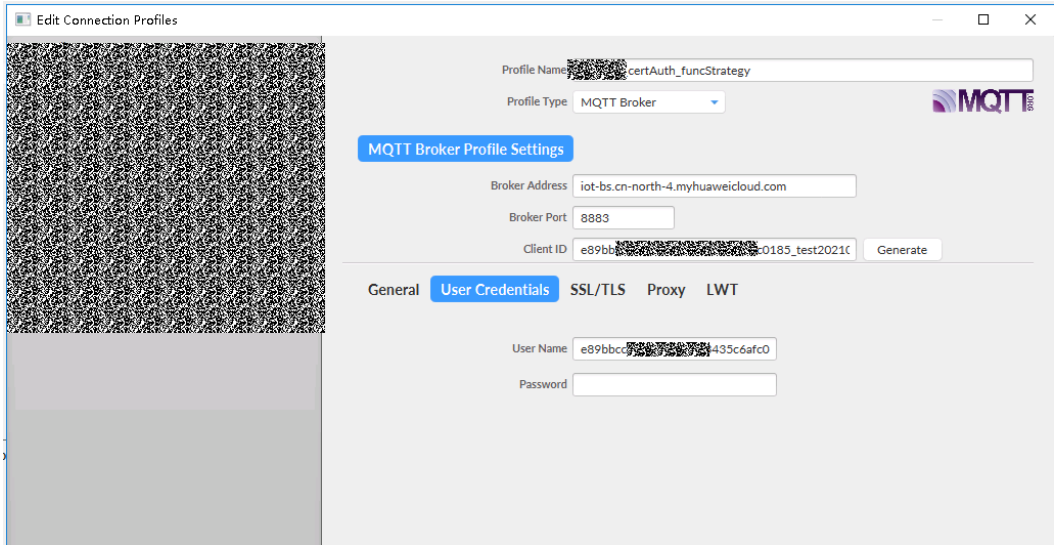
SSL/TLS

由于设备使用证书认证方式，“Enable SSL/TLS”勾选开启，选择Self signed certificates，CA File填写用于签发设备发放设备侧接入SSL/TLS通道证书的CA证书路径，Client Certificate File和Client Key File填写使用已上传并验证了的CA证书签发的设备证书和设备证书私钥路径。如证书为PEM格式，请勾选“PEM Formatted”。

具体配置请参见下图：

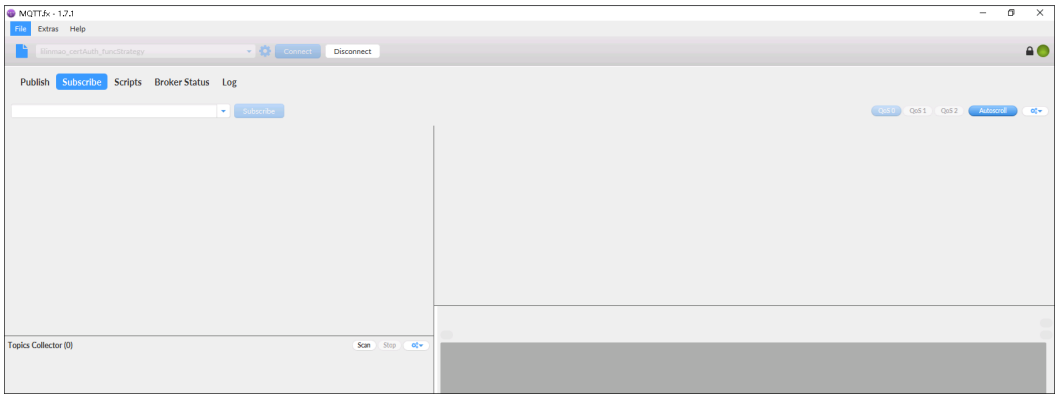


“User Credentials”的“User Name”字段填写设备ID。具体配置参见下图：



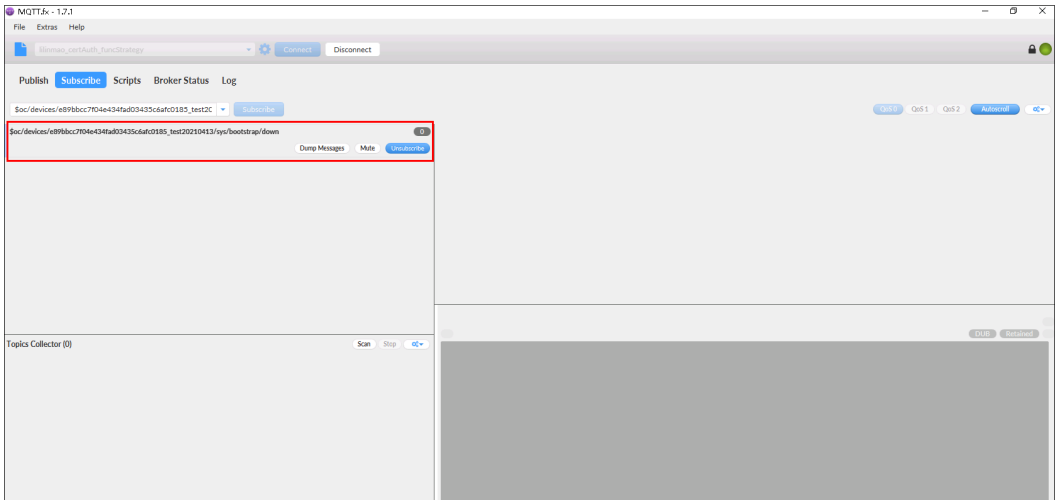
步骤2 连接鉴权。

连接参数配置完成后，单击“Connnect”。若右上角圆圈呈现绿色，说明设备已成功上设备发放平台。



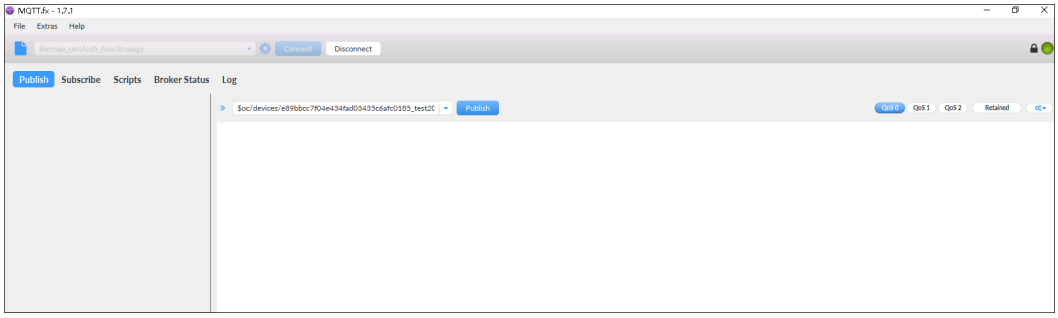
步骤3 订阅接收引导消息Topic。

参照**设备接收引导信息**填写用于接收引导地址的Topic，单击“Subscribe”订阅该Topic。若订阅Topic填写框下方的已订阅Topic列表中存在该Topic，则订阅成功。

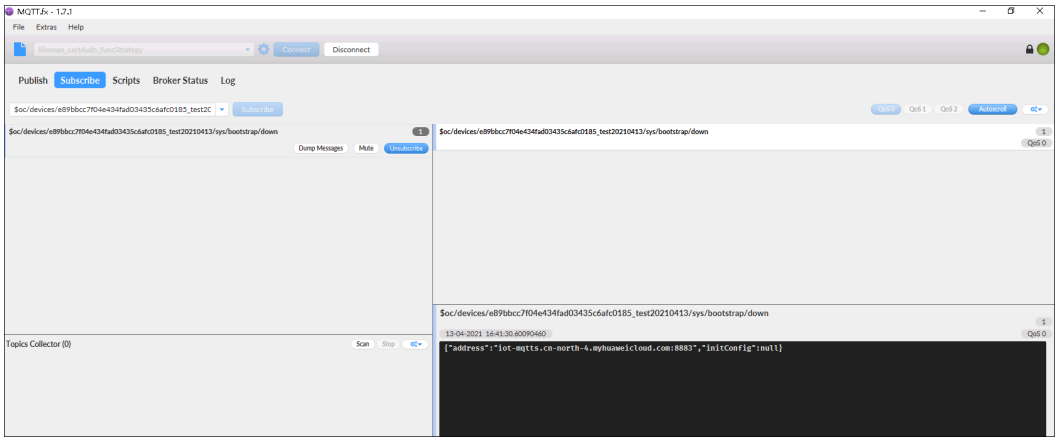


步骤4 发送请求引导消息。

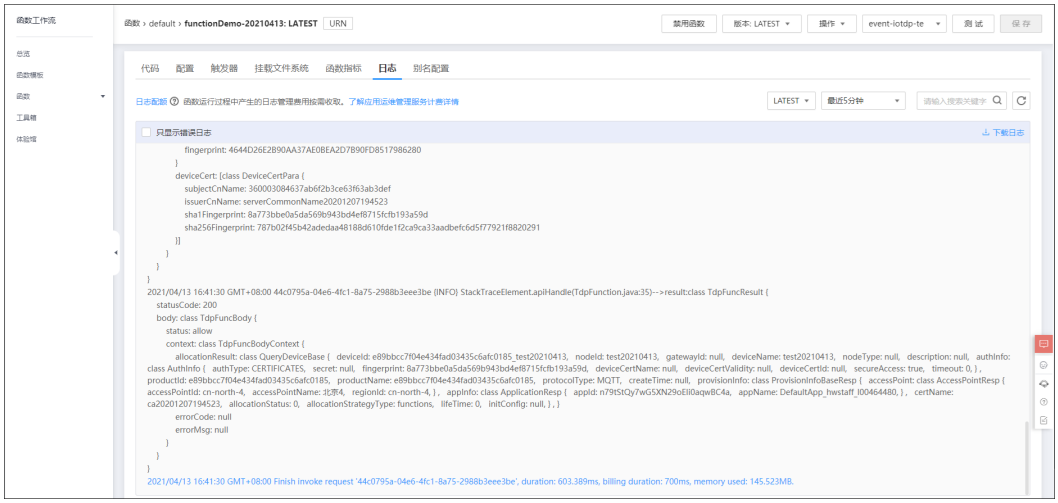
参照**设备请求引导消息**填写发送请求引导消息的Topic，单击“Publish”向该Topic推送消息。



查看订阅的Topic，很快在订阅Topic下，接收到了目的接入点的设备侧接入地址。

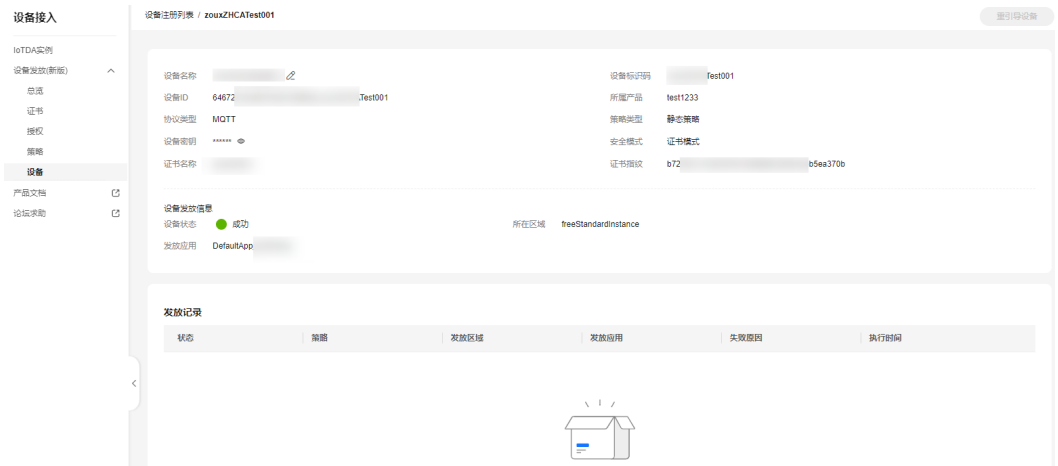


进入函数服务控制台，通过查看使用的函数的日志，可看出自定义策略发放设备过程中，该函数被成功触发且正确执行。



进入到设备发放控制台，可查看到一条该设备的发放记录。

图 2-5 设备详情页



至此，您已成功完成了，使用设备发放和函数服务将证书认证的设备通过自定义策略触发函数将设备发放到指定的设备接入实例中。

----结束

3 设备自动注册安全接入示例

场景说明

平台基于客户业务安全考虑，只有将设备的基本信息（例如设备ID、鉴权信息）注册到平台后，设备才能使用成功注册后的设备ID和鉴权信息接入物联网平台。当用户注册的设备不断增多时，如何安全且高效地注册设备变得尤为关键。

一种方案是，使用批量注册模板或者循环调用平台应用侧注册设备的API，实现高效地批量注册设备。此方案能够满足客户对批量注册设备的大部分场景需求，但要求客户业务应用侧和客户产线侧相互配合，客户应用侧对设备ID提前进行规划，应用侧完成批量注册后，客户业务产线侧需将应用侧获取到的设备ID逐一烧录到各个设备比较麻烦。

另一种更加高效的方案是，使用设备发放的注册组功能实现设备免注册安全极简地接入物联网平台。

本文基于设备发放注册组功能，使用证书策略，带您快速体验设备免注册安全极简接入物联网平台。

表 3-1 方案对比

设备注册方案	基于单个注册	批量注册	免注册安全极简接入
适用场景	● 设备数量不多； ● 对设备身份标识和认证信息可提前规划； ● 产线侧与应用侧可紧密配合。	● 设备数量多； ● 对设备身份标识和认证信息可提前规划。	● 设备数量较多； ● 设备能力强； ● 对设备注册的灵活性要求高。

设备注册方案	基于单个注册	批量注册	免注册安全极简接入
过程	1. 应用侧通过多次调用单个注册API将设备信息（包括但不限于身份标识和认证信息）逐一注册到平台； 2. 产线侧与应用侧密切配合，将设备身份标识和与之对应的认证信息逐个写入设备。	1. 下载并填写批量注册模板文件，定义设备信息（包括但不限于身份标识和认证信息）； 2. 将填写完成的批量注册模板文件上传至平台，完成批量注册过程； 3. 产线侧将设备身份标识和与之对应的认证信息逐个写入设备。	1. 基于设备发放的注册组和发放策略能力； 2. 具体过程在本文后续章节展开。
特点	• 多次调用API； • 整体流程简单。	将单个注册过程模板化。	• 无需对设备身份标识和认证信息提前规划； • 应用侧无需逐一注册设备信息； • 降低产线管理和维护成本。

整体流程

通过注册组实现设备免注册极简接入平台操作流程图如下图所示。

图 3-1 注册流程

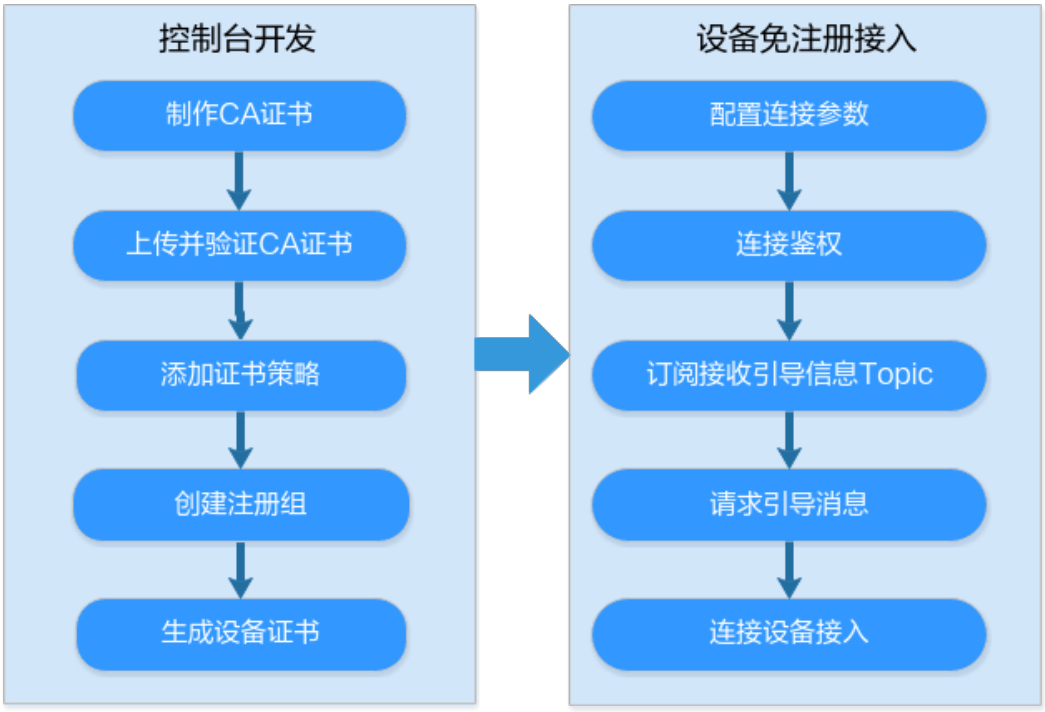
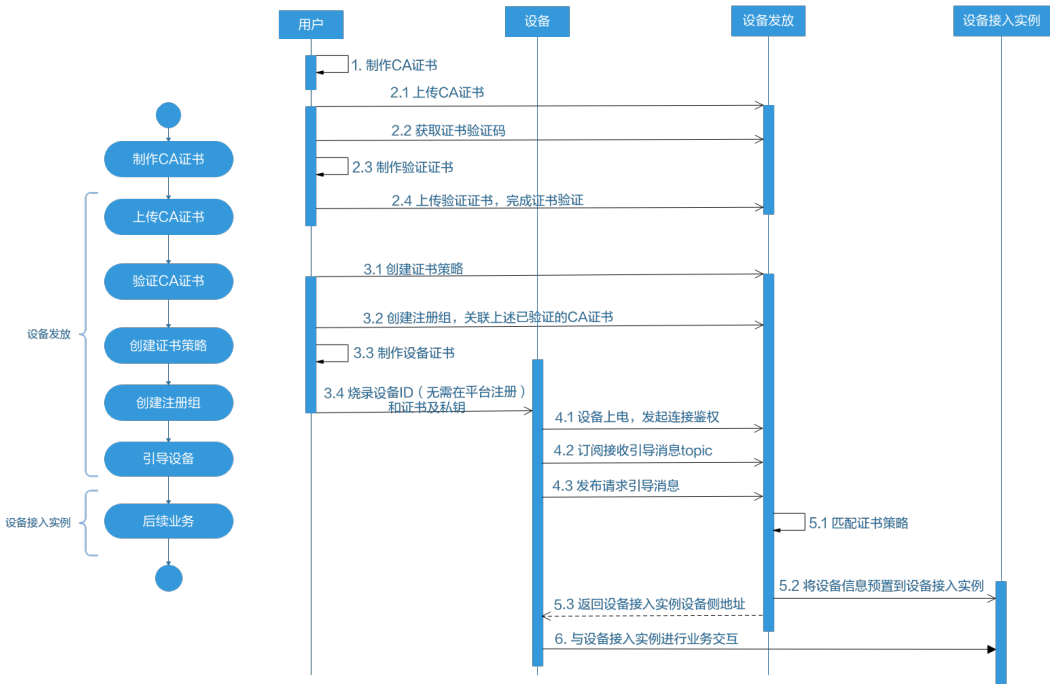


图 3-2 接入平台操作流程



制作 CA 证书

本文以Windows环境为例，介绍通过Openssl工具制作CA证书和验证证书的方法。

注意

以下“生成密钥对（rootCA.key）”和“生成CA证书（rootCA.crt）”为操作过程中需要使用到的两个文件。

具体步骤：

步骤1 在浏览器中访问这里，下载并进行安装OpenSSL工具，安装完成后配置环境变量。

步骤2 在 D:\certificates 文件夹下，以管理员身份运行cmd命令行窗口。

步骤3 生成密钥对（rootCA.key）：

说明

生成“密钥对”时输入的密码在生成“证书签名请求文件”、“CA证书”，“验证证书”以及“设备证书”时需要用到，请妥善保存。

```
openssl genrsa -des3 -out rootCA.key 2048
```

步骤4 使用密钥对生成证书签名请求文件：

说明

生成证书签名请求文件时，要求填写证书唯一标识名称（Distinguished Name，DN）信息，参数说明如下**表1** 所示。

表 3-2 参数说明

提示	参数名称	取值样例
Country Name (2 letter code) []:	国家/地区	CN
State or Province Name (full name) []:	省/市	GuangDong
Locality Name (eg, city) []:	城市	ShenZhen
Organization Name (eg, company) []:	组织机构（或公司名）	Huawei Technologies Co., Ltd.
Organizational Unit Name (eg, section) []:	机构部门	Cloud Dept.
Common Name (eg, fully qualified host name) []:	CA名称（CN）	Huawei IoTDP CA
Email Address []:	邮箱地址	/
A challenge password []:	证书密码，如您不设置密码，可以直接回车	/
An optional company name []:	可选公司名称，如您不设置，可以直接回车	/

```
openssl req -new -key rootCA.key -out rootCA.csr
```

步骤5 生成CA证书（rootCA.crt）：

```
openssl x509 -req -days 50000 -in rootCA.csr -signkey rootCA.key -out rootCA.crt
```

说明

“-days”后的参数值指定了该证书的有效天数，此处示例为50000天，您可根据实际业务场景和需要进行调整。

----结束

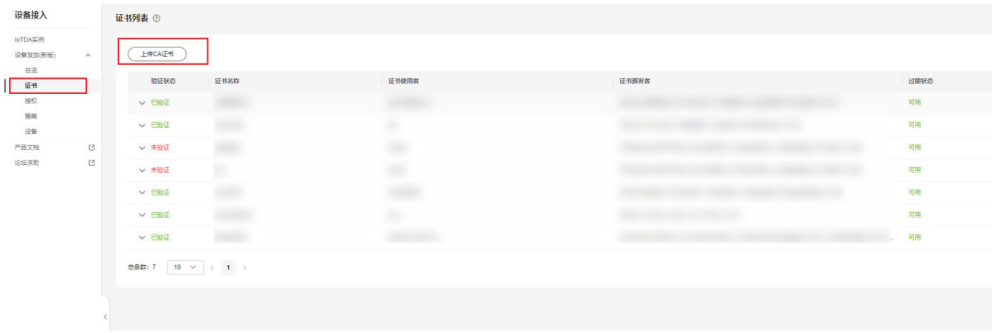
上传 CA 证书

具体操作步骤

步骤1 登录**设备发放控制台**。

步骤2 在设备发放控制台，左侧导航窗格中，选择“证书”，单击右上方的“上传CA证书”。

图 3-3 上传 CA 证书



步骤3 在“上传CA证书”页面，填写“证书名称”，单击“添加文件”，上传此前“**制作CA证书**”步骤中生成的“CA证书（rootCA.crt文件）”，单击“确定”。

图 3-4 上传 CA 证书详情页

----结束

说明

上传的CA证书初始状态为“未验证”，需要完成“**验证CA证书**”过程，方可正常使用该CA证书。

表 3-3 证书状态表

CA证书状态	说明
已验证	可正常使用。
未验证	不可正常使用，待验证通过后，方可正常使用。
已过期	CA证书已过期，需更新，但不影响平台使用该CA证书验证对应的设备证书。
即将过期	CA证书30天内即将过期，需及时更新。

验证 CA 证书

对于已上传的CA证书，平台要求用户完成“验证CA证书”过程，以验证用户具备该CA证书的签发能力。

操作步骤

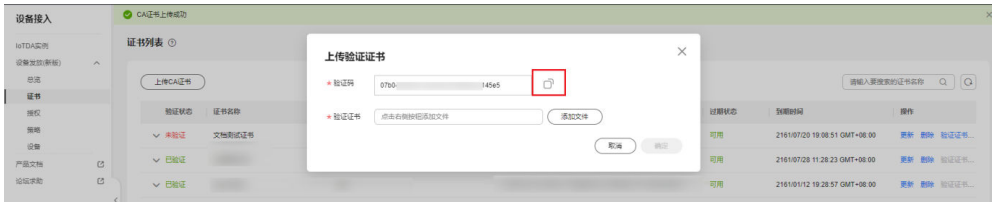
步骤1 登录**设备发放控制台**。

步骤2 在设备发放控制台，左侧导航窗格中，选择“证书”，单击“证书列表”条目的操作栏中的“验证证书”。

图 3-5 上传 CA 证书完成页

步骤3 在上传验证证书页面，单击“生成验证码”，单击“复制图标”复制此CA证书的随机验证码。

图 3-6 复制验证码



说明

CA证书验证码有效期为一天，请及时使用验证码生成验证证书并完成验证。

验证码的生成为替换机制，即对于一个CA证书，即使此前的验证码未过期，也将被新生成的验证码替换。

步骤4 使用OpenSSL工具为验证证书生成密钥对。

```
openssl genrsa -out verificationCert.key 2048
```

步骤5 利用此验证码生成证书签名请求文件CSR。

```
openssl req -new -key verificationCert.key -out verificationCert.csr
```

说明

CSR文件的Common Name (e.g. server FQDN or YOUR name) 需要填写此验证码。

步骤6 使用CA证书、CA证书私钥和上一步骤中生成的CSR文件创建验证证书（verificationCert.crt）。

```
openssl x509 -req -in verificationCert.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out verificationCert.crt -days 36500 -sha256
```

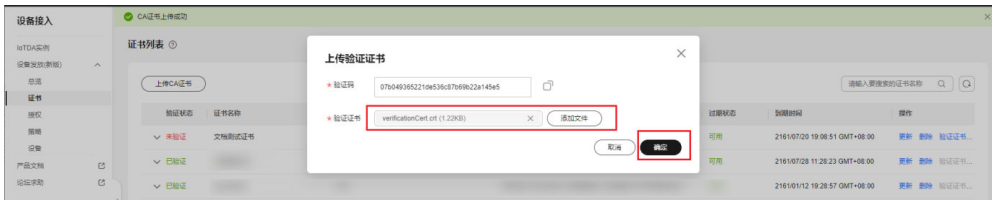

说明

生成验证证书用到的“rootCA.crt”和“rootCA.key”这两个文件，为“**制作CA证书**”中所生成的两个文件。

“-days”后的参数值指定了该证书的有效天数，此处示例为36500天，您可根据实际业务场景和需要进行调整。

步骤7 上传验证证书进行验证。

图 3-7 上传验证证书



----结束

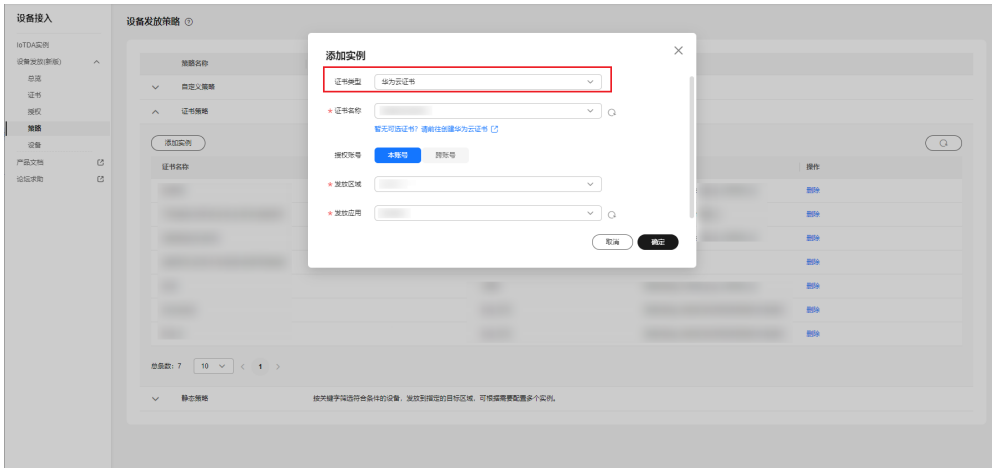
添加证书策略

步骤1 添加证书策略，发放CA证书到指定的IoTDA，并且由此CA签发的设备证书都会发放到指定的IoTDA。

图 3-8 添加证书策略

步骤2 进入“策略”界面，单击展开“证书策略”，单击“添加实例”。

图 3-9 添加证书策略详情



步骤3 按照下方参数说明填写关键参数信息后，单击“确定”。

表 3-4 参数信息

参数名称	说明	示例
证书名称	即所要根据证书属性将设备发放到指定的目标区域，选择对应的证书。	将需要通过证书“certificates”发放的设备发放至华北-北京四的物联网平台。 • 需通过证书“certificates”发放的设备：WaterMeter-Beijing0001、WaterMeter-Beijing0002 • 证书名称：certificates • 发放区域：华北-北京四 • 发放应用：beijing-app1
发放区域	发放到指定区域后，设备将接入对应区域的设备接入服务。 所选区域未开通设备接入服务时，如果确定添加实例，系统将自动为您开通设备接入服务。不同区域设备接入服务价格不同，收费详情请参考价格说明。	
发放应用	选择对应设备接入服务区域已创建的应用。在物联网平台中，设备由应用统一管理。 如果对应设备接入服务区域未创建应用，需要前往对应服务创建应用。	

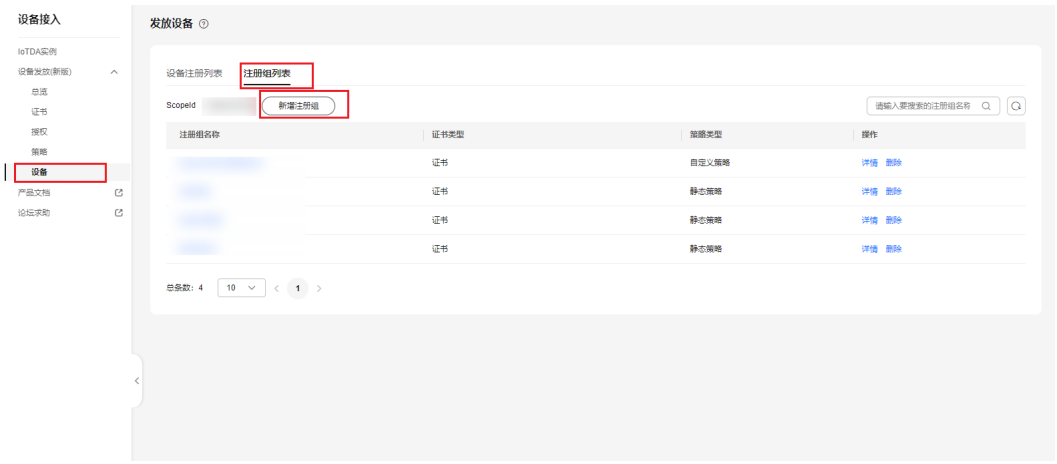
----结束

创建注册组

MQTT证书接入的设备，可以在设备发放创建一个注册组，绑定对应的CA证书和自定义策略，可以实现批量设备的自注册，实现设备一键上电即可上云的动作，可在注册组详情中查看该注册组下所有的设备。

步骤1 进入“设备-注册组”界面，单击右上角“新增注册组”。

图 3-10 新增注册组



步骤2 按照下方参数说明填写关键参数信息后，完成创建。

表 3-5 参数信息

参数名称	说明
注册组名称	注册组的唯一标识。
选择证书	用于和注册组绑定，同一个证书只能同时绑定一个注册组，不能同时绑定多个注册组。
发放策略	当前只支持“自定义策略”，同时需要选择所要运行的函数。

----结束

生成设备证书

步骤1 使用OpenSSL工具为设备证书生成密钥对（设备私钥）：

```
openssl genrsa -out deviceCert.key 2048
```

步骤2 使用设备密钥对，生成证书签名请求文件：

```
openssl req -new -key deviceCert.key -out deviceCert.csr
```

说明

生成证书签名请求文件时，要求填写证书唯一标识名称（Distinguished Name，DN）信息，参数说明如下表2所示。

表 3-6 参数说明

提示	参数名称	取值样例
Country Name (2 letter code) []:	国家/地区	CN
State or Province Name (full name) []:	省/市	GuangDong
Locality Name (eg, city) []:	城市	ShenZhen
Organization Name (eg, company) []:	组织机构（或公司名）	Huawei Technologies Co., Ltd.
Organizational Unit Name (eg, section) []:	机构部门	Cloud Dept.
Common Name (eg, fully qualified host name) []:	CA名称（CN）	Huawei IoTDP CA
Email Address []:	邮箱地址	/
A challenge password []:	证书密码，如您不设置密码，可以直接回车	/

提示	参数名称	取值样例
An optional company name []:	可选公司名称，如您不设置，可以直接回车	/

步骤3 使用CA证书、CA证书私钥和CSR文件创建设备证书（deviceCert.crt）。

```
openssl x509 -req -in deviceCert.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out deviceCert.crt -days 36500 -sha256
```

说明

生成设备证书用到的“rootCA.crt”和“rootCA.key”这两个文件，为“制作CA证书”中所生成的两个文件，且需要完成“上传并验证CA证书”。

“-days”后的参数值指定了该证书的有效天数，此处示例为36500天，您可根据实际业务场景和需要进行调整。

----结束

设备免注册接入

步骤1 下载并修改华为SDK示例代码进行设备引导（这里以**java sdk代码**为示例）。

说明

使用IDEA/Eclipse打开SDK代码工程，修改iot-device-demo目录下的DEMO示例“BootstrapSelfRegSample”中的参数。

1. ScopeId从设备发放的注册组页面中获取；
2. deviceId由客户自主规划且未在平台注册（需注意平台要求deviceId全局唯一）；
3. 设备证书指定为“控制台开发”中生成的设备证书；
4. bootstrapUri为上述**终端节点**。

图 3-11 修改 demo 示例

```
/**
 * 演示自注册场景（证书方式），设备启动时，通过引导服务获取真实的服务器地址
 */
public class BootstrapSelfRegSample extends BaseBootstrapSample {
    /**
     * ScopeID，与租户相关，请从注册组页面获取
     */
    private static String scopeId = "[Please input your scope id here, example:f67f8df43c4a]";

    /**
     * 设备ID（自注册场景下，设备ID无需提前在设备发放上注册）
     */
    private static String deviceId = "[Please input your device id here, example:702b1038-a174-4a1d-969f-f67f8df43c4a]";

    /**
     * 设备证书信息
     */
    private static String DEVICE_CERT
        = "[Please input your device cert path here, example:D:\\SDK\\cert\\deviceCert.pem]";

    private static String DEVICE_CERT_KEY
        = "[Please input your device cert key path here, example:D:\\SDK\\cert\\deviceCert.key]";

    private static String DEVICE_CERT_KEY_PWD
        = "[Please input your device cert key pwd here, example:yourpwd. If not set, input empty string]";

    public static void main(String[] args) throws Exception {
        // 读取pem格式设备证书
        KeyStore keyStore = CertificateUtil.getKeyStore(DEVICE_CERT, DEVICE_CERT_KEY, DEVICE_CERT_KEY_PWD);

        // 创建引导客户端，发起引导
        BootstrapClient bootstrapClient = new BootstrapClient(BOOTSTRAP_URI, deviceId, keyStore, DEVICE_CERT_KEY_PWD, scopeId, PLATFORM_CA_PROVIDER);
        bootstrapClient.bootstrap(new SimpleBootstrapActionListener(bootstrapClient));
    }
}
```


步骤2 运行DEMO程序，看到如下日志，代表设备发放成功，并且已经收到设备发放下发的设备接入地址。如果程序运行正常，在对应的设备接入实例可以看到该设备，且该设备已在线。

图 3-12 日志信息

步骤3 收到设备发放下发的设备接入地址后，需要关闭设备侧的设备发放的连接，用的新的URL地址与设备接入通信，进行相关业务。

图 3-13 关闭连接示例代码

```
@Override
public void onSuccess(Object context) {
    // 引导成功，获取到iot平台的地址
    String address = (String) context;
    log.info( message: "bootstrap success, the address is {}", address);

    // 引导成功后关闭客户端
    bootstrapClient.close();

    // 与iot平台建立连接，上报消息
    IoTDevice device = bootstrapClient.getIoTDevice( serverUri: "ssl://" + address);
    if (device == null || device.init() != 0) {
        return;
    }
    device.getClient().reportDeviceMessage(new DeviceMessage("hello"), listener: null);
}
```

说明

说明：

1. DeviceId即设备ID，用于唯一标识一个设备。设备需使用未注册的DeviceId进行免注册接入，不同设备的DeviceId不同。
2. 在客户实际业务场景中，推荐一个设备使用一个设备证书，不建议多个设备共用一个设备证书。

----结束

