技术文档 / TECHNICAL DOCUMENT

OnBoard™ IoT Security 用户手册

为万物互联,筑安全之基

版本号	V3.2.2
编制日期	2026-02-09



版本：v3.2.2

OnBoard™ IoT Security 产品介绍

概述

OnBoard™ IoT Security (OBIS) 是一个基于授权的安全平台，旨在解决在全球范围内大规模部署安全设备时的核心挑战，同时保持 OEM 对密码信任的控制。保护产品数字资产的完整性、机密性及其授权使用。OBIS 以云平台的形式交付，提供灵活的部署选项，以满足不同的数据驻留和运行需求。OBIS 不是 PKI 服务，也不是工厂安全系统。它是一个安全控制平台（security control plane），用于管理产品数字资产在设备制造生命周期中的使用方式。

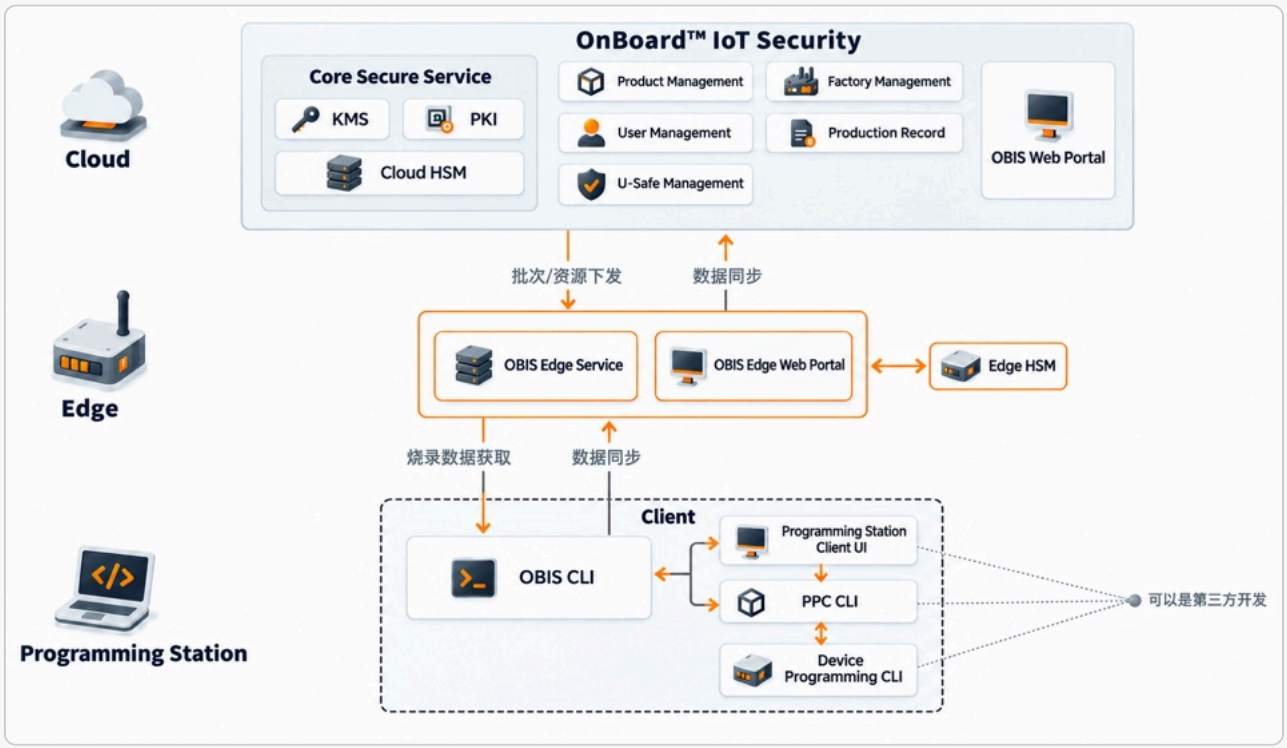
OBIS 基于 **授权模型（authorization-based model）** 构建，采用 **集中管控（centralized control ）** 与 **分布式实施（distributed execution）** 相结合的方式，使 OEM 能够管理所有产品的数字资产以及每一个安全关键操作，覆盖任意数量的工厂和合作伙伴，同时不会暴露敏感资料。

这种架构使 OEM 可以：

- 快速接入新的工厂
- 更换生产合作伙伴或芯片供应商
- 跨地区扩展生产规模

同时仍然保持集中化的控制与监控能力。与传统模式需要"信任工厂"不同，OBIS 的设计理念是：通过架构设计让"信任"变得不再必要。

产品架构



OBIS 由云端（Cloud），边缘端（Edge）和烧录端（Programming Station）三大部分构成：

端类型	说明

云端	物联网安全（OnBoard™ IoT Security）包含：
	• 核心安全服务（Core Secure Service）负责提供基础的安全服务和管理功能，包含：
	  ◦ 密钥管理系统（KMS）
	  ◦ 公钥基础设施（PKI）
	  ◦ 云硬件安全模块（Cloud HSM）
	• 产品管理（Product Management）
	• 工厂管理（Factory Management）
	• 用户管理（User Management）
	• 生产记录（Production Record）
	• U-Safe 管理（U-Safe Management）
	• OBIS 门户网站（OBIS Web Portal）：用户可以通过该网站进行产品相关的操作和管理，例如产品管理，工厂管理，用户管理，密钥管理，证书管理等。

边缘端	• OBIS 边缘服务 (OBIS Edge Service)：提供工厂本地端的生产管理能力。 • OBIS 边缘门户网站 (OBIS Edge Web Portal)：工厂管理员可通过该网站实现工厂本地的可视化管理。 • 边缘 HSM (Edge HSM)：用于产线服务器上的硬件安全设备。
烧录端	客户端（Client）提供 OBIS CLI（Command Line Interface），Programming Station Client UI，PPC（Programming Process Control） CLI 和 Device Programming CLI，用于执行设备烧录。

核心功能

- **数字资产协同化管理**

  以"产品"为核心纽带配置，管理数字资产，支持 OEM，ODM 等多角色协同管理，不同角色权限隔离，以保证数字资产的安全。通过产品配置与任务管理，完成从建模到量产前的全流程协作，主要包括：

  - **产品建模**：定义并维护产品基础信息及参数。
  - **团队协作**：邀请多方（可跨租户）成员共同参与产品管理。
  - **资源管理**：统一管理芯片参数，Matter 配置，固件，证书，密钥等资源包。
  - **生产烧录管理**：通过版本控制与批次管理，标准化每一次生产烧录任务。

- **安全烧录**

  提供从云端配置到边缘端执行的端到端安全烧录流程，覆盖密钥写入，证书生成，设备烧录，安全启动（Secure Boot），安全调试（Secure Debug）等关键环节，保障每台设备在生产阶段的参数可信与数据安全。

- **设备生命周期管理**

  支持对已烧录设备的持续管理，提供设备状态监控，固件安全升级等能力，实现设备从生产到投产后的全生命周期可查看，可控与可追溯。

- **安全基础设施**

  OBIS 提供标准的 KMS 和 PKI 服务，并结合 HSM 实现边缘端（工厂侧）的分布式 KMS 和 PKI 服务，同时保证 OBIS 端到端安全。

OBIS 定位与系统边界

OBIS 为 OEM 提供一个用于产品数字资产与授权管理的集中式控制平台，定位于上游芯片制造与下游设备生命周期管理之间的关键层。

- **上游：信任资产接入（Trust Artifact Ingestion）**

  OBIS 可接收在半导体制造过程中生成的信任资产，例如硬件 Root of Trust 材料或由芯片厂商注入的初始设备密钥。这些资产会被纳入 OEM 的产品安全模型，并在 OBIS 中与其他产品数字资产一起进行统一管理。

- **制造环节：生产批次执行（Production Batch Execution）**

  OBIS 向 OEM 自有或合作伙伴运营的工厂下发明确的生产批次，定义可生产的内容、生产地点、时间以及规模。

- **下游：信任资产交付（Trust Artifact Handoff）**

  OBIS 汇集所有生产结果，包括每台设备的安全资产以及不可篡改的生产记录，并将其交付给 OEM 管理或授权的设备管理平台，从而实现安全的设备接入并支持下游的生命周期运营。

  OBIS 并不是工厂执行系统、设备管理平台，也不是 MES 的替代品。它是一个以授权驱动的产品数字资产管理与安全治理层，使 OEM 能够通过单一控制平台，在所有制造合作伙伴之间统一协调产品安全。

角色分类与权限说明

支持基于角色的权限管理，当前角色分为三大类别：OBIS角色，产品角色与工厂角色。下文将通过六个表格，分别阐述每个角色的说明及其所对应的操作权限。

OBIS 角色

OBIS 角色由系统预定义，用于满足常用权限需求。以下角色可适配大多数使用场景，用户可根据实际需求配置一个或多个角色。如果有特殊权限控制需求，请联系雪球创建对应的特殊角色。

角色	说明
System Manager	拥有所有角色的权限。
Product Manager	拥有查看和操作产品功能页面的权限。
Resource Manager	拥有查看和操作 KMS 和 PKI 功能页面的权限。
Factory Manager	拥有查看和操作工厂功能页面的权限。
Normal User	拥有查看所有功能页面的权限。

下表汇总了 OBIS 各角色在不同功能页面上的访问与操作权限。

功能页面	权限点	System Manager	Product Manager	Factory Manager	Resource Manager	Normal User
产品	页面访问	✓	✓			✓
	页面操作	✓	✓			

功能页面	权限点	System Manager	Product Manager	Factory Manager	Resource Manager	Normal User
工厂	页面访问	✓		✓		✓
	页面操作	✓		✓		
记录	页面访问	✓	✓	✓	✓	✓
KMS	页面访问	✓			✓	✓
	页面操作	✓			✓	
PKI	页面访问	✓			✓	✓
	页面操作	✓			✓	
账号	页面访问	✓				✓
	页面操作	✓				

功能页面	权限点	System Manager	Product Manager	Factory Manager	Resource Manager	Normal User
用户组	页面访问	✓				✓
	页面操作	✓				

产品角色

产品角色用于控制产品资源的访问与操作权限，用户可根据实际需求配置一个或多个角色。

角色	说明
Product Owner	默认为产品的创建人，拥有该产品的全部操作权限。
Product Manager	拥有该产品的全部操作权限。
Resource Manager	拥有查看该产品和管理资源的权限。
Version Manager	拥有查看该产品和管理版本的权限。
Batch Manager	拥有查看该产品和管理批次的权限。
Member	默认用户角色，拥有查看该产品的权限。

下表汇总了产品各角色在产品功能的操作权限。

功能	操作	Product Owner	Product Manager	Resource Manager	Version Manager	Batch Manager	Member
产品	配置产品	✓	✓	✓			
	查看产品	✓	✓	✓	✓	✓	✓
	管理产品	✓	✓				
	管理成员	✓	✓				
	创建版本	✓	✓		✓		

功能	操作	Product Owner	Product Manager	Resource Manager	Version Manager	Batch Manager	Member
	查看版本	✓	✓	✓	✓	✓	✓
	管理版本	✓	✓		✓		
	创建批次	✓	✓			✓	
	查看批次	✓	✓	✓	✓	✓	✓
	查看烧录设备	✓	✓	✓	✓	✓	✓

工厂角色

工厂角色用于管理和操作工厂相关业务，用户可根据实际需求进行配置（本地工厂 & 云工厂）。

角色	说明
Factory Manager	拥有工厂在云端和边缘端的全部操作权限。

下表汇总了 Factory Manager 在云端和边缘端的操作权限。

端类型	功能	工厂类型	操作	Factory Manager
云端	工厂	本地工厂 & 云工厂	创建工厂	✓
		本地工厂 & 云工厂	配置与管理工厂	✓
		本地工厂 & 云工厂	配置与管理烧录站	✓
		云工厂	管理批次与分配烧录站	✓
		本地工厂 & 云工厂	查看工厂	✓
边缘端	工厂	本地工厂	管理工厂	✓

端类型	功能	工厂类型	操作	Factory Manager
	烧录站	本地工厂	配置与管理烧录站	✓
	批次	本地工厂	管理批次与分配烧录站	✓

版本：v3.2.2

开通与登录 OnBoard™ IoT Security

欢迎使用 OnBoard™ IoT Security (OBIS)！本文将指导你快速完成 OBIS 账号开通并登录，让你立即开始使用 OBIS 功能。请按照以下步骤操作：

开通 OBIS 账号

请联系我们并提供以下信息，我们将协助你快速完成 OBIS 账号的开通。

- 公司名称
- 姓名
- 邮箱

提交信息后，我们会尽快与你联系，完成开通流程。

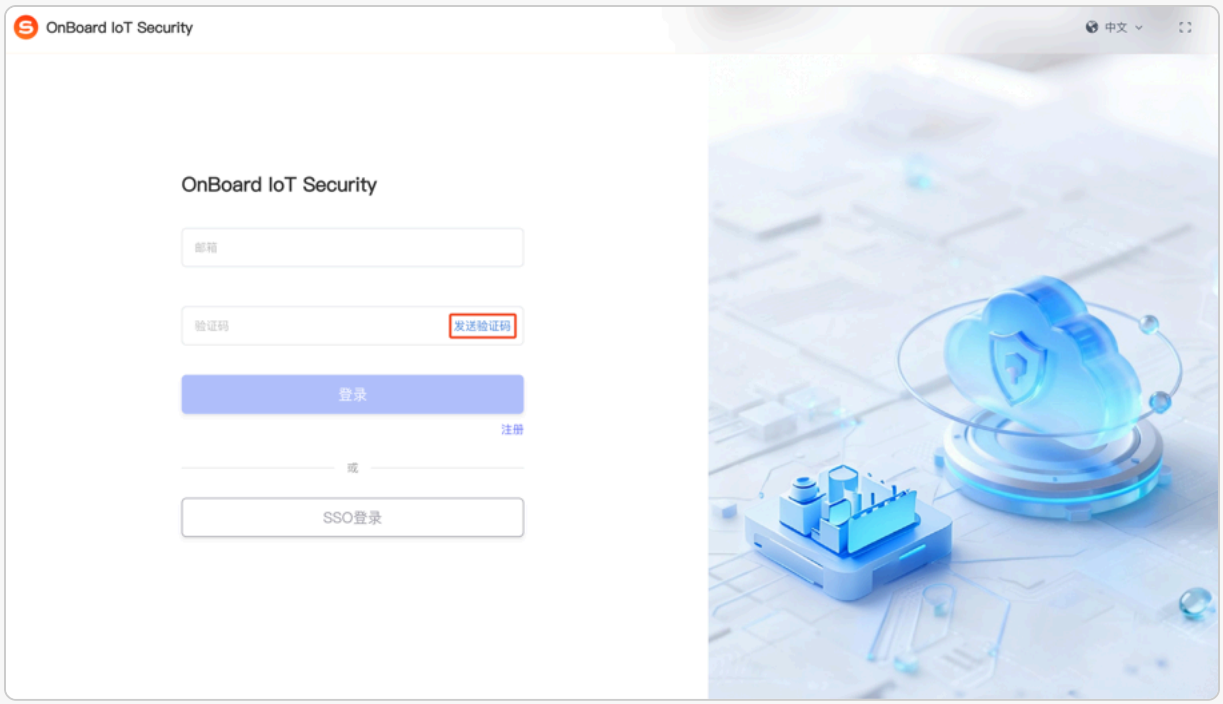
登录 OBIS

账号开通后，可通过验证码或者 SSO (Single Sign-On) 方式登录：

方式一：使用验证码登录

> **提示**
>
> 可以联系雪球启用企业自注册功能，允许用户使用公司邮箱在 OBIS 登录页面自行注册账号并使用验证码方式登录，OBIS 将自动为其分配默认角色权限。

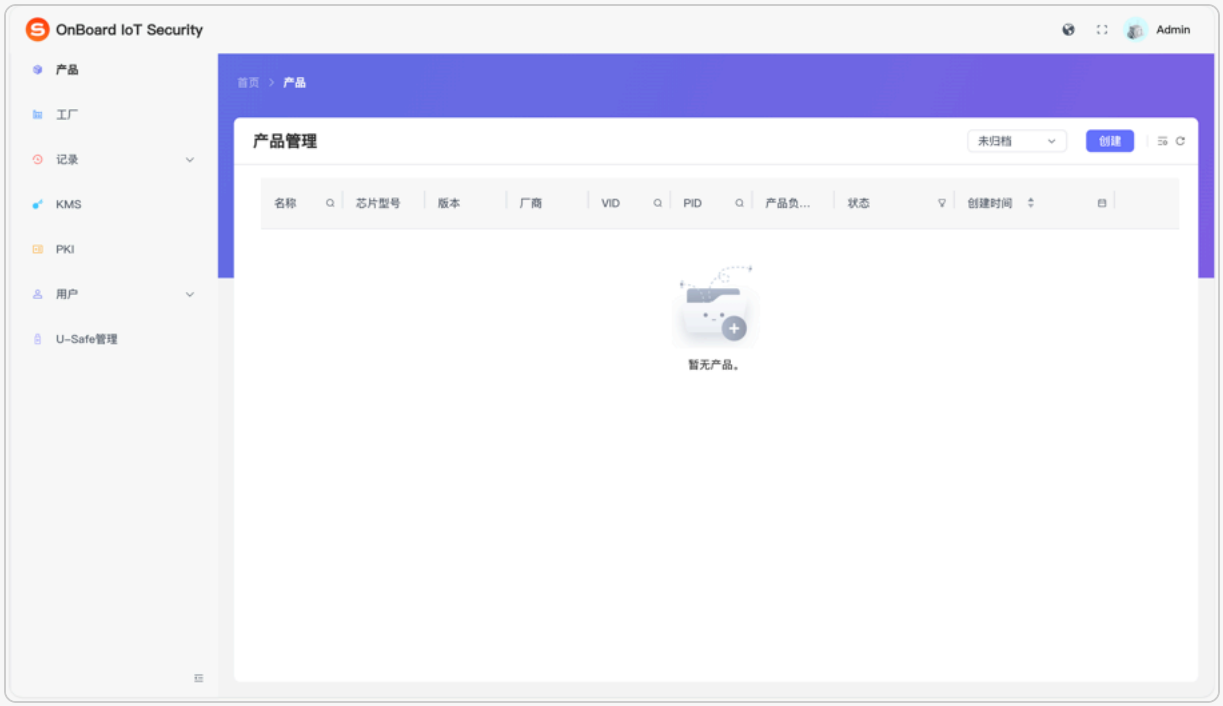
1. 在 OBIS 登录页面，输入你在注册 OBIS 账号时使用的邮箱，然后点击**发送验证码**。



2. 登录你的邮箱，查收来自 OBIS 的验证邮件并获取 6 位数字验证码。

3. 在 OBIS 登录页面，输入你邮件中收到的 6 位数字验证码，然后点击**登录**。

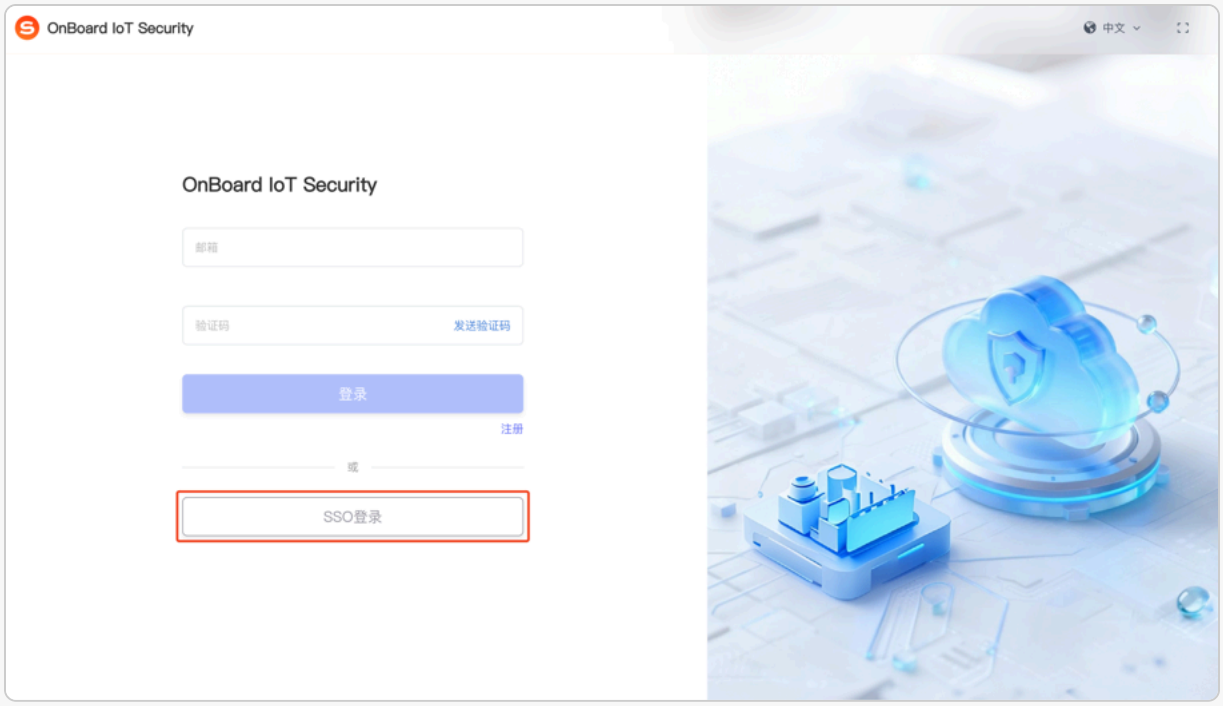
4. 登录成功后，页面会跳转至 OBIS 主界面。



方式二：使用 SSO 登录

> **提示**
>
> 可以联系雪球启用 SSO 登录功能，允许用户通过企业身份提供商（IdP） 使用公司邮箱进行登录，OBIS 将自动为其创建账号并分配默认角色权限。更多信息请参考配置 SSO 登录。

1. 在 OBIS 登录页面，选择 **SSO 登录**。



2. 输入你的公司邮箱并点击**下一步**，页面会跳转至企业 IdP 认证页面。

3. 认证成功后，页面会跳转至 OBIS 主界面。



版本：v3.2.2

产品创建与设备烧录

欢迎使用 OnBoard™ IoT Security (OBIS)！本文将引导你快速完成以下操作：

- 创建和配置产品
- 创建版本与批次
- 为批次分配烧录站并完成设备烧录

按照本文步骤操作，你将能够快速上手 OBIS，完成从产品创建到设备烧录的完整流程。

前提条件

在开始创建产品和设备烧录之前，请确保已满足以下条件：

- 你的账号已具备在 OBIS 完成产品创建到执行设备烧录流程的角色权限。
- 联系雪球获取用于身份鉴权的 HSM U-Safe 和 Identity U-Safe。
- 你已完成工厂的创建和配置。具体操作请参考：
  - 创建工厂
  - 配置工厂（本地工厂 & 云工厂）
  - 配置烧录站（本地工厂 & 云工厂）

操作示意图



第一步：创建和配置产品

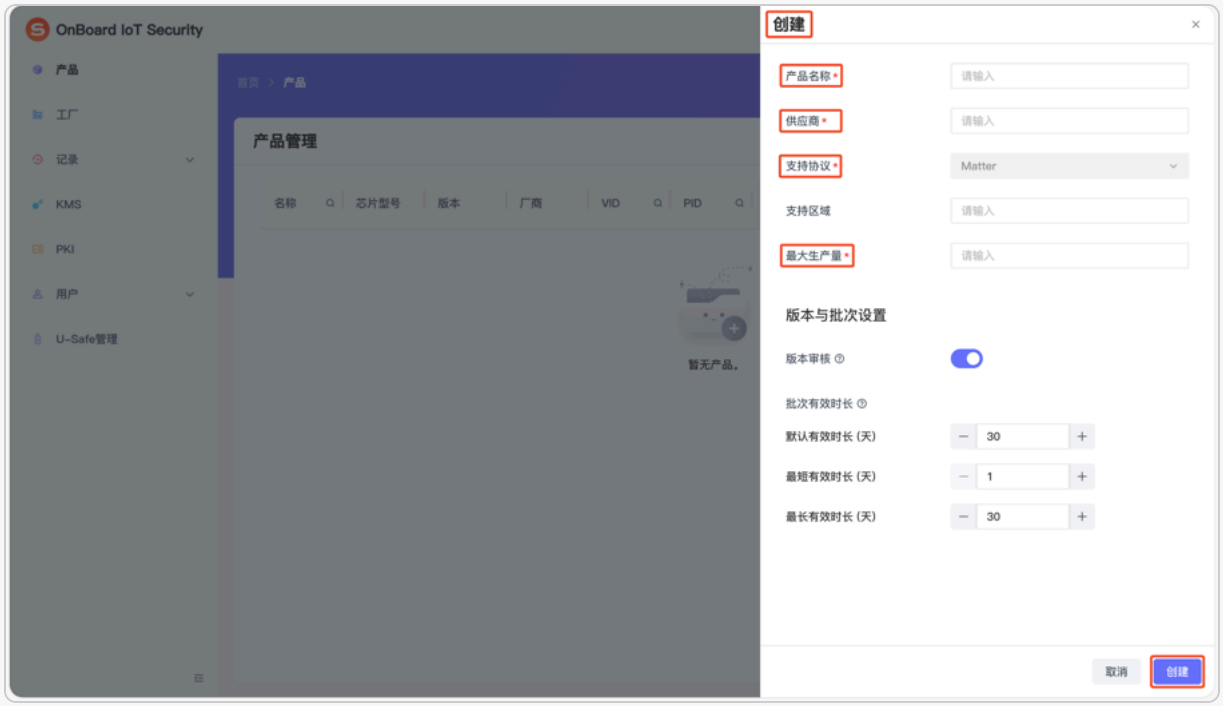
请参考以下步骤，在 OBIS 创建和配置你的产品：

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击**创建**。

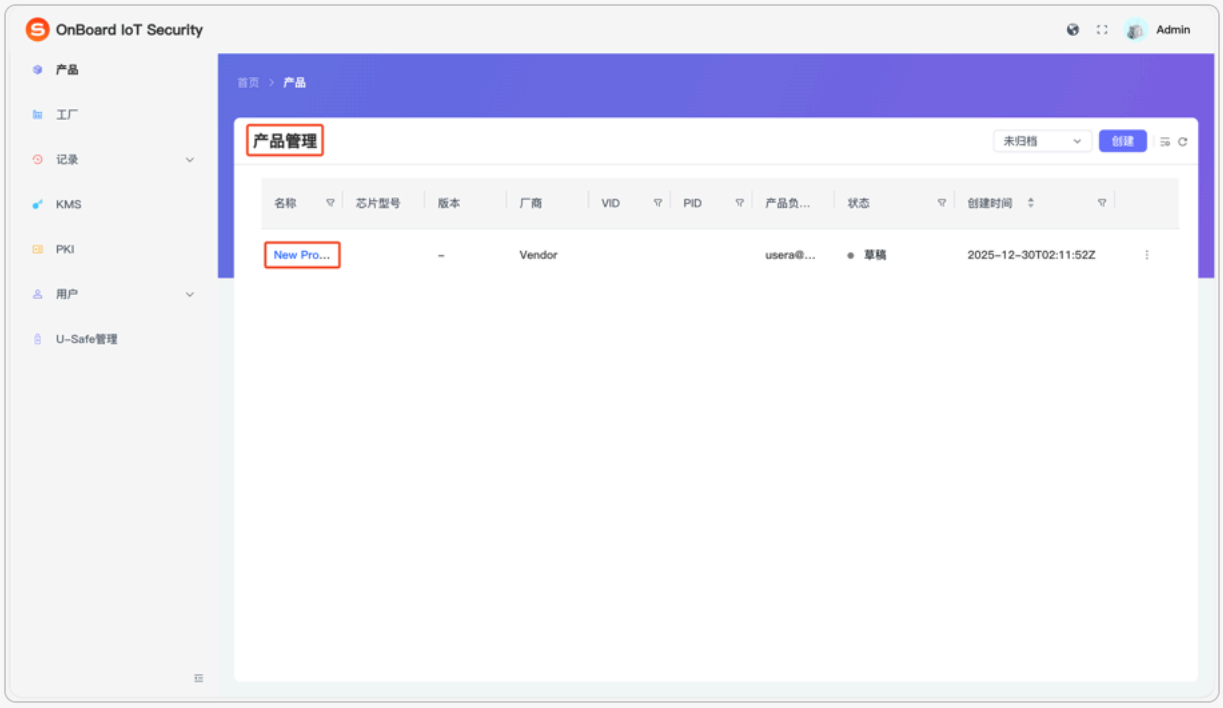


3. 在**创建**面板，输入产品必要的基础信息（**产品名称**，**供应商**，**支持协议**，**最大生产量**），然后点击**创建**。产品创建成功后，页面会跳转至产品页面。

> **备注**
>
> 更多信息请参考产品基础信息说明。



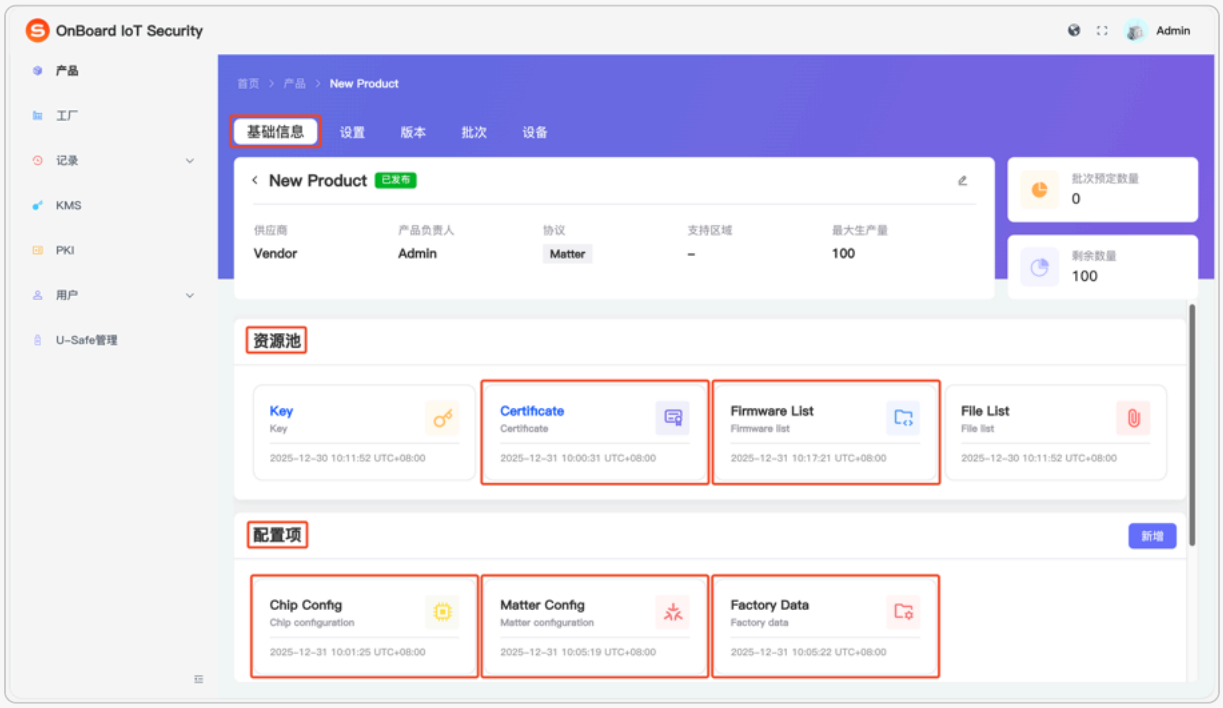
4. 在**产品管理**页面，选择创建的产品。



5. 在产品页面，选择**基础信息**标签页。找到**资源池**和**配置项**区域，分别点击并配置产品必要的物料与配置项：

    - **Certificate**
    - **Firmware List**
    - **Chip Config**
    - **Matter Config**
    - **Factory Data**

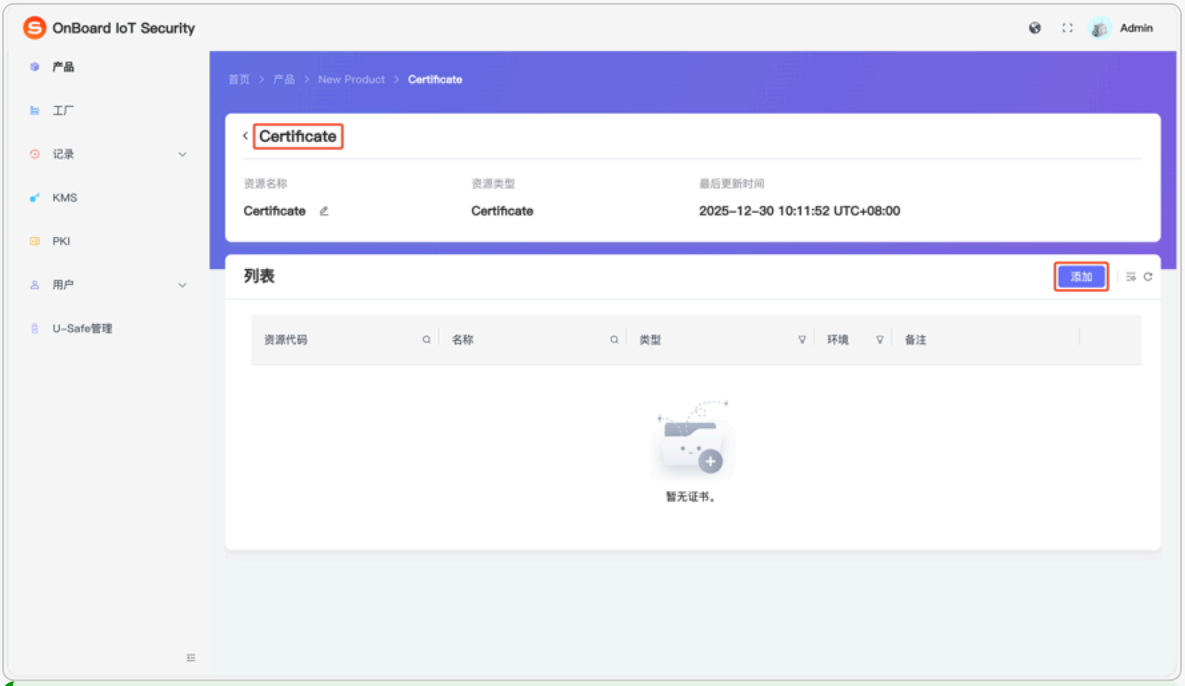
    > **提示**
    >
    > **资源池**与**配置项**区域的配置应用于创建版本。更多信息请参考配置产品的步骤。在配置 Firmware List，Matter Config 和 Factory Data 时上传的文件，之后可以随时在对应资源的配置页面进行下载。



Certificate | Firmware List | Chip Config | Matter Config | Factory Data

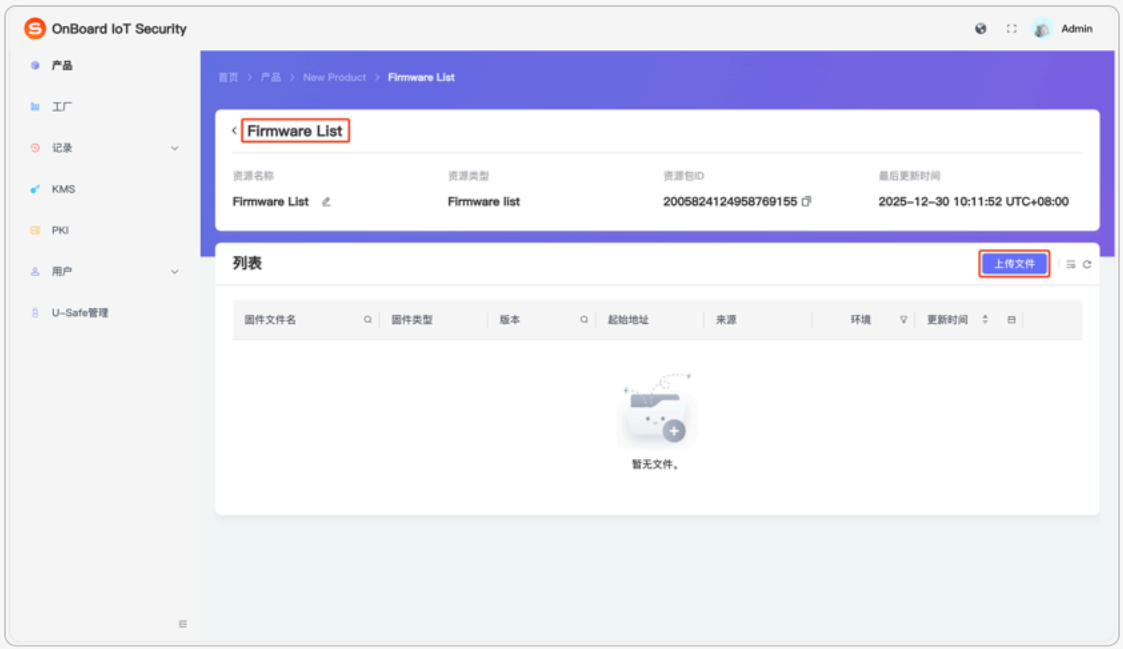
> **提示**
>
> 配置 **Certificate** 的目的是将 PKI 中的 DAC 模板添加到该产品，使其在配置 Matter Config 时可供选择。
>
> DAC 模板可以在 PKI 中创建，更多信息请参考创建证书的步骤。

在 **Certificate** 页面，点击**添加**，选择并添加 **CA 模板**。



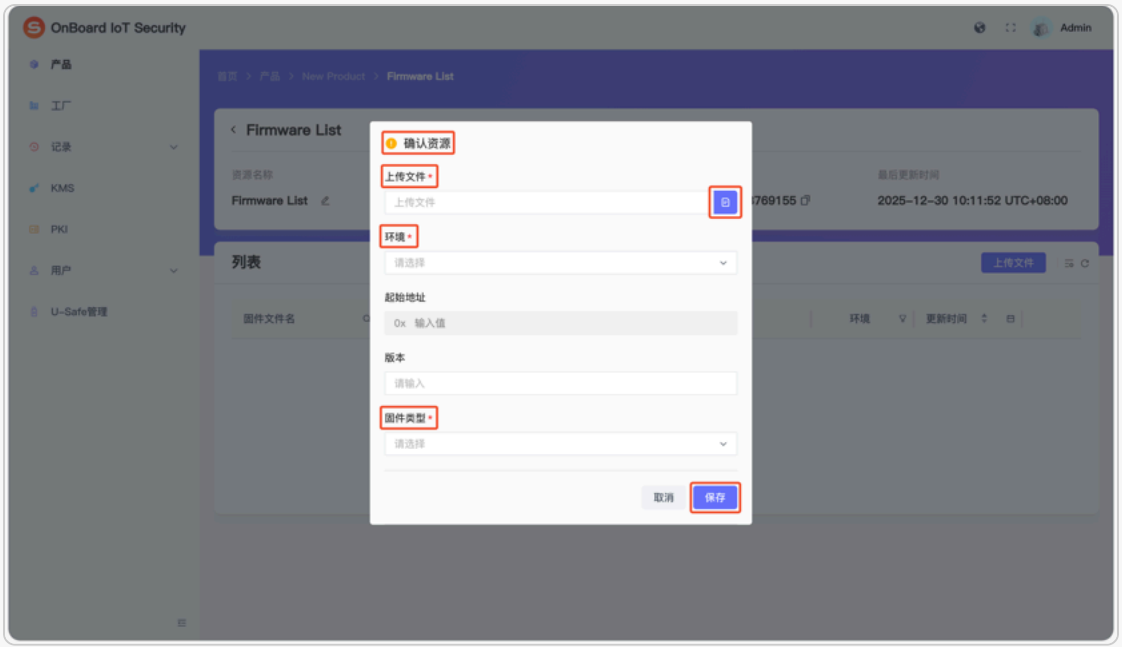
> **提示**
>
> 配置 **Firmware List** 的目的是上传该产品需要烧录的固件文件，支持上传 **.bin**, **.hex** 和 **.s37** 格式的固件。上传的固件可在创建版本时进行选择。

i. 在 **Firmware List** 页面，点击**上传文件**。



ii. 在**确认资源**窗口，点击文件 图标上传固件文件。同时，选择**环境**（**TEST** 或 **PROD**）与**固件类型**（**Application** 或 **Bootloader**）并点击**保存**。

> **备注**
>
> 更多信息请参考固件信息说明。



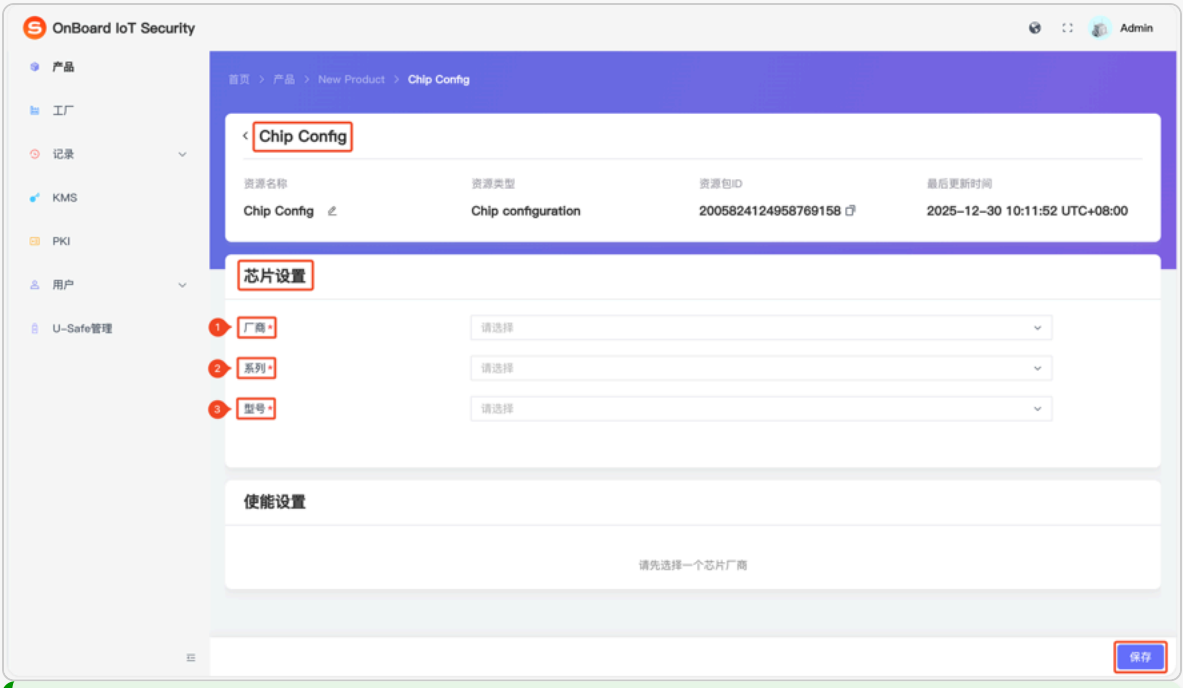
提示

配置 **Chip Config** 的目的是选择并设定该产品所使用的芯片型号和安全选项。

具体列表请参考当前支持的芯片设置。后续将根据需求新增支持的芯片厂商，系列和型号。

更多信息请参考配置 Chip Config 的步骤。

在 **Chip Config** 页面，找到**芯片设置**区域，选择芯片**厂商**，**系列**和**型号**，并点击**保存**。



提示

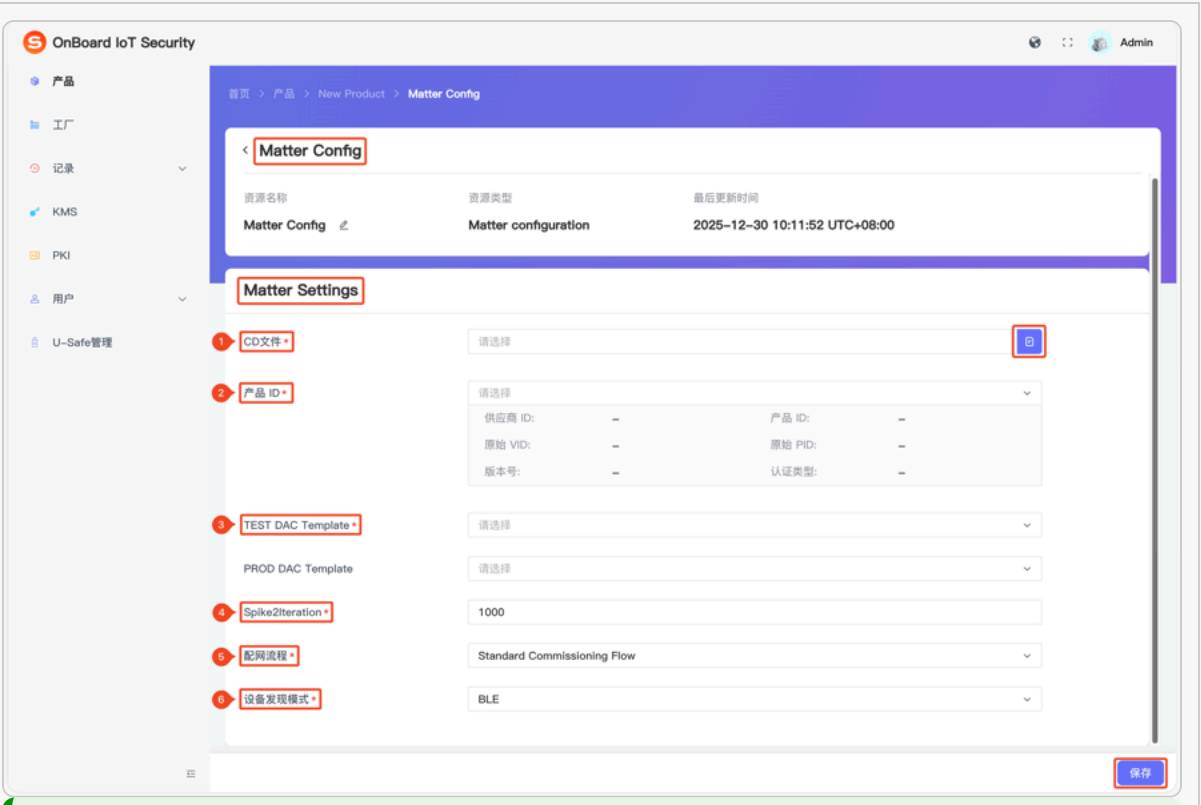
配置 **Matter Config** 的目的是选择并设定与 Matter 协议相关的设置。

在此步骤中，需要选择并上传 CD 文件，并选择在配置 **Certificate** 时添加的 DAC 模板。请确保 CD 文件与 DAC 模板中的 VID （供应商 ID）和 PID （产品 ID） 保持一致。

在后续创建版本时，你可以根据实际需求更换其他 CD 文件。

更多信息请参考 Matter Settings 信息说明。

在 **Matter Config** 页面中，找到 **Matter Settings** 区域，点击文件 图标选择并上传 CD 文件。同时，选择**产品 ID**，**TEST DAC Template**，**Spike2Iteration**，**配网流程**和**设备发现模式**并点击**保存**。

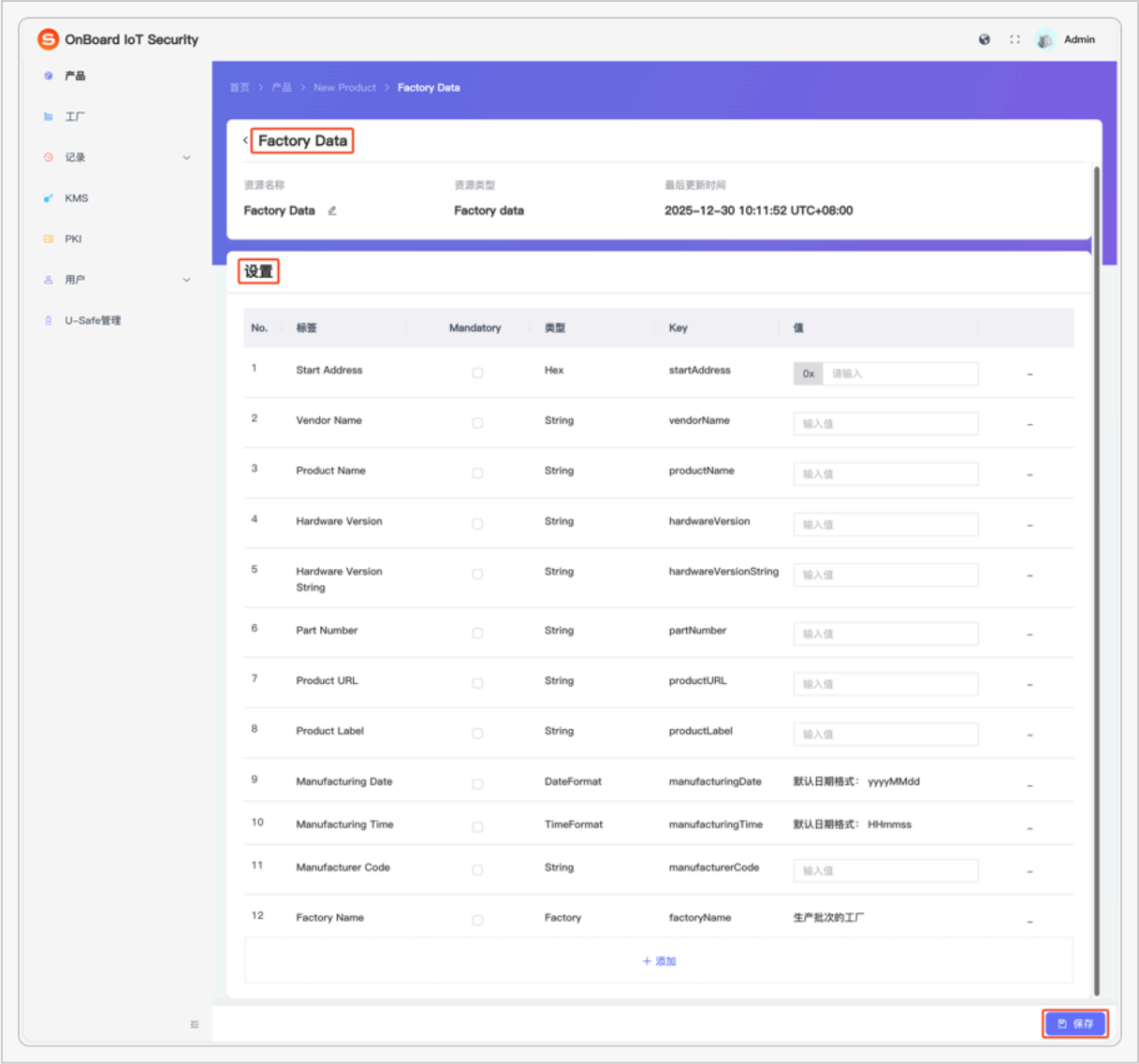


提示

配置 **Factory Data** 的目的是设定工厂的相关信息，支持 Matter 规范所需的全部字段。在后续创建版本时，你可以根据实际需求对 Factory Data 进行修改。

在此处勾选的 Factory Data 信息将在创建版本时作为必填项。更多信息请参考 Factory Data 信息说明。

在 **Factory Data** 页面，找到**设置**区域，输入 Factory Data 信息（值可为空），然后点击**保存**。

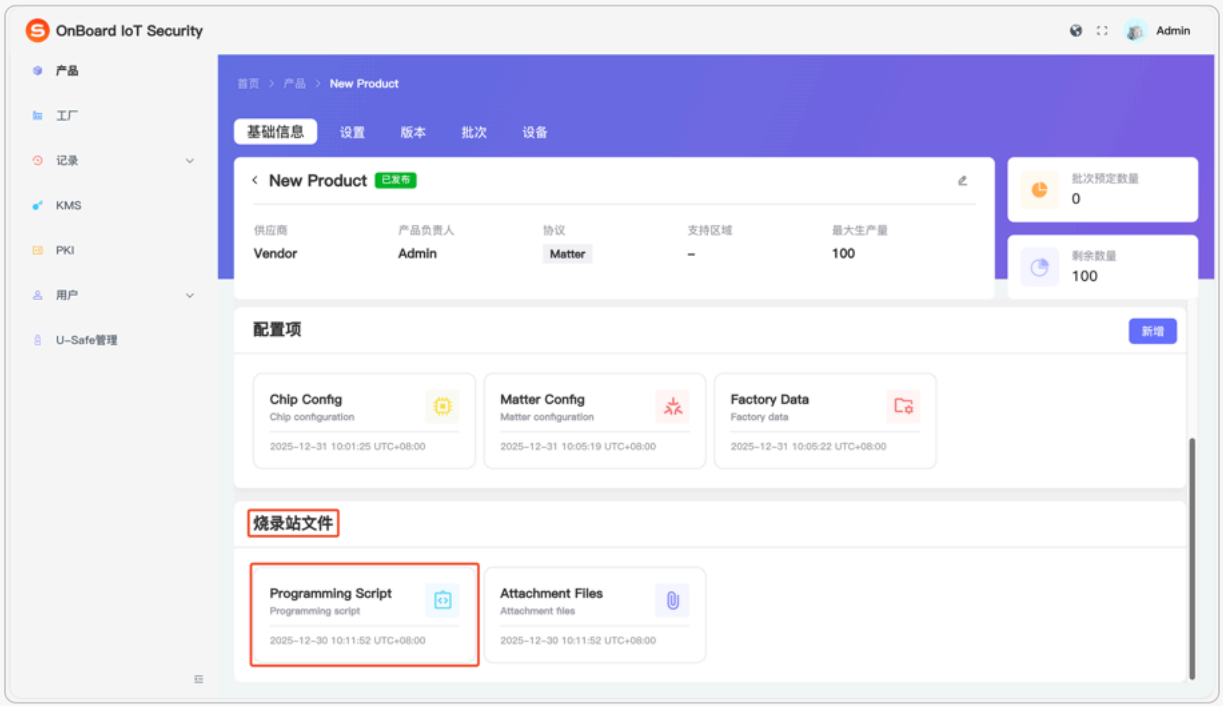


6. 在产品页面中，找到**烧录站文件**区域并上传对应的文件。

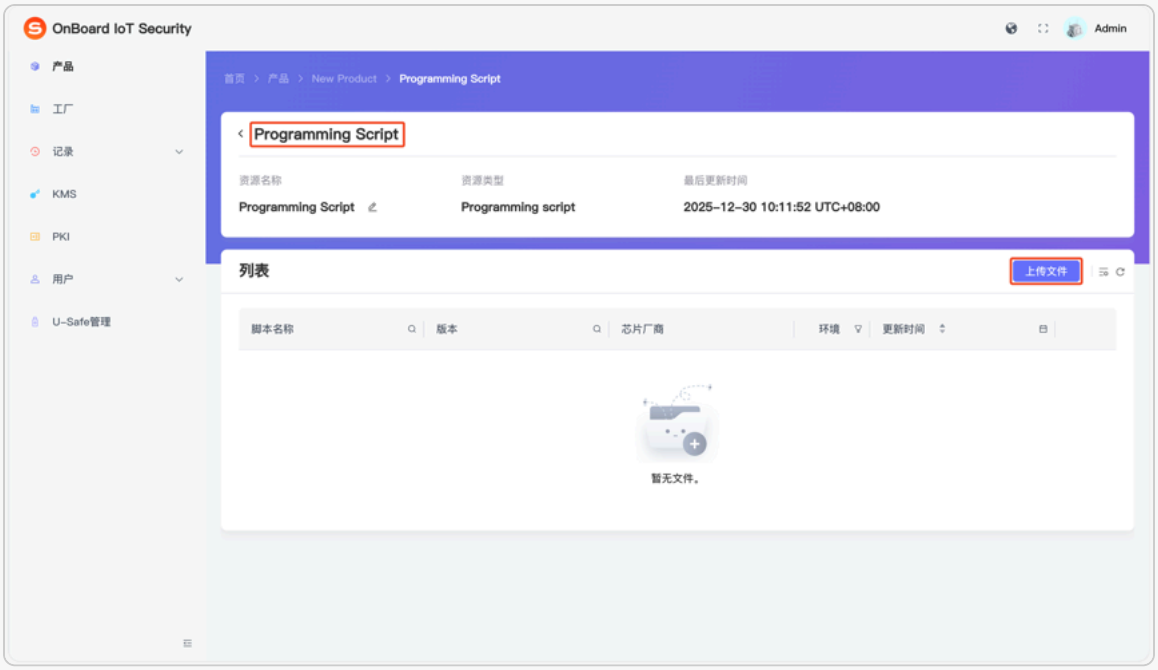
> **备注**
>
> **烧录站文件**区域的文件应用于创建批次。更多信息请参考上传烧录站文件的步骤。在配置烧录站文件时上传的文件，之后可以随时在对应资源的配置页面进行下载。

例如，请参考以下步骤上传烧录脚本。

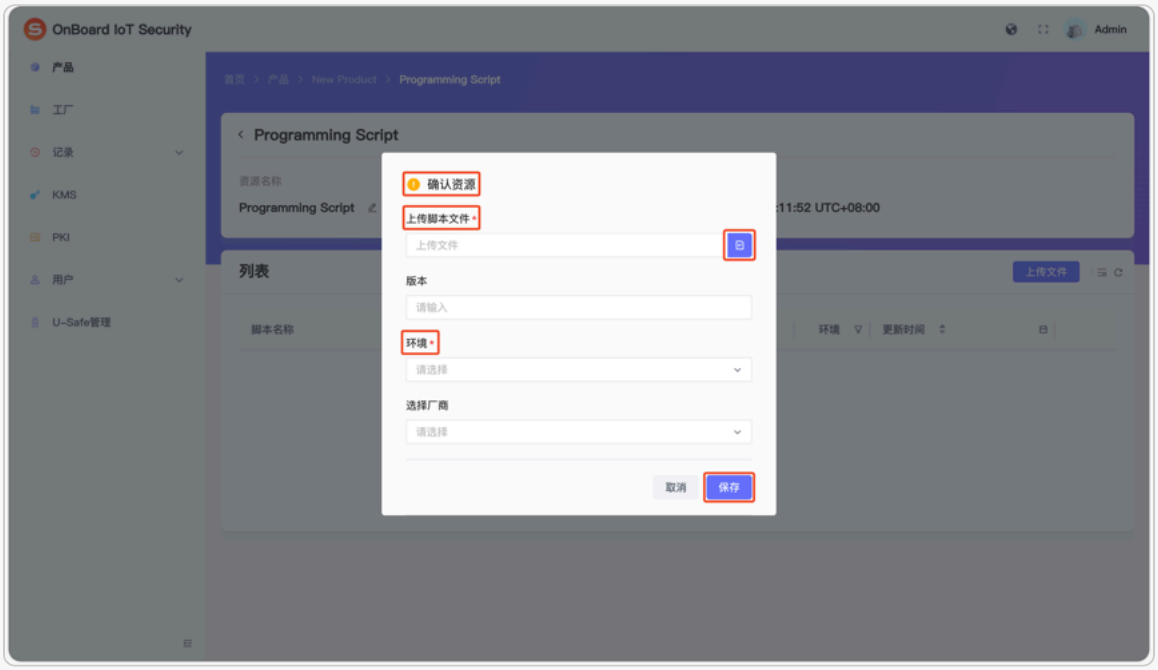
> **提示**
>
> 配置 **Programming Script** 的目的是上传设备烧录时使用的脚本文件（支持上传任意格式的烧录脚本）。上传的烧录脚本可在创建批次时进行选择。
>
> 你可以根据实际需求选择并上传烧录脚本；如果暂不需要，后续可在 Programming Station 客户端中为特定批次选择烧录脚本。



i. 在 **Programming Script** 页面，点击**上传文件**。



ii. 在**确认资源**窗口，点击文件 图标上传脚本文件。同时，选择**环境**（**TEST** 或 **PROD**）并点击**保存**。



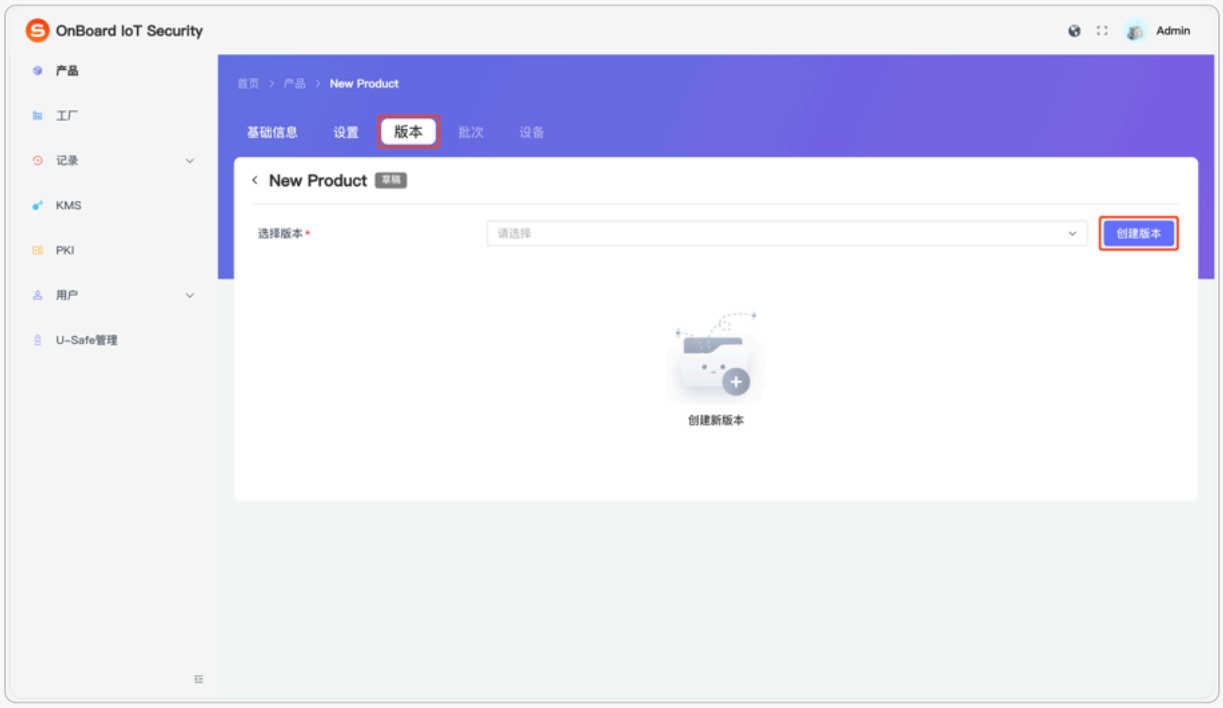
第二步：创建版本

> **提示**
>
> 一个产品可以创建多个版本。

产品创建并配置完成后，请参考以下步骤，在 OBIS 创建版本：

1. 在**产品**页面，选择**版本**标签页并点击**创建版本**。页面会跳转至**创建版本**标签页。

   > **提示**
   >
   > 如果**版本**标签页显示为灰色，表示当前不可用。请检查该产品必要的资源与配置项是否均已完成配置。当所有内容配置完成后，**版本**标签页将自动启用。



2. 在创建版本标签页：

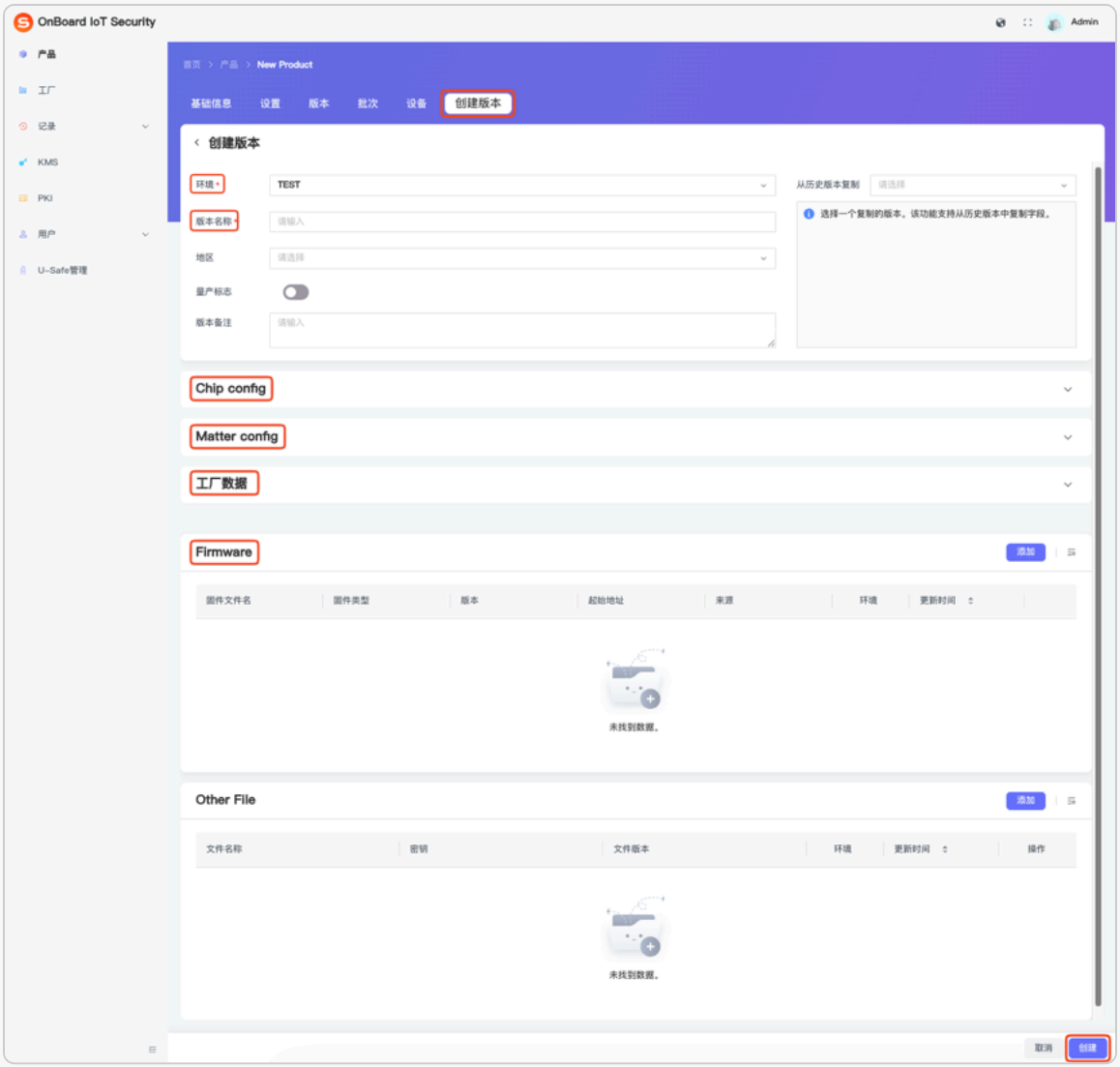
> **备注**
>
> 更多信息请参考版本信息说明。

i. 选择**环境**，并输入**版本名称**。

ii. 核对 **Chip config**，**Matter config** 和**工厂数据**区域中的信息。

iii. 在 **Firmware** 区域中点击**添加**，选择并添加在配置产品时上传的固件。

iv. 点击**创建**。版本创建成功后，页面会跳转至**版本**标签页。



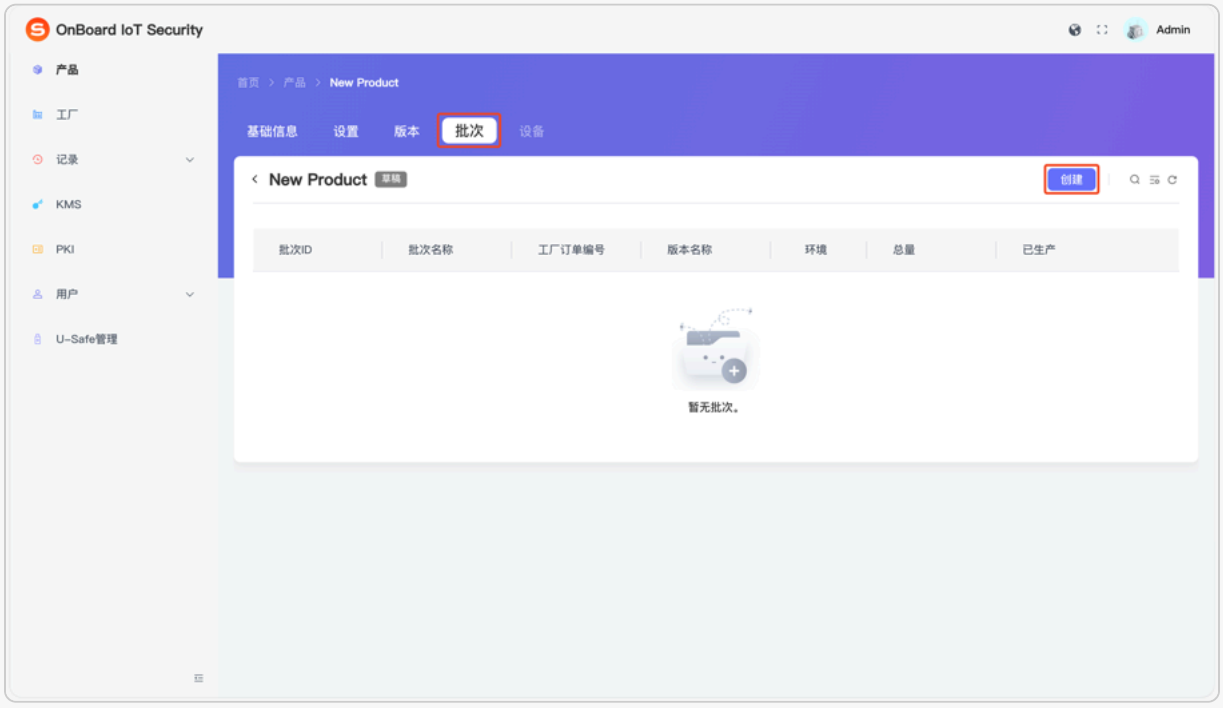
第三步：创建批次

> **提示**
>
> 一个版本可以创建多个批次，而批次则代表基于该版本的生产计划。

创建版本后，请参考以下步骤，在 OBIS 创建批次：

1. 在产品页面，选择**批次**标签页并点击**创建**。页面会跳转至**创建批次**标签页。

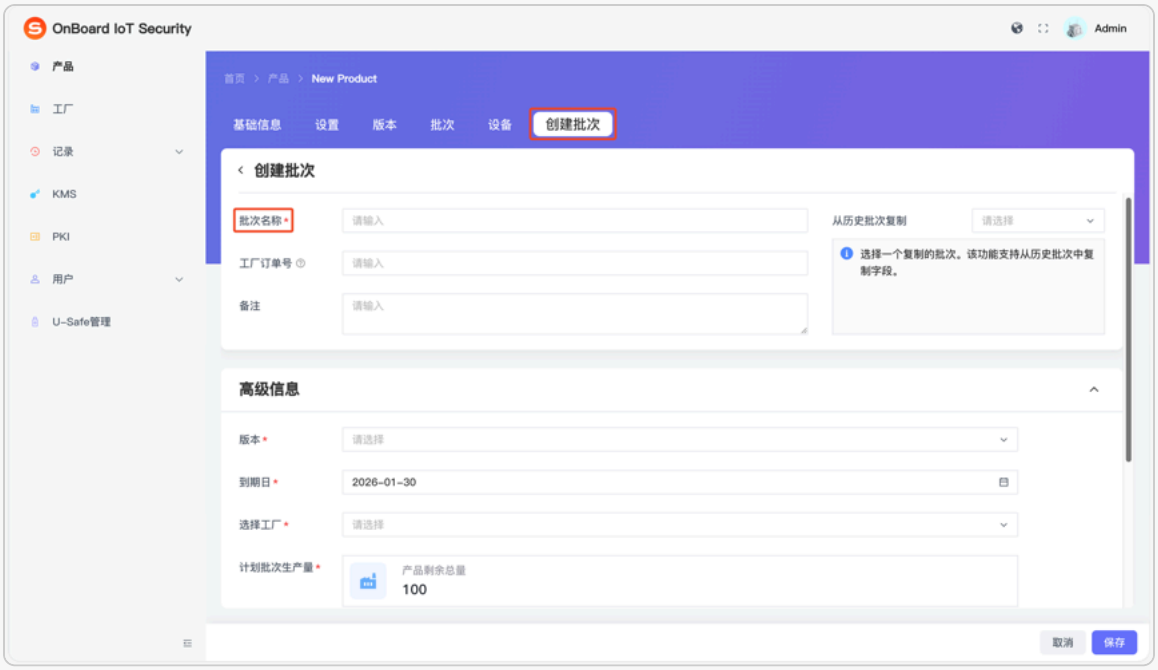
> **提示**
>
> 如果**批次**标签页显示为灰色，表示当前不可用。请检查版本是否已创建完成。创建版本后，**批次**标签页将自动启用。



2. 在**创建批次**标签页：

> **备注**
>
> 更多信息请参考批次信息说明。

   i. 输入**批次名称**

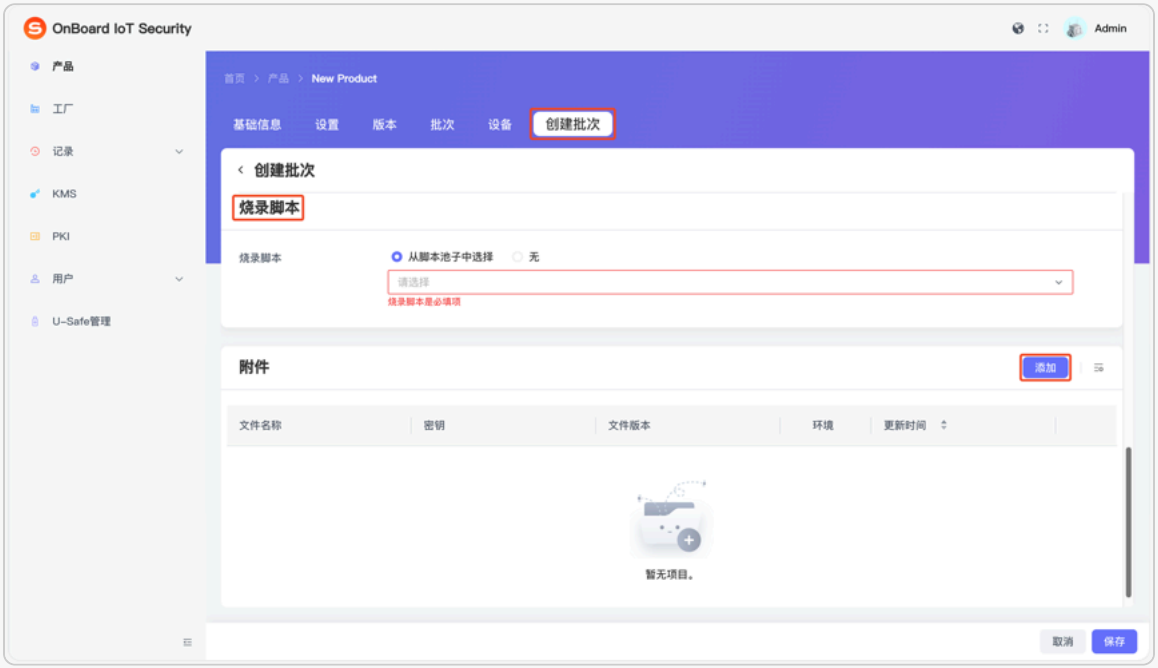


ii. 在高级信息区域：

- 选择在第二步创建的版本
- 设置**到期日**
- 选择工厂
- 输入**计划批次生产量**



iii. 在**烧录脚本**区域，你可以根据实际需求选择配置产品时上传的烧录脚本；如果暂不需要，可选择**无**，后续可在 Programming Station 客户端中为特定批次选择烧录脚本。



iv. 点击**保存**。批次创建成功后，页面会跳转至批次标签页。

第四步：为批次分配烧录站

> **提示**
>
> 为批次分配烧录站的目的是将该生产计划与烧录站进行绑定。

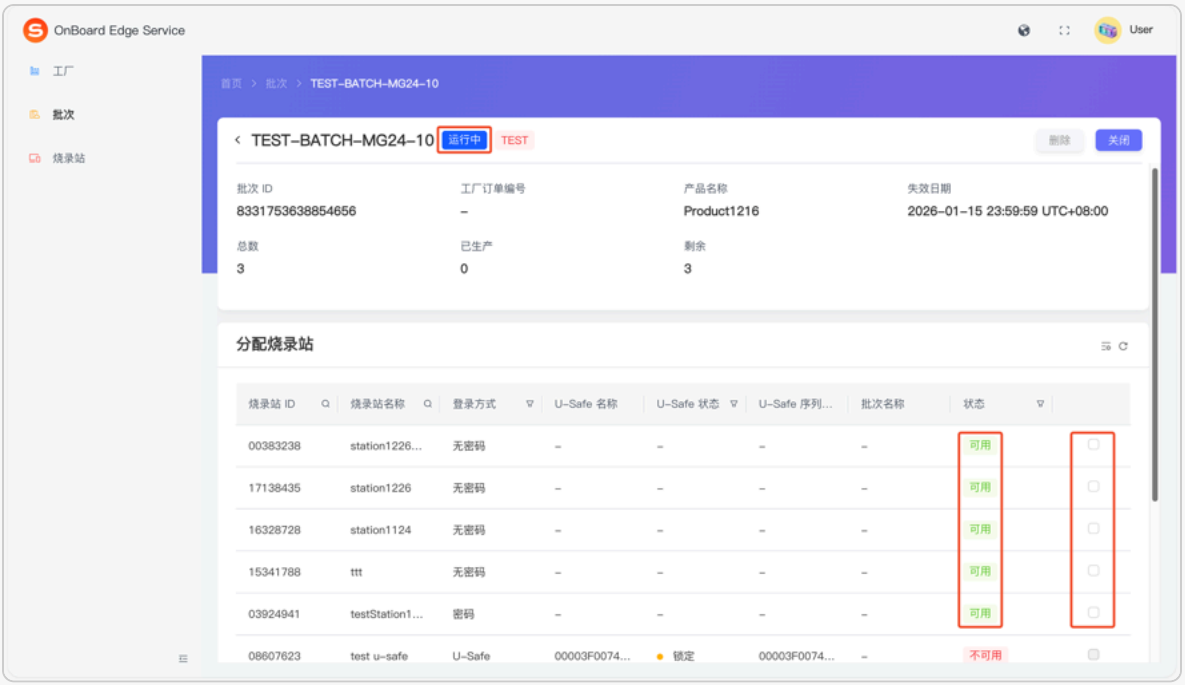
本地工厂 | 云工厂

请联系雪球协助完成 OnBoard Edge Service 在工厂侧的部署。

部署完成后，你可以通过生成的服务地址登录 OnBoard Edge Service 并管理工厂，烧录站和批次：

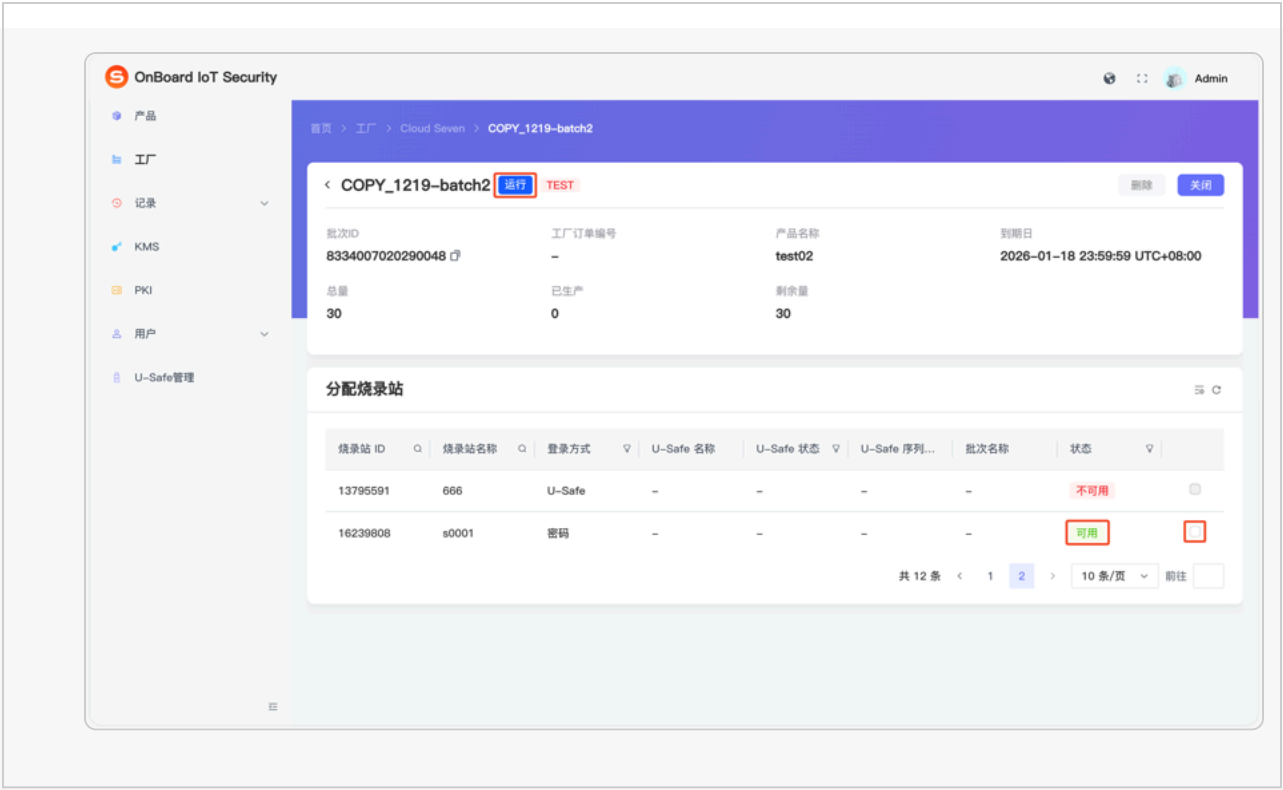
1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**批次**，在**批次列表**中点击在第三步创建的批次。
3. 在批次页面，确认批次的状态为**运行中**，并勾选状态为**可用**的烧录站进行分配。

> **备注**
>
> 有关烧录站的创建，请参考添加烧录站。



1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID 进入工厂页面并选择**批次**标签页。
3. 在**批次列表**中点击在第三步创建的批次，确认批次的状态为**运行**，并勾选状态为**可用**的烧录站进行分配。

> **备注**
>
> 有关烧录站的创建，请参考添加烧录站。



第五步：设备烧录

Programming Station 客户端用于设备烧录。在 OnBoard Edge Service 为批次分配烧录站后，请参考以下步骤开始烧录：

1. 安装并登录 Programming Station 客户端，确认设备的状态为**就绪**。

> **提示**
>
> 如果在创建批次时未选择烧录脚本，或者想更换为其他烧录脚本，可点击脚本右侧的文件夹图标，从本地选择所需的烧录脚本（任意格式）。

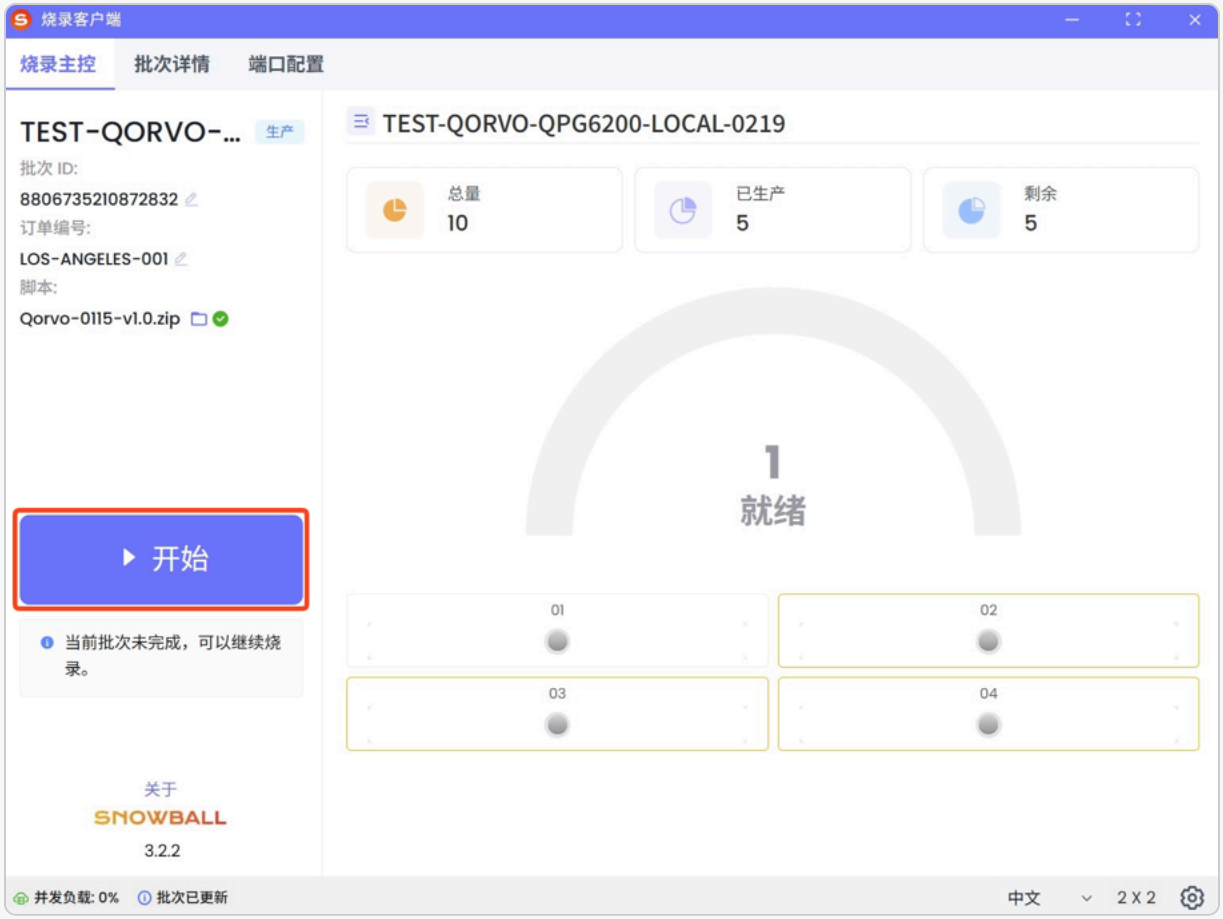


2. 在 Programming Station 客户端的主界面中，选择**端口配置**标签页，根据实际需求在**布局**区域设置烧录器连接数量。

> **提示**
>
> 设定烧录器连接数量从而调整同时烧录的设备数量。当前系统支持设置的烧录器连接数量由布局的行数与列数的乘积决定，需同时满足以下条件：
>
> - 行数和列数必须为整数
> - 行数 × 列数 ≤ 24
> - 行数和列数均不得超过 16
> - 当行的数大于 4 时，列数仅可为 1 或 2



3. 在 Programming Station 客户端的主界面中，点击**开始**，进行设备烧录。



4. 烧录结束后，点击对应的端口即可查看设备的烧录详情。

> **提示**
>
> 同时，也可在 OBIS 上查看设备档案及相关日志信息。


烧录客户端
烧录主控
批次详情
端口配置
TEST-QORVO-...
生产
批次 ID:
8806735210872832
订单编号:
LOS-ANGELES-001
脚本:
Qorvo-0115-v1.0.zip
1200000508
100%
成功
详情
hex"
"C:\Users\maryw\AppData\Roaming\SNB\ppcli\programmings\Qorvo\matter_factory_data_1200000508.hex"
[15:18:42] hexmerge:
[15:18:42] [Progress]:80% Executing command 9: program-hex
"C:\Users\maryw\AppData\Roaming\SNB\ppcli\programmings\Qorvo\qorvo_merged_QPG6200_1200000508.hex" --device QPG6200 --emulator-serialnumber 1200000508 --emulator-logs
"C:\Users\maryw\AppData\Roaming\SNB\Qorvo\emulator_logs"
[15:19:04] output:Programming
C:\Users\maryw\AppData\Roaming\SNB\ppcli\programmings\Qorvo\qorvo_merged_QPG6200_1200000508.hex Programming took 20.52 seconds. [Progress]:90%
[15:19:06] [Progress]:100% [Finish].
[15:19:08] out:b'{\r\n "success": true,\r\n "respCode": "8000",\r\n "respMsg": "NoError",\r\n "respData": null,\r\n "respTime": "1770189547"\r\n}\r\n'
[15:19:08] [EXIT] 0
[QRCode]: MT:634J0-C714EPGS7WE10
QRcode link
开始
当前批次未完成，可以继续烧录。
关于
SNOWBALL
3.2.2
并发负载: 0%
批次已更新
中文
2 X 2

版本：v3.2.2

功能指南 - OnBoard™ IoT Security

OnBoard™ IoT Security 提供了一套完整的功能，覆盖 IoT 设备安全烧录全流程的每个环节—包括产品管理、工厂管理、密钥与证书管理、用户管理、U-Safe 管理及设备记录追踪。选择下方功能模块，快速了解各模块的核心能力与相关操作。

产品

- 创建产品
- 配置产品
- 管理产品
- 创建版本
- 创建批次

工厂

本地工厂

- 配置和管理工厂
- 配置和管理烧录站

云工厂

- 配置和管理工厂
- 配置和管理烧录站
- 管理批次与分配烧录站

记录

- 查看设备历史

KMS

- 创建密钥
- 查看密钥
- 管理密钥

PKI

- 创建证书
- 查看证书
- 管理证书

用户

账号

- 创建账号
- 查看账号
- 管理账号

用户组

- 创建用户组
- 查看用户组
- 管理用户组

U-Safe 管理

- 查看 U-Safe

版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建产品

> **信息**
>
> 目前产品创建后无法被删除。

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中创建产品从而配置产品的基础信息，为后续配置产品物料做准备。

前提条件

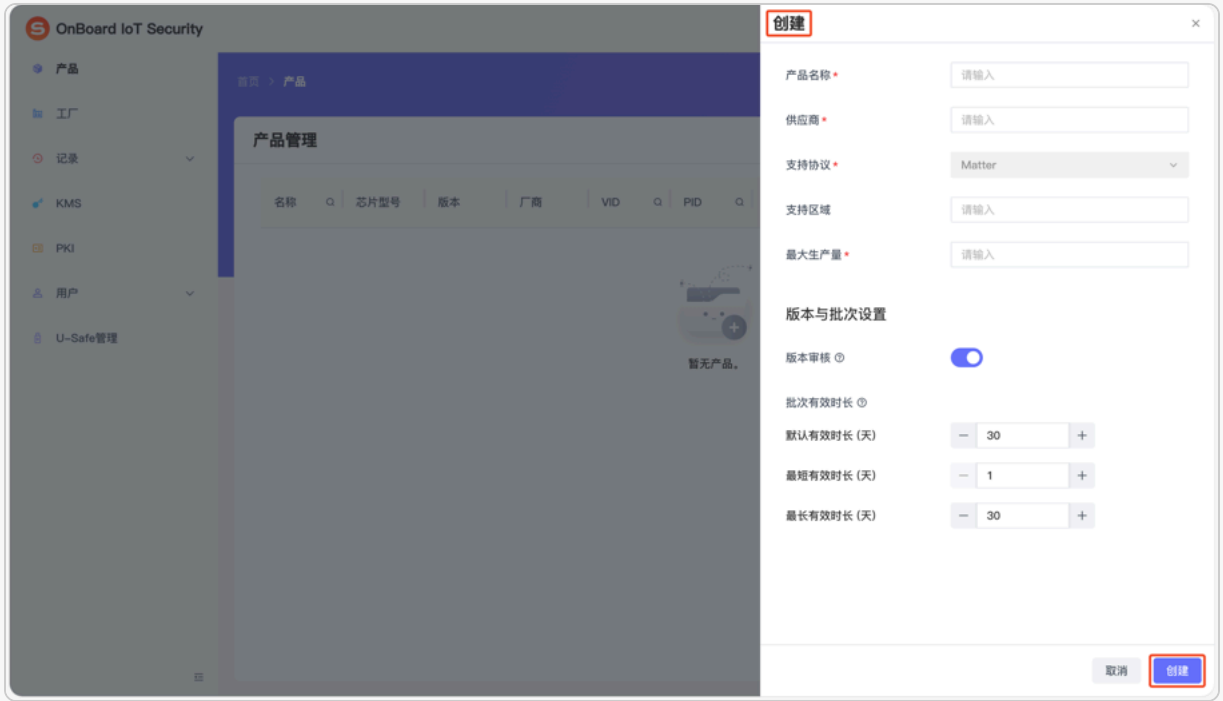
在开始之前，请确保你已拥有 System Manager 或 Product Manager 的 OBIS 角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击**创建**。



3. 在**创建**面板，选择并输入所需的产品基础信息。确认信息无误后，点击**创建**即可完成产品的创建。



产品基础信息说明：

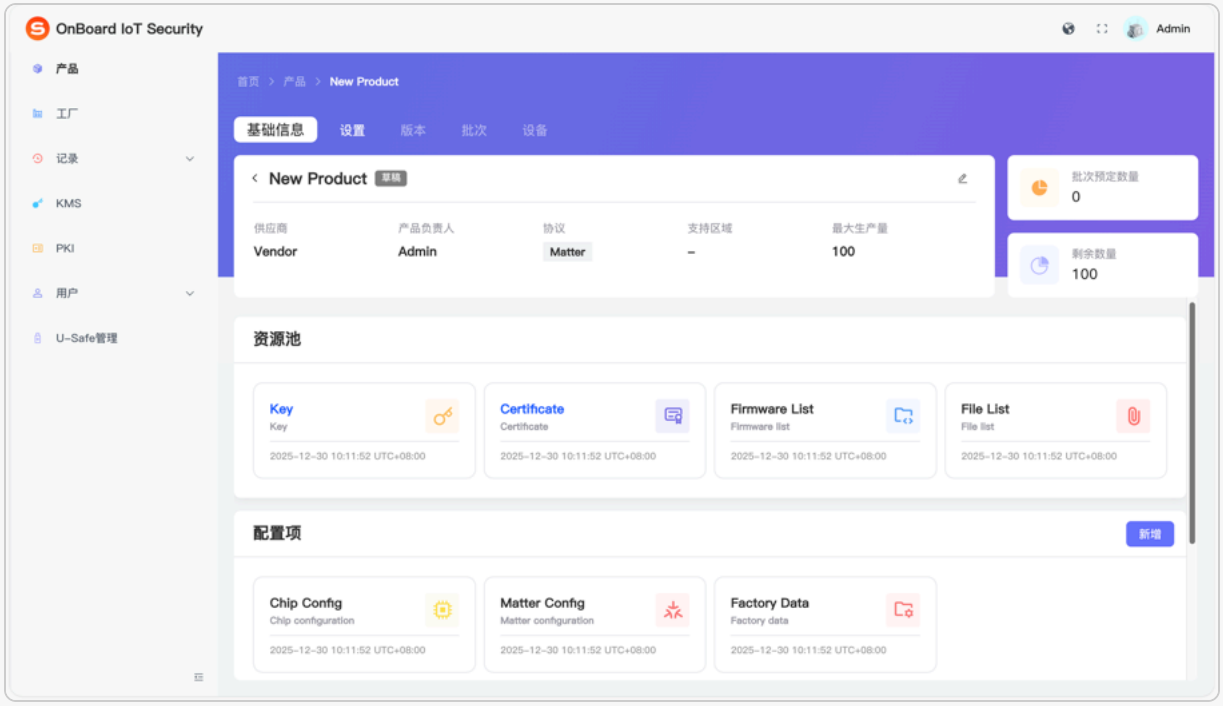
信息	是否必填	说明
产品名称	是	产品名称必须唯一。
供应商	是	根据租户自动填写供应商名称，保证信息准确，减少手动错误。如果需要更改，你也可以手动填写其他供应商名称。
支持协议	是	目前仅支持 Matter 协议。
支持区域	否	指定产品支持的地区。输入地区后，按**回车键**生成地区标签，可添加多个地区标签。 **注释**：创建版本时，地区可选范围将限定于这些地区。

信息	是否必填	说明
最大生产量	是	产品的最大生产量不能超过 999,999,999。批次数量的总和不能超过此数量。
版本与批次设置	否	◦ **版本审核**：此功能默认开启以确保所有用于设备生产的版本数据都经过确认。功能开启后，PROD 环境下的版本创建后将需要 Product Owner 或 Product Manager 进行审核，审核通过后才可用于创建批次。 ◦ **批次有效时长**：可设置**默认有效时长**，**最短有效时长**和**最长有效时长**，单位为天。创建批次时，到期日需在此设置范围内。

4. 产品创建后，页面会跳转至产品页面。

> **信息**
>
> 产品创建完成后，产品的初始状态为**草稿**状态。创建版本后，产品的状态将变更为**已发布**状态。更多信息请参考产品状态说明。

> **提示**
>
> 一般情况下，默认产品创建人为产品负责人。如果要更改产品负责人，可在产品页面点击编辑 图标，在**编辑**面板中选择新的产品负责人，并点击**保存**即可。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product
基础信息
设置
版本
批次
设备
New Product
草稿
供应商
Vendor
产品负责人
Admin
协议
Matter
支持区域
-
最大生产量
100
批次预定数量
0
剩余数量
100
资源池
Key
Key
2025-12-30 10:11:52 UTC+08:00
Certificate
Certificate
2025-12-30 10:11:52 UTC+08:00
Firmware List
Firmware list
2025-12-30 10:11:52 UTC+08:00
File List
File list
2025-12-30 10:11:52 UTC+08:00
配置项
新增
Chip Config
Chip configuration
2025-12-30 10:11:52 UTC+08:00
Matter Config
Matter configuration
2025-12-30 10:11:52 UTC+08:00
Factory Data
Factory data
2025-12-30 10:11:52 UTC+08:00

版本：v3.2.2

如何在 OnBoard™ IoT Security 中配置产品

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作从而配置产品物料，为后续创建版本做准备：

- 配置资源池
- 设定配置项
- 上传烧录站文件

前提条件

在开始之前，请确保：

- 你已拥有对应产品的产品角色权限：Product Owner，Product Manager 或 Resource Manager。
- 已完成证书和密钥的创建。

产品物料分类

每个产品均拥有独立的物料管理系统，用于统一管理各类产品物料。用户可上传固件，配置文件，烧录文件，关联密钥与证书模板，并预先配置产品参数。产品物料分为三个大类并包含不同的类型：

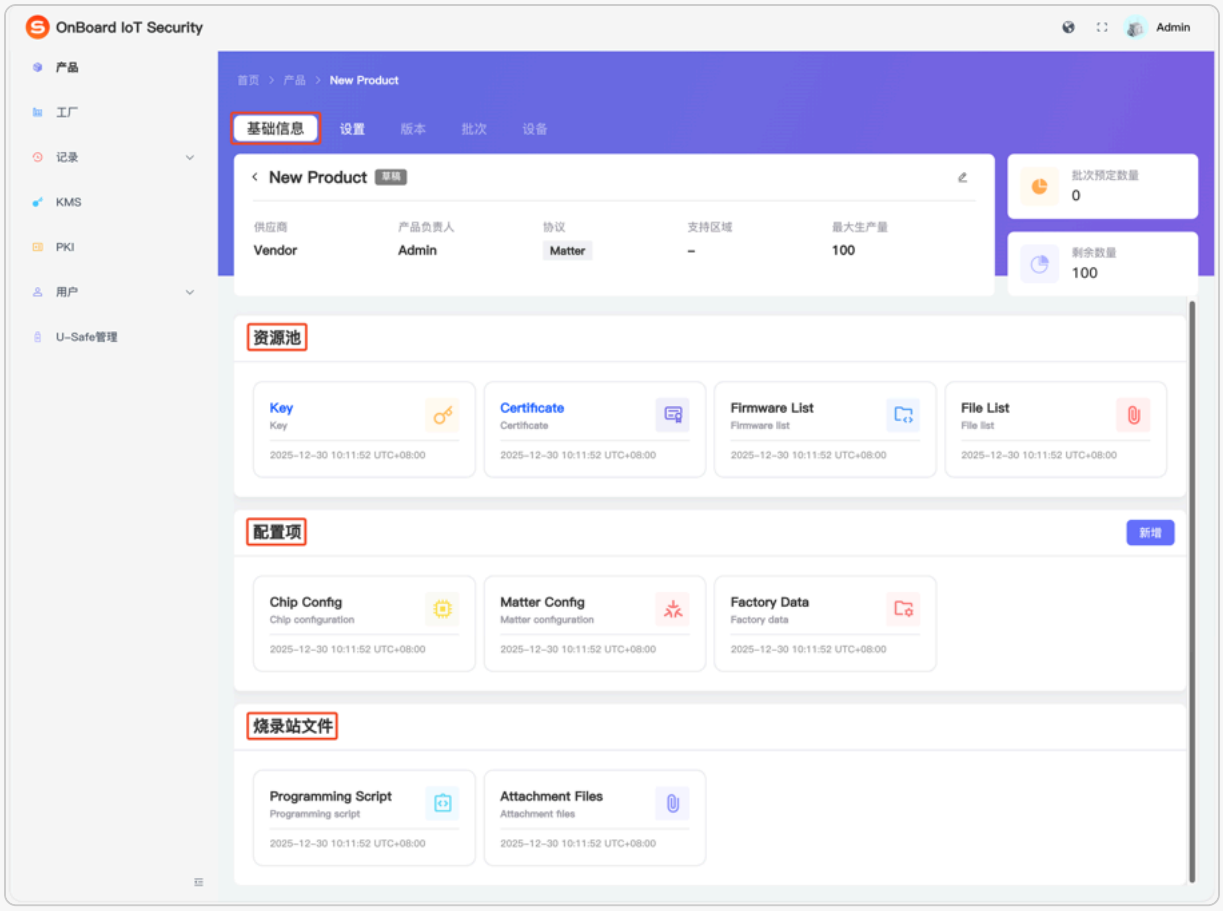
分类	物料类型	说明
资源池	Key	将 KMS 中的密钥添加到该产品。
	Certificate	将 PKI 中的 CA 模板添加到该产品。
	Firmware List	上传该产品需要烧录的固件文件，支持上传 **.bin**, **.hex** 和 **.s37** 格式的固件。
	File List	上传除固件外，创建版本所需的其他文件（支持上传任意格式的文件）。
配置项	Chip Config	选择并设定产品所使用的芯片型号和安全选项。
	Matter Config	选择并设定 Matter 协议相关的设置。
	Factory Data	设定工厂的相关信息。
	Custom certificate configuration	设定自定义证书相关的设置。
	Google certificate configuration	设定 Google Cast 协议相关的设置。
烧录站文件	Programming Script	上传设备烧录时（支持上传任意格式的烧录脚本）使用的脚本文件。
	Attachment Files	上传除烧录脚本外，创建批次所需的其他文件（支持上传任意格式的文件）。

操作步骤

1. 登录 OBIS。

2. 点击左侧菜单中的产品并选择对应的产品，然后在产品页面的**资源池**，**配置项**和**烧录站文件**区域中进行配置。

   > **信息**
   >
   > - 资源池和配置项区域的配置用于创建版本。在**资源池**和**配置项**区域，配置产品必要的资源与配置项（**Certificate**，**Firmware List**，**Chip Config**，**Matter Config**，**Factory Data**）。未完成必要配置时，无法创建版本，产品处于**草稿**状态。
   > - 烧录站文件区域的文件用于创建批次。在**烧录站文件**区域，上传烧录必要的文件（**Programming Script**）。或者你也可以后续在 Programming Station 客户端中为特定批次选择烧录脚本。



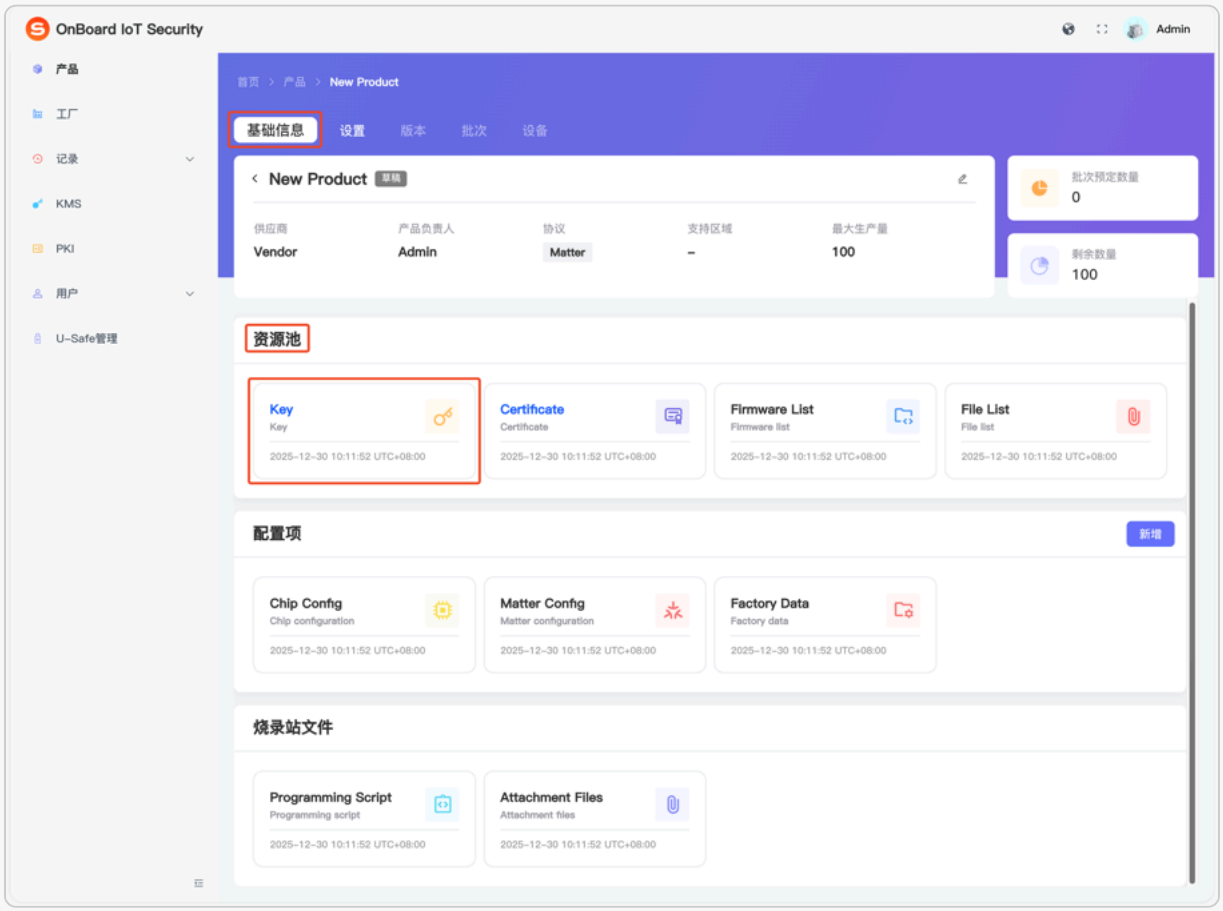
第一步：配置资源池

> **提示**
>
> **资源池**区域的配置用于创建版本。在配置 Firmware List 和 File List 时上传的文件，之后可以随时在对应资源的配置页面进行下载。

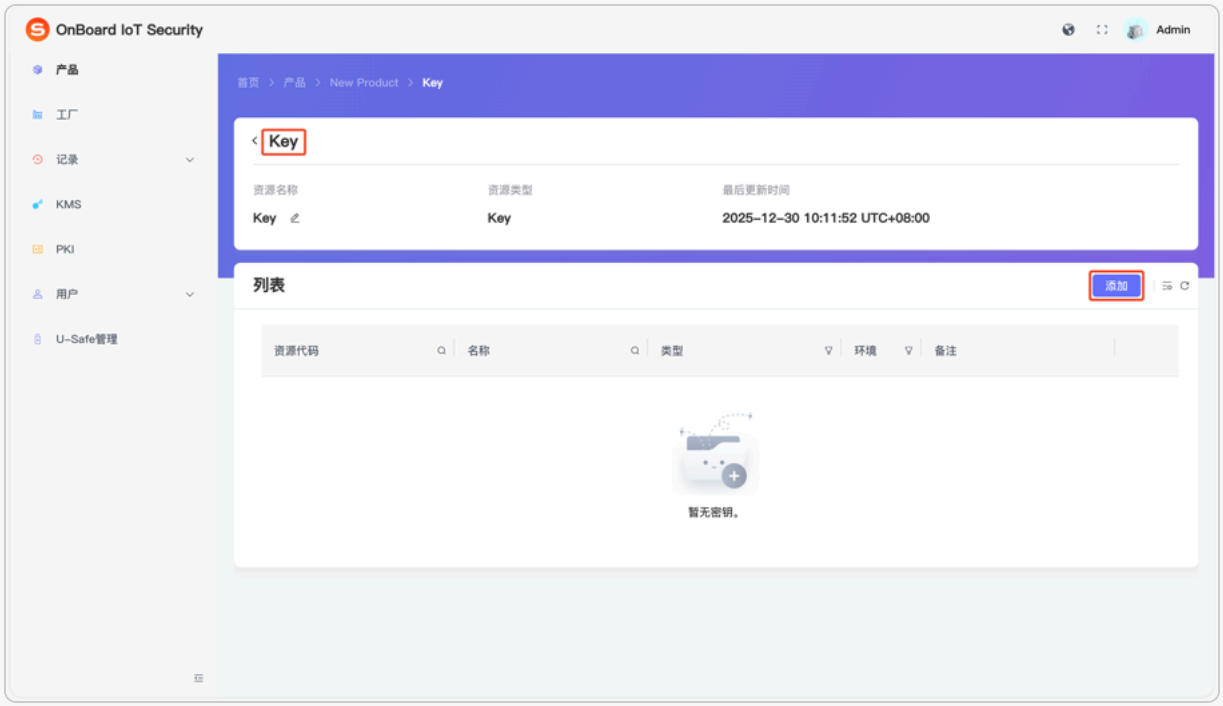
Key（可选）

> **提示**
>
> 配置 **Key** 的目的是将 KMS 中的密钥添加到该产品，添加的密钥可在配置 Chip Config 时进行选择。
>
> 密钥可以在 KMS 中创建，更多信息请参考创建密钥的步骤。

1. 在产品页面，点击**资源池**区域的 **Key**，进入 **Key** 页面。



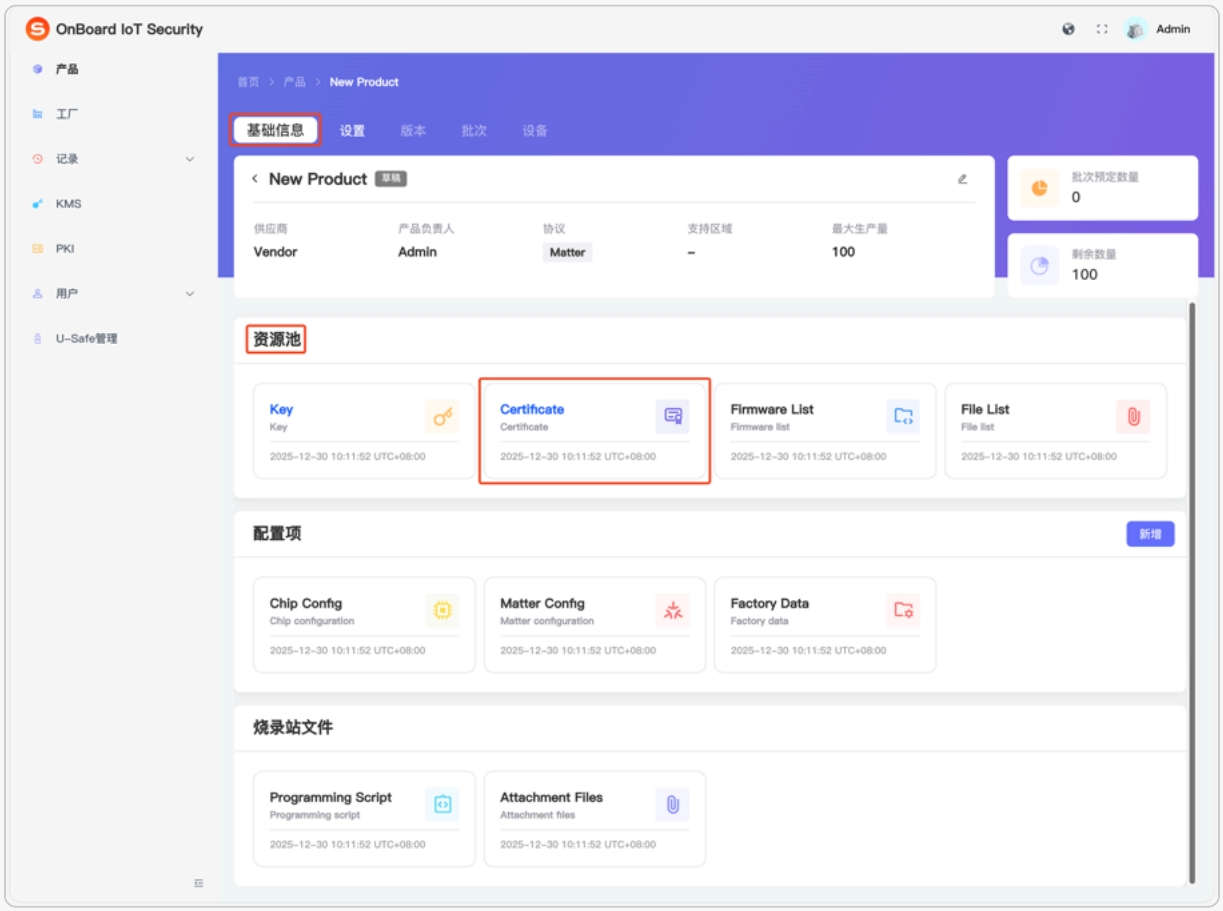
2. 在 **Key** 页面，点击**添加**，选择并添加密钥。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product > Key
Key
资源名称
Key
资源类型
Key
最后更新时间
2025-12-30 10:11:52 UTC+08:00
列表
添加
资源代码
名称
类型
环境
备注
暂无密钥。

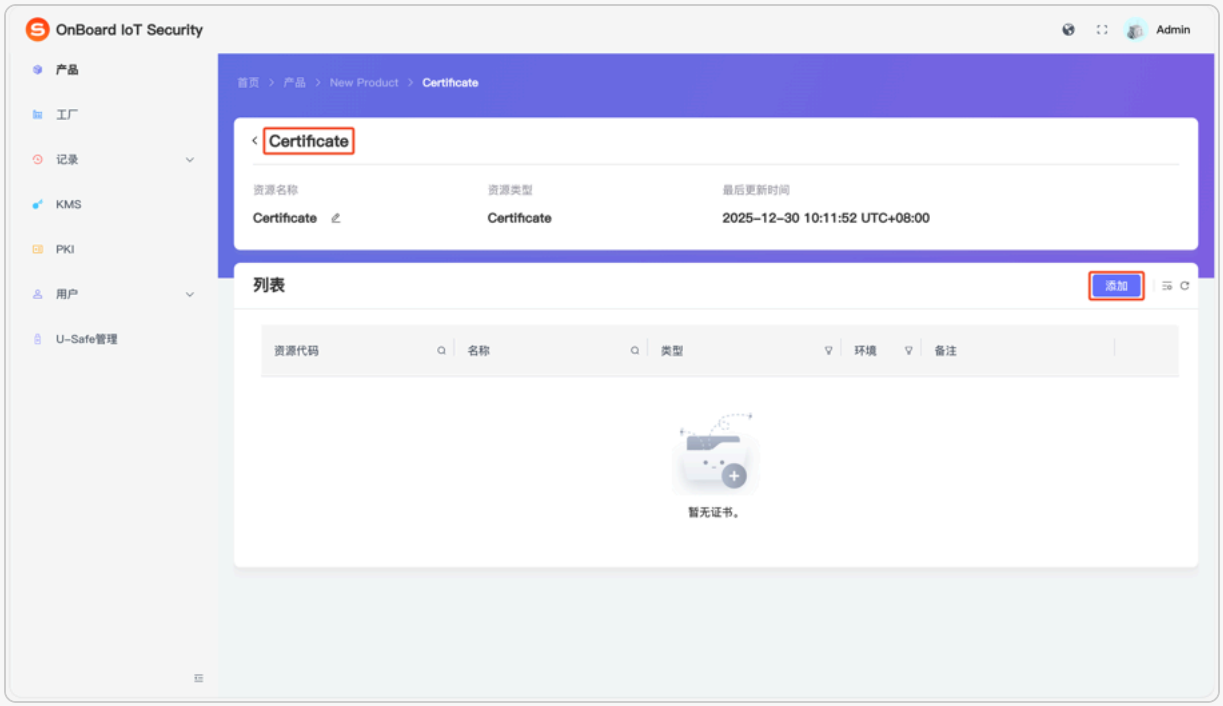
Certificate（必要）

> **提示**
>
> 配置 **Certificate** 的目的是将 PKI 中的 CA 模板（终端实体模板，DAC 模板或 Device 模板）添加到该产品，添加的 CA 模板可在配置 Matter Config 时进行选择。
>
> CA 模板可以在 PKI 中创建，更多信息请参考创建证书的步骤。

1. 在产品页面，点击资源池区域的 **Certificate**，进入 **Certificate** 页面。



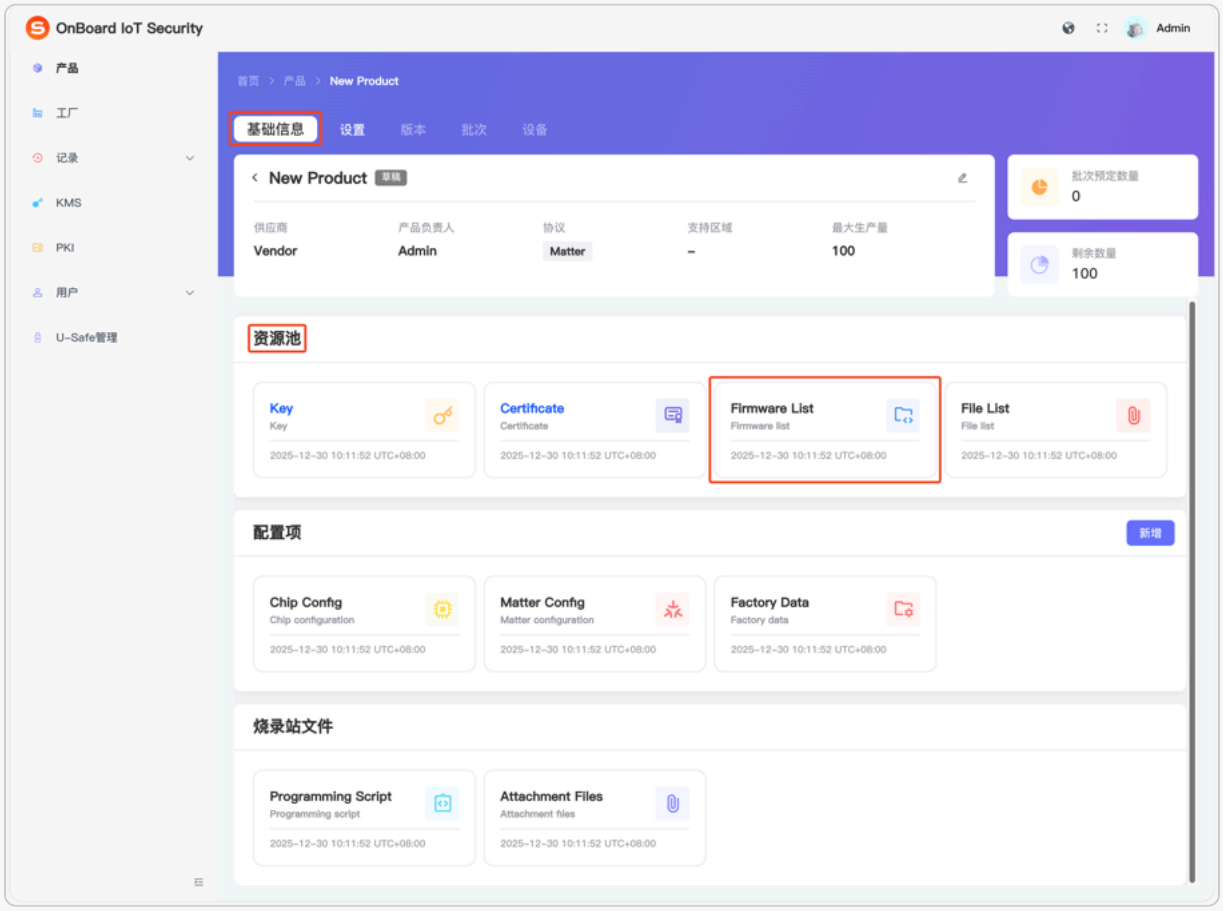
2. 在 **Certificate** 页面，点击**添加**，选择并添加 CA 模板。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product > Certificate
Certificate
资源名称
Certificate
资源类型
Certificate
最后更新时间
2025-12-30 10:11:52 UTC+08:00
列表
添加
资源代码
名称
类型
环境
备注
暂无证书。

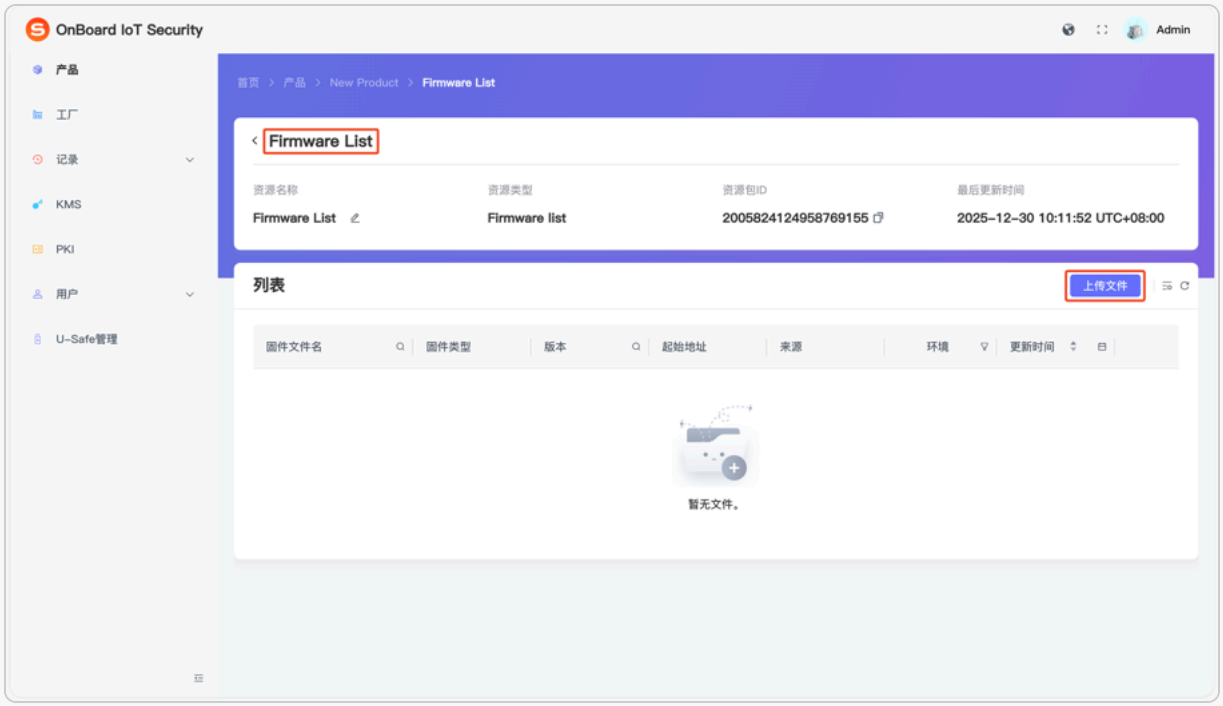
Firmware List（必要）

> **提示**
>
> 配置 **Firmware List** 的目的是上传该产品需要烧录的固件文件，支持上传 **.bin**, **.hex** 和 **.s37** 格式的固件。 上传的固件可在创建版本时进行选择。

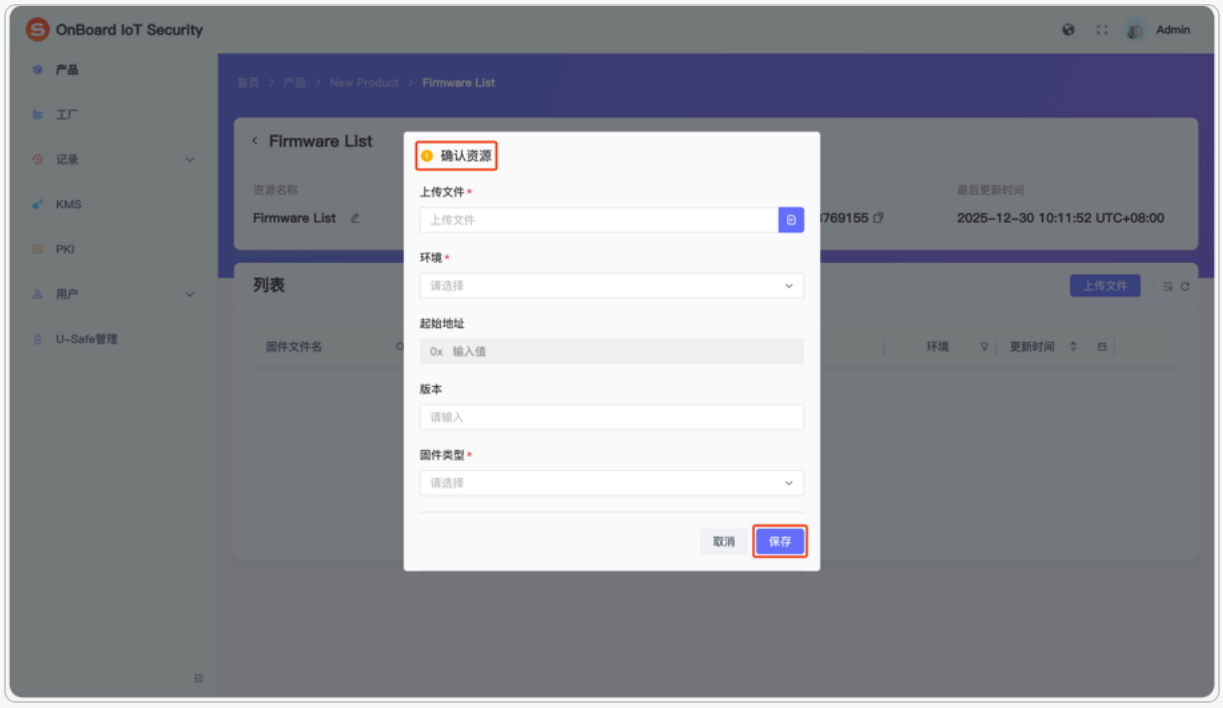
1. 在产品页面，点击资源池区域的 **Firmware List**，进入 **Firmware List** 页面。



2. 在 **Firmware List** 页面，点击**上传文件**。



3. 在**确认资源**窗口，输入固件相关信息。



固件信息说明：

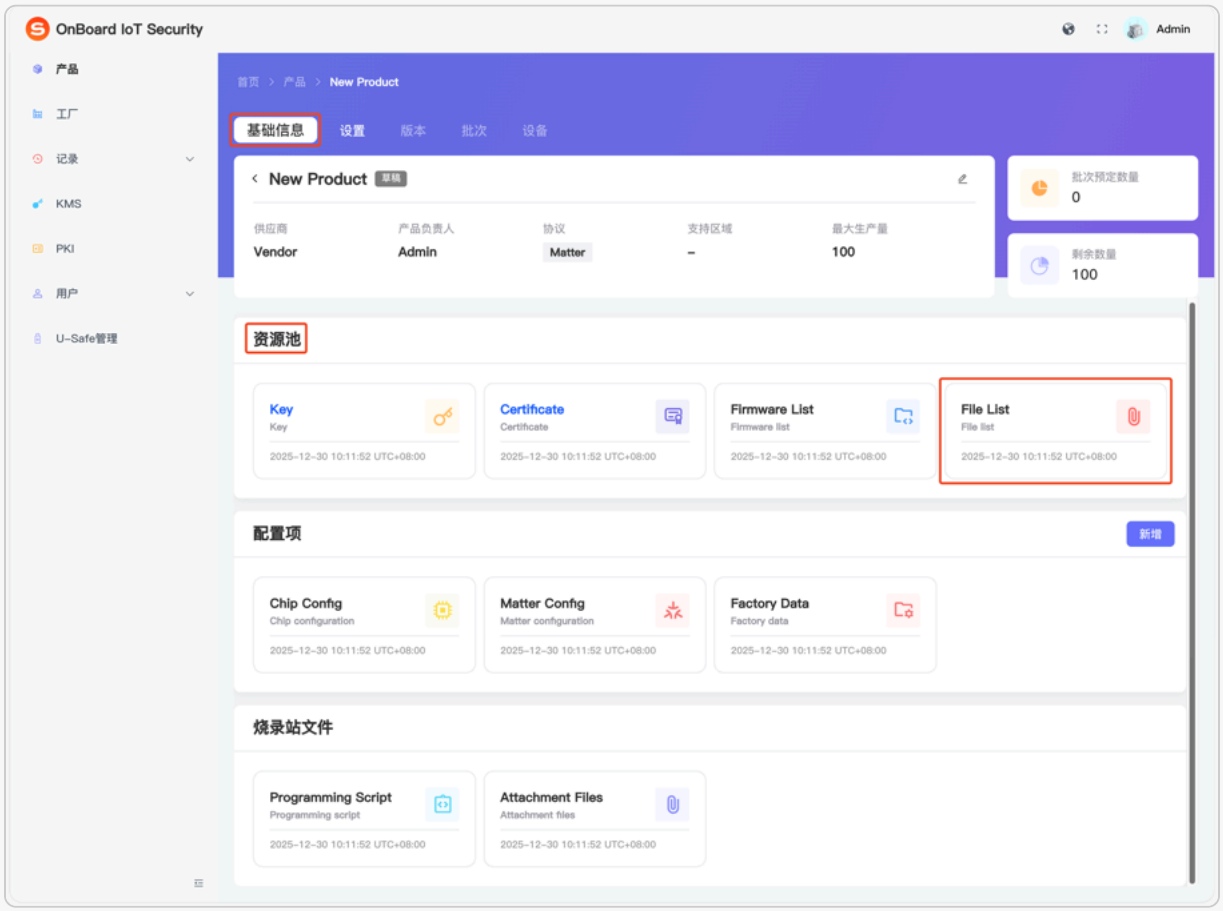
信息	是否必填	说明
上传文件	是	通过选择本地的固件文件进行上传。
环境	是	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。
起始地址	否	固件的起始地址。
版本	否	标识固件版本。
固件类型	是	支持以下两种类型： ◦ **Application**：用于实现设备功能，依赖 Bootloader 提供的环境。 ◦ **Bootloader**：用于硬件初始化，固件加载与更新。

4. 点击**保存**，完成 Firmware List 的配置。

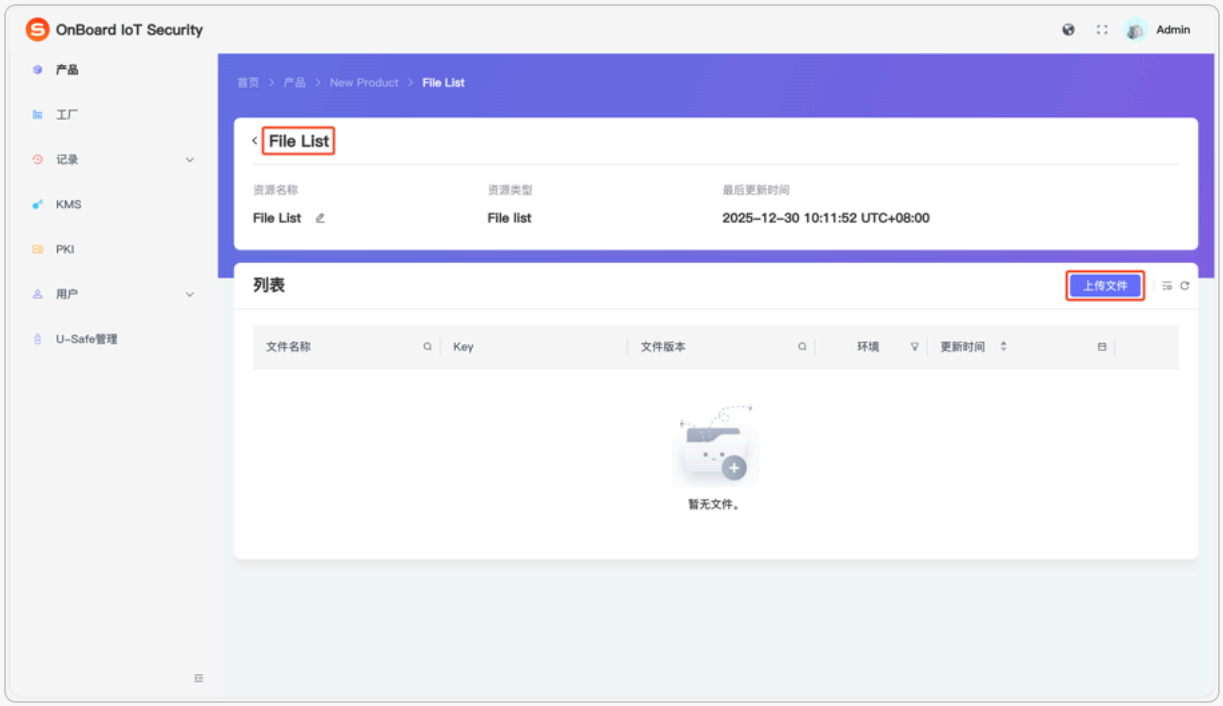
File List（可选）

> **提示**
>
> 配置 **File List** 的目的是上传除固件外，创建版本所需的其他文件（支持上传任意格式的文件）。
>
> 上传的文件可在创建版本时进行选择。

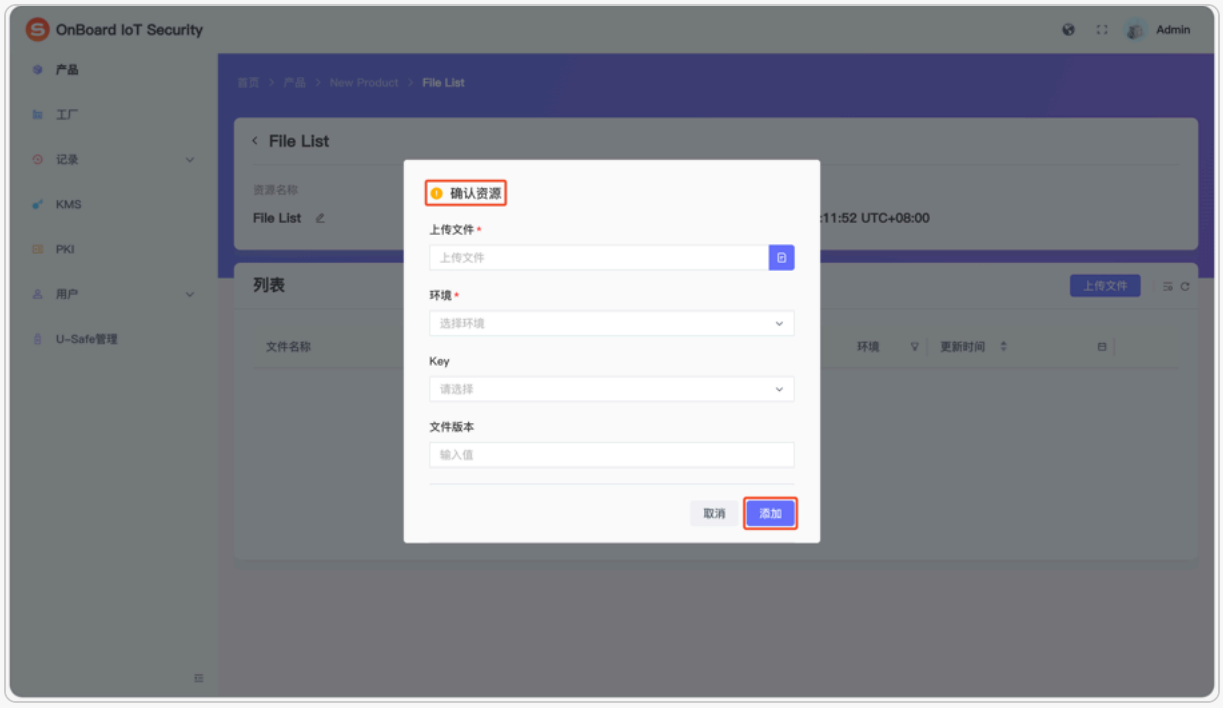
1. 在产品页面，点击**资源池**区域的 **File List**，进入 **File List** 页面。



2. 在 **File List** 页面，点击**上传文件**。



3. 在**确认资源**窗口，选择并输入相关信息。



File List 信息说明：

信息	是否必填	说明
上传文件	是	通过选择本地的文件进行上传。
环境	是	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。
Key	否	为文件分配 Key。后期烧录时，烧录脚本可通过 Key 调取并使用对应文件。 **注释**：一个文件可分配多个不重复的 Key。同时，同一个 key 可被分配到多个文件。可新增 Key 或从已有 Key 中进行选择。
文件版本	否	标识文件版本并用于跟踪文件变更。

4. 点击**添加**，即可完成 File List 的配置。

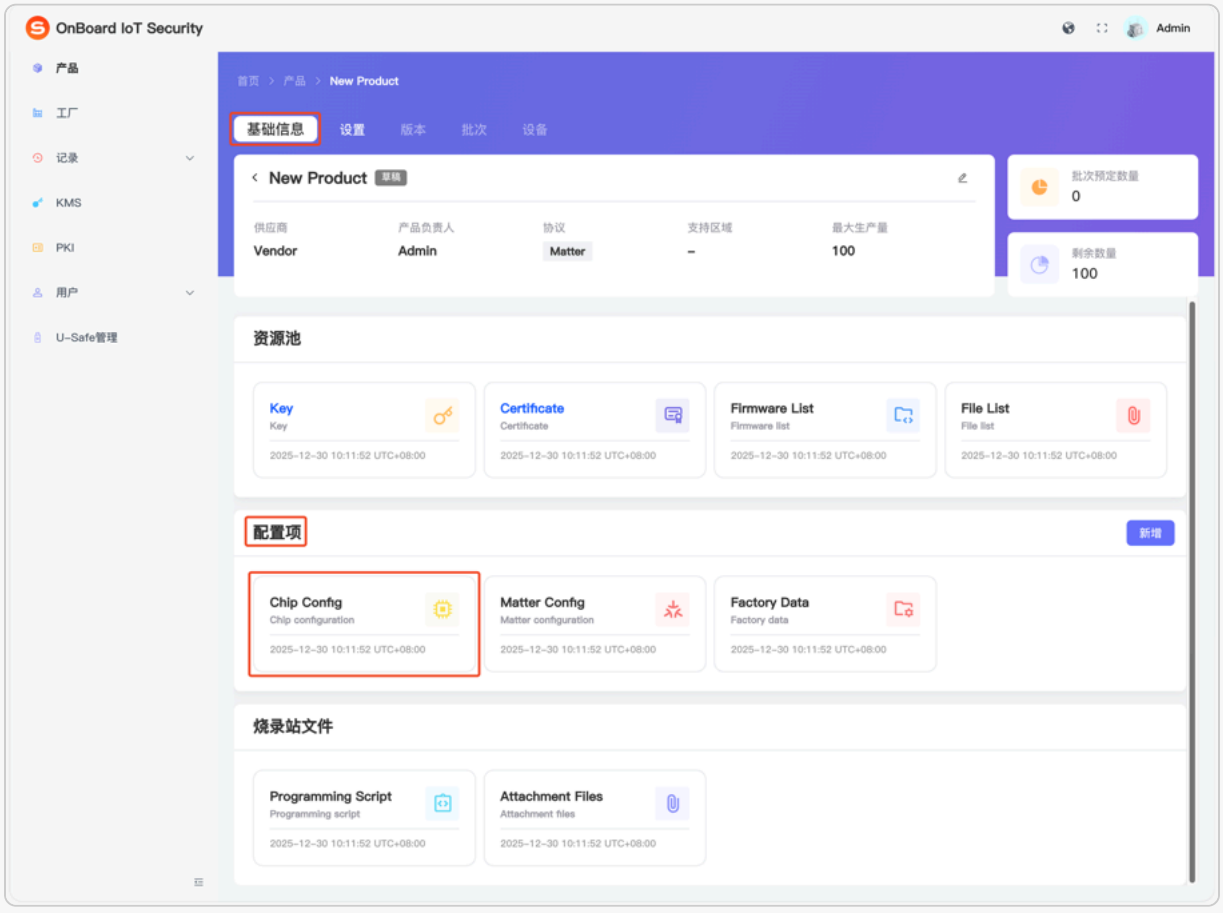
第二步：设定配置项

> **提示**
>
> **配置项**区域的配置用于创建版本。在配置 Matter Config 和 Factory Data 时上传的文件，之后可以随时在对应资源的配置页面进行下载。

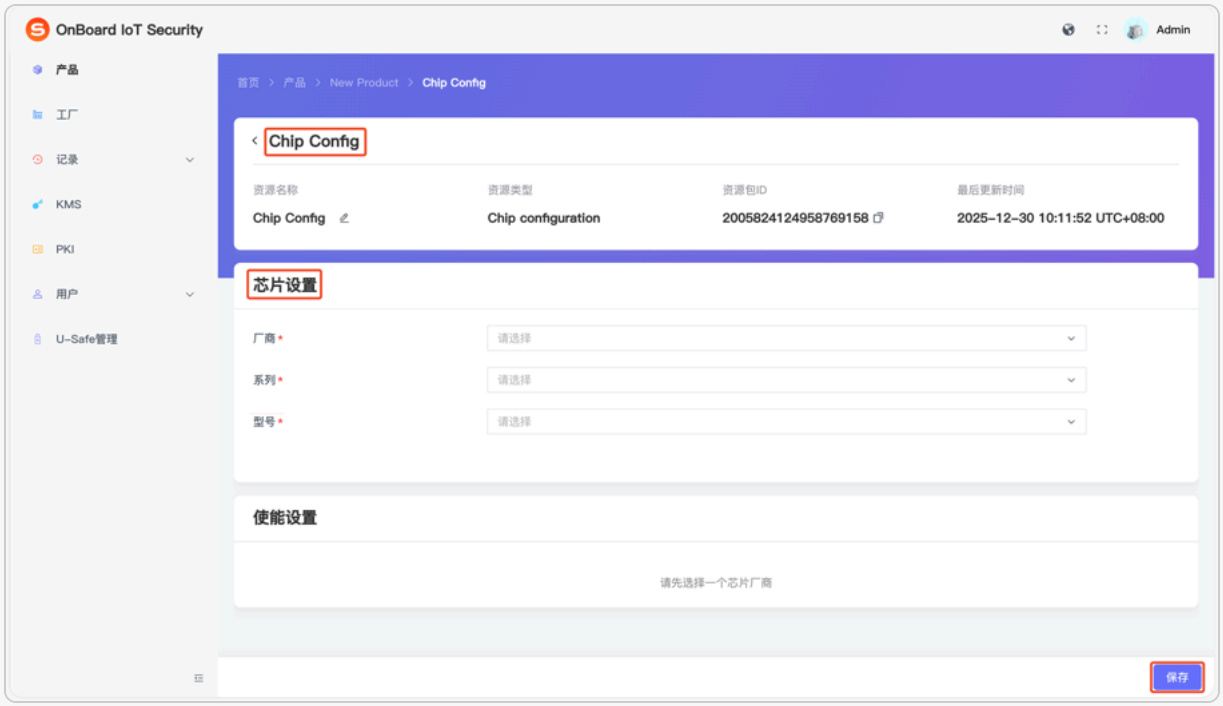
Chip Config（必要）

> **提示**
>
> 配置 **Chip Config** 的目的是选择并设定产品所使用的芯片型号和安全选项。不同的芯片型号在使能设置中会有对应的安全选项，你可以根据实际需求进行配置。
>
> 具体列表请参考当前支持的芯片设置。后续将根据需求新增支持的芯片厂商，系列和型号。

1. 在产品页面，点击**配置项**区域的 **Chip Config**，进入 **Chip Config** 页面。



2. 在 **Chip Config** 页面，找到**芯片设置**区域，选择对应的芯片**厂商**，**系列**和**型号**，然后点击**保存**。



当前支持的芯片设置：

厂商	系列	型号
Beken	BK7238	BK7238
NXP	s32k1	s32k144
Qorvo	QPG6200	QPG6200

厂商	系列	型号
Silicon Labs	EFR32MG24	◦ EFR32MG24B210F1536IM48 ◦ EFR32MG24B020F1536IM48 ◦ EFR32MG24A110F1024IM48 ◦ EFR32MG24A020F1536IM48 ◦ EFR32MG24A420F1536IM48 ◦ EFR32MG24B020F1536IM40 ◦ EFR32MG24A010F1024IM40 ◦ EFR32MG24A020F1536IM40 ◦ EFR32MG24A420F1536IM40 ◦ EFR32MG24B010F1024IM48 ◦ EFR32MG24A010F1024IM48 ◦ EFR32MG24B110F1536IM48 ◦ EFR32MG24A021F1024IM40 ◦ EFR32MG24B010F1536IM40 ◦ EFR32MG24B220F1536IM48 ◦ EFR32MG24A020F1024IM40 ◦ EFR32MG24A010F1536IM40 ◦ EFR32MG24B010F1536IM48 ◦ EFR32MG24A410F1536IM48 ◦ EFR32MG24A010F1536IM48 ◦ EFR32MG24B120F1536IM48 ◦ EFR32MG24A410F1536IM40 ◦ EFR32MG24B310F1536IM48 ◦ EFR32MG24B020F1024IM48

厂商	系列	型号
		◦ EFR32MG24A020F1024IM48
	SIMG301	SiMG301M113WIH
Telink	TLSR92x	TLSR9268J

3. 在**使能设置**中，你可根据实际需求在生产（PROD）或测试（TEST）环境勾选并启用**安全启动**和**安全调试**功能。TEST 环境下，**安全启动**和**安全调试**功能的启用可以在版本中进行更改。

   > **提示**
   >
   > 在创建批次时，如果需要启用重新打开安全调用功能实现芯片的再次烧录，必须启用**安全调试**功能。

   使能设置项说明（不同的芯片设置会显示不同的设置项）：

设置项	说明
安全启动	防止未授权软件加载的安全保护机制。
安全调试	防止未授权调试的安全保护机制。
Secure Boot Key	选择在配置 **Key** 时添加的密钥。用于验证固件的密钥，确保设备的安全启动。
Secure Boot Key Hash	Secure Boot Key 的哈希值。

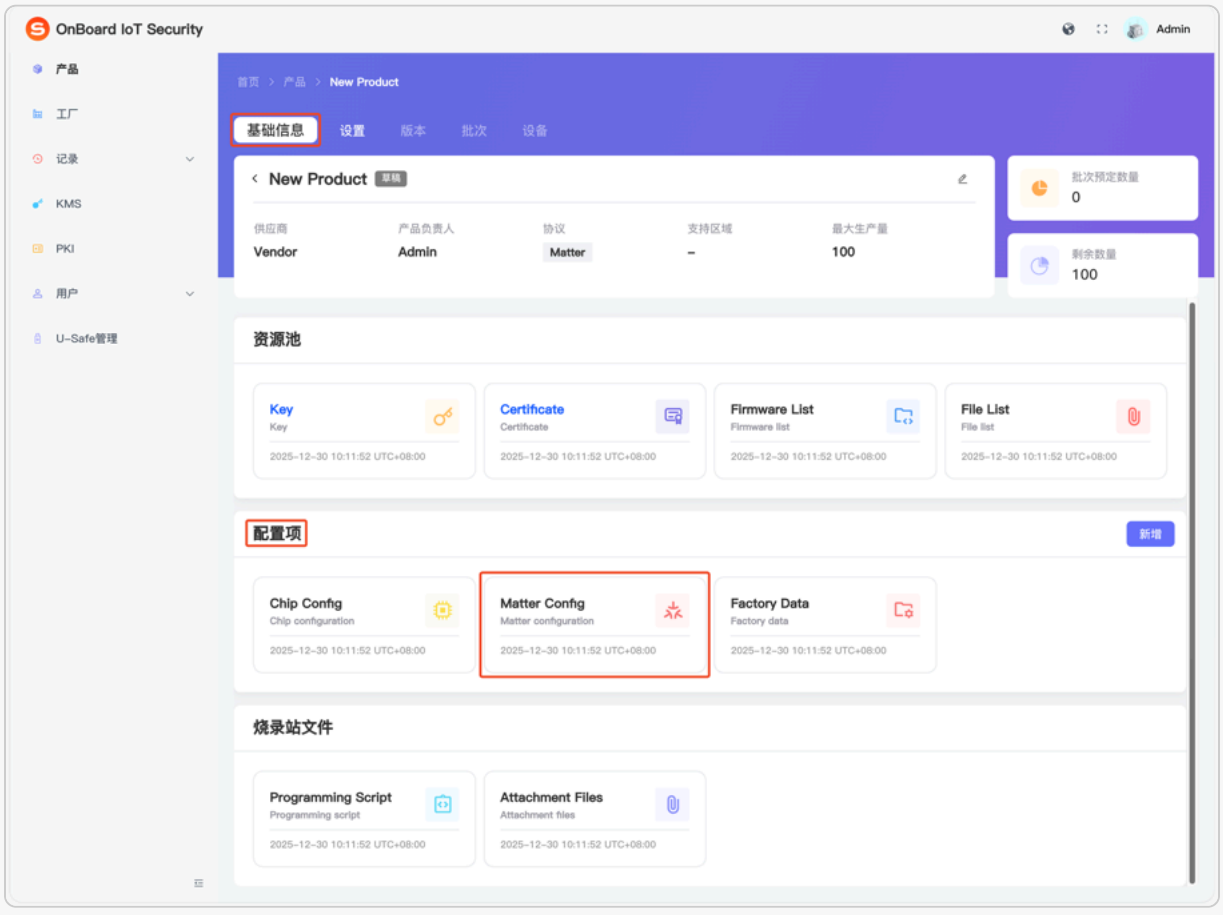
设置项	说明
Secure Debug Key	选择在配置 **Key** 时添加的密钥。用于授权调试访问。 **注释**：对于 Telink 芯片，该密钥将由 MCU Boot Key 与 Debug Plaintext 计算得出。
MCU Boot Key	选择在配置 **Key** 时添加的密钥。用于设备身份认证和数据加密。
Debug Plaintext	未加密的调试数据。
DAC Partition Address	设定 DAC 证书下载到闪存的储存地址。

4. 点击**保存**，即可完成 Chip Config 的配置。

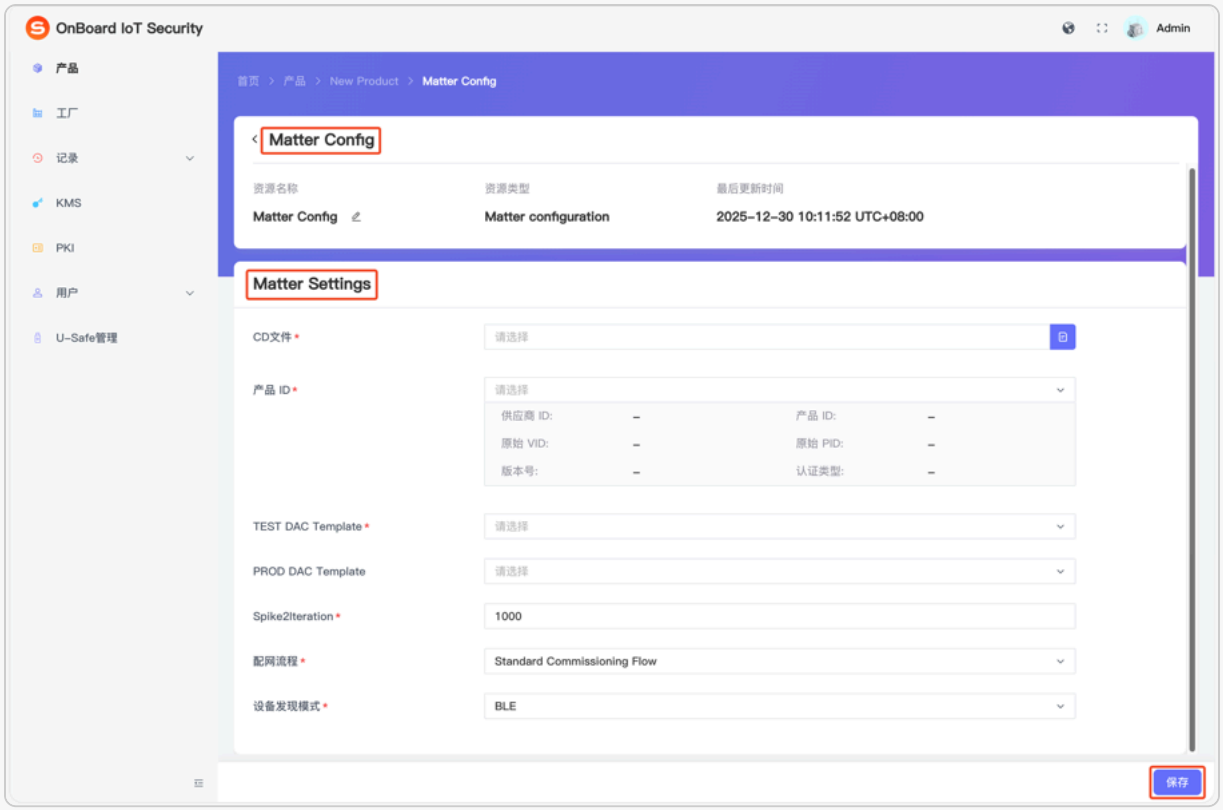
Matter Config（必要）

> **提示**
>
> 配置 **Matter Config** 的目的是选择并设定 Matter 协议相关的设置。
>
> 在此步骤中，需要选择并上传 CD 文件 ，并选择在配置 **Certificate** 时添加的 DAC 模板。请确保 CD 文件与 DAC 模板中的 VID （供应商 ID）和 PID （产品 ID）保持一致。
>
> 在后续创建版本时，你可以根据实际需求更换其他 CD 文件。

1. 在产品页面，点击**配置项**区域的 **Matter Config**，进入**Matter Config** 页面。



2. 在 **Matter Config** 页面，选择或输入 **Matter Setting** 区域的相关信息。



Matter Settings 信息说明：

信息	是否必填	说明
CD 文件	是	上传时需注意： ◦ CD 文件中的 VID 和 PID 必须与 DAC 模板中的一致。 ◦ 如果 CD 文件包含多个 PID，则其中至少有一个必须与 DAC 模板中的 PID 相同。 **注释**：仅支持上传 **.der** 格式的 CD 文件。

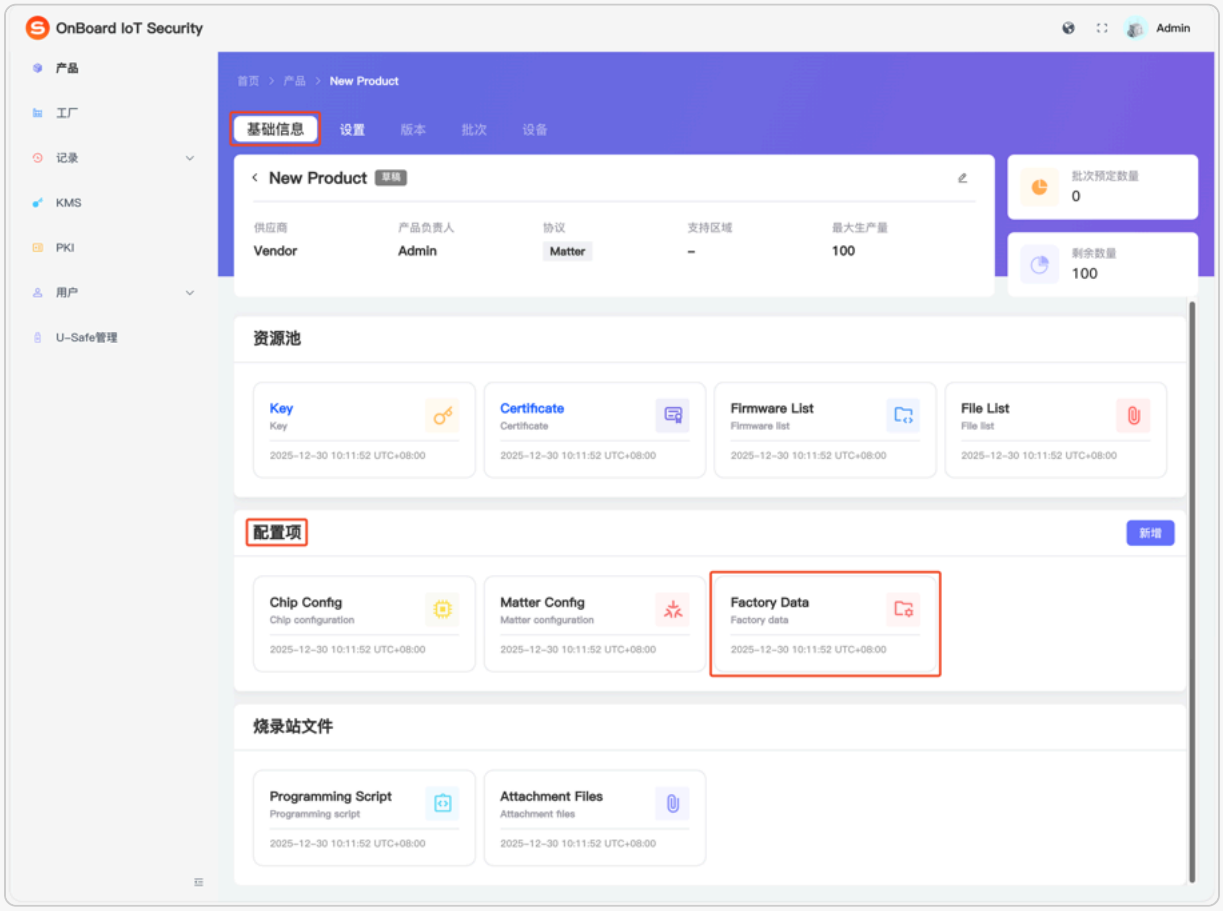
信息	是否必填	说明
产品 ID	是	上传 CD 文件后会出现对应的 PID。
TEST DAC Template	是	选择在配置 **Certificate** 时 TEST 环境下的 DAC 模板，DAC 模板中的 VID 和 PID 必须与CD 文件中的一致。
PROD DAC Template	否	选择在配置 **Certificate** 时 PROD 环境下的 DAC 模板，DAC 模板中的 VID 和 PID 必须与 CD 文件中的一致。
Spike2Iteration	是	Matter 配网过程中 SPAKE2+ 协议的一个参数，保障配网过程的安全性。
配网流程	是	将设备连网配置的过程，适配不同场景： ◦ Standard Commissioning Flow ◦ User-Intent Commissioning Flow ◦ Custom Commissioning Flow
设备发现模式	是	支持以下两种发现模式： ◦ BLE：蓝牙低功耗。 ◦ On IP network：通过 IP 网络发现。

3. 点击保存，即可完成 Matter Config 的配置。

Factory Data（必要）

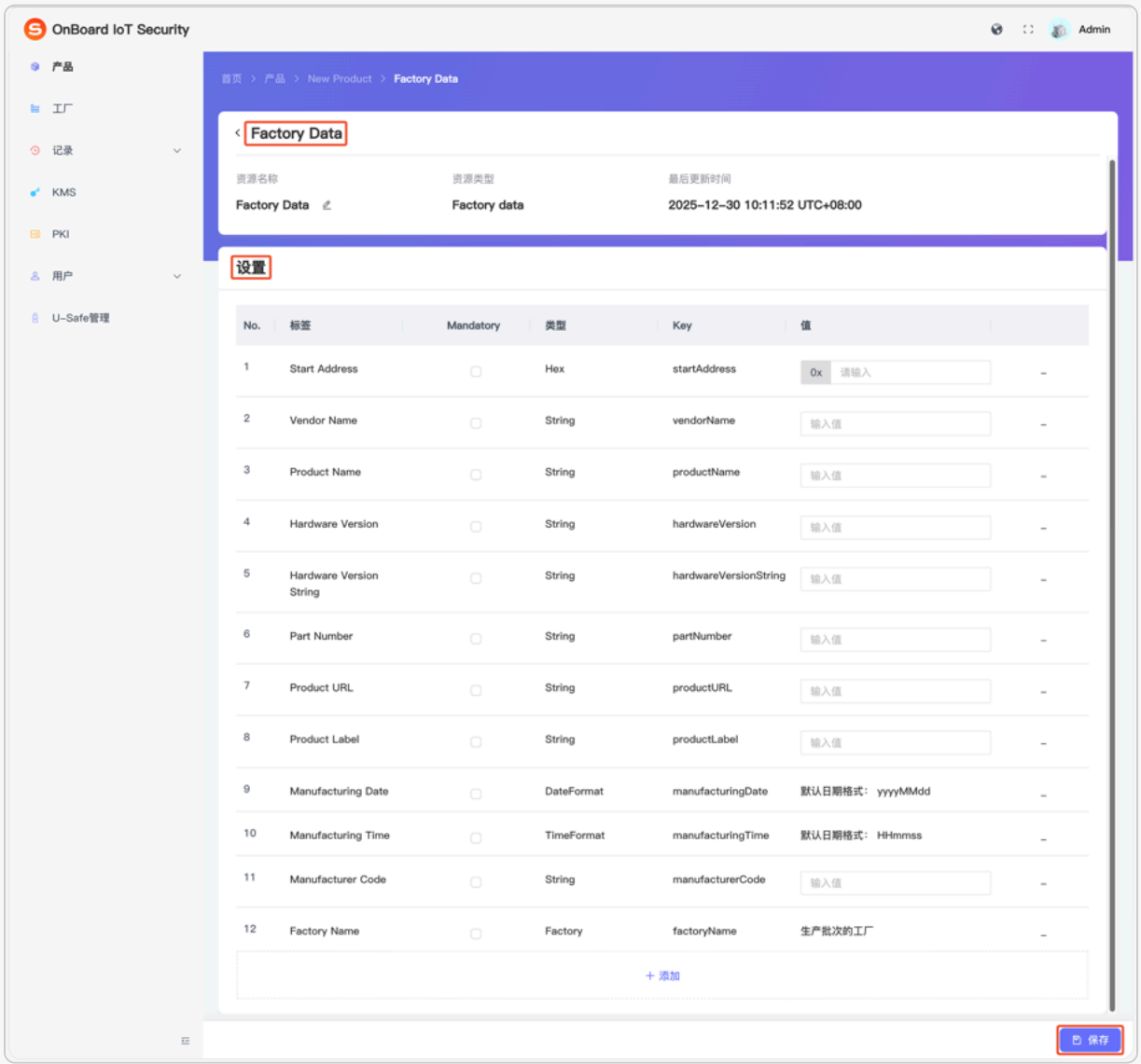
> **提示**
>
> 配置 **Factory Data** 的目的是设定工厂的相关信息，支持 Matter 规范所需的全部字段。在后续创建版本时，你可以根据实际需求对 Factory Data 进行修改。

1. 在产品页面，点击**配置项**区域的 **Factory Data**，进入 **Factory Data** 页面。



2. 在 **Factory Data** 页面，找到**设置**区域，输入产品的 Factory Data 信息（值可为空）。

> **备注**
>
> 如果勾选某个工厂数据 **Mandatory** 列的复选框，成员在创建版本时必须输入该工厂数据的值。



Factory Data 信息说明：

> **提示**
>
> 支持的数据类型包括：String，Number，Hex，Factory，DateFormat，TimeFormat 和 File。

信息	类型	说明
Start Address	Hex	起始地址。
Vendor Name	String	供应商名称。
Product Name	String	产品名称。
Hardware Version	String	硬件版本。
Hardware Version String	String	以字符串形式呈现的硬件版本信息，可能包含更详细的版本描述，例如，HW_V1.0_20240101。
Part Number	String	型号编号。
Product URL	String	产品链接。
Product Label	String	产品标签。
Manufacturing Date	DateFormat	生产日期，默认格式为 yyyyMMdd。

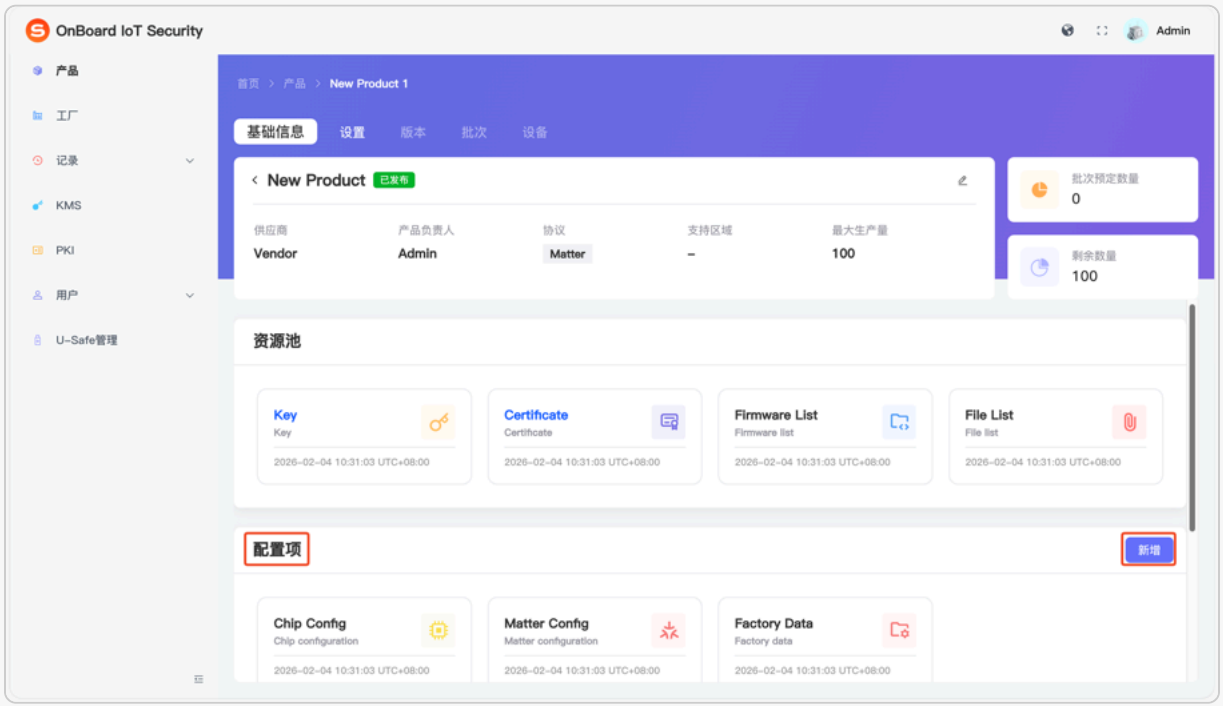
信息	类型	说明
Manufacturing Time	TimeFormat	生产时间，默认格式为 HHmmss
Manufacturing Code	String	生产代码。
Factory Name	Factory	生产批次的工厂名称。

3. 点击**保存**，即可完成 Factory Data 的配置。

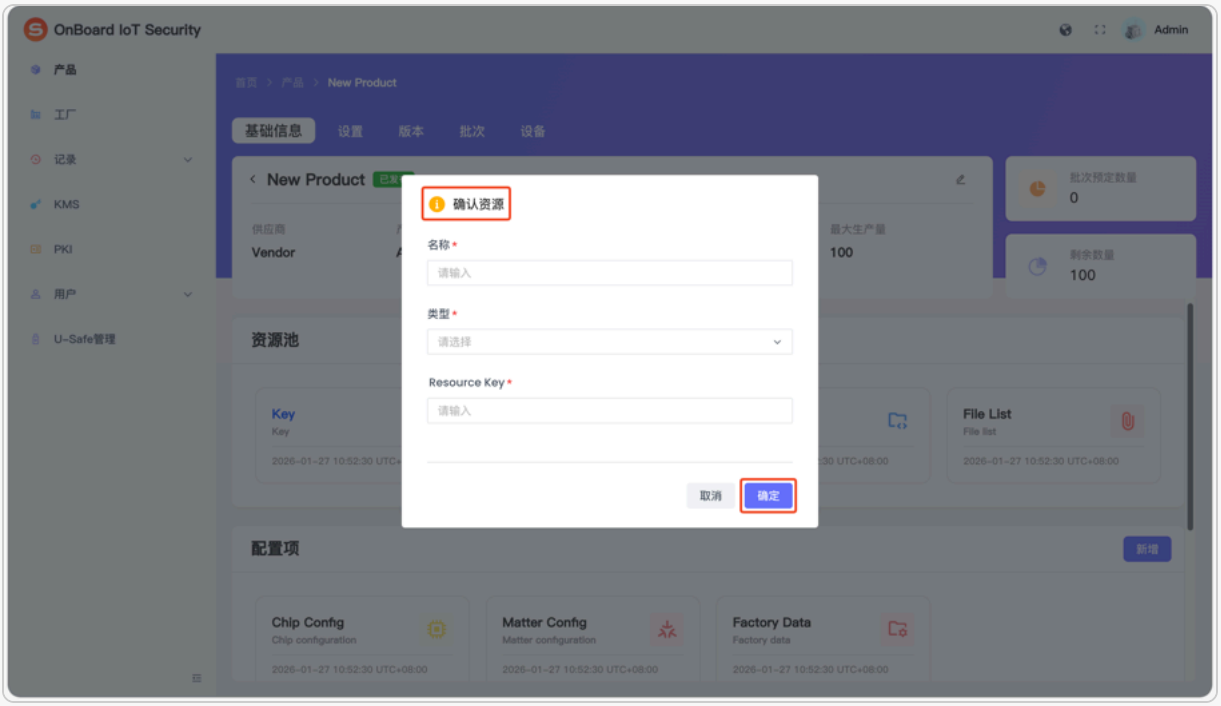
Custom certificate configuration（可选）

> **提示**
>
> 配置 **Custom certificate configuration** 的目的是设定自定义证书相关的设置。

1. 在产品页面，点击**配置项**区域的**新增**。



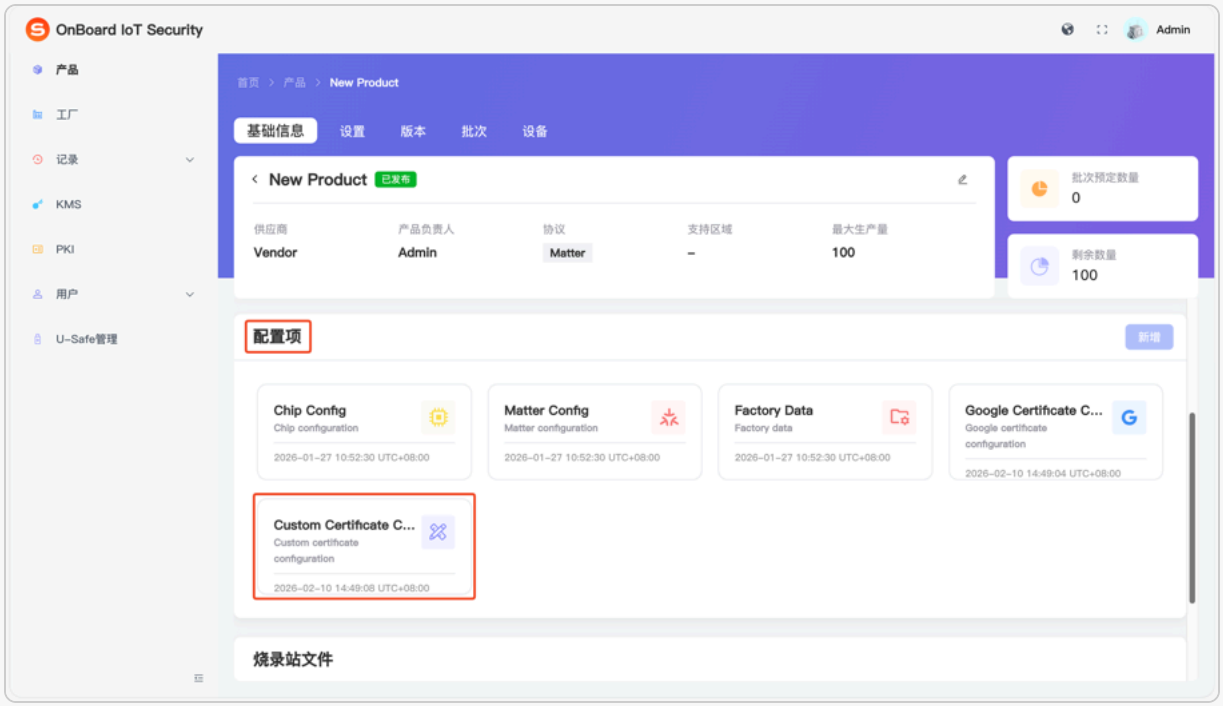
2. 在**确认资源**窗口，设置**名称**，**类型**和 **Resource key**。



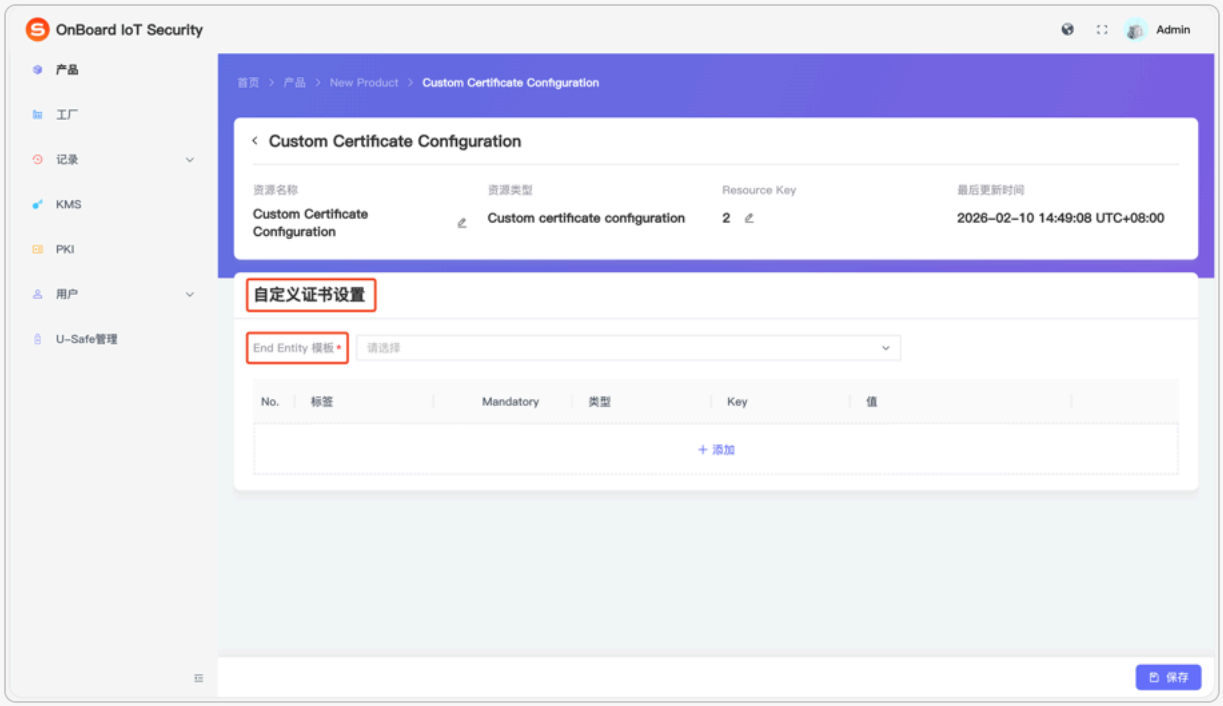
资源信息说明：

信息	说明
名称	资源名称。
类型	资源类型。
Resource Key	Resource key 必须唯一。

3. 在**配置项**区域点击刚新增的配置项。



4. 在**自定义证书设置**区域, 选择在配置 **Certificate** 时添加的终端实体模板。

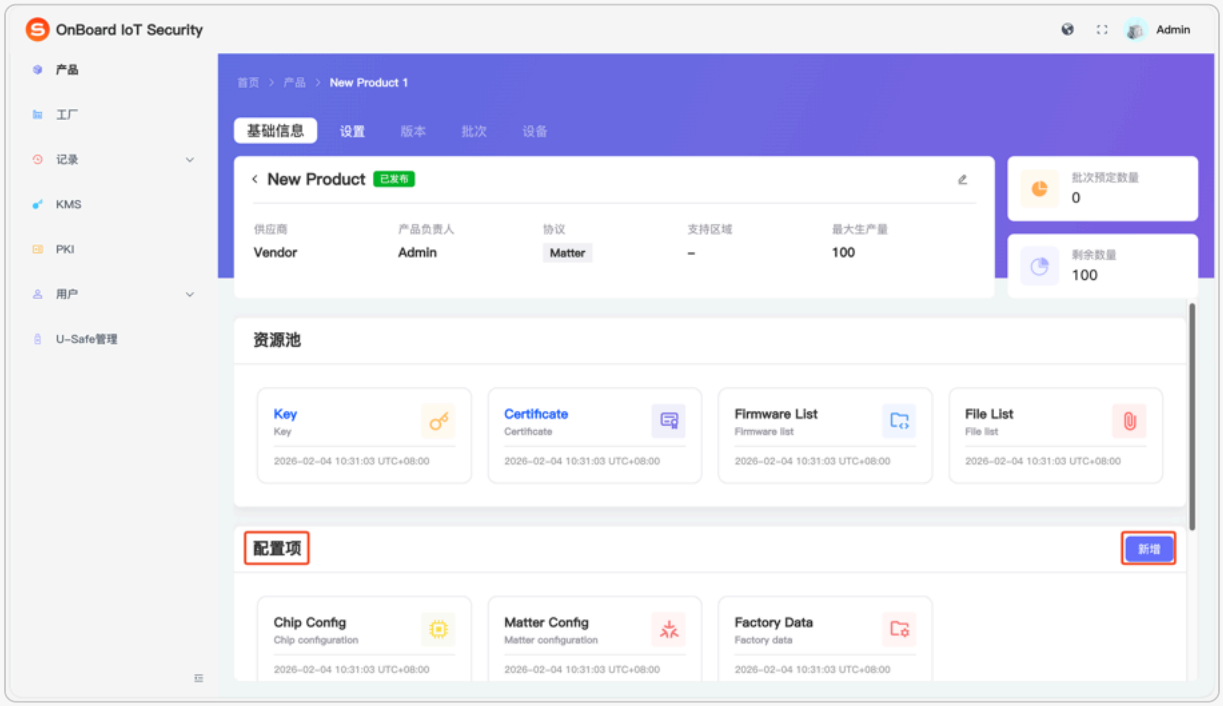


5. 点击**保存**，即可完成 Custom certificate configuration 的配置。

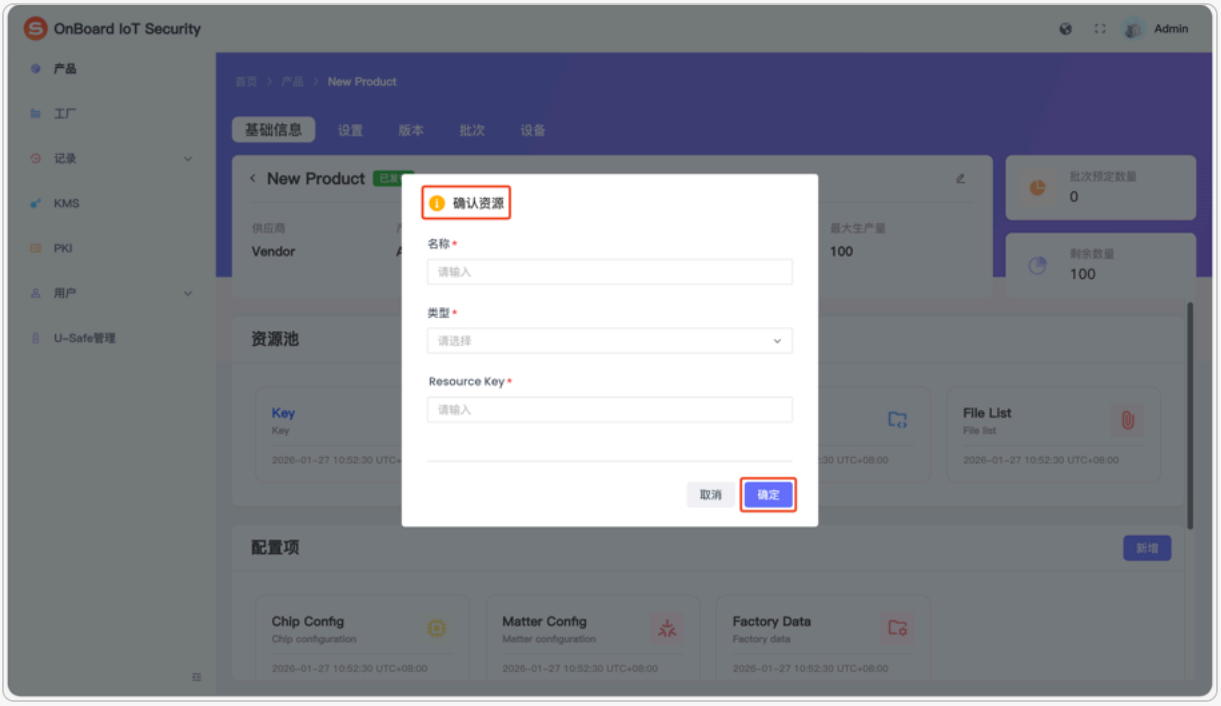
Google certificate configuration（可选）

> **提示**
>
> 配置 **Google certificate configuration** 的目的是设定 Google Cast 协议相关的设置。

1. 在产品页面，点击**配置项**区域的**新增**。



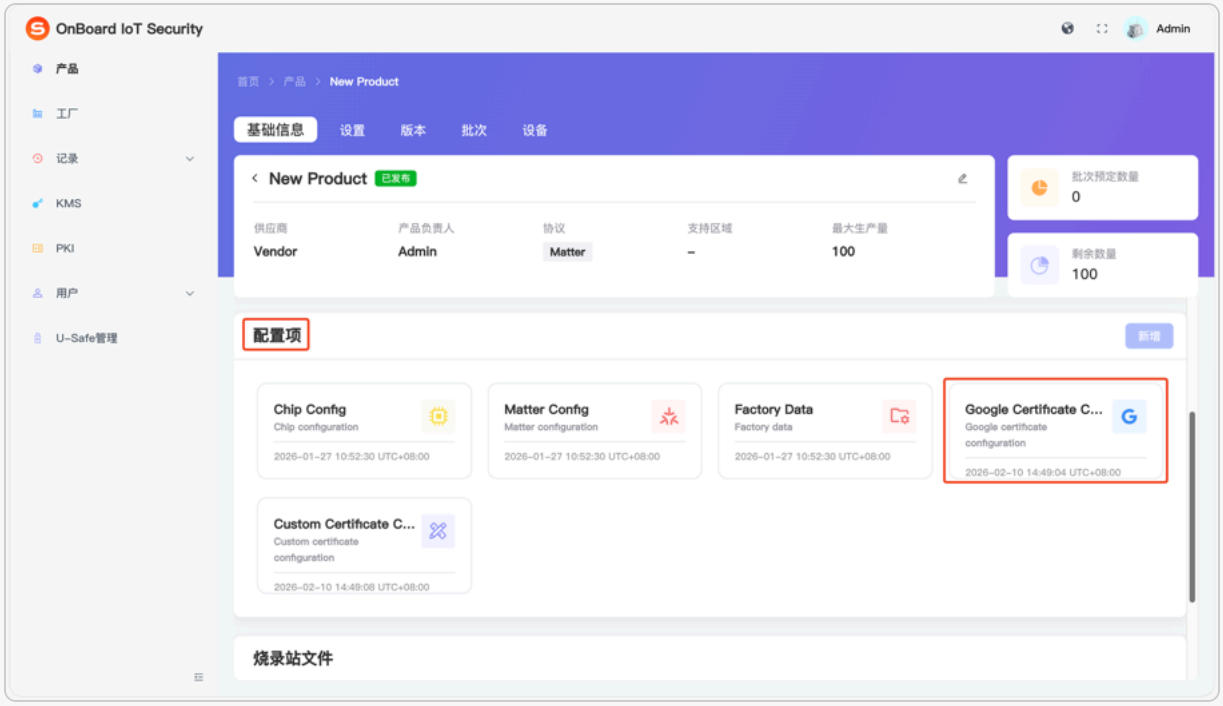
2. 在**确认资源**窗口，设置**名称**，**类型**和 **Resource key**。



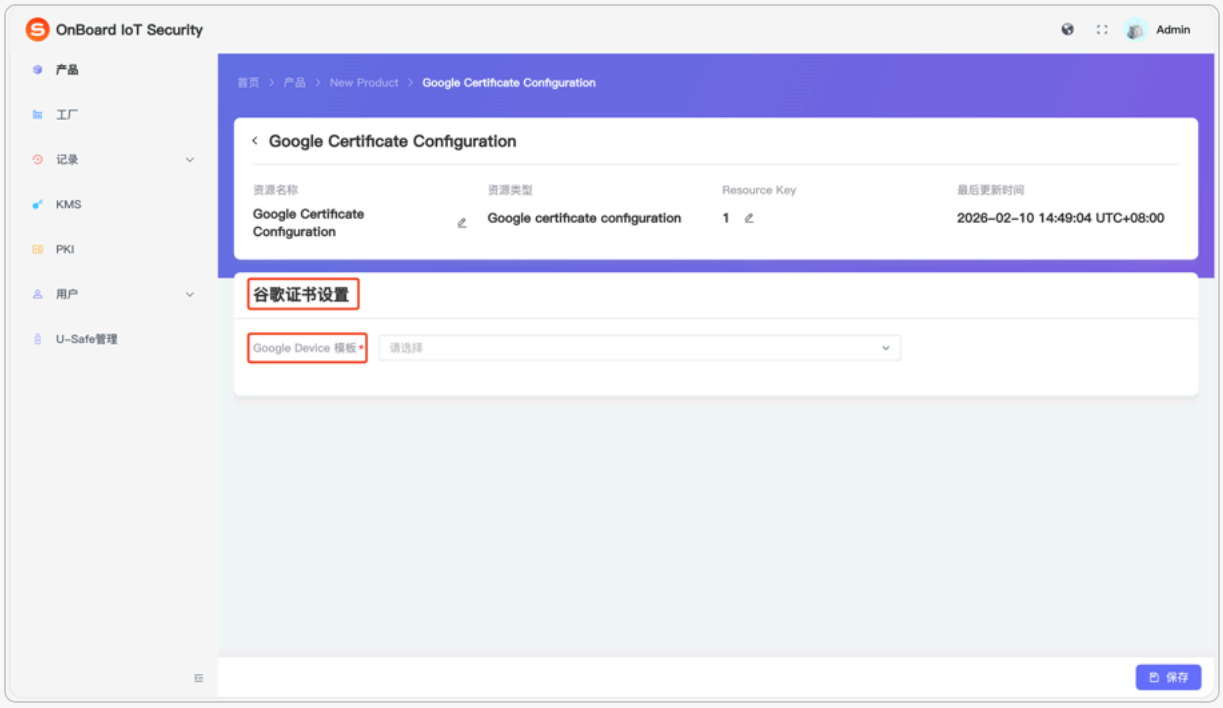
资源信息说明：

信息	说明
名称	资源名称。
类型	资源类型。
Resource Key	Resource key 必须唯一。

3. 在**配置项**区域点击刚新增的配置项。



4. 在**谷歌证书设置**区域, 选择在配置 **Certificate** 时添加的 Device 模板。



5. 点击**保存**，即可完成 Google certificate configuration 的配置。

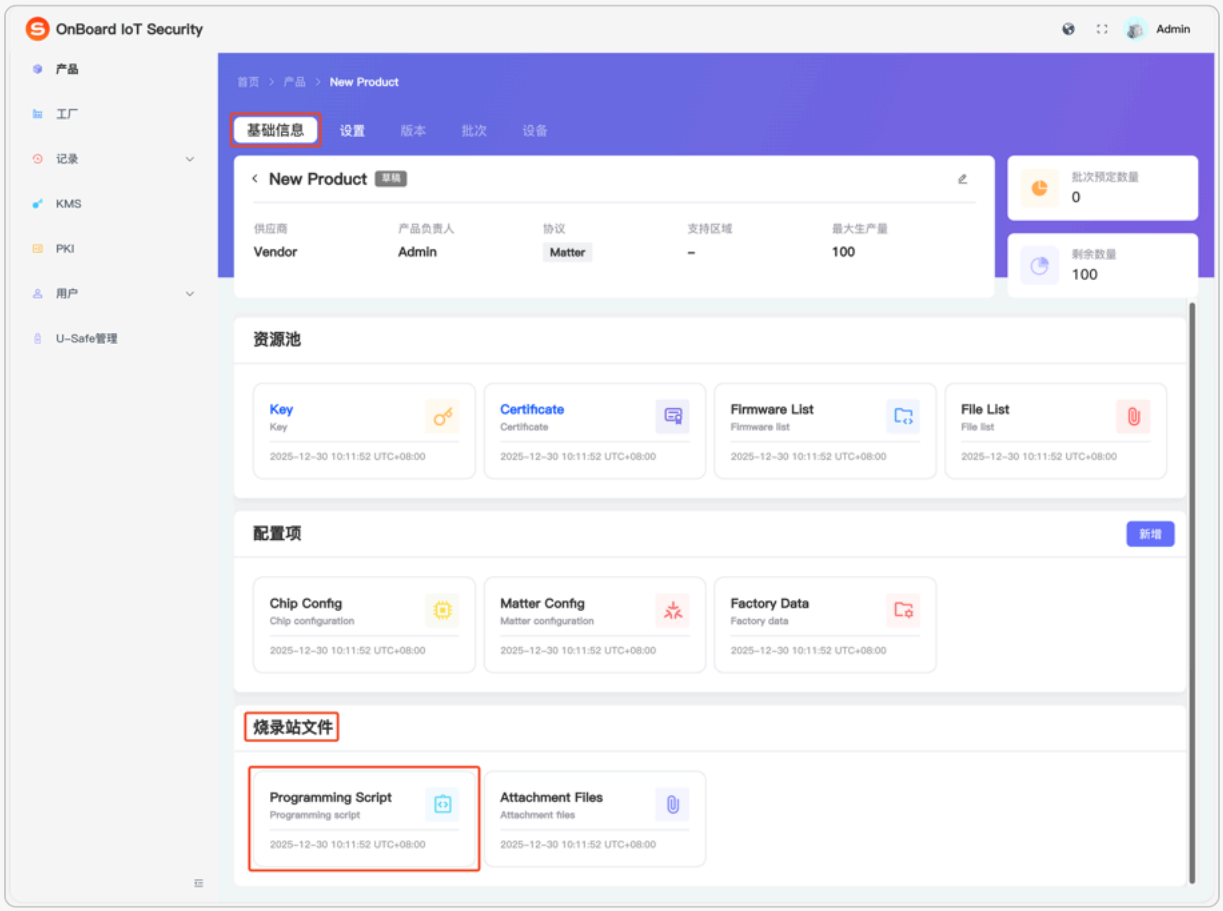
第三步：上传烧录站文件

> **提示**
>
> **烧录站文件**区域的文件用于创建批次。在配置 Programming Script 和 Attachment Files 时上传的文件，之后可以随时在对应资源的配置页面进行下载。

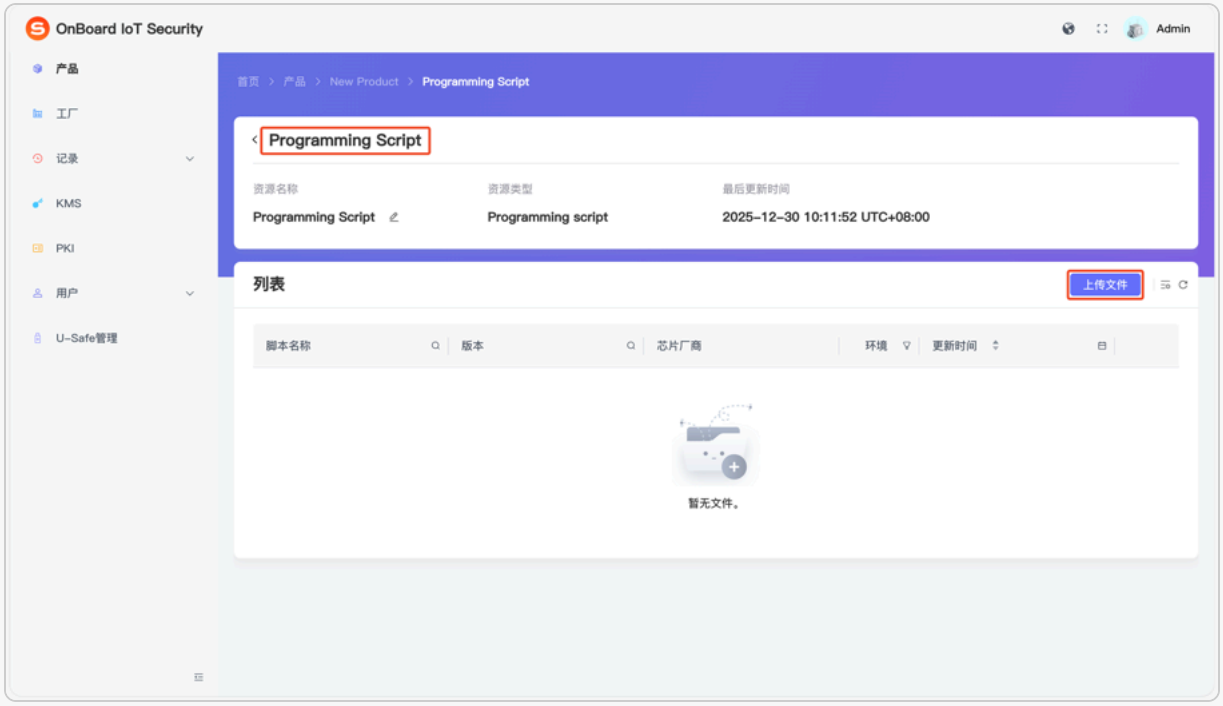
Programming Script（可选）

> **提示**
>
> 配置 **Programming Script** 的目的是上传设备烧录时使用的脚本文件（支持上传任意格式的烧录脚本）。上传的烧录脚本可在创建批次时进行选择。
>
> 你可以根据实际需求选择并上传烧录脚本；如果暂不需要，后续可在 Programming Station 客户端中为特定批次选择烧录脚本。

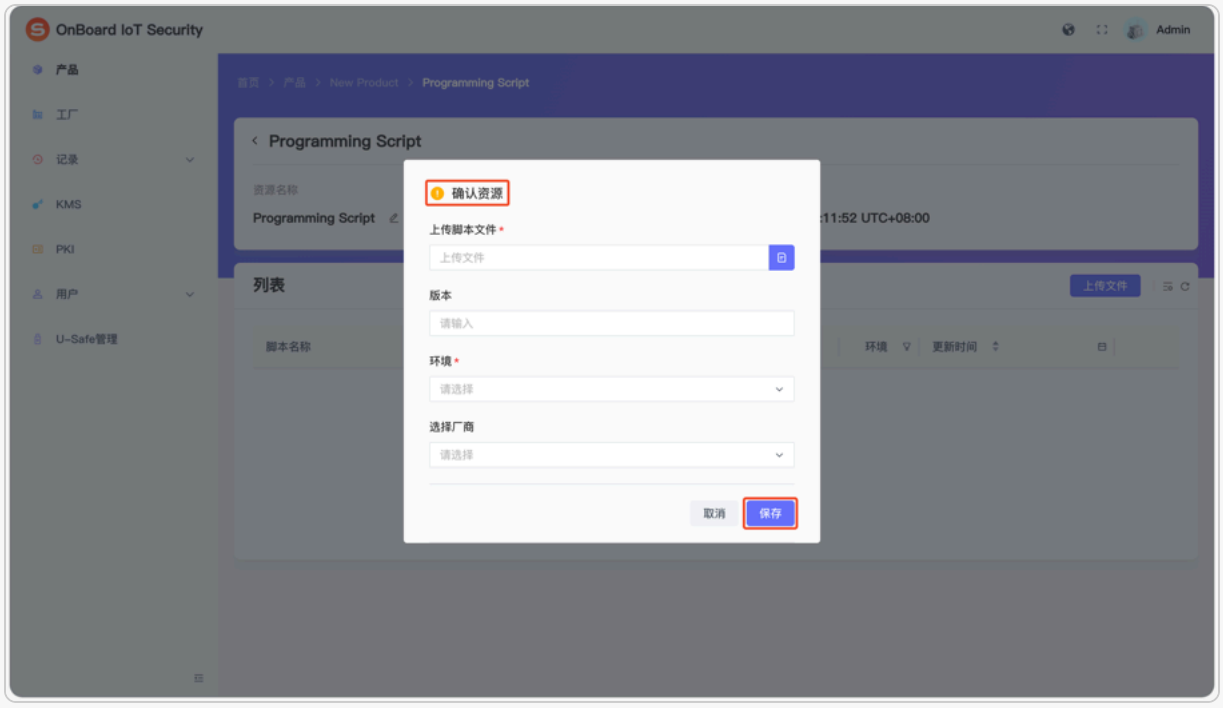
1. 在产品页面，点击**烧录站文件**区域的 **Programming Script**，进入 **Programming Script** 页面。



2. 在 **Programming Script** 页面，点击**上传文件**。



3. 在**确认资源**窗口，选择并输入烧录脚本相关信息。



Programming Script 信息说明：

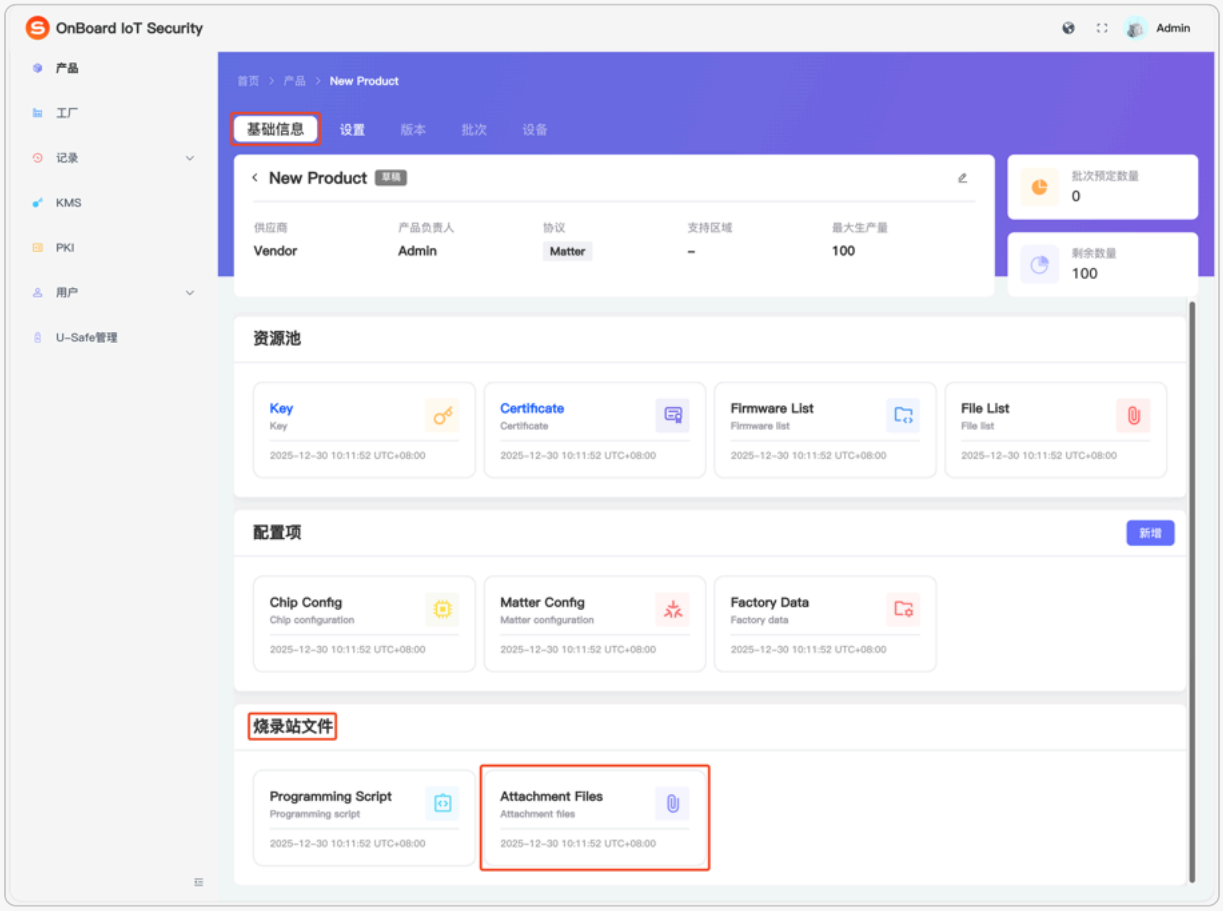
信息	是否必填	说明
上传脚本文件	是	用户通过选择本地的烧录脚本进行上传。
版本	否	标识烧录脚本的版本。
环境	是	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。
选择厂商	否	选择对应的芯片厂商。

4. 点击**保存**，即可完成 Programming Script 的配置。

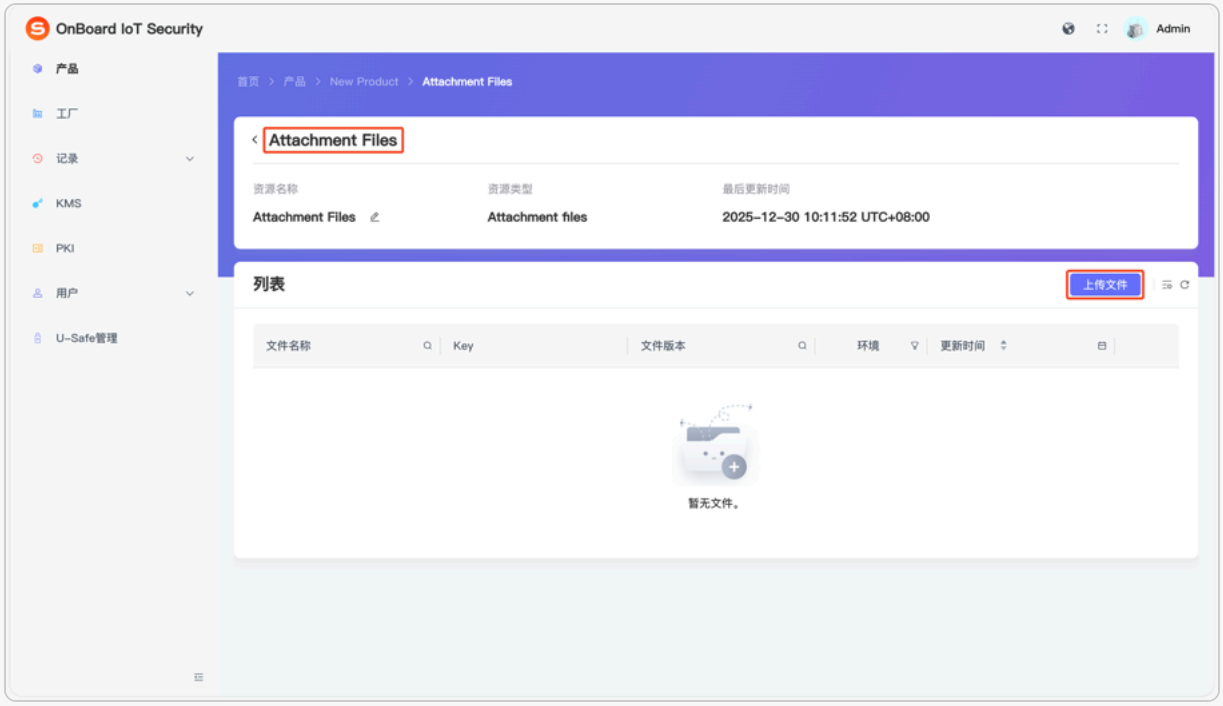
Attachment Files（可选）

> **提示**
>
> 配置 **Attachment Files** 的目的是上传除烧录脚本外，创建批次所需的其他文件（支持上传任意格式的文件）。
>
> 上传文件后，可在创建批次时进行选择。

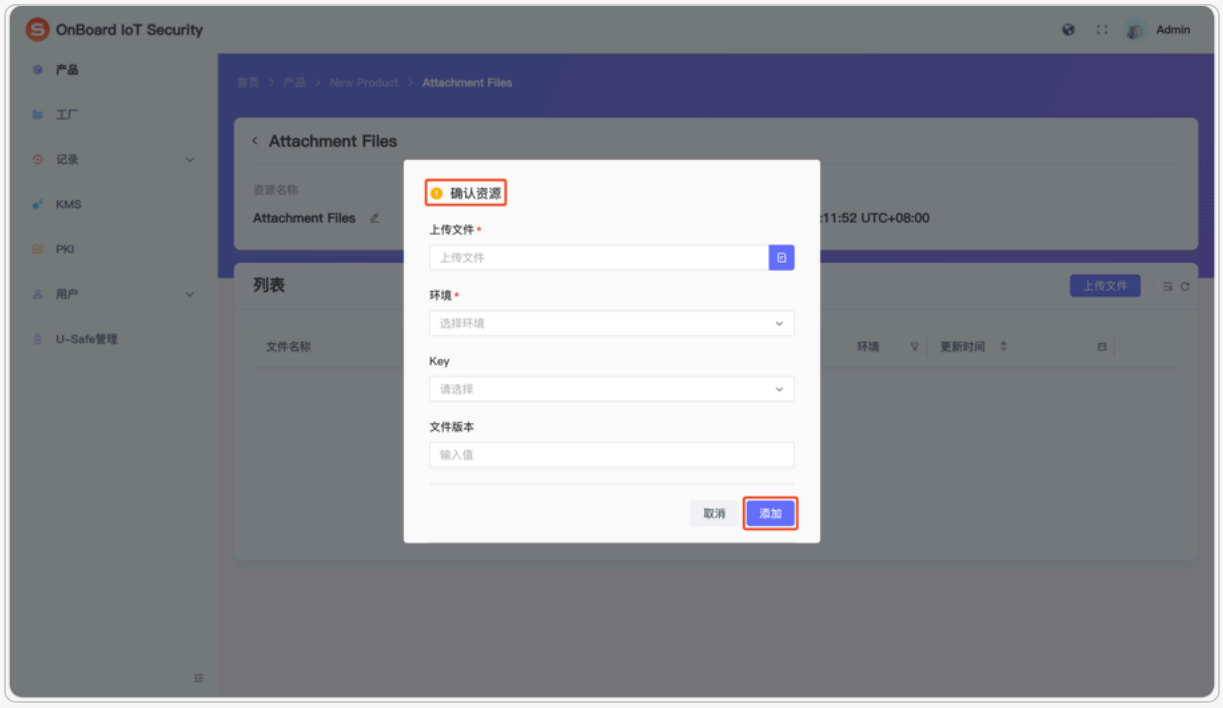
1. 在产品页面，点击**烧录站文件**区域的 **Attachment Files**，进入 **Attachment Files** 页面。



2. 在 **Attachment Files** 页面，点击**上传文件**。



3. 在**确认资源**窗口，选择并输入相关信息。



Attachment Files 信息说明：

信息	是否必填	说明
上传文件	是	通过选择本地的文件进行上传。
环境	是	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。
Key	否	为文件分配 Key。后期烧录时，烧录脚本可通过 Key 调取并使用对应文件。 **注释**：一个文件可分配多个不重复的 Key。同时，同一个 key 可被分配到多个文件。可新增 Key 或从已有 Key 中进行选择。
文件版本	否	标识文件版本并用于跟踪文件变更。

4. 点击**添加**，即可完成 Attachment Files 的配置。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看产品

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看产品的基础信息和配置。

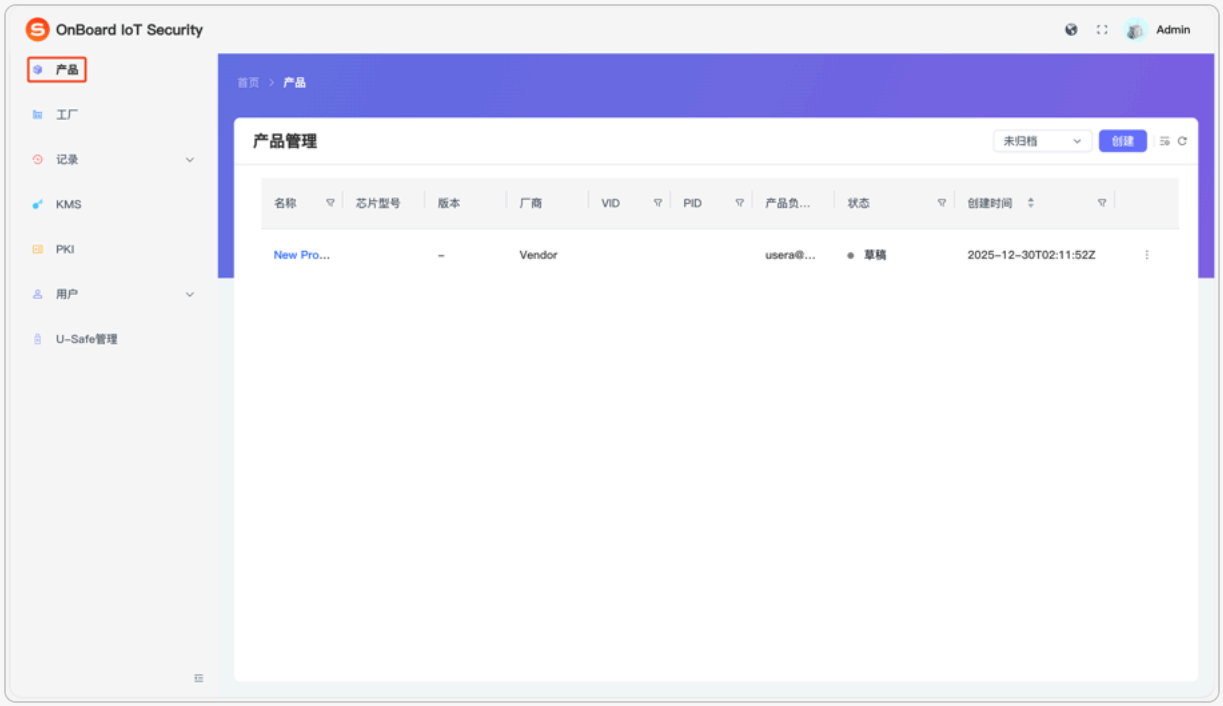
前提条件

在开始之前，请确保你已拥有对应产品的产品角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品，进入产品页面。

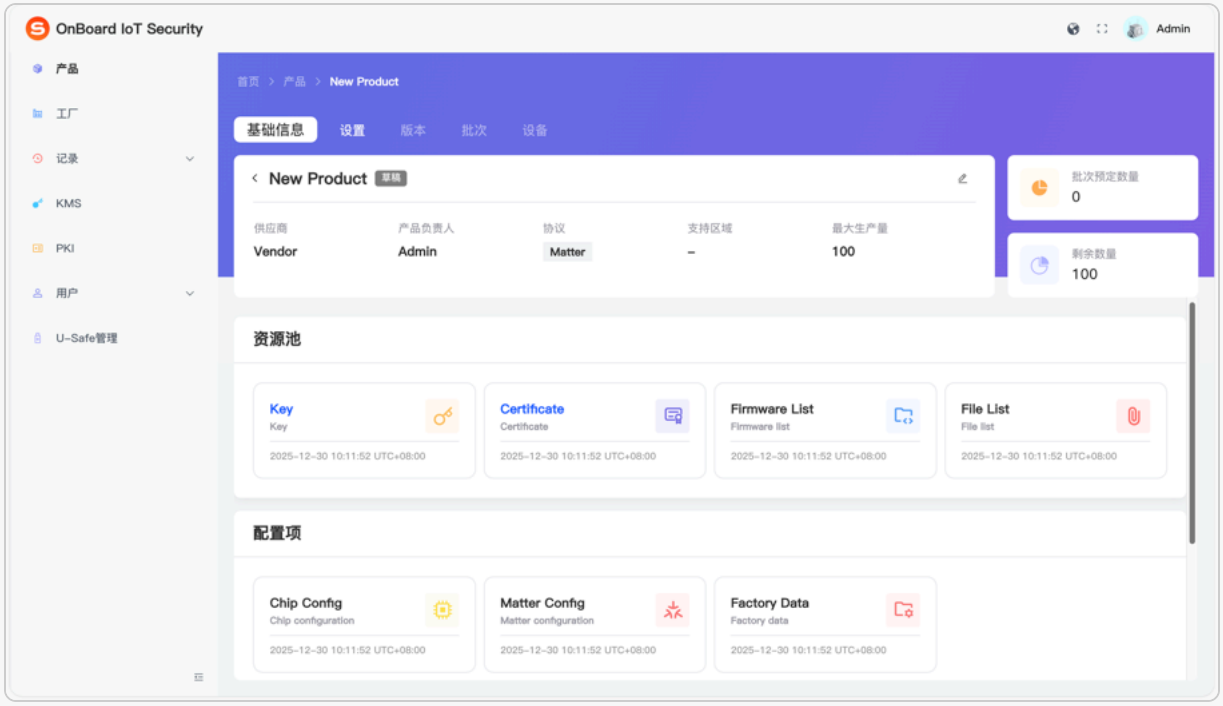
> **提示**
>
> 产品列表包含你参与的及租户内所有已创建的产品，默认显示未归档的产品。



产品状态说明：

状态	说明
草稿	产品尚未创建版本前的状态。
已发布	产品创建版本后的状态。 **注释**：如果删除产品所有的版本，产品状态将变更为**草稿**状态。

3. 在产品页面，即可查看产品相关信息（**供应商**，**产品负责人**和**协议**等）。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product
基础信息
设置
版本
批次
设备
New Product
草稿
供应商
Vendor
产品负责人
Admin
协议
Matter
支持区域
-
最大生产量
100
批次预定数量
0
剩余数量
100
资源池
Key
Key
2025-12-30 10:11:52 UTC+08:00
Certificate
Certificate
2025-12-30 10:11:52 UTC+08:00
Firmware List
Firmware list
2025-12-30 10:11:52 UTC+08:00
File List
File list
2025-12-30 10:11:52 UTC+08:00
配置项
Chip Config
Chip configuration
2025-12-30 10:11:52 UTC+08:00
Matter Config
Matter configuration
2025-12-30 10:11:52 UTC+08:00
Factory Data
Factory data
2025-12-30 10:11:52 UTC+08:00

版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理产品

> **提示**
>
> 在对应资源的配置页面，可以下载在配置 Firmware List，File List，Matter Config，Factory Data，Programming Script 和 Attachment Files 时上传的文件。

本文将指导你在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

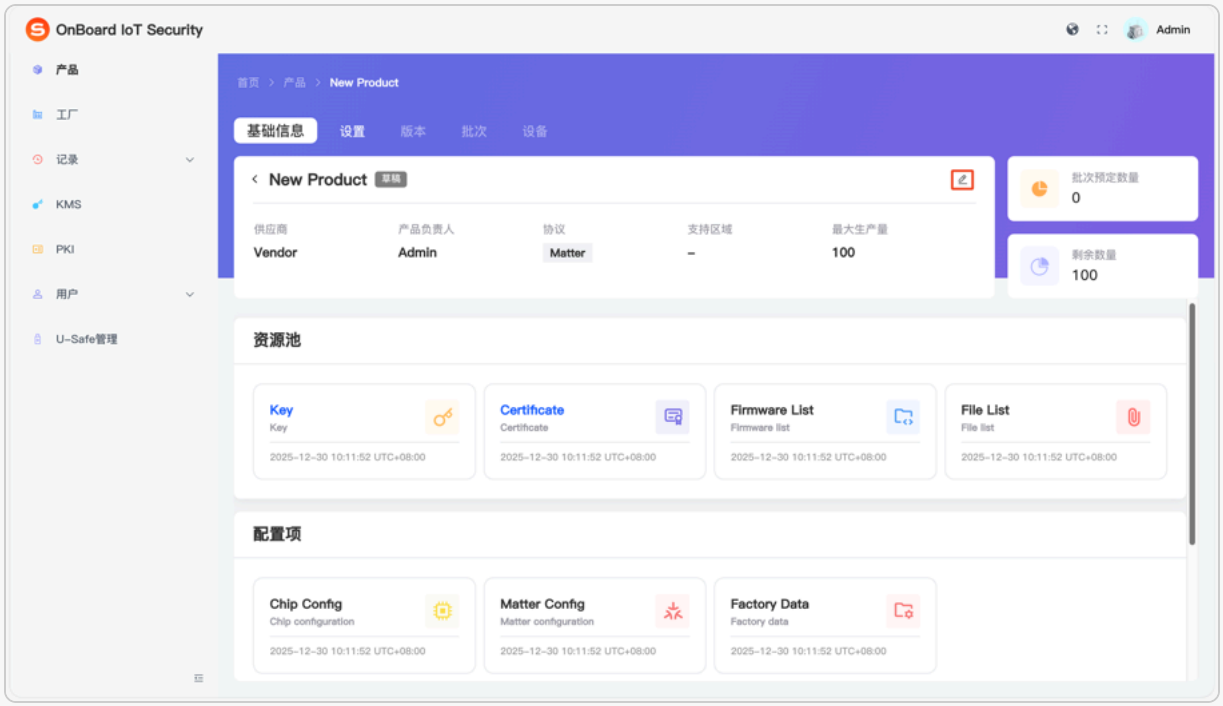
- 编辑产品信息
- 管理产品归档

前提条件

在开始之前，请确保你已拥有对应产品的产品角色权限：Product Owner 或 Product Manager。

编辑产品信息

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在**基础信息**标签页，点击编辑 图标即可更改产品信息（**产品名称**，**供应商**和**产品负责人**等）。

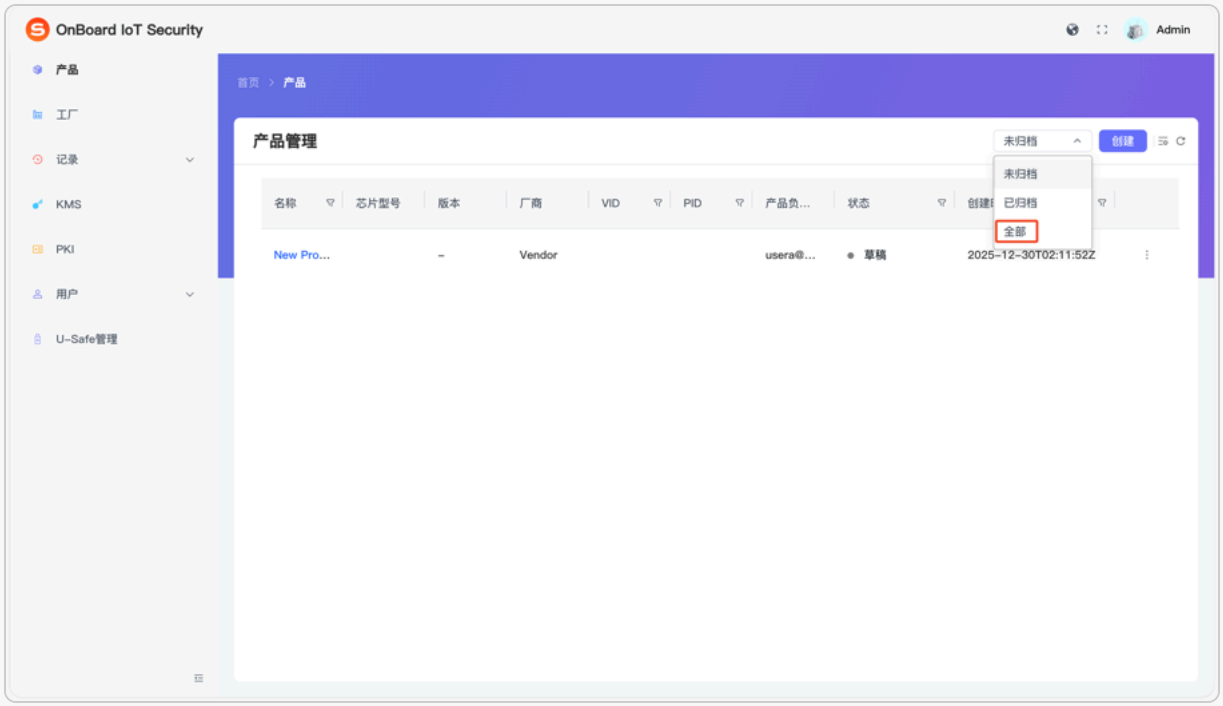


管理产品归档

> **提示**
>
> 产品列表默认显示未归档的产品。产品归档后将无法进行任何编辑，包括信息修改，资源配置，版本创建及批次创建。

1. 登录 OBIS。

2. 点击左侧菜单中的**产品**，并在**产品管理**页面的右上角选择**全部**。



3. 点击对应产品的选项 ⋮ 图标并选择**归档**或**取消归档**，即可归档或取消归档该产品。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理产品业务成员

> **提示**
>
> 你可以在**设置**标签页的**日志**区域查看成员在产品中的操作记录。

本文将指导你在 OnBoard™ IoT Security (OBIS) 中完成以下操作从而调整用户对产品的访问与操作权限：

- 添加用户
- 更新用户角色
- 移除用户
- 添加用户组
- 更新用户组角色
- 移除用户组

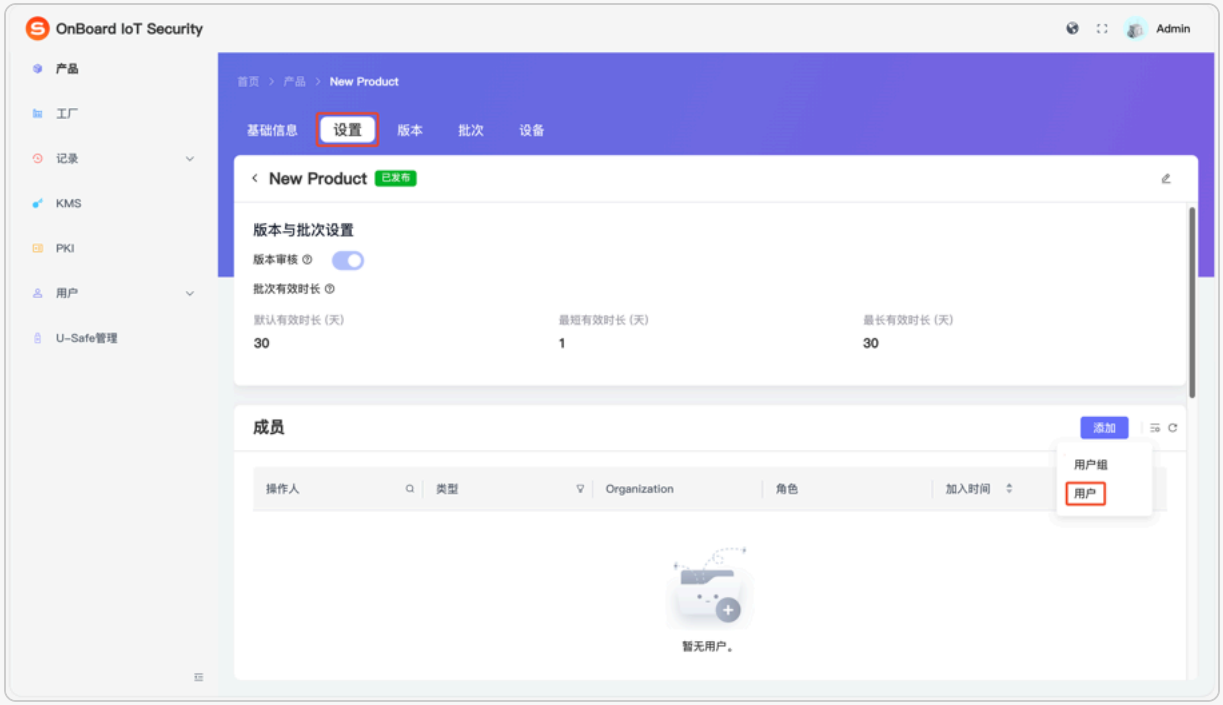
前提条件

在开始之前，请确保：

- 已完成用户和用户组的创建。
- 你已拥有对应产品的产品角色权限：Product Owner 或 Product Manager。

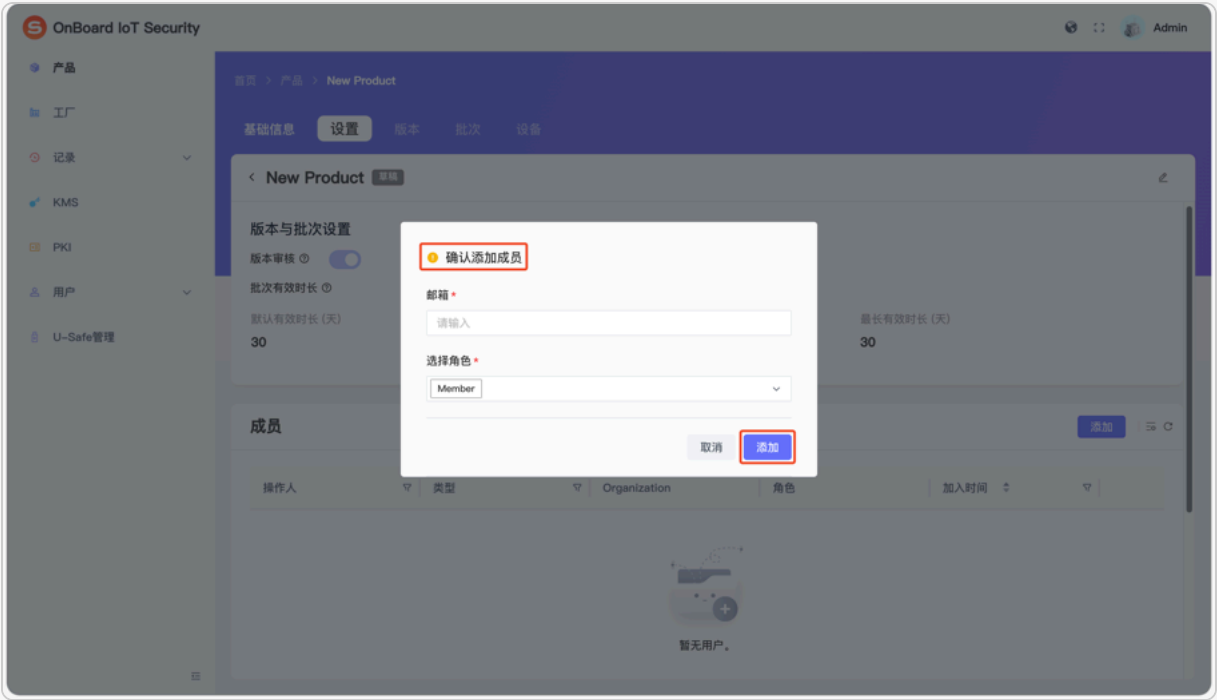
添加用户

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在产品页面，选择**设置**标签页，在**成员**区域中点击**添加**并选择**用户**。



4. 在**确认添加成员**窗口，输入用户的邮箱并选择角色。

> **提示**
>
> - 一个用户可以被赋予多个角色。更多信息请参考产品角色权限说明。
> - 被添加的用户将收到一封包含链接的电子邮件，可通过该链接查看产品。
> - 被添加的用户必须是状态为**启用**的用户。支持跨租户添加，例如：Product Manager 来自租户 A，可以添加来自租户 B 的用户加入其产品团队。



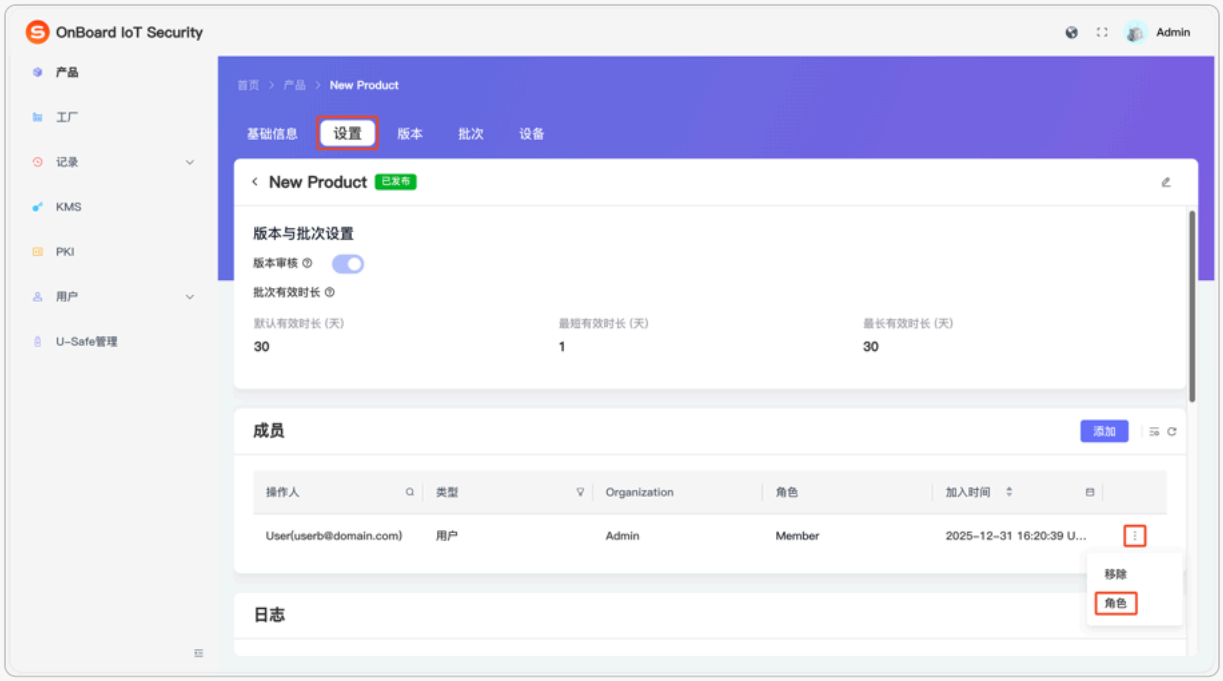
5. 点击**添加**，即可完成成员的添加。

更新用户角色

1. 登录 OBIS。

2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。

3. 在产品页面，选择**设置**标签页，在**成员**区域点击对应用户的选项 ⋮ 图标并选择**角色**。

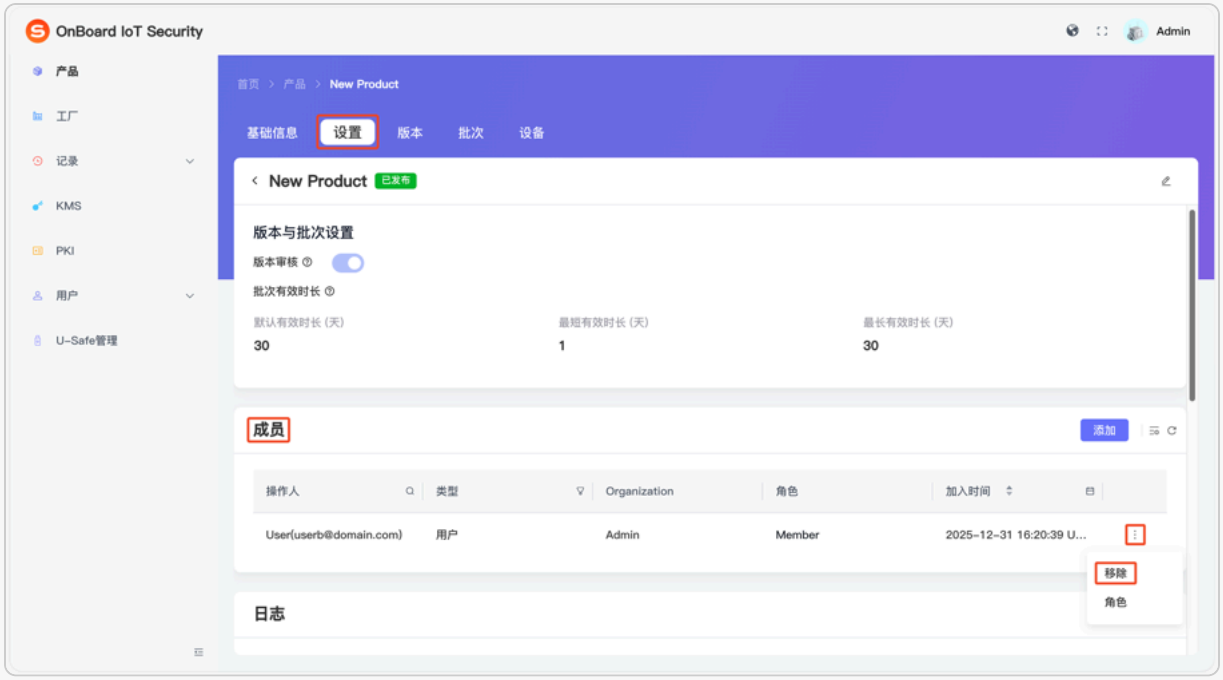


4. 在**确认更新角色**窗口，选择对应的角色，点击**添加**即可完成更新。

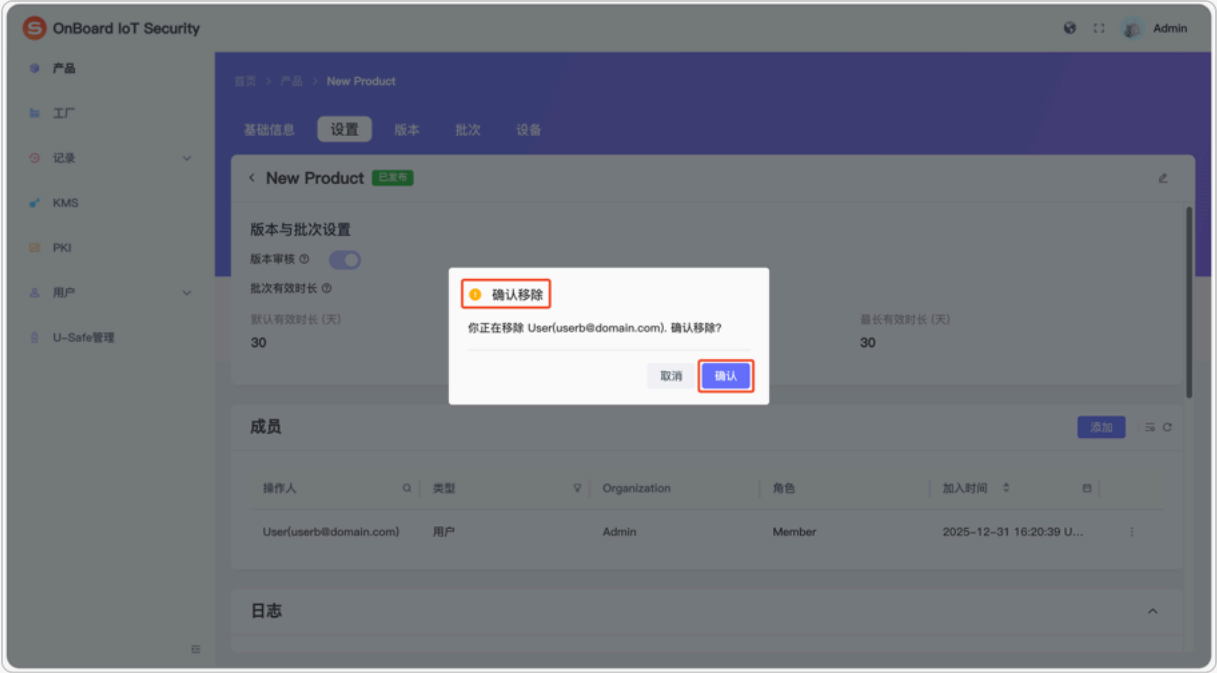


移除用户

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在产品页面，选择**设置**标签页，在**成员**区域点击对应用户的选项 ⋮ 图标并选择**移除**。

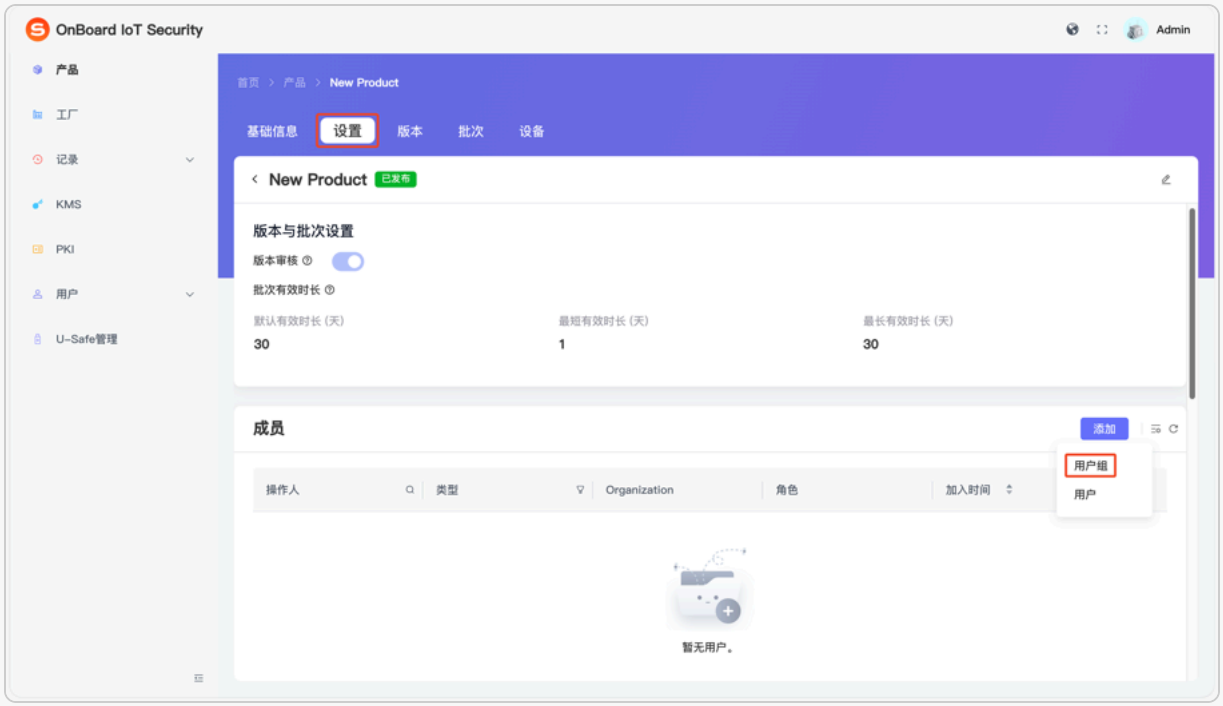


4. 在**确认移除**窗口，仔细核对信息，点击**确认**即可移除该用户。



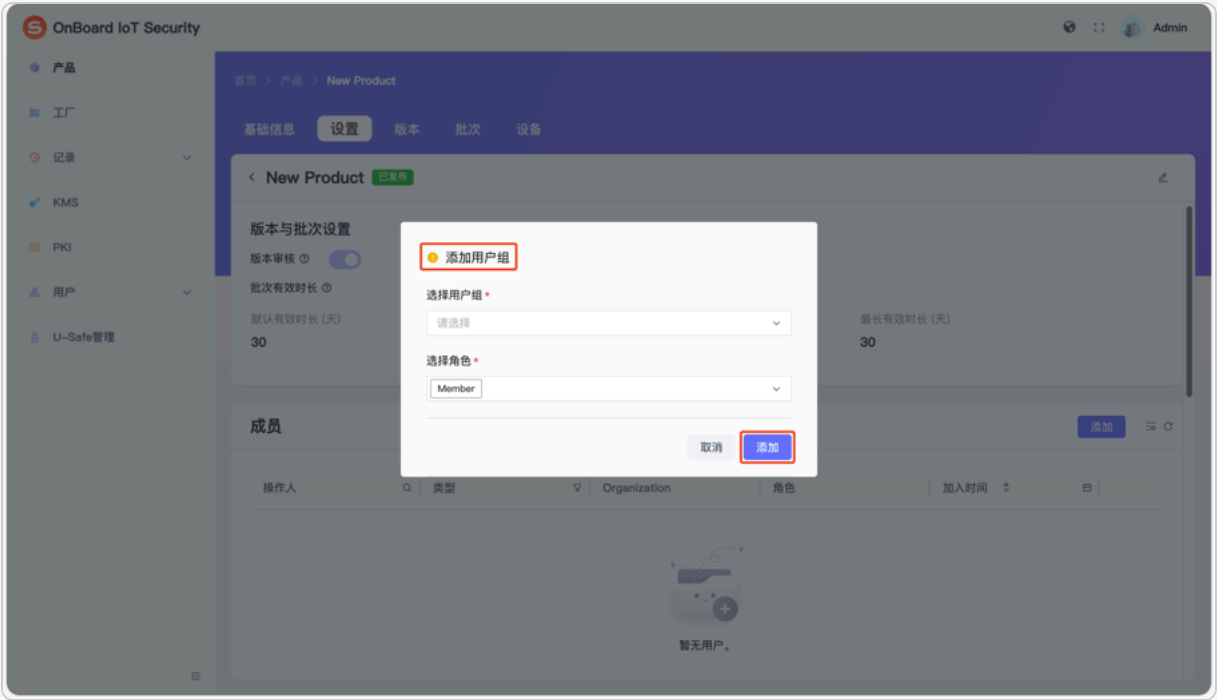
添加用户组

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在产品页面，选择**设置**标签页，在**成员**区域中点击**添加**并选择**用户组**。



4. 在**添加用户组**窗口，选择用户组和角色。

> **提示**
>
> - 一个用户组可以被赋予多个角色。更多信息请参考产品角色权限说明。
> - 用户组内的用户将收到一封包含链接的电子邮件，可通过该链接查看产品。
> - 只可添加同一租户下的用户组。



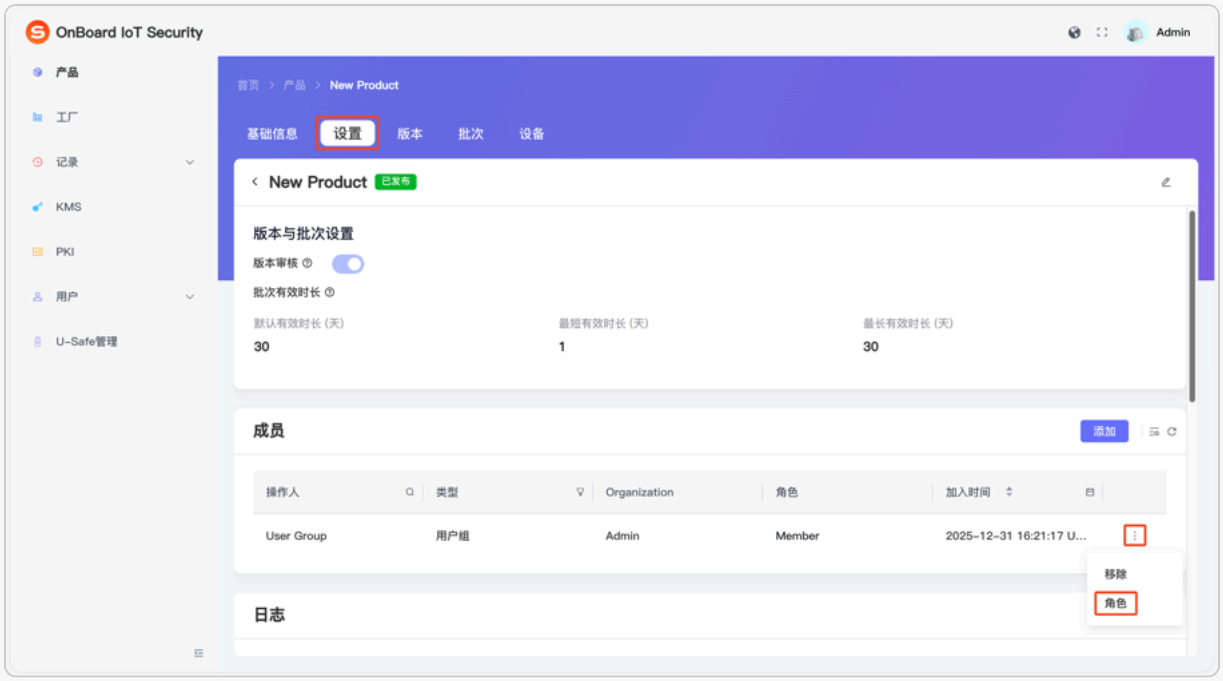
5. 点击**添加**，即可完成用户组的添加。

更新用户组角色

1. 登录 OBIS。

2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。

3. 在产品页面，选择**设置**标签页，在**成员**区域点击对应用户的选项 ⋮ 图标并选择**角色**。



4. 在**确认更新角色**窗口，选择对应的角色，点击**添加**即可完成更新。

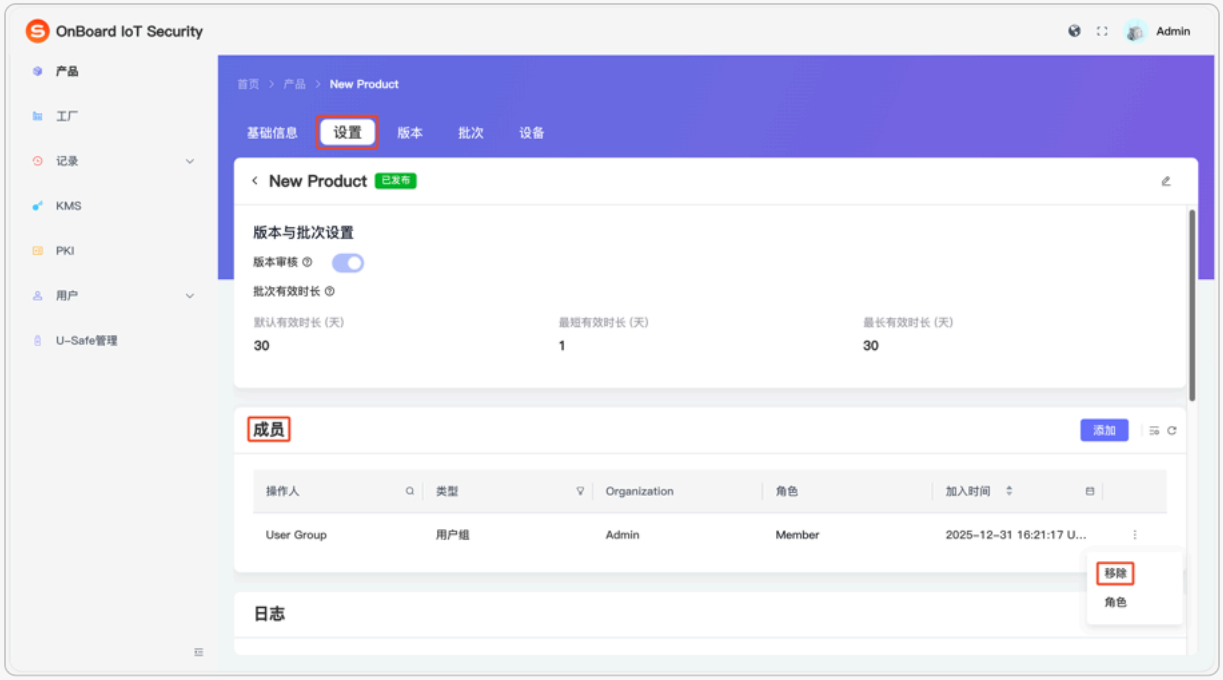


移除用户组

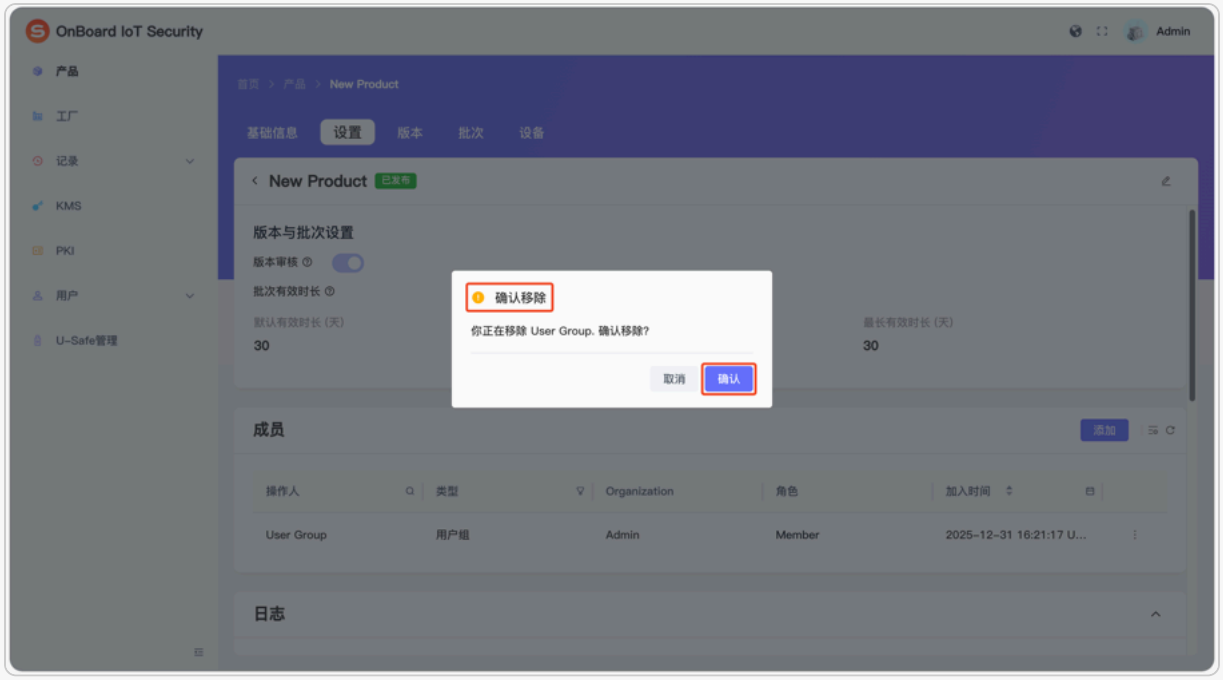
1. 登录 OBIS。

2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。

3. 在产品页面，选择**设置**标签页，在**成员**区域点击对应用户的选项 ⋮ 图标并选择**移除**。



4. 在**确认移除**窗口，仔细核对信息，点击**确认**即可移除该用户组。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product
基础信息
设置
版本
批次
设备
New Product
已发布
版本与批次设置
版本审核
批次有效时长
默认有效时长 (天)
30
最长有效时长 (天)
30
确认移除
你正在移除 User Group. 确认移除?
取消
确认
成员
添加
操作人
类型
Organization
角色
加入时间
User Group
用户组
Admin
Member
2025-12-31 16:21:17 U...
日志

版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建版本

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中创建版本，为后续创建批次做好准备。每个产品可以创建多个版本。

前提条件

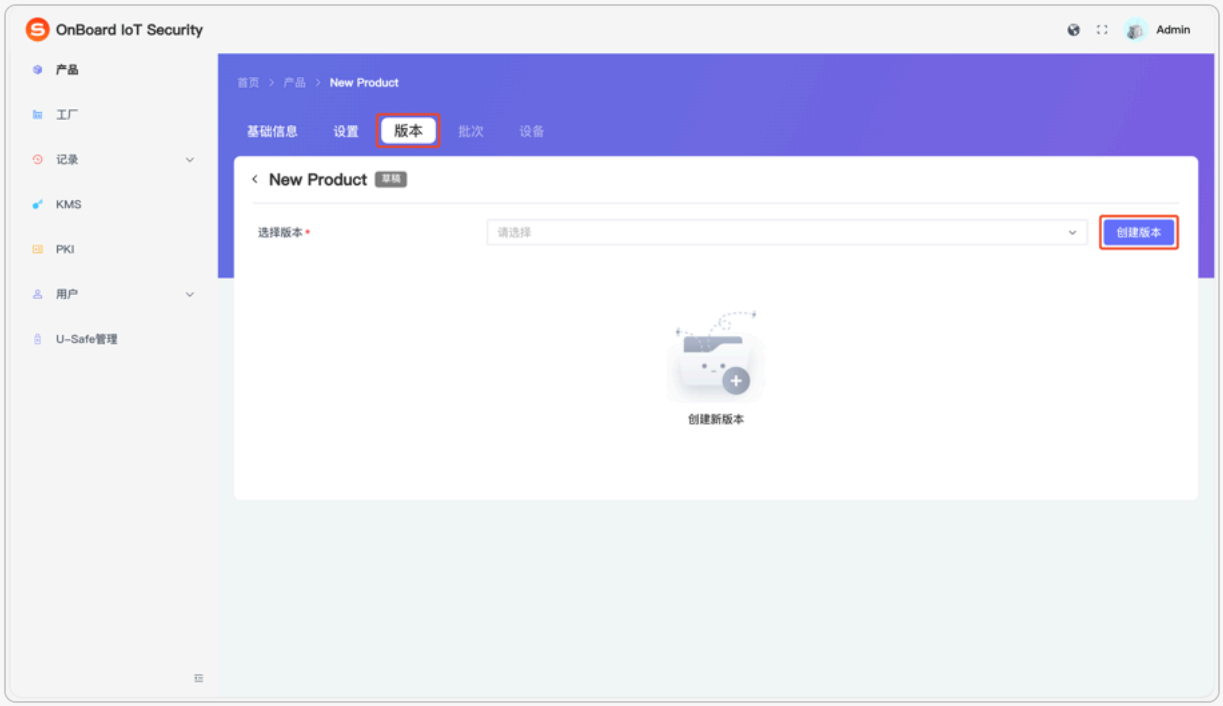
在开始之前，请确保：

- 已完成产品的配置。
- 你已拥有对应产品的产品角色权限：Product Owner，Product Manager 或 Version Manager。

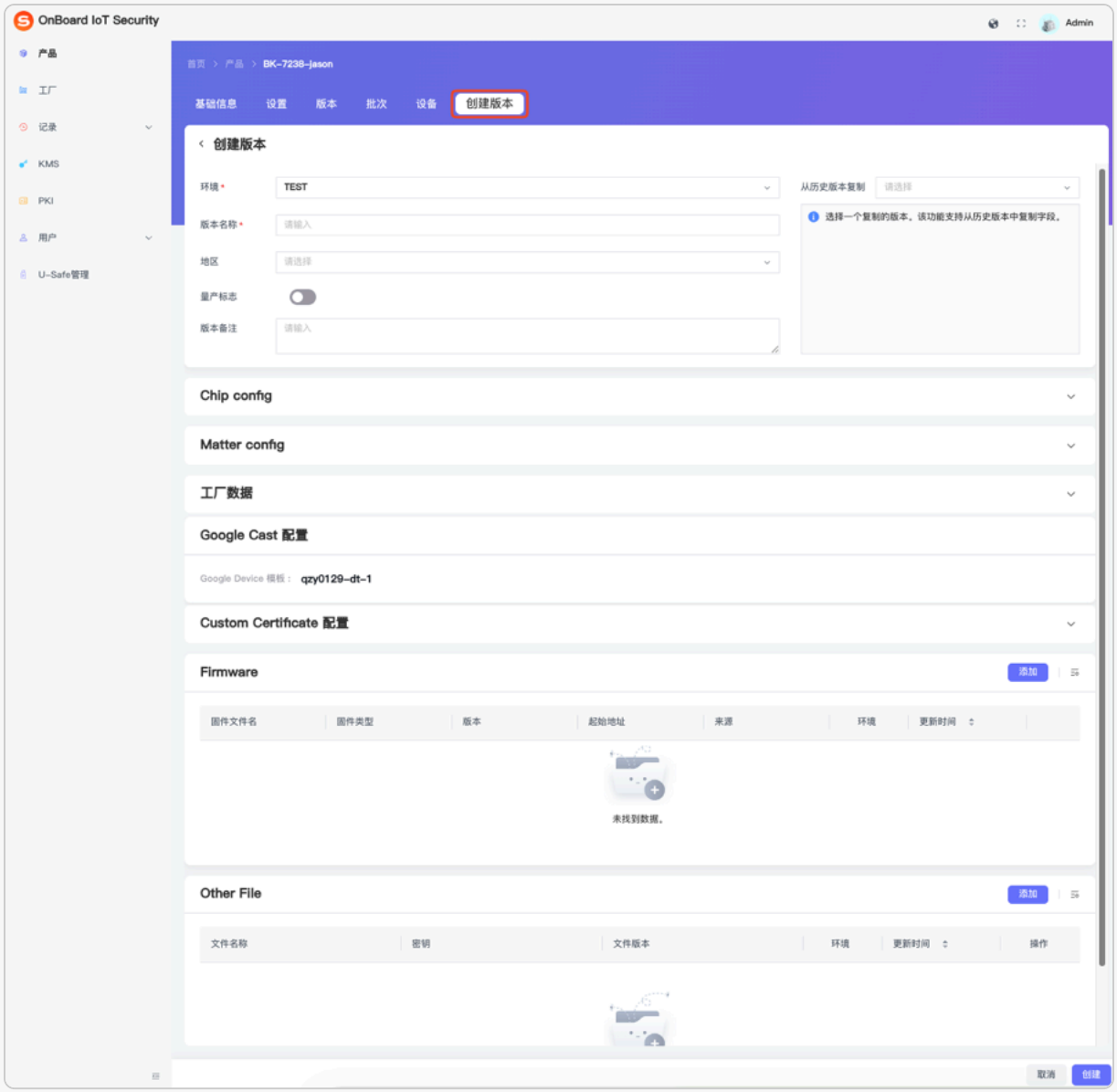
操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在产品页面，选择**版本**标签页并点击**创建版本**。页面会跳转至**创建版本**标签页。

> **提示**
>
> 如果**版本**标签页显示为灰色，表示当前不可用。请检查该产品必要的资源与配置项是否均已完成配置。当所有内容配置完成后，**版本**标签页将自动启用。



4. 在**创建版本**标签页，选择并输入版本信息。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > BK-7238-jason
基础信息
设置
版本
批次
设备
创建版本
创建版本
环境 *
TEST
从历史版本复制
请选择
版本名称 *
请输入
选择一个复制的版本，该功能支持从历史版本中复制字段。
地区
请选择
量产标志
版本备注
请输入
Chip config
Matter config
工厂数据
Google Cast 配置
Google Device 模板：
qzy0129-dt-1
Custom Certificate 配置
Firmware
添加
固件文件名
固件类型
版本
起始地址
来源
环境
更新时间
未找到数据。
Other File
添加
文件名称
密钥
文件版本
环境
更新时间
操作
取消
创建

版本信息说明：

信息	是否必填	说明
环境	是	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。 **注释**：如需配置 PROD 环境的版本，在 Matter Config 配置项中必须配置 PROD DAC Template。
版本名称	是	版本名称在同一环境下不可重复。
地区	否	选择并设定版本的地区。 **注释**：可选地区请参考创建产品时设置的支持区域。
量产标志	否	标记版本是否可用于大规模量产。通常仅在完成所有测试后，才会为 PROD 环境的版本启用**量产标志**功能。
版本备注	否	用于补充说明版本相关的额外信息，如特殊要求和注意事项等。
从历史版本复制	否	可以选历史版本作为模板修改。 **注释**：从历史版本复制后，不可切换环境。
Chip config	是	产品配置项区域配置的 Chip Config。如果在 TEST 环境，且启用了安全启动或安全调试功能，此处可取消勾选；PROD 环境下则不可修改。
Matter config	是	产品配置项区域配置的 Matter Config。其中 **CD 文件**和**产品 ID** 可以重新选择，其余信息则不可修改。

信息	是否必填	说明
工厂数据	是	产品配置项区域配置的 Factory Data。值均可修改。 **注释**：在配置 Factory Data 时标记为必填的工厂数据都必须提供有效值，否则无法完成创建。
Custom Certificate 配置	否	产品配置项区域配置的 Custom certificate configuration。
Google Cast 配置	否	产品配置项区域配置的 Google certificate configuration。
Firmware	是	产品物料区域配置的 Firmware List。选择该版本需要的相关固件文件。
Other File	否	产品物料区域配置的 File List。选择该版本需要的相关文件。

5. 确认所有信息无误后，点击页面底部的**创建**，即可完成版本的创建。

> **提示**
>
> 如果在创建产品时开启了版本审核，PROD 环境下的版本创建后需要 Product Owner 或 Product Manager 进行审核。
>
> - 审核通过前，版本状态为**待审核**，无法用于创建批次。
> - 审核通过后，版本状态变为**已通过**。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看版本

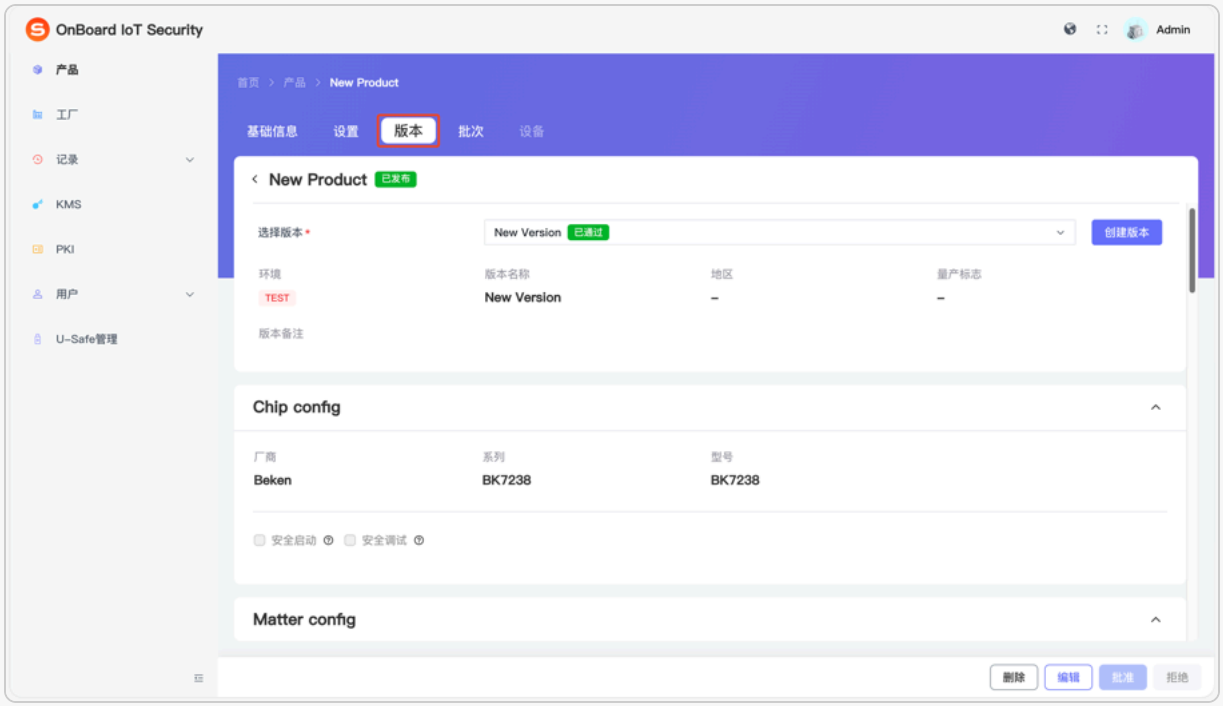
本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看版本的信息和配置。

前提条件

在开始之前，请确保你已拥有对应产品的产品角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在产品页面，选择**版本**标签页。在**版本**标签页选择对应的版本，即可查看版本相关信息（**Chip config**，**Matter config** 和 **Firmware** 等）。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product
基础信息
设置
版本
批次
设备
New Product
已发布
选择版本 *
New Version
已通过
创建版本
环境
TEST
版本名称
New Version
地区
–
量产标志
–
版本备注
Chip config
厂商
Beken
系列
BK7238
型号
BK7238
安全启动
安全调试
Matter config
删除
编辑
批准
拒绝

版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理版本

本文将指导你在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

- 审核版本
- 编辑版本
- 删除版本

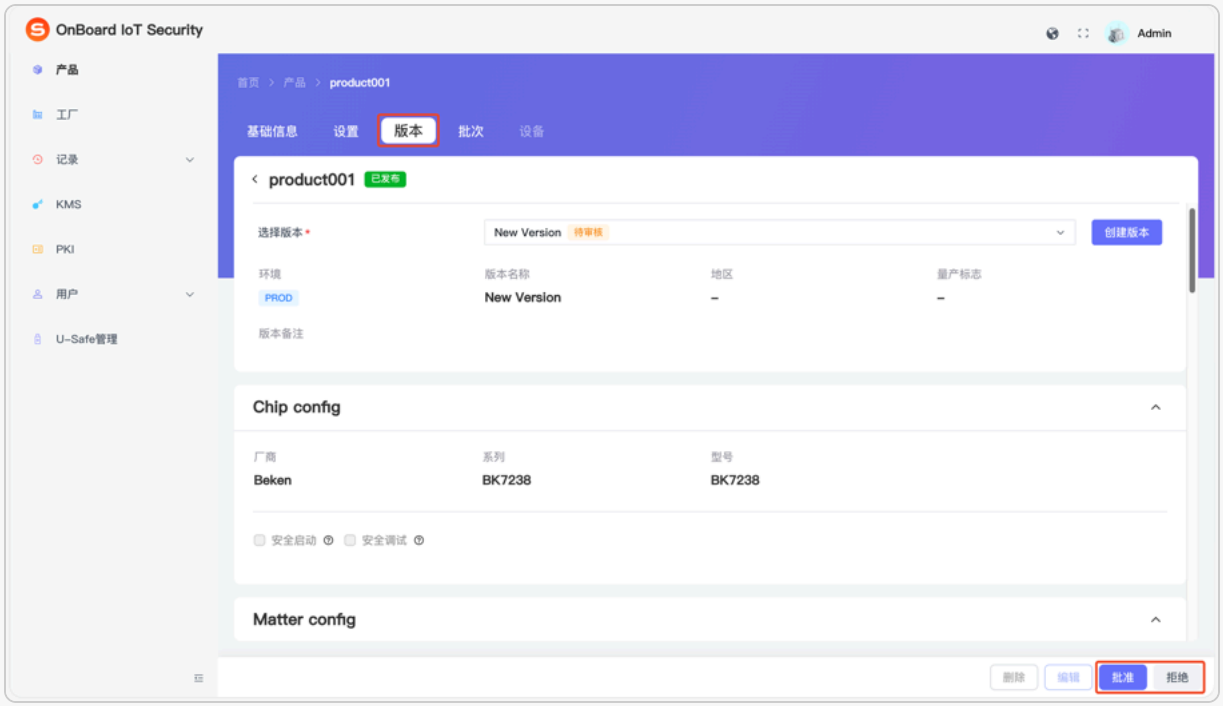
前提条件

在开始之前，请确保你已拥有对应产品的产品角色权限：Product Owner，Product Manager 或 Version Manager。

审核版本

> **提示**
>
> 如果在创建产品时开启了版本审核，PROD 环境下的版本创建后将需要 Product Owner 或 Product Manager 进行审核（批准或拒绝）。
>
> - 批准后，版本状态变为**已通过**并可用于创建批次。
> - 拒绝后，版本状态变为**未通过**，需要修改并重新审核。
>
> 如果需要关闭版本审核，Product Owner 或 Product Manager 可以在产品的**设置**标签页面进行关闭。

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 选择**版本**标签页并选择需要审核的版本，然后点击页面底部的**批准**或**拒绝**。



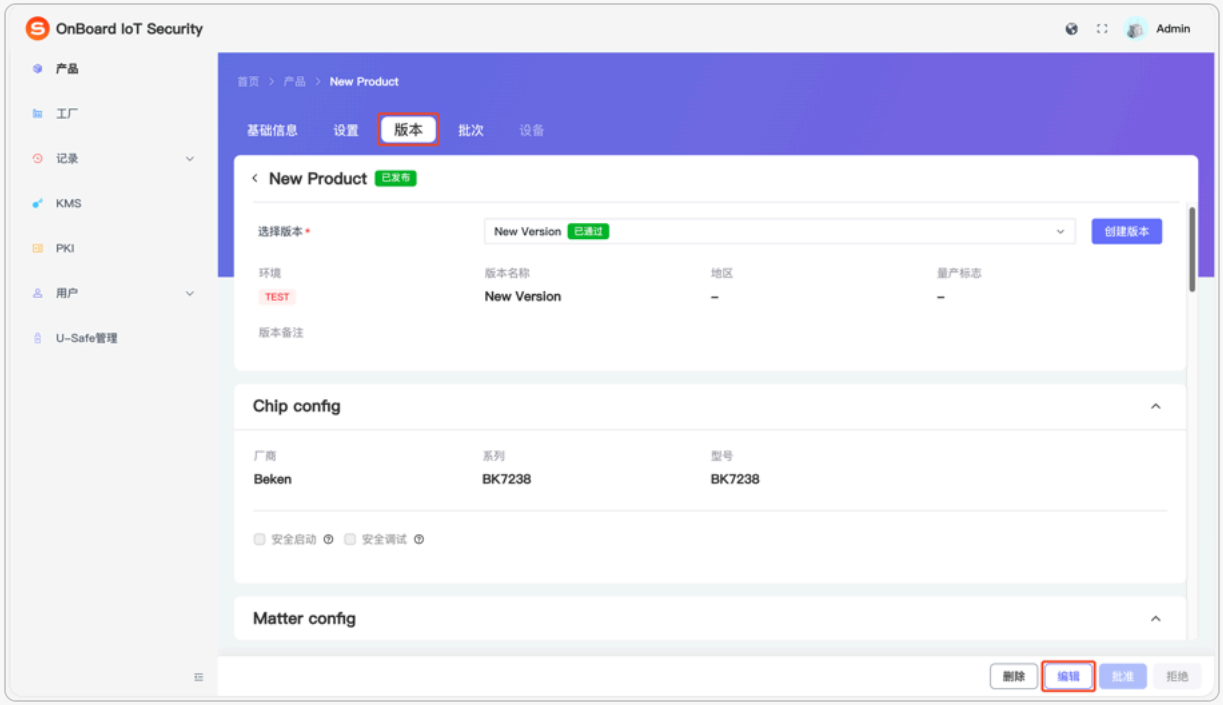
编辑版本

> **提示**
>
> 编辑版本前请确保：
>
> - 此版本状态不是**待审核**（如果版本状态是**待审核**，此版本将无法编辑）。
> - 此版本未关联任何批次（如果使用此版本创建批次后，此版本将无法编辑）。

1. 登录 OBIS。

2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。

3. 选择**版本**标签页并选择需要编辑的版本，然后点击页面底部的**编辑**。



4. 确认更改的信息无误后，点击**保存**，即可更改版本信息。

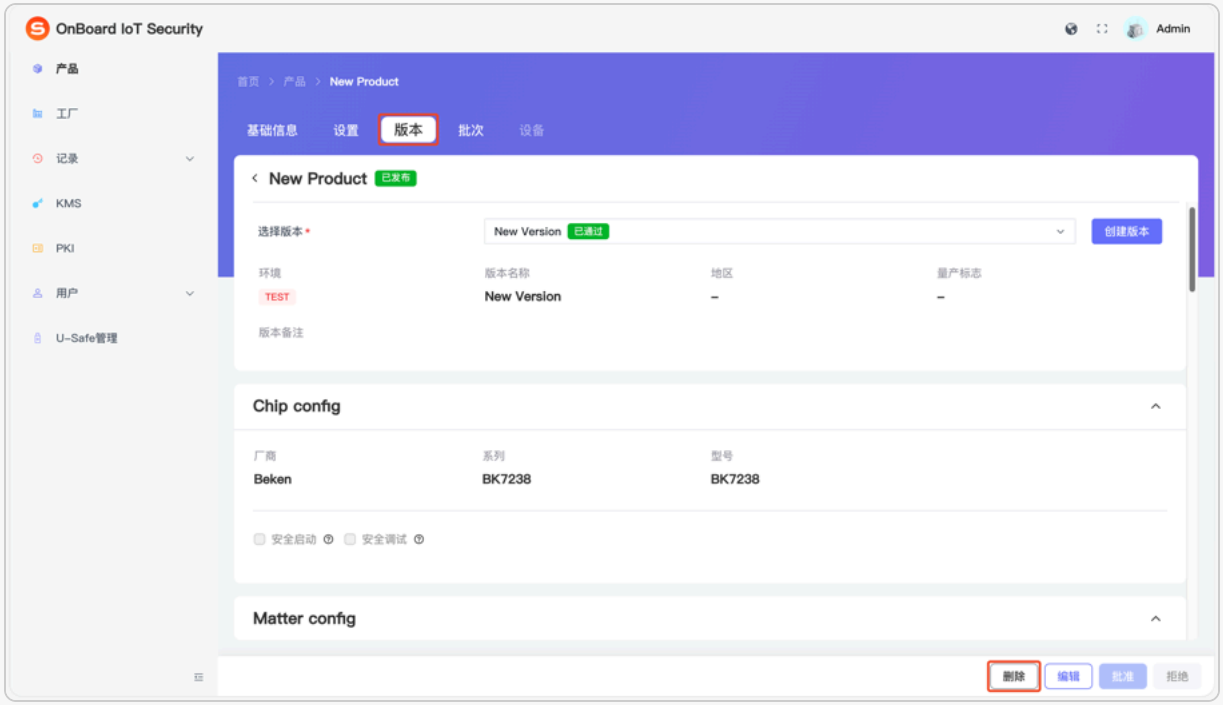
删除版本

> **提示**
>
> 删除版本前请确保：
>
> - 此版本状态不是**待审核**（如果版本状态是**待审核**，此版本将无法删除）。
> - 此版本未关联任何批次（如果使用此版本创建批次后，此版本将无法删除）。

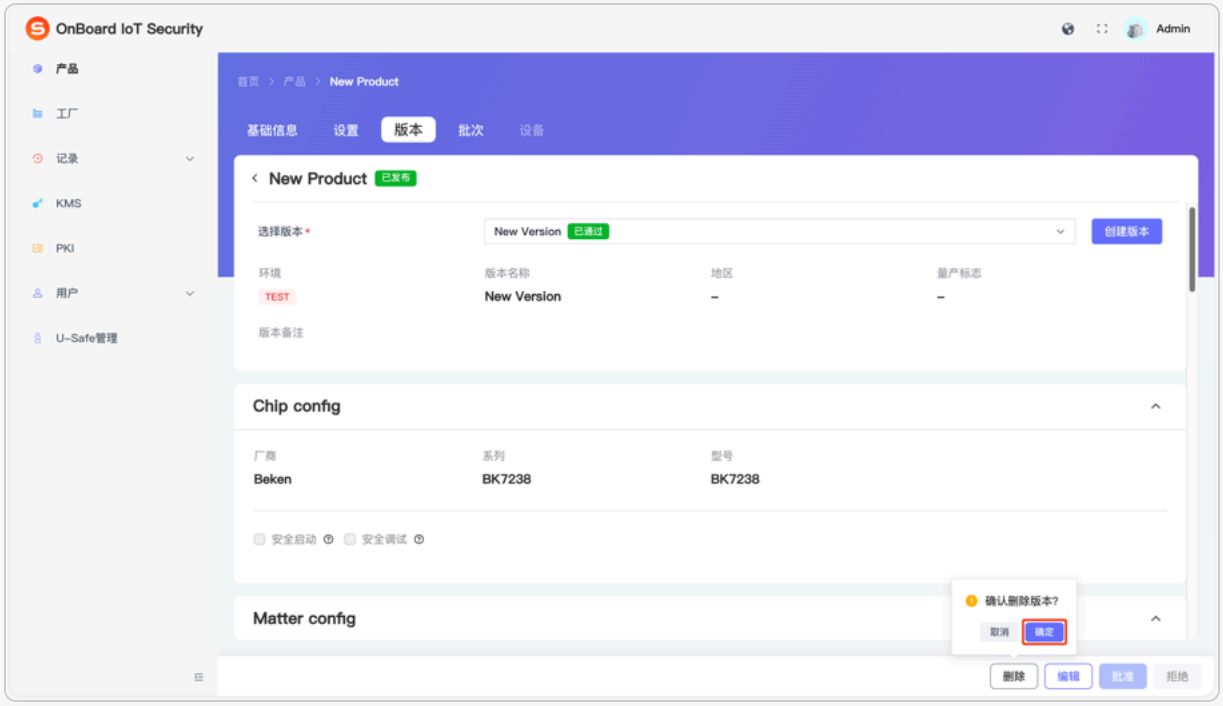
1. 登录 OBIS。

2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。

3. 选择**版本**标签页并选择需要删除的版本，然后点击页面底部的**删除**。



4. 在弹出的窗口中点击**确定**，即可删除该版本。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product
基础信息
设置
版本
批次
设备
New Product
已发布
选择版本 *
New Version
已通过
创建版本
环境
TEST
版本名称
New Version
地区
-
量产标志
-
版本备注
Chip config
厂商
Beken
系列
BK7238
型号
BK7238
安全启动
安全调试
Matter config
确认删除版本?
取消
确定
删除
编辑
批准
拒绝

版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建批次

> **信息**
>
> 目前批次创建后，无法修改和删除。

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中创建批次，后续可为该批次分配烧录站（本地工厂 & 云工厂）用于烧录。每个版本可以创建多个批次，而批次则代表基于该版本的生产计划。

前提条件

在开始之前，请确保：

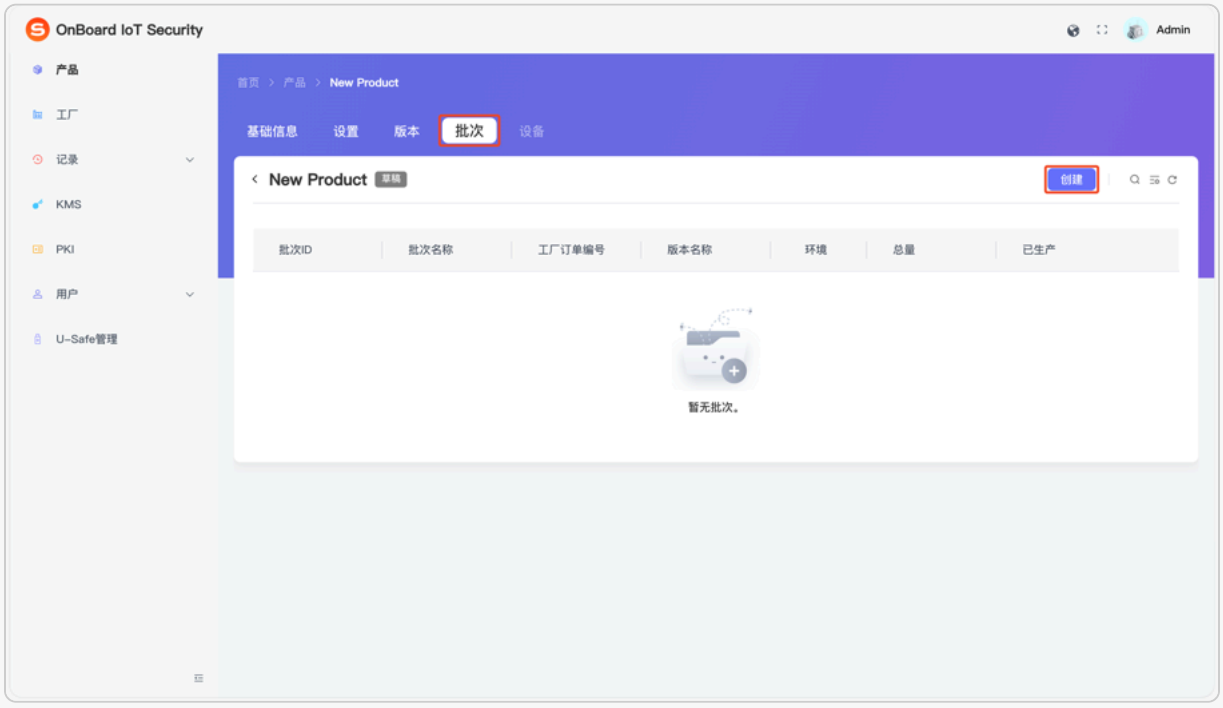
- 已完成版本的创建，并且版本状态为**已通过**。（如果在创建产品时开启了版本审核，PROD 环境下的版本创建后将需要 Product Owner 或 Product Manager 进行审核）
- 你已拥有对应产品的产品角色权限：Product Owner，Product Manager 或 Batch Manager。
- 已完成工厂的创建和添加 HSM U-Safe（本地工厂），并且工厂状态为**正常**。

操作步骤

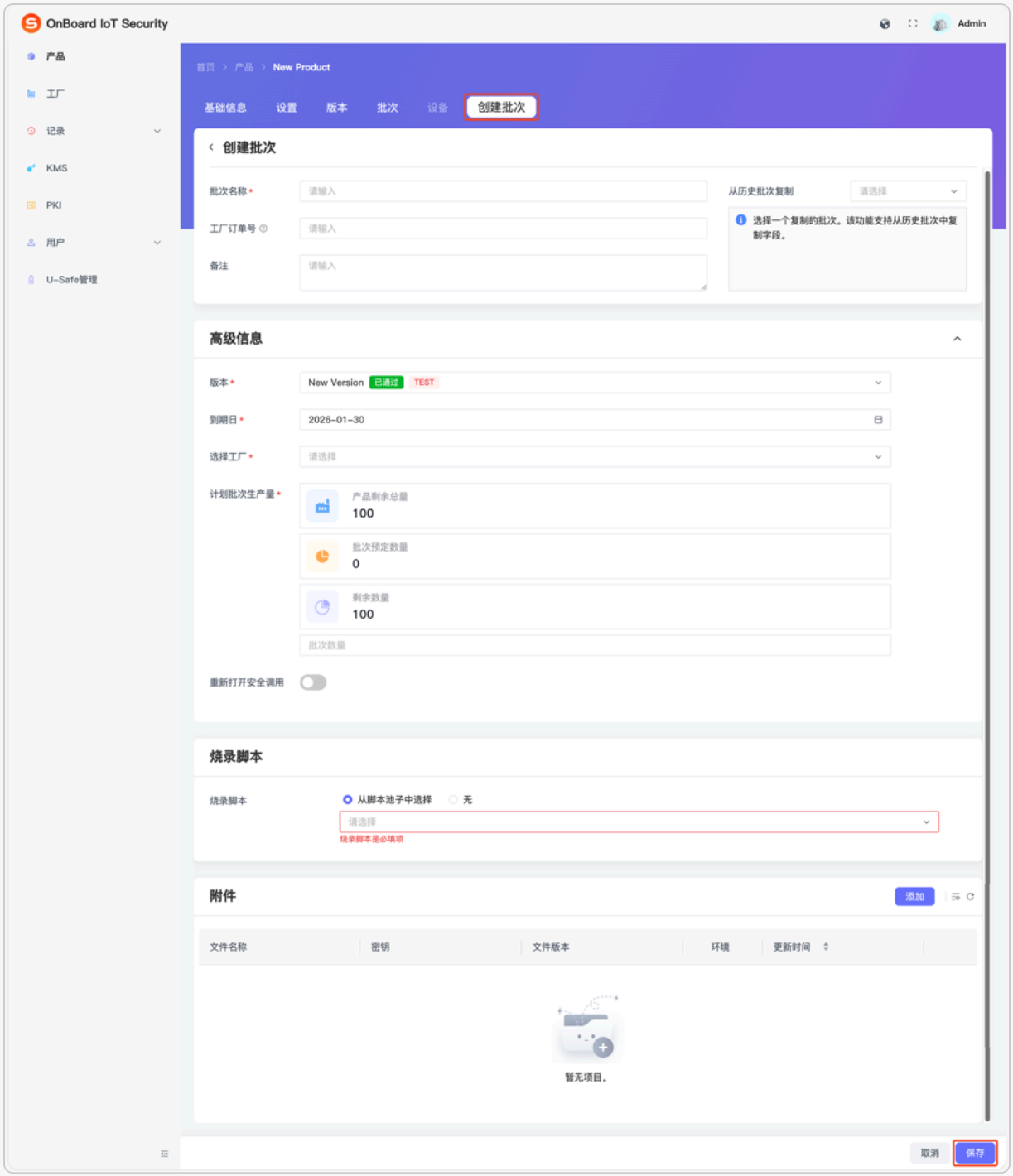
1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。

3. 在产品页面，选择**批次**标签页并点击**创建**。页面会跳转至**创建批次**标签页。

> **提示**
>
> 如果**批次**标签页显示为灰色，表示当前不可用。请检查版本是否已创建完成。创建版本后，**批次**标签页将自动启用。



4. 在**创建批次**标签页，选择并输入批次信息。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product
基础信息
设置
版本
批次
设备
创建批次
创建批次
批次名称 *
请输入
工厂订单号
请输入
备注
请输入
从历史批次复制
请选择
选择一个复制的批次。该功能支持从历史批次中复制字段。
高级信息
版本 *
New Version
已通过
TEST
到期日 *
2026-01-30
选择工厂 *
请选择
计划批次生产量 *
产品剩余总量
100
批次预定数量
0
剩余数量
100
批次数量
重新打开安全调用
烧录脚本
烧录脚本
从脚本池子中选择
无
请选择
烧录脚本是必填项
附件
添加
文件名称
密钥
文件版本
环境
更新时间
暂无项目。
取消
保存

批次信息说明:

信息	是否必填	说明
批次名称	是	批次名称必须唯一。
工厂订单号	否	用于关联特定的工厂订单。
备注	否	用于补充说明批次相关的额外信息，如特殊要求，注意事项等。
从历史批次复制	否	可以选一个历史批次作为模板修改。
版本	是	选择状态为**已通过**的版本。
到期日	是	指定批次的有效期或完成期限。 **注释**：到期后的批次无法再使用。在 OnBoard Edge Service 中，批次状态会显示为**已过期**。可选时间请参考创建产品时设置的批次有效时长。
选择工厂	是	选择状态为**正常**的工厂。 **注释**：批次数据会同步更新到工厂的批次列表中。

信息	是否必填	说明
计划批次生产量	是	设定批次数量。上方显示**产品剩余总量**（创建产品时配置的最大生产量），**批次预订数量**和**剩余数量**。 ◦ 如果此批次关联 TEST 环境的版本，此数量不会从总量中扣除 ◦ 如果此批次关联 PROD 环境的版本，此数量将被扣除（如果批次过期，未生产的数量也不会回收）。并且，**批次数量**必须小于或等于**产品剩余总量**。
重新打开安全调用	否	此功能通过调用已配置的 Secure Debug Key 解锁芯片调试接口，并擦除芯片中的原有数据，从而实现芯片的再次烧录。 **注释**：在配置产品时，勾选并启用**安全调试**功能才能在创建批次时启用**重新打开安全调用**功能。关于启用**安全调试**功能，请参考使能设置项说明。
烧录脚本	是	你可以根据实际需求选择在烧录站文件区域配置的 Programming script；如果暂不需要，可选择**无**，后续可在 Programming Station 客户端中为特定批次选择烧录脚本。
附件	否	你可以根据实际需求选择在烧录站文件区域配置的 Attachment Files。

5. 确认所有信息无误后，点击页面底部的**保存**，即可完成批次的创建。

提示

批次创建后会自动下发到工厂。如果下发失败，可在产品的**批次**标签页，点击对应批次的选项 ⋮ 图标并选择**重新下发**。更多信息请参考批次状态和批次同步状态说明。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看批次

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看批次的信息和配置。

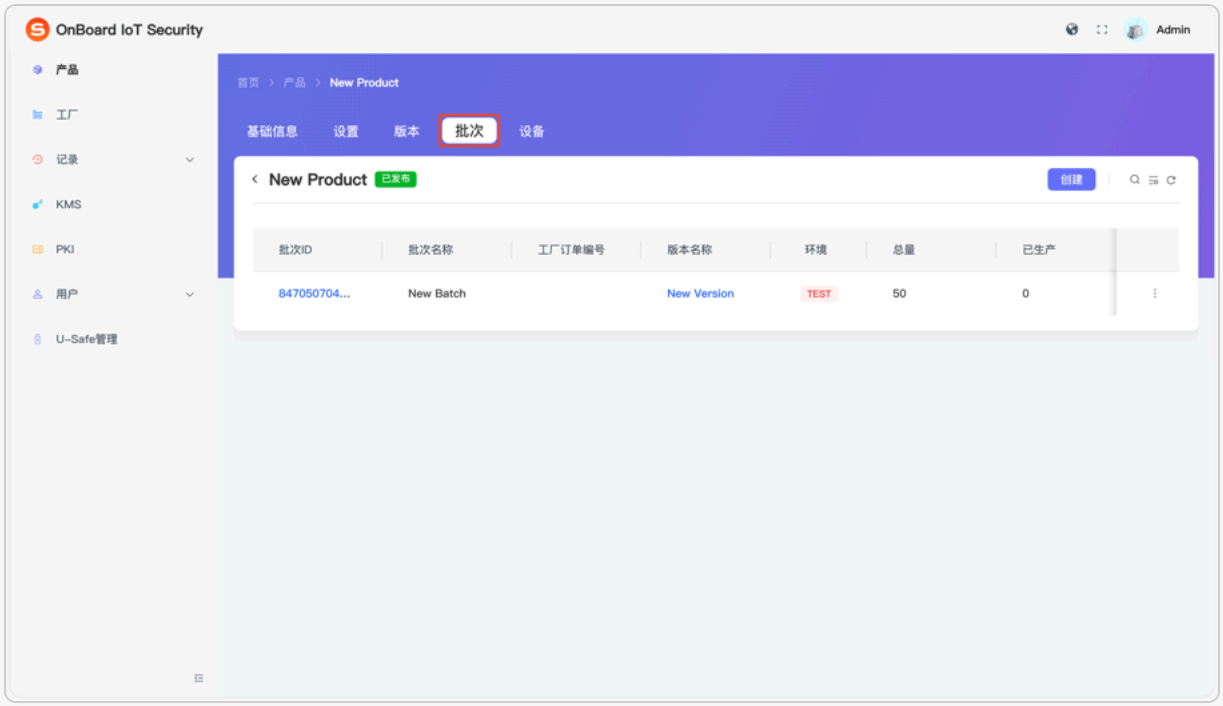
前提条件

在开始之前，请确保你已拥有对应产品的产品角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在产品页面，选择**批次**标签页即可查看已创建的批次。点击对应的批次 ID，进入批次页面。

> **提示**
>
> 目前支持使用以下条件搜索批次：**批次 ID**，**批次名称**，**版本名称**，**环境**，**创建时间**，**批次状态**，**批次同步状态**和**烧录数据同步状态**。
>
> 点击搜索图标可以显示或隐藏搜索条件。



批次状态说明：

状态	说明
生成中	批次创建后正在生成时的状态。
生成失败	批次生成失败后的状态。可尝试重新生成：在产品的**批次**标签页，点击对应批次的选项 ⋮ 图标并选择**重新下发**。
已生成	批次成功生成后的状态。
运行中	批次成功下发到工厂后的状态。

状态	说明
已停止	批次被关闭后的状态。
已完成	批次完成烧录后的状态。
已过期	批次在完成烧录前已超过有效期时的状态。

批次同步状态说明：

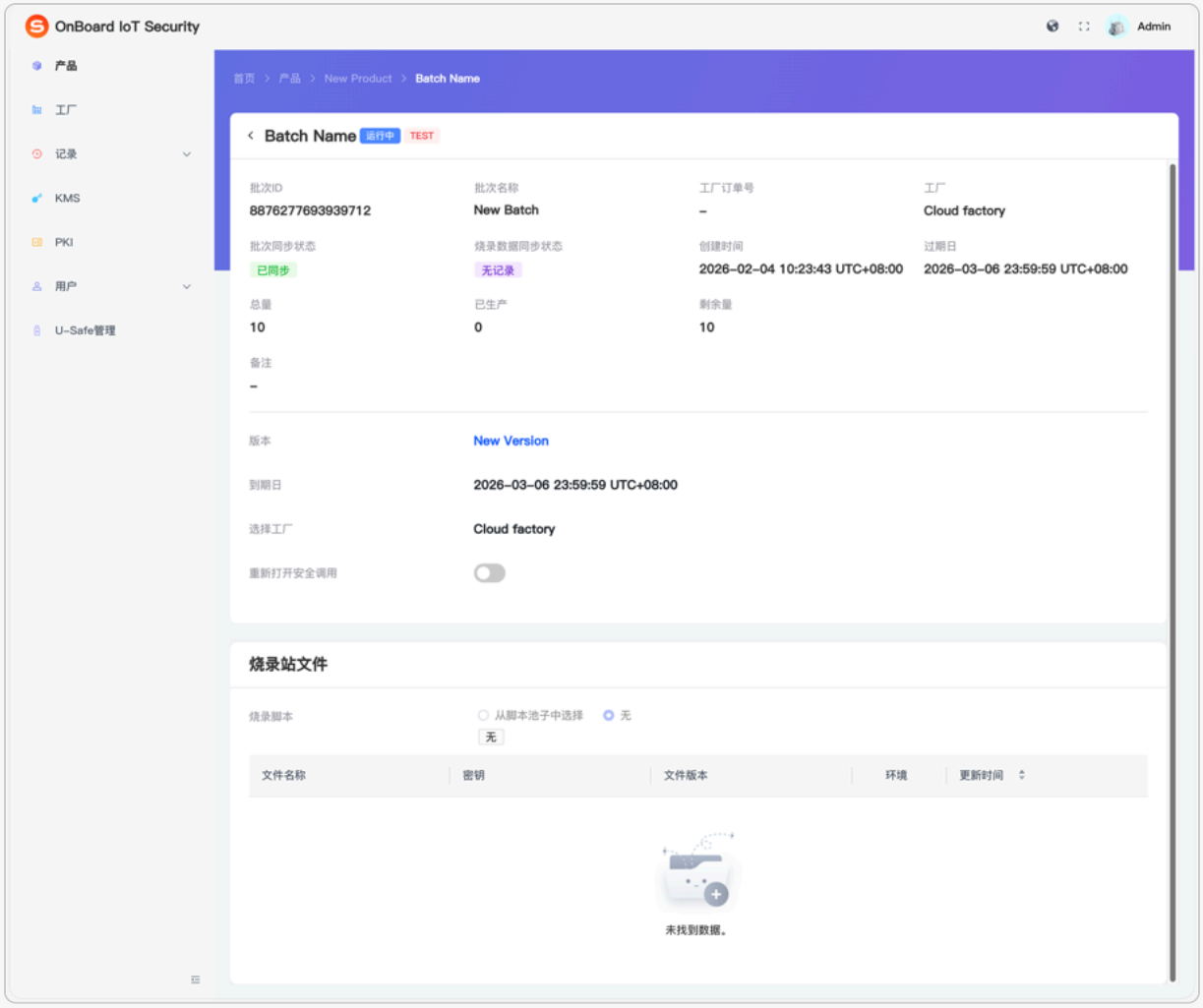
状态	说明
等待中	批次成功生成前的状态。
下发中	批次成功生成后，开始下发到工厂时的状态。
下发失败	批次下发到工厂失败后的状态。可尝试重新下发：在产品的**批次**标签页，点击对应批次的选项 ⋮ 图标并选择**重新下发**。

状态	说明
已同步	批次成功下发到工厂后的状态。
已删除	批次被删除后的状态。

烧录数据同步状态说明：

状态	说明
无记录	批次尚未开始烧录前的状态。
上报中	批次已开始烧录但尚未完成，正在上传烧录数据时的状态。
已上报	批次已完成烧录，且烧录数据都已全部上传后的状态。

4. 在批次页面，即可查看批次相关信息（**批次名称**，**版本**和**到期日**等）。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > 产品 > New Product > Batch Name
Batch Name 运行中 TEST
批次ID
8876277693939712
批次名称
New Batch
工厂订单号
–
工厂
Cloud factory
批次同步状态
已同步
烧录数据同步状态
无记录
创建时间
2026-02-04 10:23:43 UTC+08:00
过期日
2026-03-06 23:59:59 UTC+08:00
总量
10
已生产
0
剩余量
10
备注
–
版本
New Version
到期日
2026-03-06 23:59:59 UTC+08:00
选择工厂
Cloud factory
重新打开安全调用
烧录站文件
烧录脚本
从脚本池子中选择
无
无
文件名称
密钥
文件版本
环境
更新时间
未找到数据。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看设备烧录

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看租户内烧录设备的信息从而了解设备的档案和日志。

前提条件

在开始之前，请确保：

- 已完成设备烧录。
- 你已拥有对应产品的产品角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**产品**，然后点击对应的产品名称，进入产品页面。
3. 在产品页面，选择**设备**标签页，即可查看设备烧录信息。
4. 设置搜索条件并点击**搜索**，设备历史列表中将返回所有符合条件的设备历史。

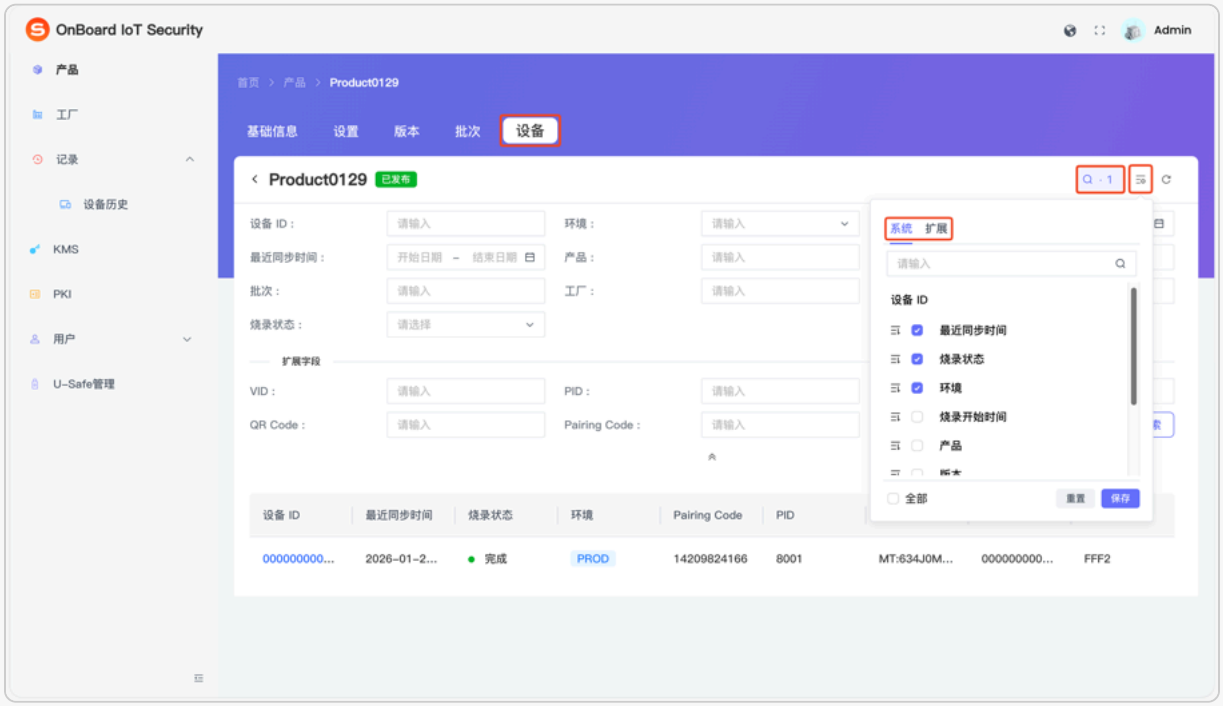
提示

目前支持使用以下条件搜索设备历史记录：

- 系统字段：**设备 ID，环境，烧录开始时间，最近同步时间，产品，版本，批次，工厂，烧录站**和**烧录状态**
- 扩展字段：**VID，PID，Chip UID，QR Code** 和 **Pairing Code**

点击搜索 图标可以隐藏或显示搜索条件。图标右侧的数字表示当前已设置的搜索条件个数。

同时，点击列选项 图标可以自定义显示哪些列以及列的排列顺序。



设备烧录信息说明：

信息	说明
设备 ID	设备标识符。
环境	版本对应的环境，例如，**TEST** 环境和 **PROD** 环境。
烧录开始时间	设备开始烧录的日期时间，格式为 **YYYY-MM-DD HH:MM:SS**。
最近同步时间	设备档案最近同步的日期时间，格式为 **YYYY-MM-DD HH:MM:SS**。
产品	设备对应的产品名称。
版本	设备对应的版本名称。
批次	设备对应的批次名称。
工厂	设备对应的工厂名称。
烧录站	设备对应的烧录站名称。
烧录状态	设备结果的状态，例如，**未上报**，**完成**和**失败**。
VID	供应商标识符。
PID	产品标识符。
Chip UID	设备内部芯片的唯一标识符。
QR Code	设备二维码，用于实现安全快速的配网。

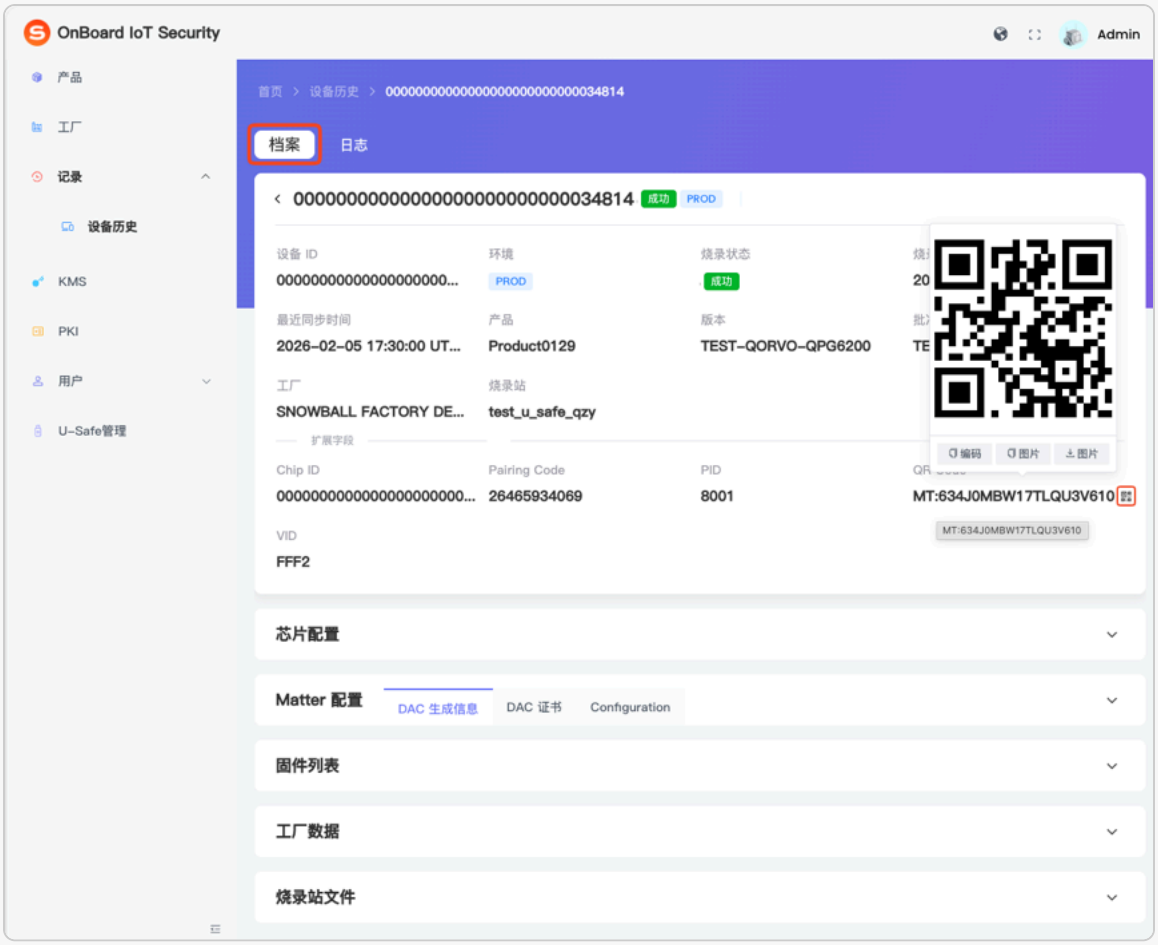
信息	说明
Pairing Code	用户手动输入以完成设备配对的临时验证码。

5. 点击对应的设备 ID，进入设备页面，即可查看设备相关信息。

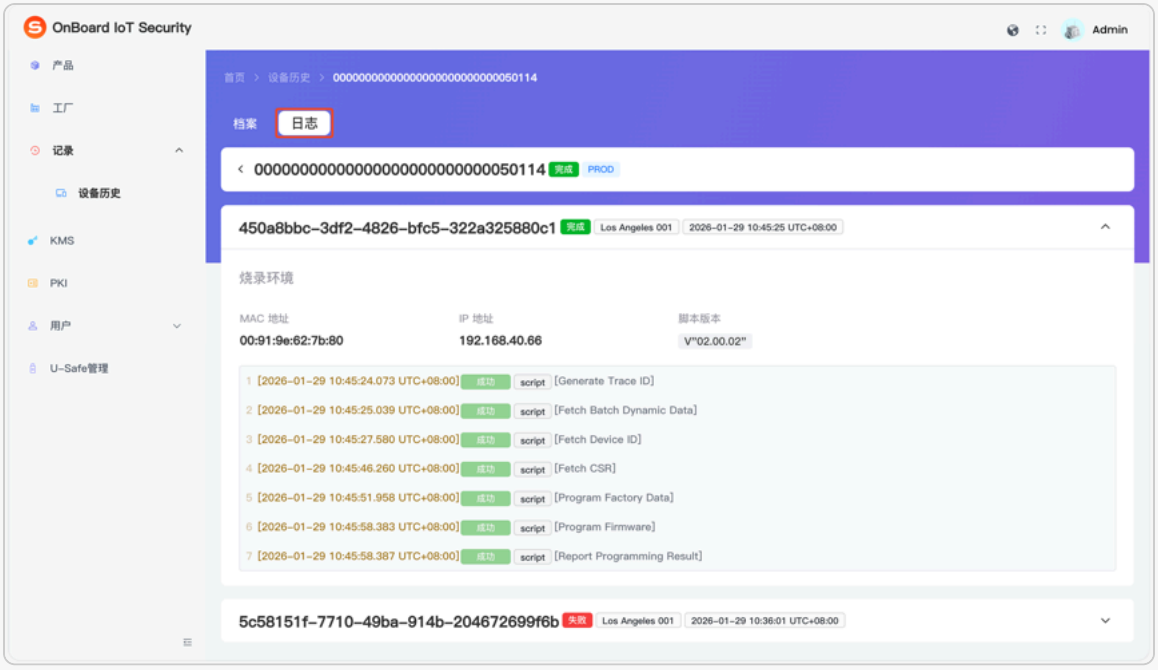
    - **档案**标签页显示了设备的档案数据，可以查看设备的完整信息。

        > **提示**
        >
        > 将鼠标悬停在 图标上，即可显示设备二维码，该二维码用于实现安全快速的配网。
        >
        > - **DAC 生成信息**区域显示了 DAC 相关信息（**DAC 序列号**，**生效日期**和**失效日期**等）。**DAC 序列号**可用于在证书中吊销对应设备的认证状态。
        > - **DAC 证书**区域显示了证书的编码。可通过解析编码提取出证书的颁发者信息，有效期，公钥等关键证书属性。

        - 系统字段和扩展字段
        - 芯片配置
        - Matter 配置（DAC 生成信息，DAC 证书和 Configuration）
        - 固件列表
        - 工厂数据
        - 烧录站文件



- **日志**标签页显示了设备的的烧录日志（每次烧录的日志都会被保存），可以排查故障和定位烧录问题。


OnBoard IoT Security
Admin
产品
工厂
记录
设备历史
KMS
PKI
用户
U-Safe管理
首页 > 设备历史 > 00000000000000000000000000050114
档案
日志
00000000000000000000000000050114 完成 PROD
450a8bbc-3df2-4826-bfc5-322a325880c1 完成 Los Angeles 001 2026-01-29 10:45:25 UTC+08:00
烧录环境
MAC 地址
00:91:9e:62:7b:80
IP 地址
192.168.40.66
脚本版本
V"02.00.02"
1 [2026-01-29 10:45:24.073 UTC+08:00] 成功 script [Generate Trace ID]
2 [2026-01-29 10:45:25.039 UTC+08:00] 成功 script [Fetch Batch Dynamic Data]
3 [2026-01-29 10:45:27.580 UTC+08:00] 成功 script [Fetch Device ID]
4 [2026-01-29 10:45:46.260 UTC+08:00] 成功 script [Fetch CSR]
5 [2026-01-29 10:45:51.958 UTC+08:00] 成功 script [Program Factory Data]
6 [2026-01-29 10:45:58.383 UTC+08:00] 成功 script [Program Firmware]
7 [2026-01-29 10:45:58.387 UTC+08:00] 成功 script [Report Programming Result]
5c58151f-7710-49ba-914b-204672699f6b 失败 Los Angeles 001 2026-01-29 10:36:01 UTC+08:00

版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建工厂

> **信息**
>
> 目前工厂创建后无法被删除。

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中创建工厂从而配置工厂的基础信息，为后续配置工厂管理员，HSM U-Safe，Identity U-Safe 和烧录站做准备。

工厂的类型

OBIS 支持创建本地工厂和云工厂：

- 本地工厂：需要在工厂侧部署 OnBoard Edge Service（通常部署在工厂本地服务器上），从而提供工厂本地端的生产管理能力。每个本地工厂均对应一个 OnBoard Edge Service。批次信息同步至 OnBoard Edge Service 后，工厂可执行离线烧录，烧录过程无需实时联网。
- 云工厂：生产管理能力由云端提供，烧录过程中需保持实时联网，以保证批次信息下发、生产数据回传及状态同步。

> **提示**
>
> 你可以联系雪球协助完成 OnBoard Edge Service 的部署。

前提条件

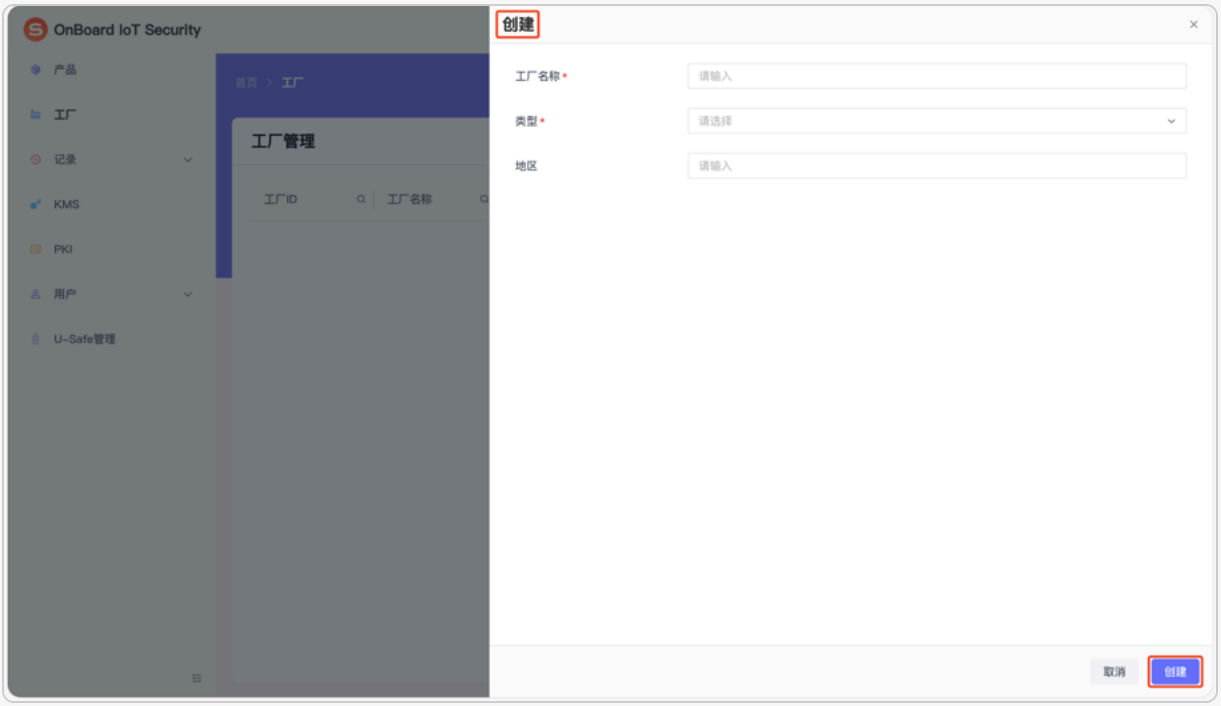
在开始之前，请确保你已拥有 System Manager 或 Factory Manager 的 OBIS 角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击**创建**。



3. 在**创建**面板，选择并输入工厂信息。



工厂基础信息说明：

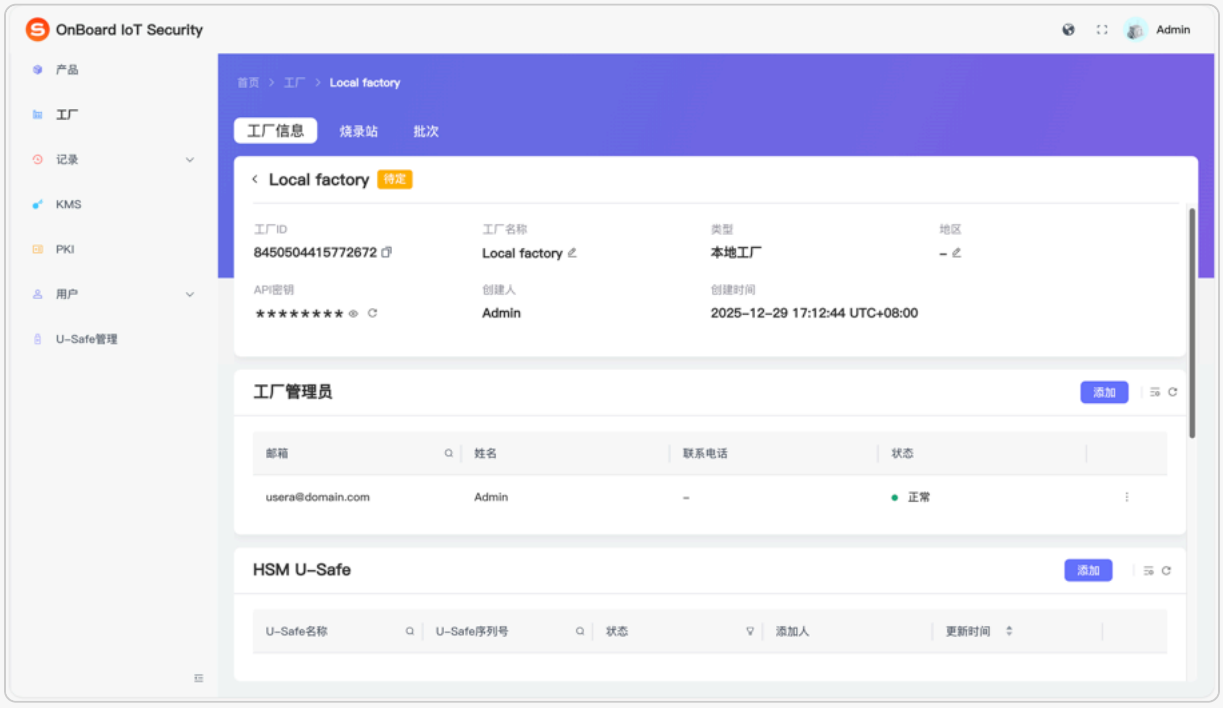
信息	是否必填	说明
工厂名称	是	工厂名称必须唯一。
类型	是	有两种类型可以选择，分别是**本地工厂**和**云工厂**。
地区	否	建议填写工厂地址。

4. 点击**创建**，即可完成工厂的创建。创建完成后，在**工厂管理**页面，点击对应的工厂ID，即可在工厂页面中查看工厂相关信息。

提示

工厂创建后，你仅可在工厂页面中修改**工厂名称**和**地区**。

- 本地工厂的初始状态为**待定**，在本地工厂添加 HSM U-Safe 后，本地工厂的状态会由**待定**变更为**正常**。更多信息请参考工厂状态说明。
- 云工厂的初始状态为**正常**。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中配置和管理工厂

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

- 添加工厂管理员
- 重置工厂管理员密码
- 删除工厂管理员
- 添加 HSM U-Safe
- 解绑，锁定或解锁 HSM U-Safe
- 添加 Identity U-Safe
- 解绑，锁定或解锁 Identity U-Safe
- 锁定或解锁工厂

前提条件

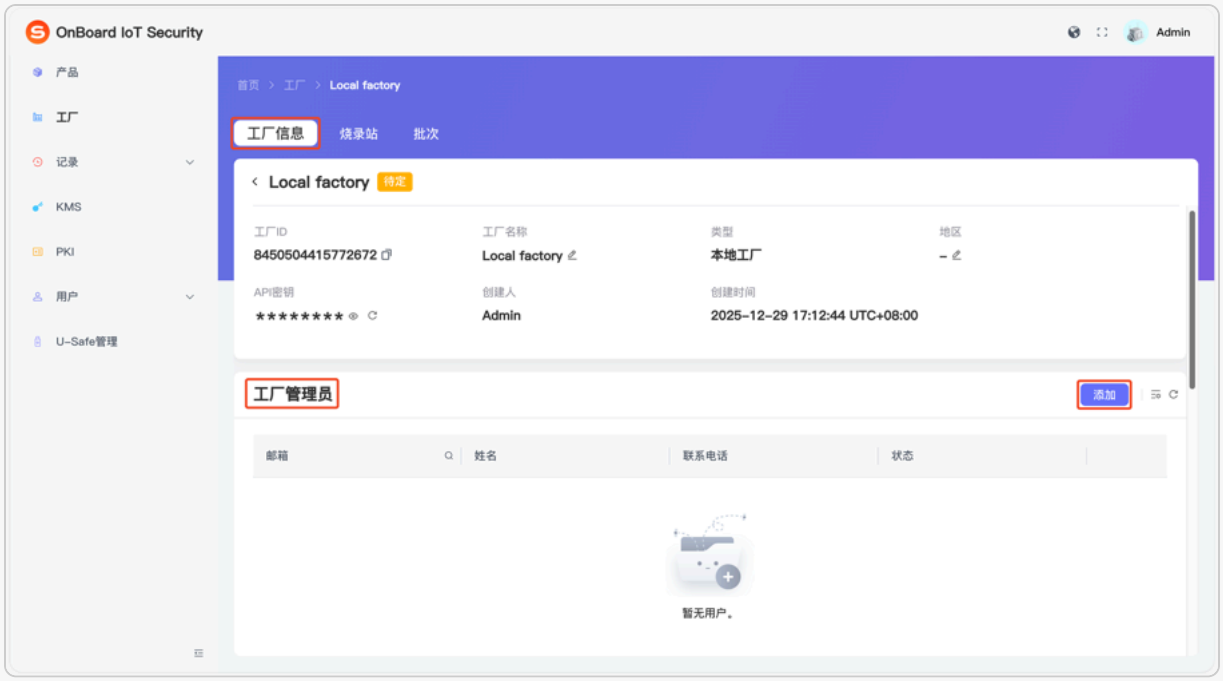
在开始之前，请确保：

- 你已拥有对应工厂的 Factory Manager 角色权限。
- 已从雪球获取用于身份鉴权的 HSM U-Safe 和 Identity U-Safe。

添加工厂管理员

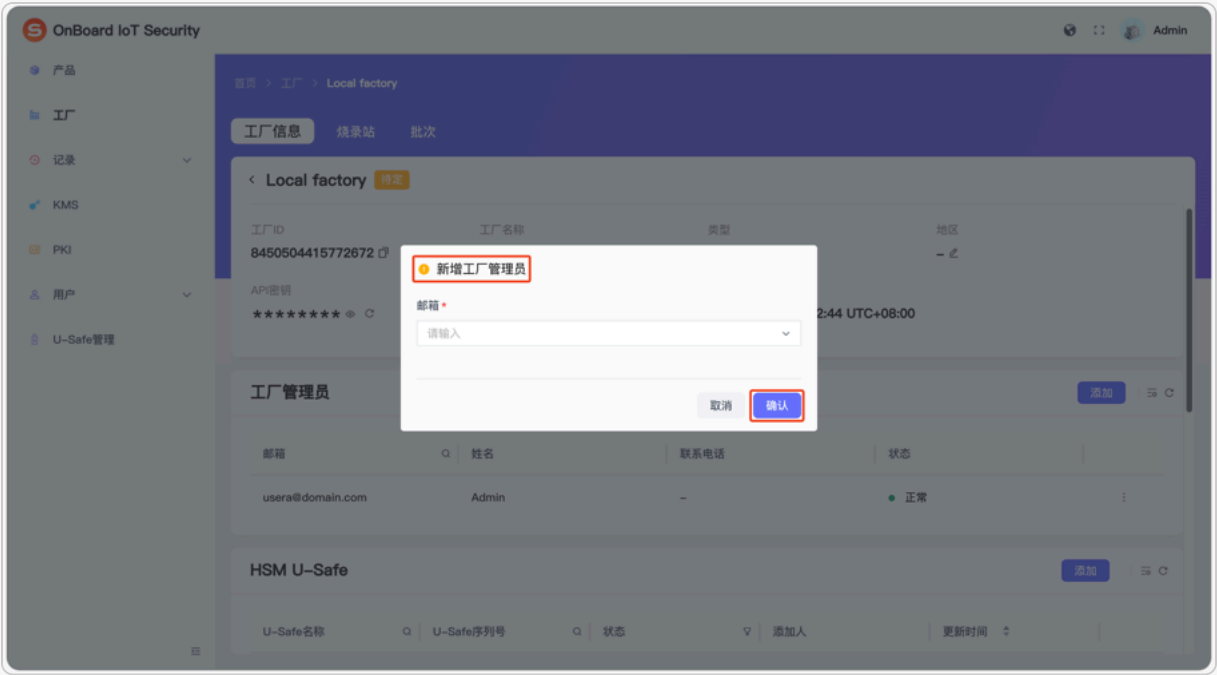
> **提示**
>
> 工厂管理员可以管理和操作工厂相关业务，并使用 "邮箱 + 密码" 或 "邮箱 + 验证码" 的方式登录到 OnBoard Edge Service。
>
> **注释：**
>
> - 在断开外网环境下，OnBoard Edge Service 仅支持使用 "邮箱 + 密码" 的方式进行身份验证。
> - 在添加工厂管理员后，系统会自动生成初始密码并发送到对应的邮箱。目前，该随机密码可以重置但暂不支持自定义修改。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，找到**工厂管理员**区域，点击**添加**。



4. 在**新增工厂管理员**窗口，选择并输入新增工厂管理员的邮箱地址。

> **提示**
>
> 被添加的工厂管理员必须是同一租户内状态为**启用**的用户。

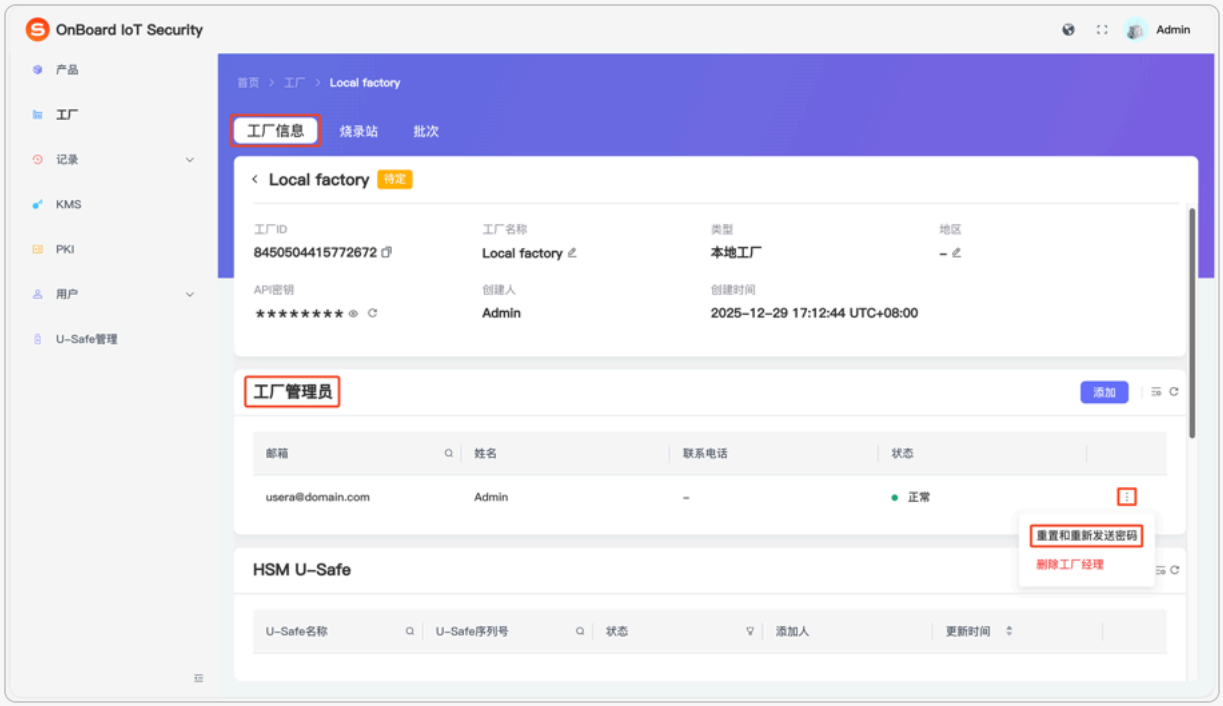


5. 点击**确认**，即可完成工厂管理员的添加。

重置工厂管理员密码

> **提示**
>
> 密码重置后，系统会自动生成随机密码并发送到对应的邮箱。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，找到**工厂管理员**区域，点击对应工厂管理员的选项 ⋮ 图标并选择**重置和重新发送密码**。



4. 在**确认重置**窗口中点击**确认**，即可重置工厂管理员密码。

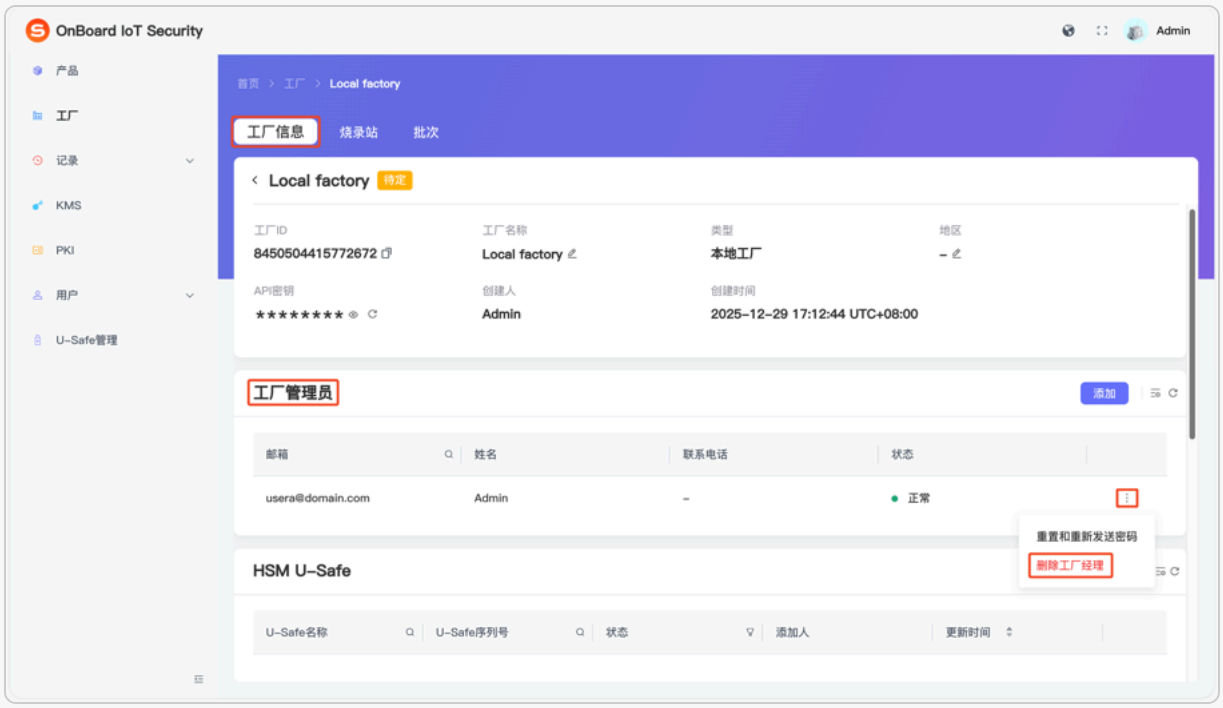


删除工厂管理员

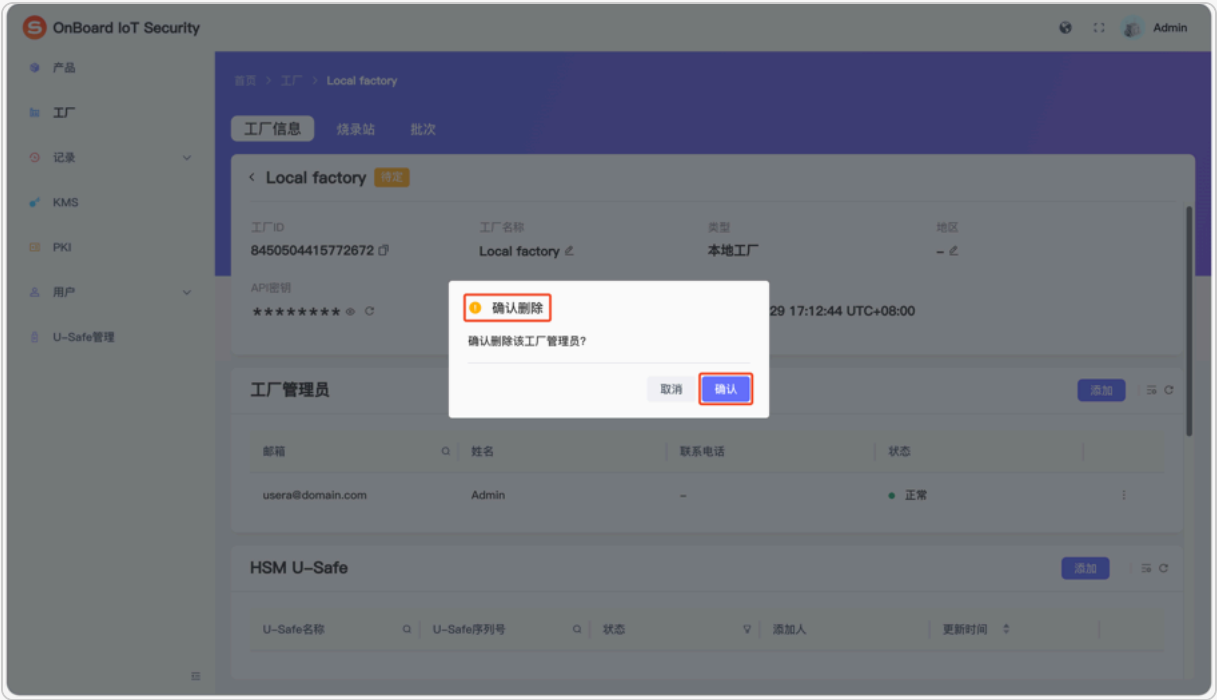
1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，找到**工厂管理员**区域，点击对应工厂管理员的选项 ⋮ 图标并选择**删除工厂经理**。



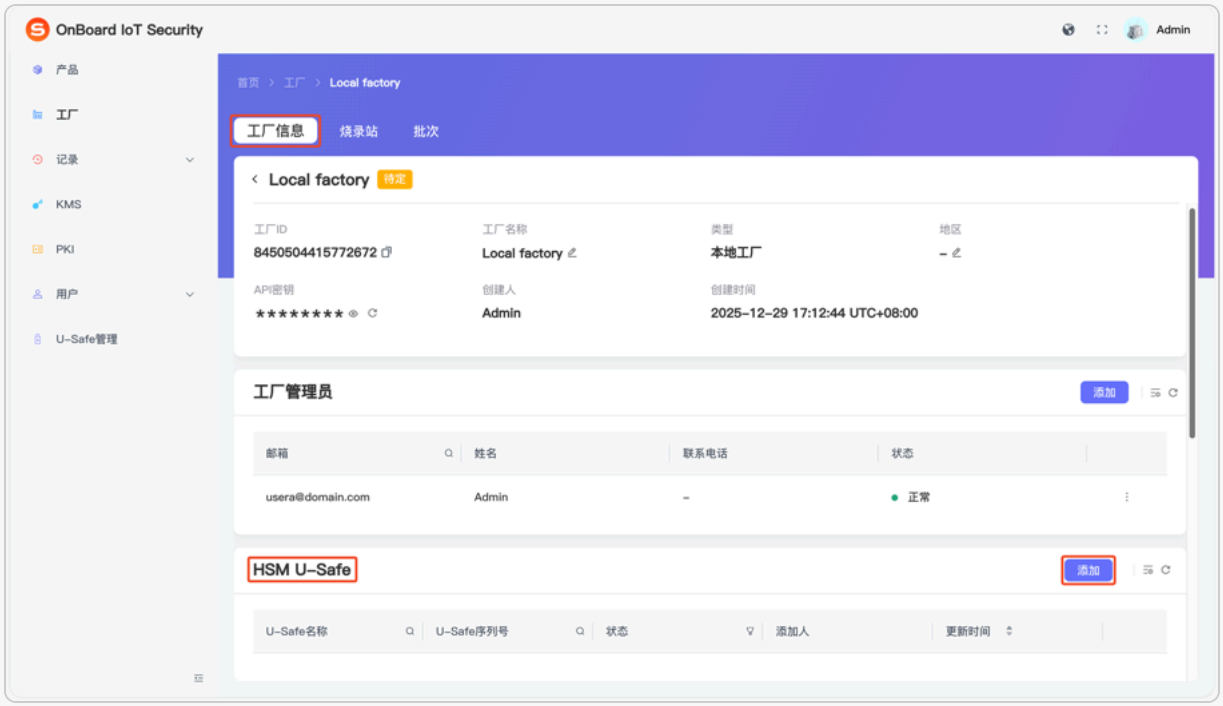
4. 在**确认删除**窗口中点击**确认**，即可删除该工厂管理员。



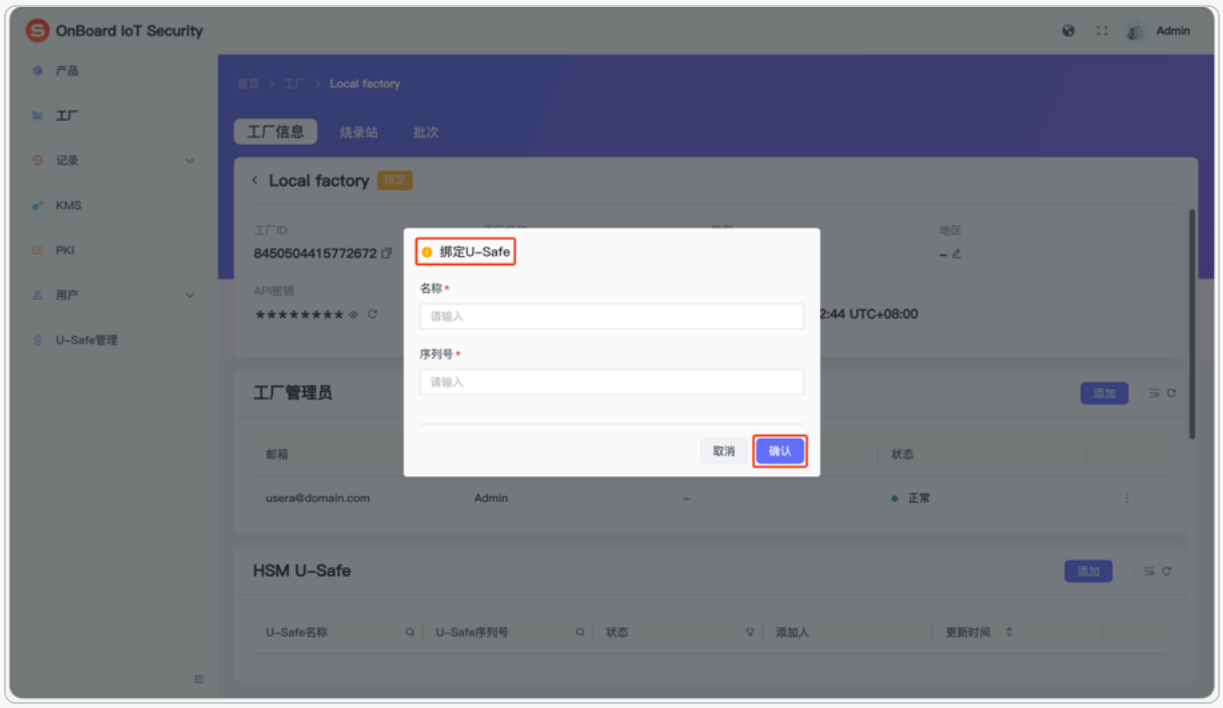
添加 HSM U-Safe

> **提示**
>
> 添加 HSM U-Safe 的目的是为了将其与对应的工厂进行绑定，并在后续用于登录 OnBoard Edge Service。同时，通过管理 HSM U-Safe 从而实现密钥的安全存储与设备烧录支持。
>
> 添加 HSM U-Safe 后，工厂状态会由**待定**变更为**正常**。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，找到 **HSM U-Safe** 区域，点击**添加**。



4. 在**绑定 U-Safe** 窗口中，输入需要绑定的 HSM U-Safe 的**名称**（可自定义）和**序列号**（可在设备本体上查看）。



5. 点击**确认**，即可绑定 HSM U-Safe。

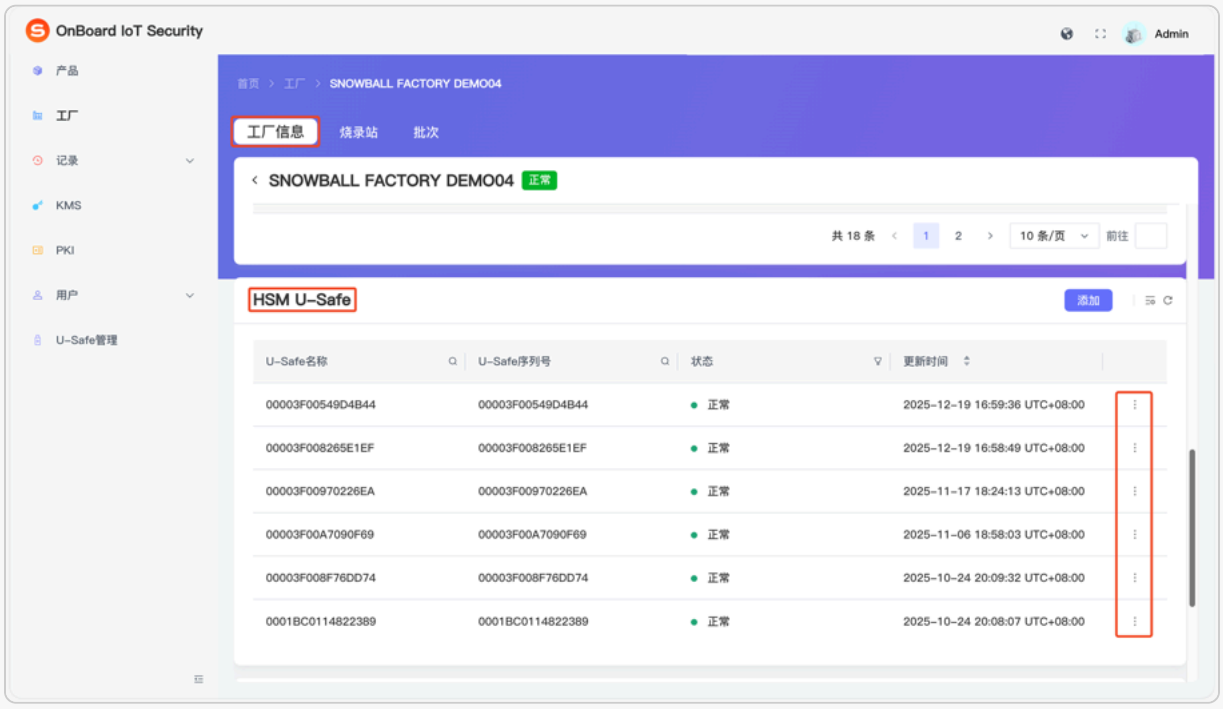
解绑，锁定或解锁 HSM U-Safe

> **提示**
>
> 锁定 HSM U-Safe 以禁止其使用和解绑。锁定后，状态会变更为**锁定**。

1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，找到 **HSM U-Safe** 区域，点击对应 HSM U-Safe 的选项 ⋮ 图标并选择**解绑**，**锁定**或**解锁**。

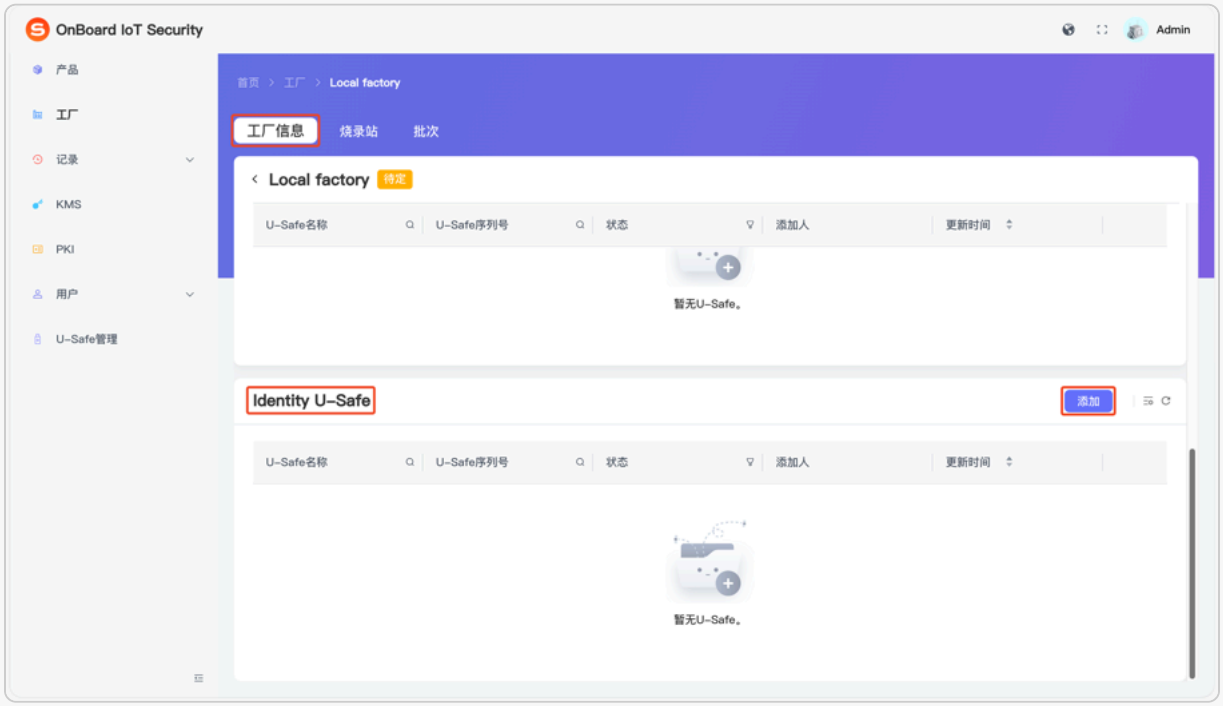


4. 在弹出的窗口中点击**确认**，即可解绑，锁定或解锁该 HSM U-Safe。

添加 Identity U-Safe

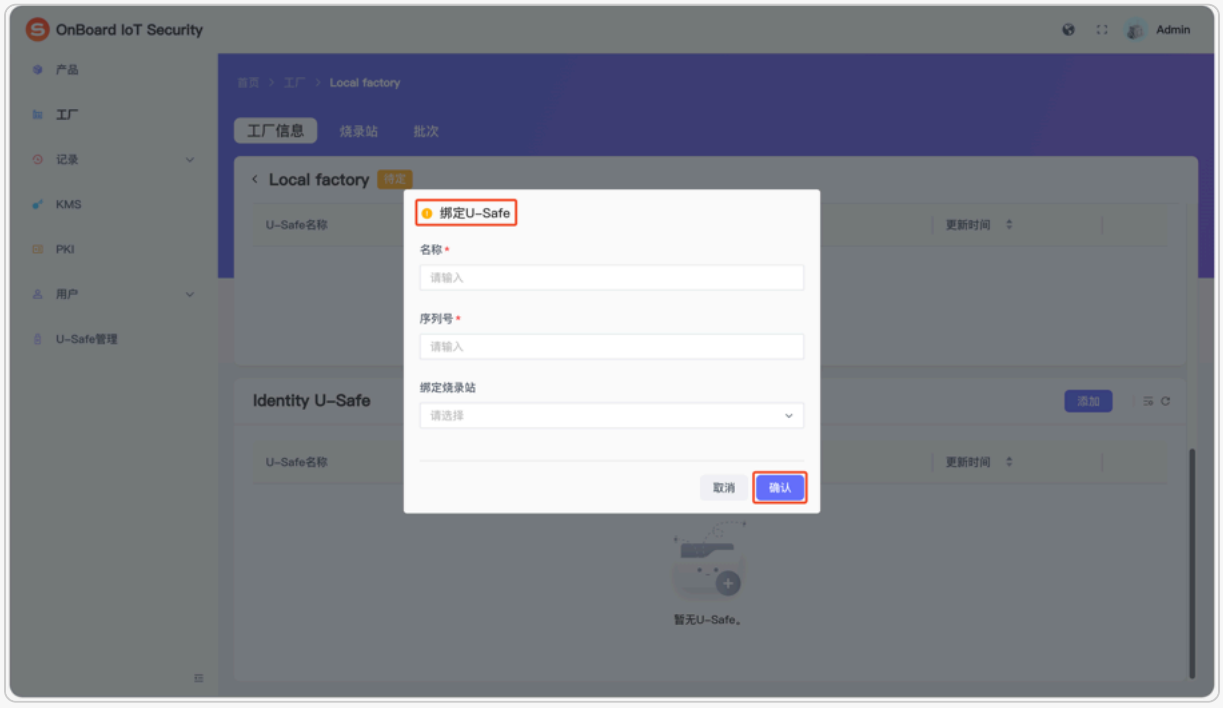
> **提示**
>
> 添加 Identity U-Safe 的目的是为了将其与对应的工厂进行绑定，并在 OBIS 或 OnBoard Edge Service 将其与烧录站进行绑定后，可用于登录 Programming Station。同时，通过管理 Identity U-safe 从而实现烧录站的登录认证。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，找到 **Identity U-Safe** 区域，点击**添加**。



4. 在**绑定 U-Safe** 窗口中，输入需要绑定的 Identity U-Safe 的**名称**（可自定义）和**序列号**（可在设备本体上查看）。

> **提示**
>
> - 如果已添加烧录站，可以选择并绑定对应的烧录站。
> - 如果还未添加烧录站，你可以在添加烧录站时进行绑定。



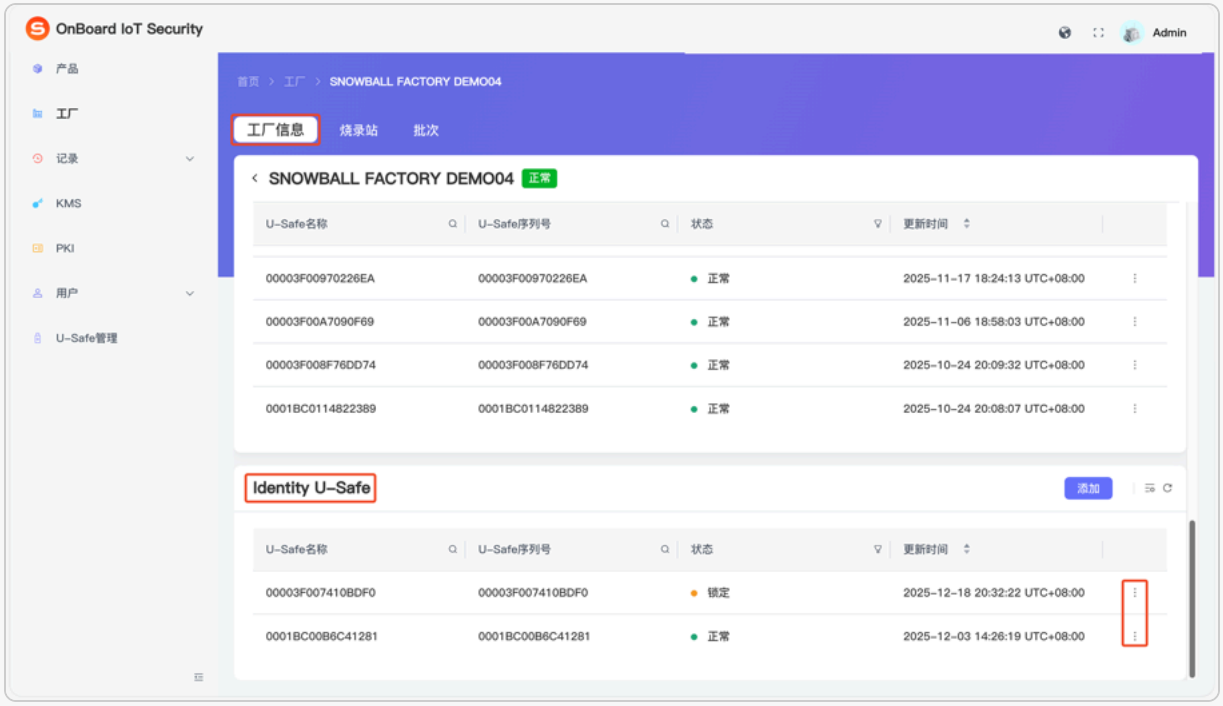
5. 点击**确认**，即可绑定 Identity U-Safe。

解绑，锁定或解锁 Identity U-Safe

> **提示**
>
> 锁定 Identity U-Safe 以禁止其使用和解绑。锁定后，状态会变更为**锁定**。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，找到 **Identity U-Safe** 区域，点击对应 Identity U-Safe 的选项 ⋮ 图标并选择**解绑**，**锁定**或**解锁**。



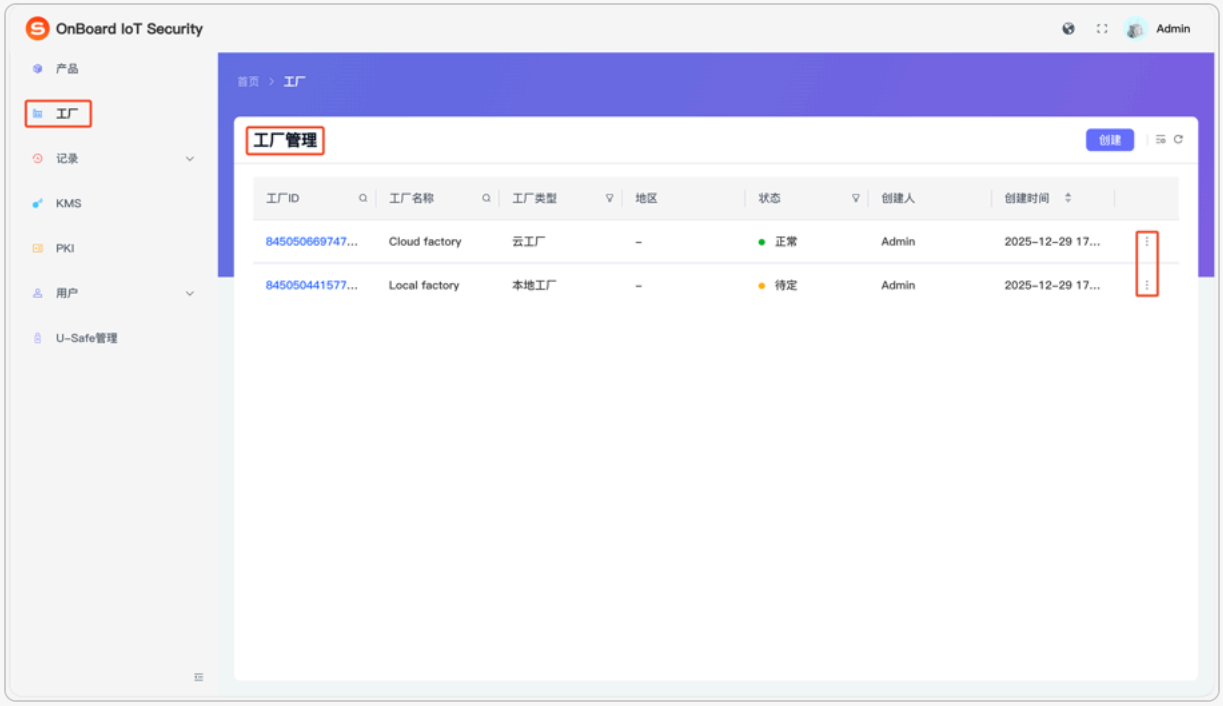
4. 在弹出的窗口中点击**确认**，即可解绑，锁定或解锁该 Identity U-Safe。

锁定或解锁工厂

> **提示**
>
> 锁定工厂以禁止其使用。锁定后，状态会变更为**锁定**。

1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，进入**工厂管理**页面，点击对应工厂的选项 ⋮ 图标并选择**锁定**或**解锁**。



3. 在弹出的窗口中点击**确认**，即可锁定或解锁该工厂。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中配置和管理烧录站

> **提示**
>
> 你可以根据实际需求，参考本文步骤在 OnBoard™ IoT Security (OBIS) 配置和管理烧录站，或者在 OnBoard Edge Service 配置和管理烧录站。

本文将指导你如何在 OBIS 中完成以下操作：

- 创建烧录站
- 为烧录站绑定 Identity U-Safe
- 为烧录站解绑 Identity U-Safe
- 删除烧录站
- 查看烧录站密码

前提条件

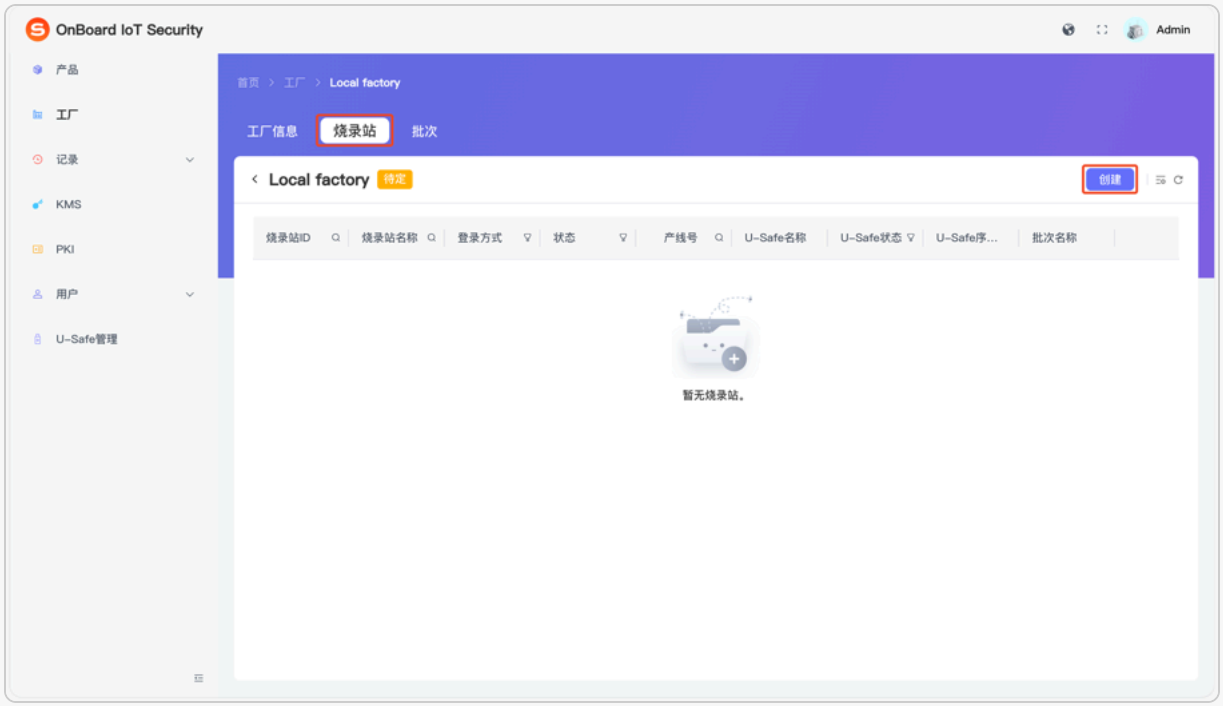
在开始之前，请确保：

- 已完成工厂的配置。
- 你已拥有对应工厂的 Factory Manager 角色权限。
- 已从雪球获取用于身份鉴权的 Identity U-Safe。

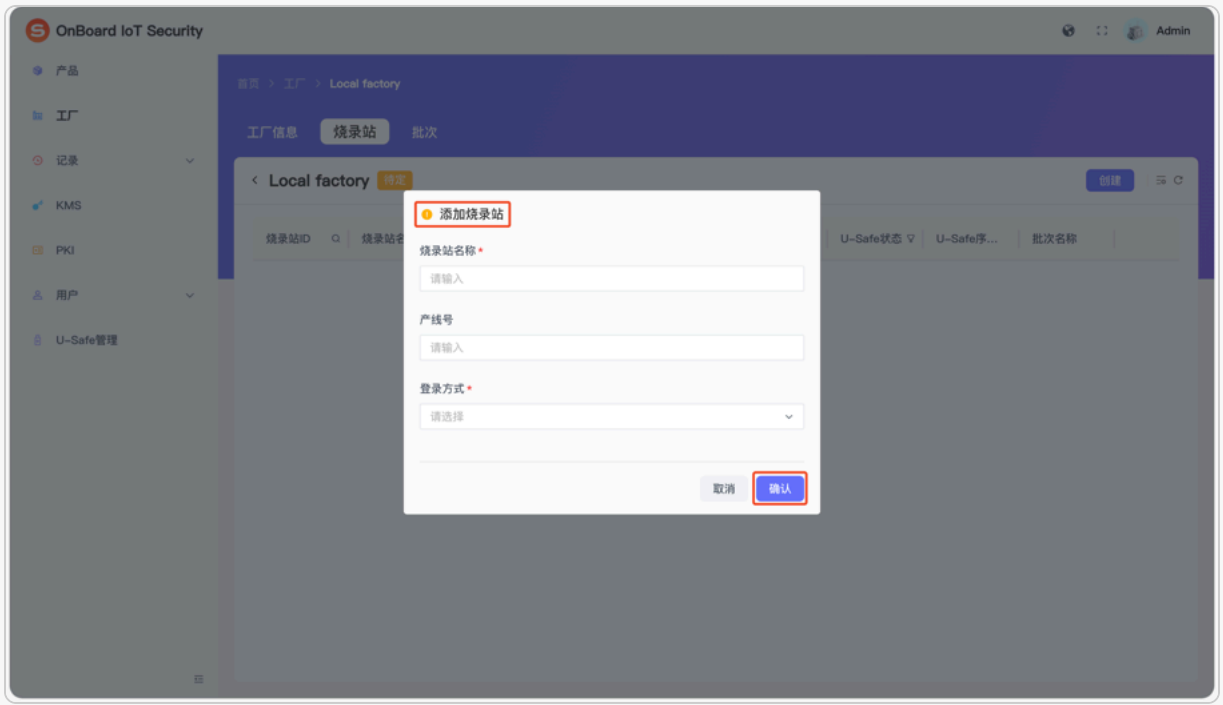
创建烧录站

> **提示**
>
> 为工厂创建烧录站后，你可以根据实际需求在 OnBoard Edge Service 为批次分配具体的烧录站。执行设备烧录前，需要根据烧录站 ID 查询并获取对应的批次信息和烧录数据。
>
> 支持两种烧录方式：
>
> - 使用烧录客户端（Programming Station）
> - 使用命令行工具（OBIS Programming CLI）

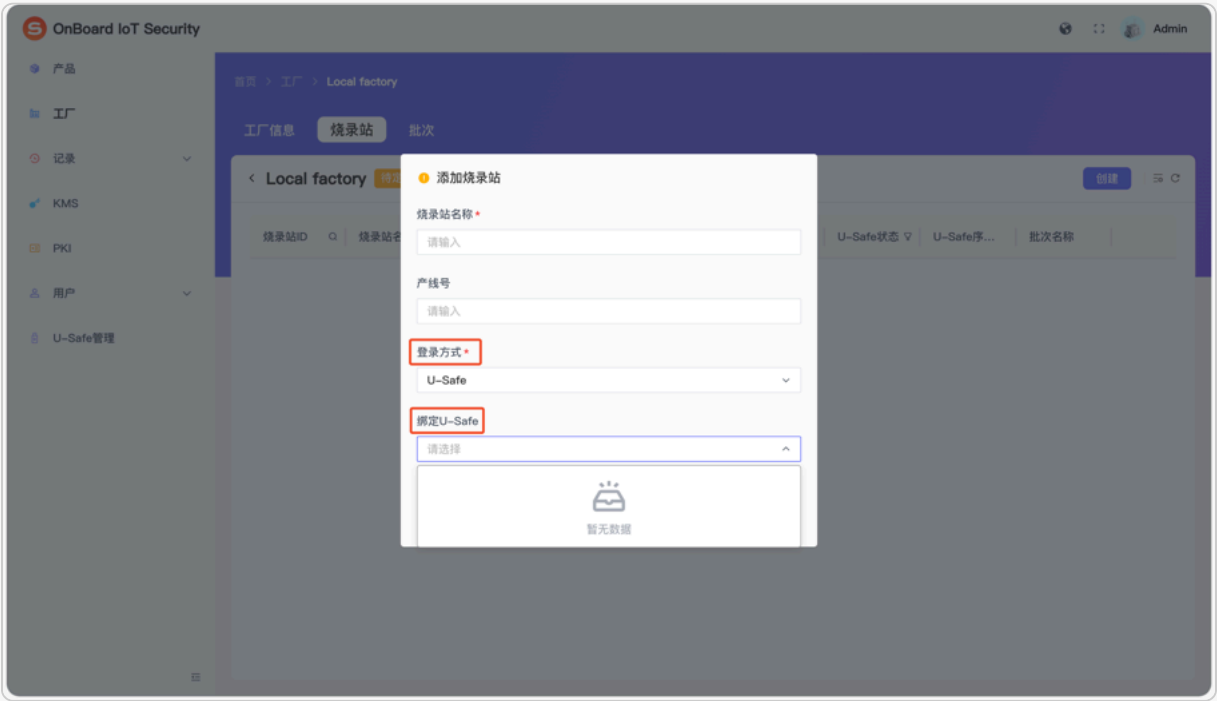
1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击**创建**。



4. 在**添加烧录站**窗口，输入**烧录站名称**和**产线号**，并选择**登录方式**。**登录方式**可选**无密码**，**密码**或 **U-Safe**。



> **提示**
>
> 如果登录方式为 **U-Safe**，你可以在**绑定 U-Safe** 下拉框中选择已添加的 Identity U-Safe 进行绑定，也可以在烧录站添加后再进行绑定。



5. 点击**确认**，即可完成烧录站的创建。

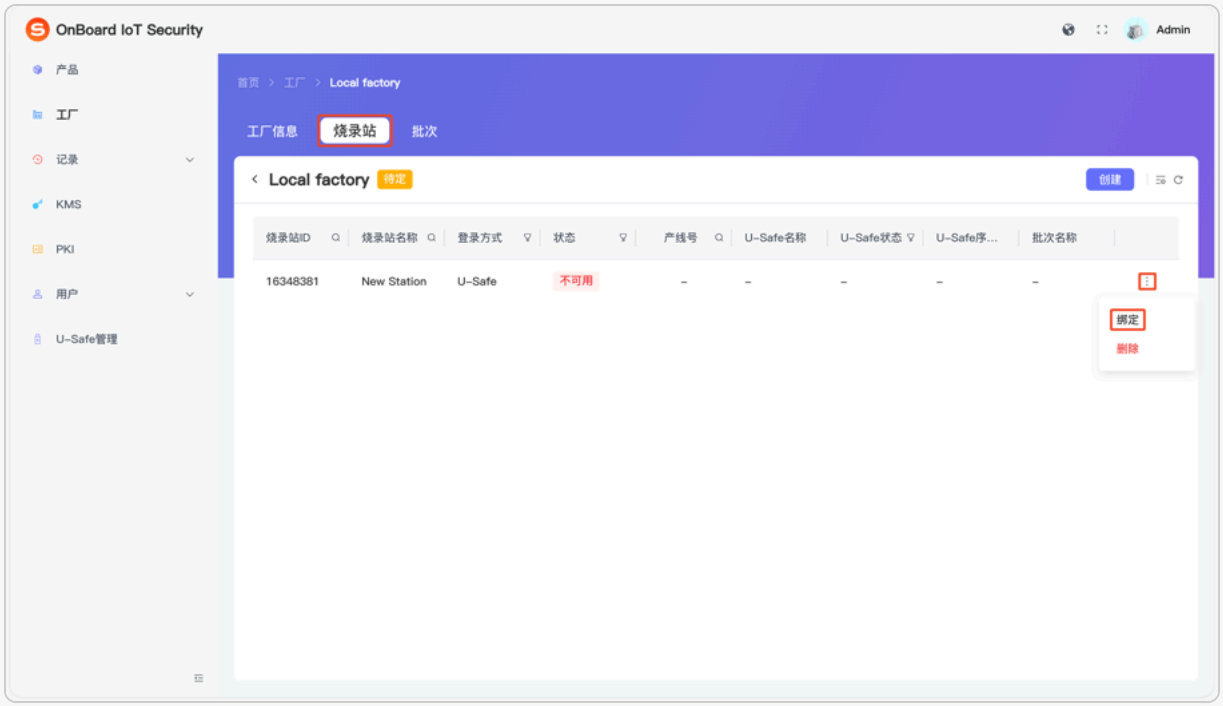
为烧录站绑定 Identity U-Safe

> **提示**
>
> 绑定后，可使用 Identity U-Safe 登录 Programming Station 客户端。

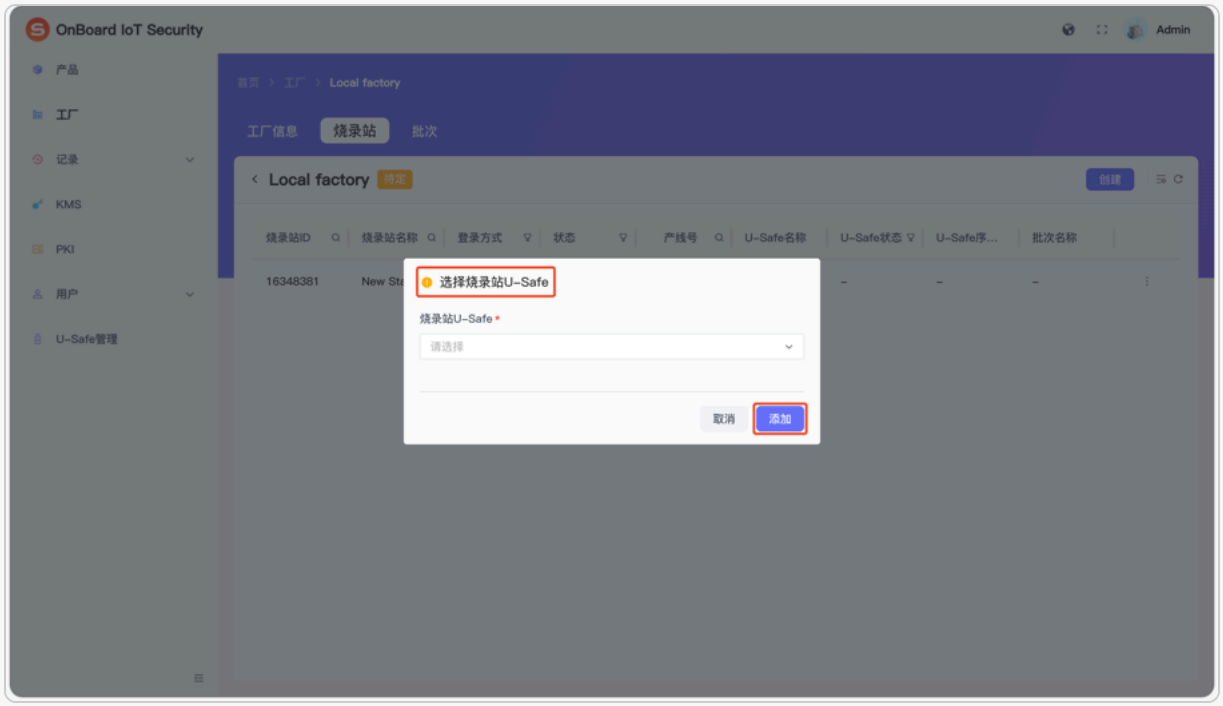
1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**绑定**。



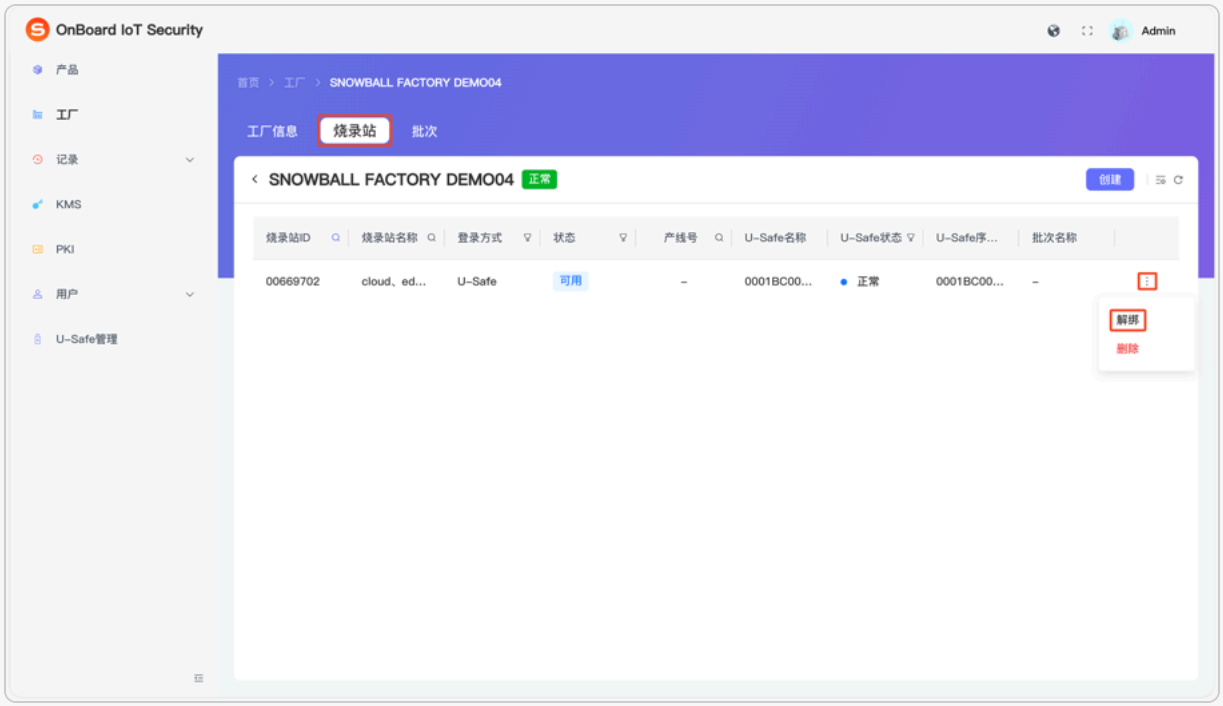
4. 在**选择烧录站 U-Safe** 窗口中，选择要绑定的 Identity U-Safe 并点击**添加**，即可为该烧录站绑定 Identity U-Safe。



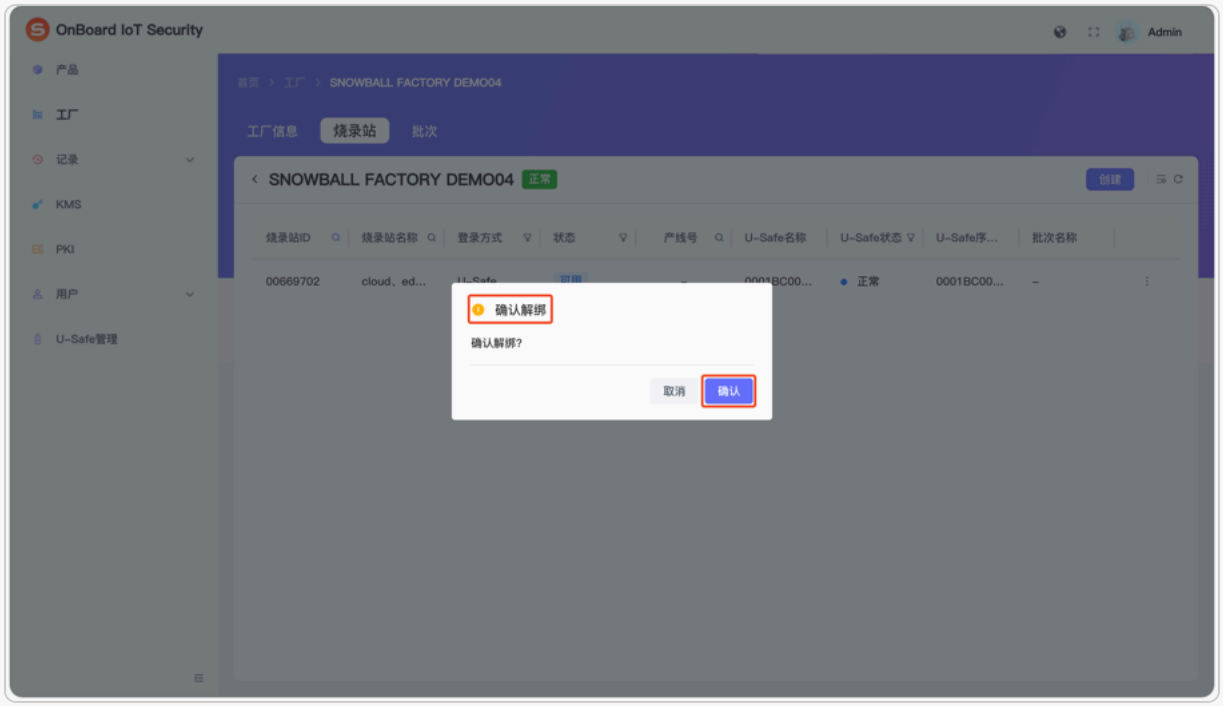
为烧录站解绑 Identity U-Safe

> **注意**
>
> 解绑后，将无法使用 Identity U-Safe 登录 Programming Station 客户端。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**解绑**。

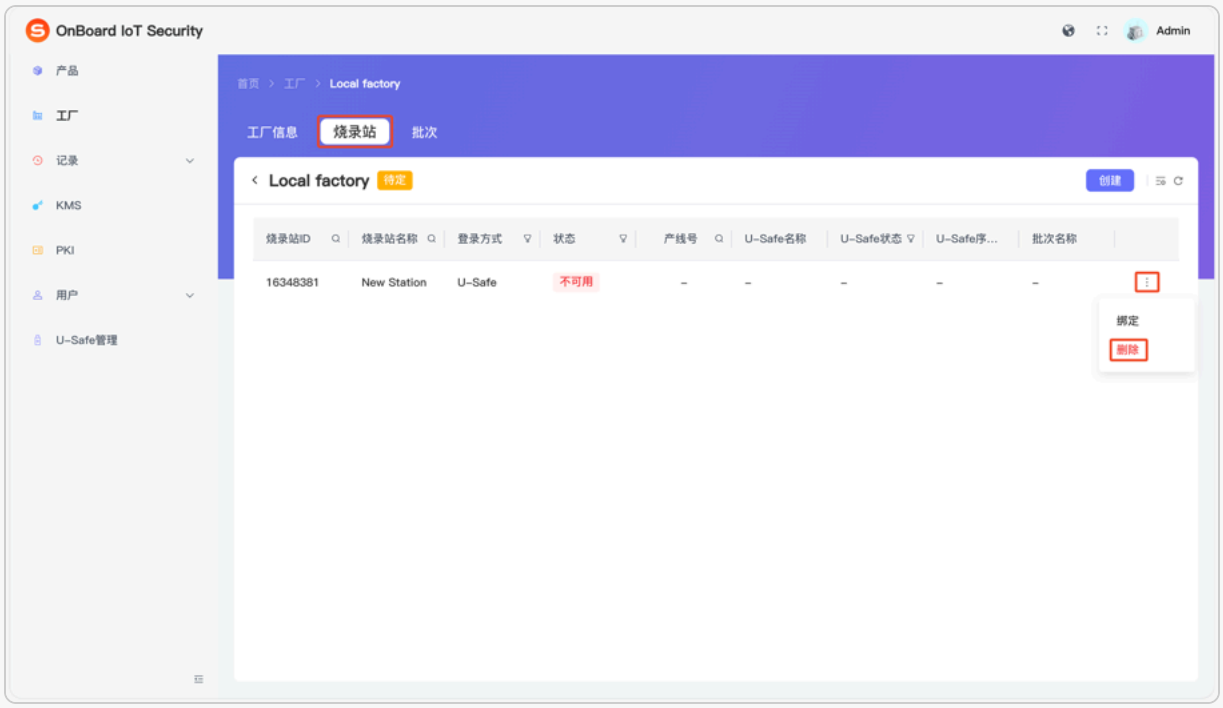


4. 在**确认解绑**窗口中点击**确认**，即可为该烧录站解绑 Identity U-Safe。

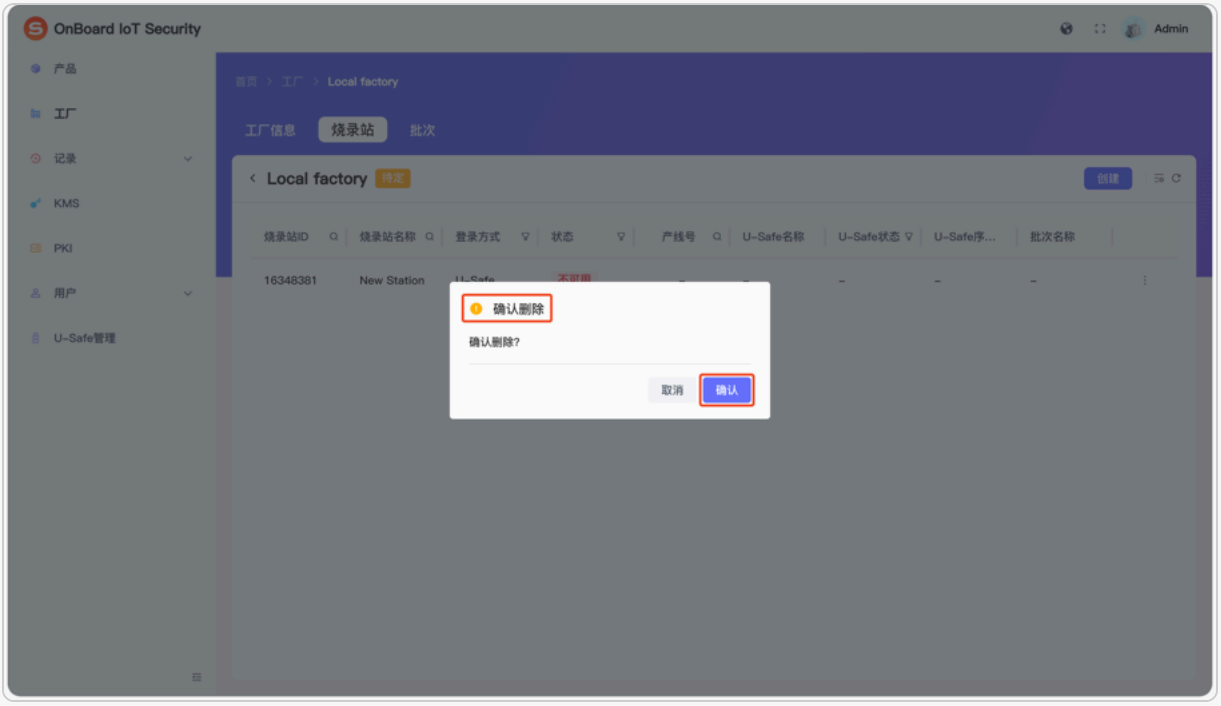


删除烧录站

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**删除**。



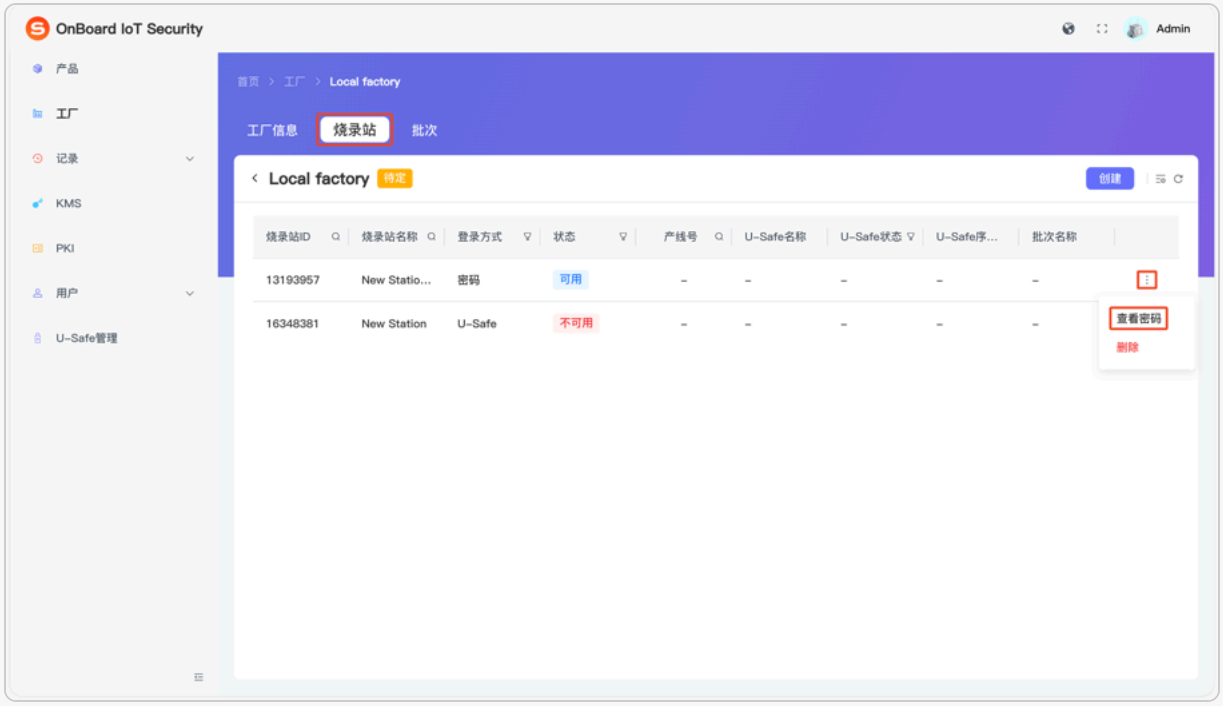
4. 在**确认删除**窗口中点击**确认**，即可删除该烧录站。



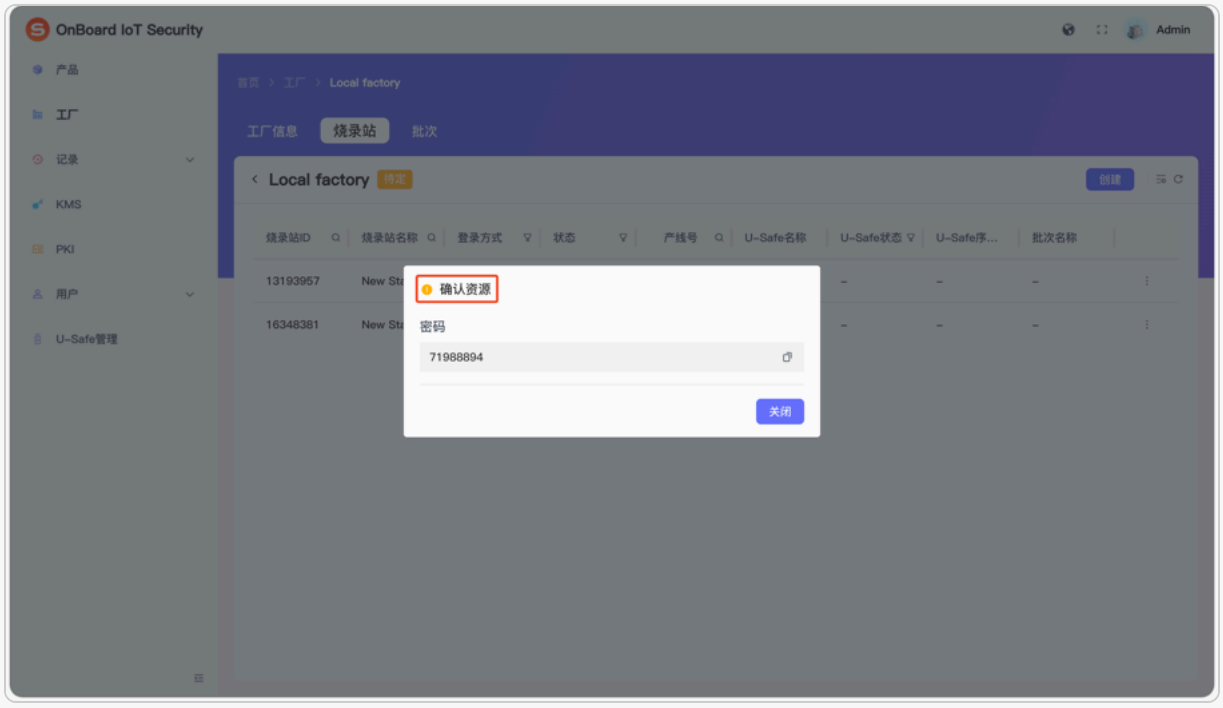
查看烧录站密码

> **提示**
>
> 当烧录站的登录方式为密码登录，系统会生成一个随机密码，该密码用于登录 Programming Station 客户端。该随机密码目前暂不支持修改。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**查看密码**。



4. 在**确认资源**窗口，即可查看并复制烧录站的登录密码。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中配置和管理工厂

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

- 添加工厂管理员
- 重置工厂管理员密码
- 删除工厂管理员
- 添加 Identity U-Safe
- 解绑，锁定或解锁 Identity U-Safe
- 锁定或解锁工厂

前提条件

在开始之前，请确保：

- 你已拥有对应工厂的 Factory Manager 角色权限。
- 已从雪球 Identity U-Safe。

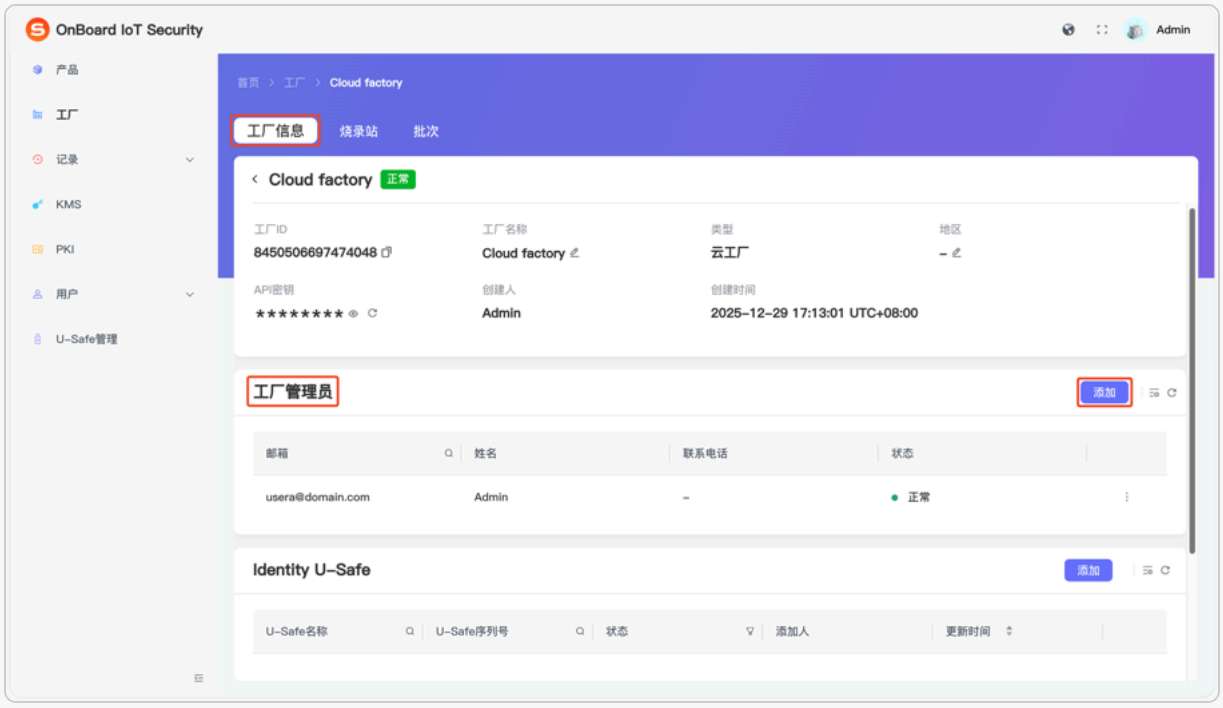
添加工厂管理员

> **提示**
>
> 工厂管理员可以管理和操作工厂相关业务。在添加工厂管理员后，系统会自动生成初始密码并发送到对应的邮箱。目前，该随机密码可以重置但暂不支持自定义修改。

1. 登录 OBIS。

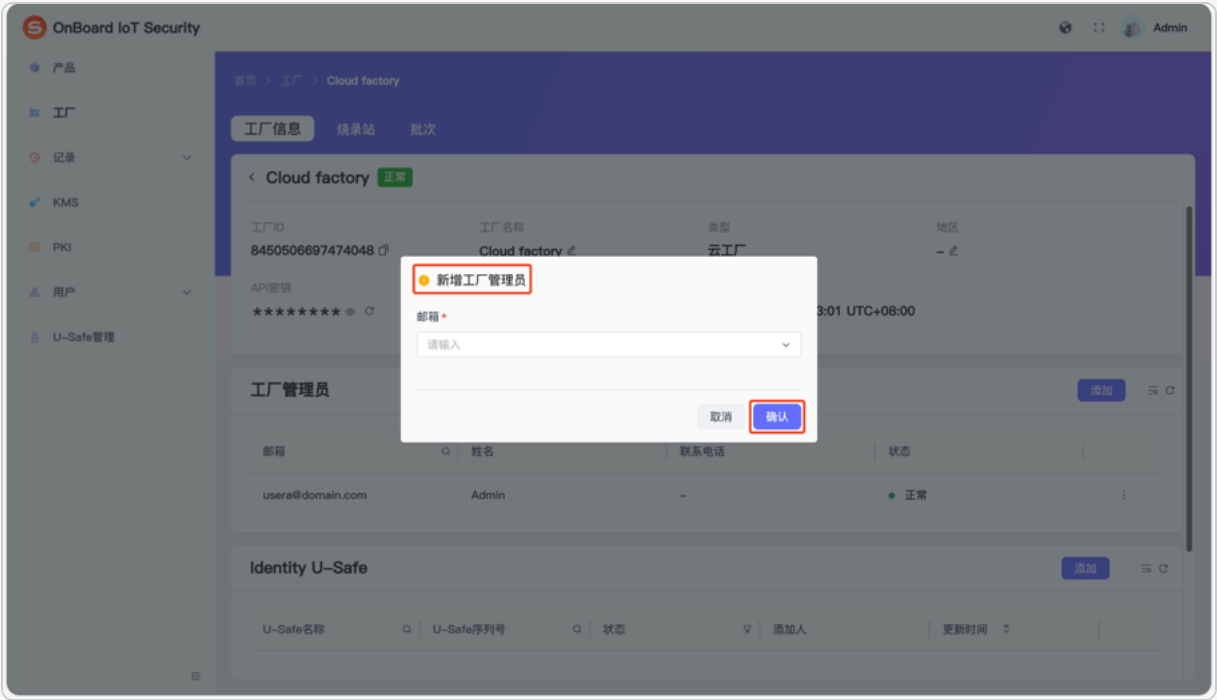
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，找到**工厂管理员**区域，点击**添加**。



4. 在**新增工厂管理员**窗口，选择并输入新增工厂管理员的邮箱地址。

> **提示**
>
> 被添加的工厂管理员必须是同一租户内状态为**启用**的用户。

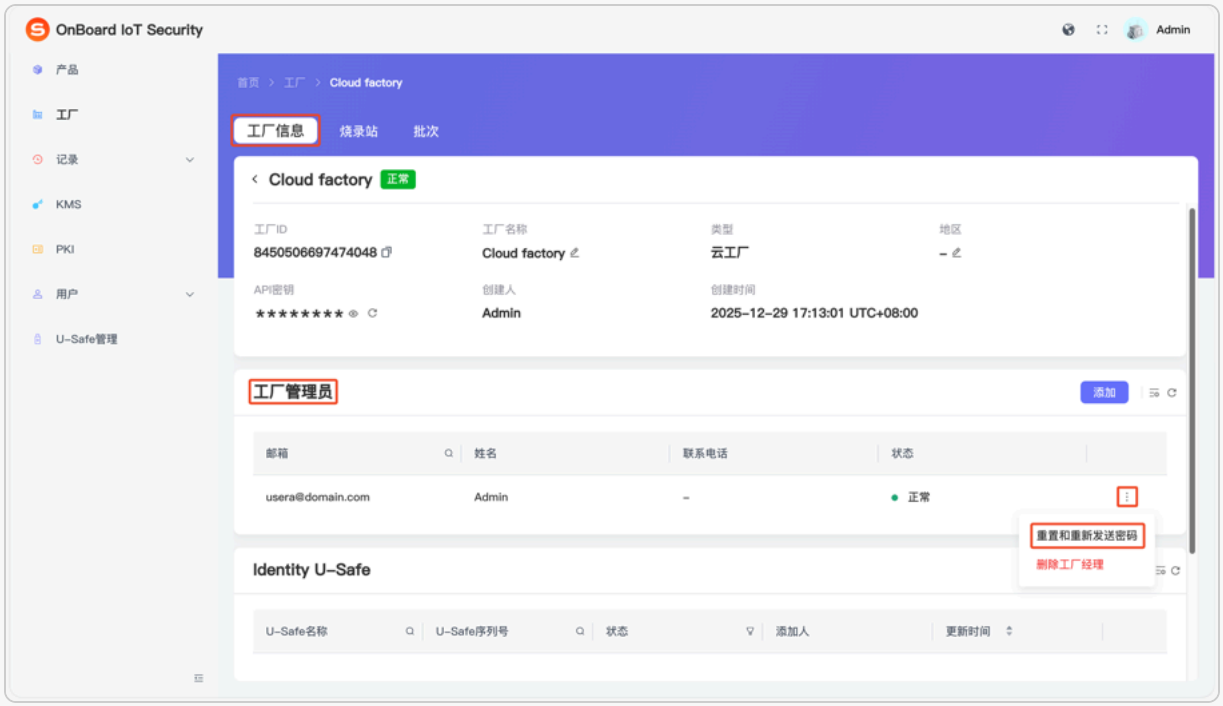


5. 点击**确认**，即可完成工厂管理员的添加。

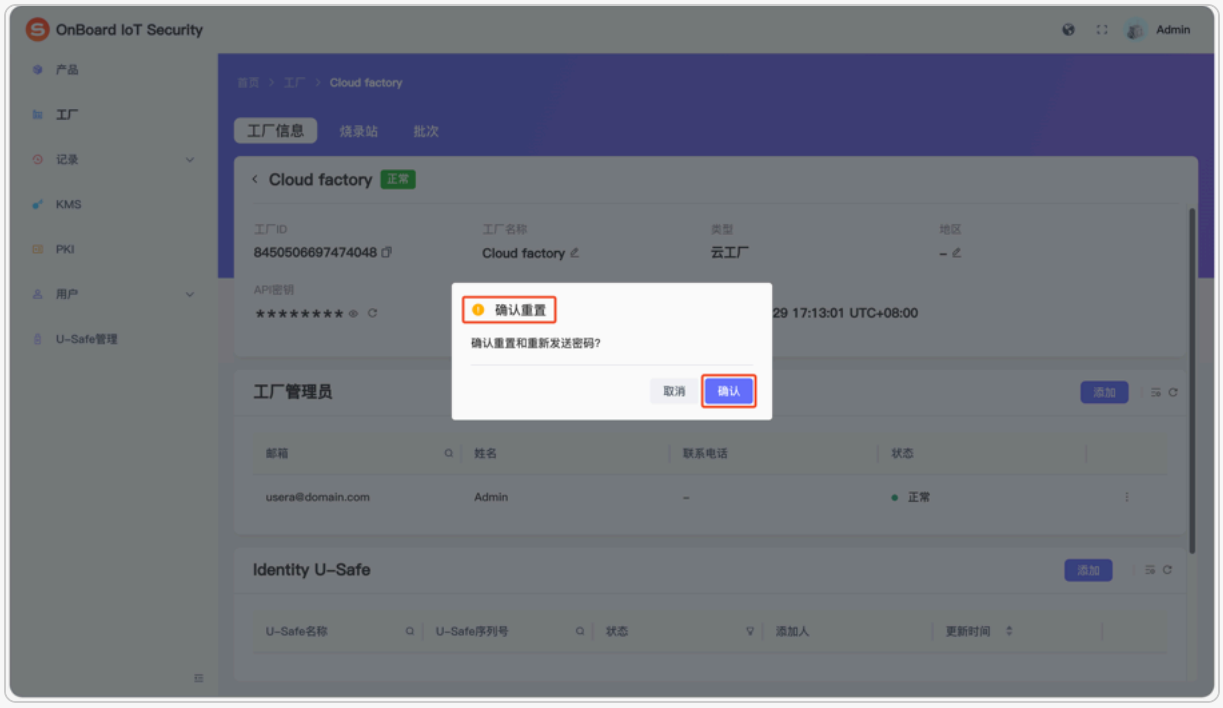
重置工厂管理员密码

> **提示**
>
> 密码重置后，系统会自动生成随机密码并发送到对应的邮箱。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，找到**工厂管理员**区域，点击对应工厂管理员的选项 ⋮ 图标并选择**重置和重新发送密码**。



4. 在**确认重置**窗口中点击**确认**，即可重置工厂管理员密码。

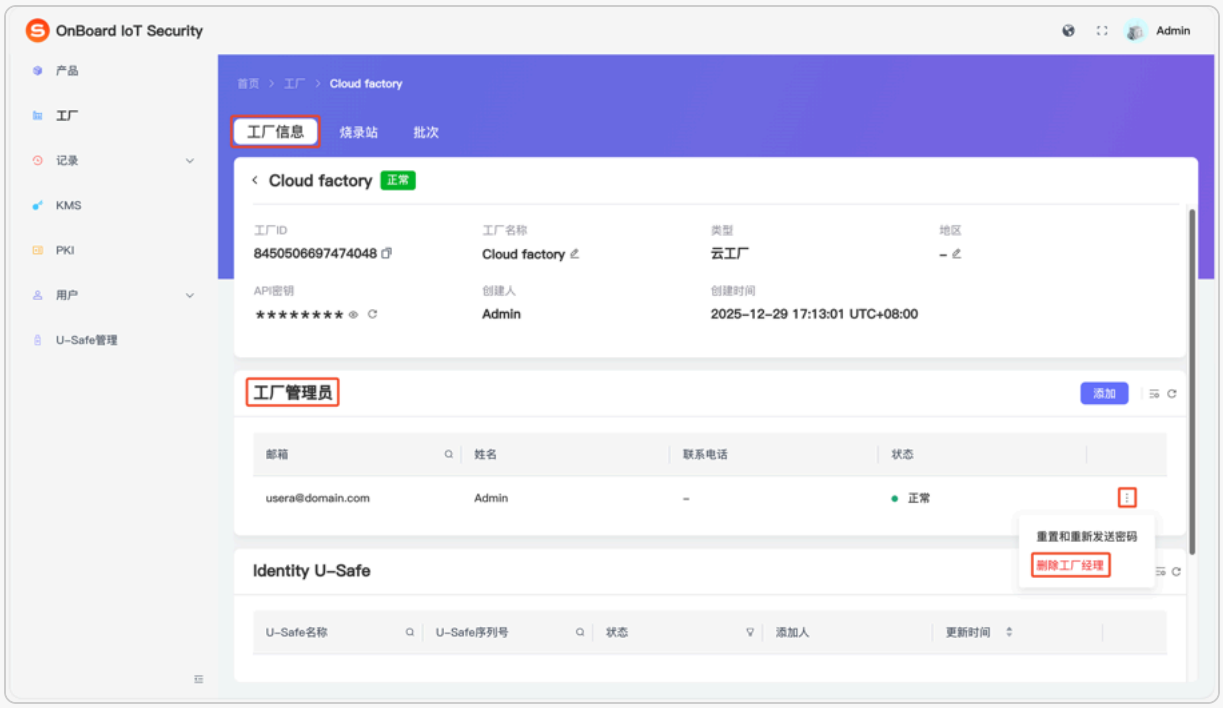


删除工厂管理员

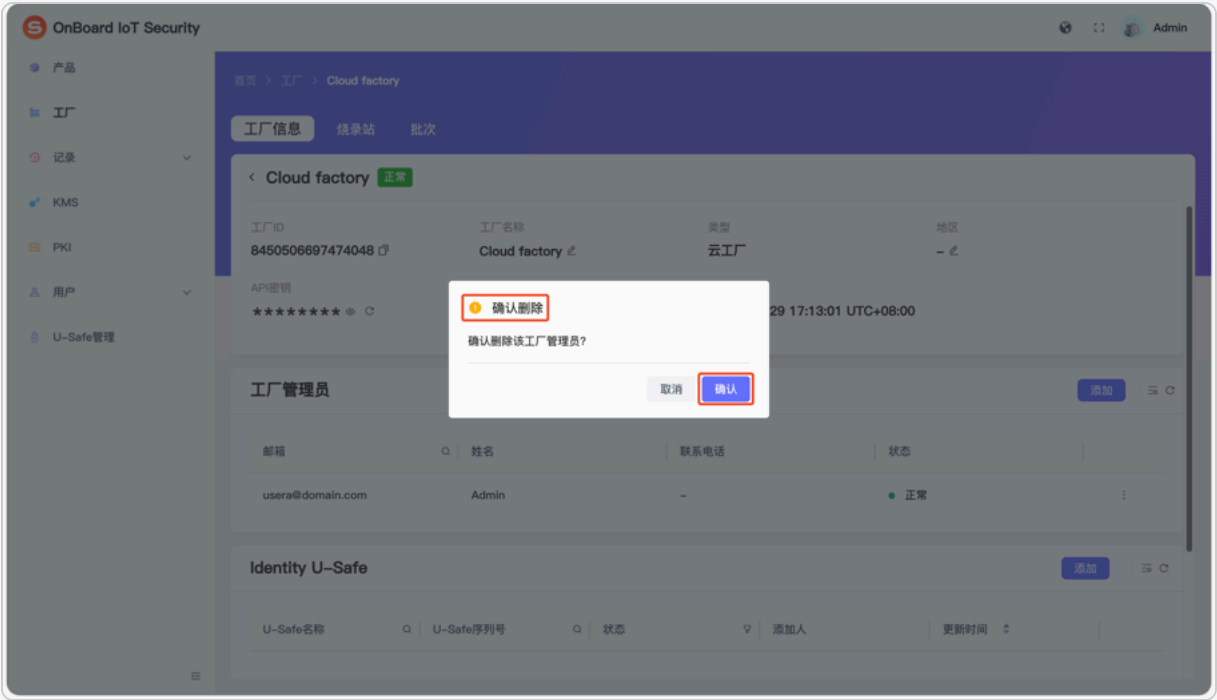
1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，找到**工厂管理员**区域，点击对应工厂管理员的选项 ⋮ 图标并选择**删除工厂经理**。



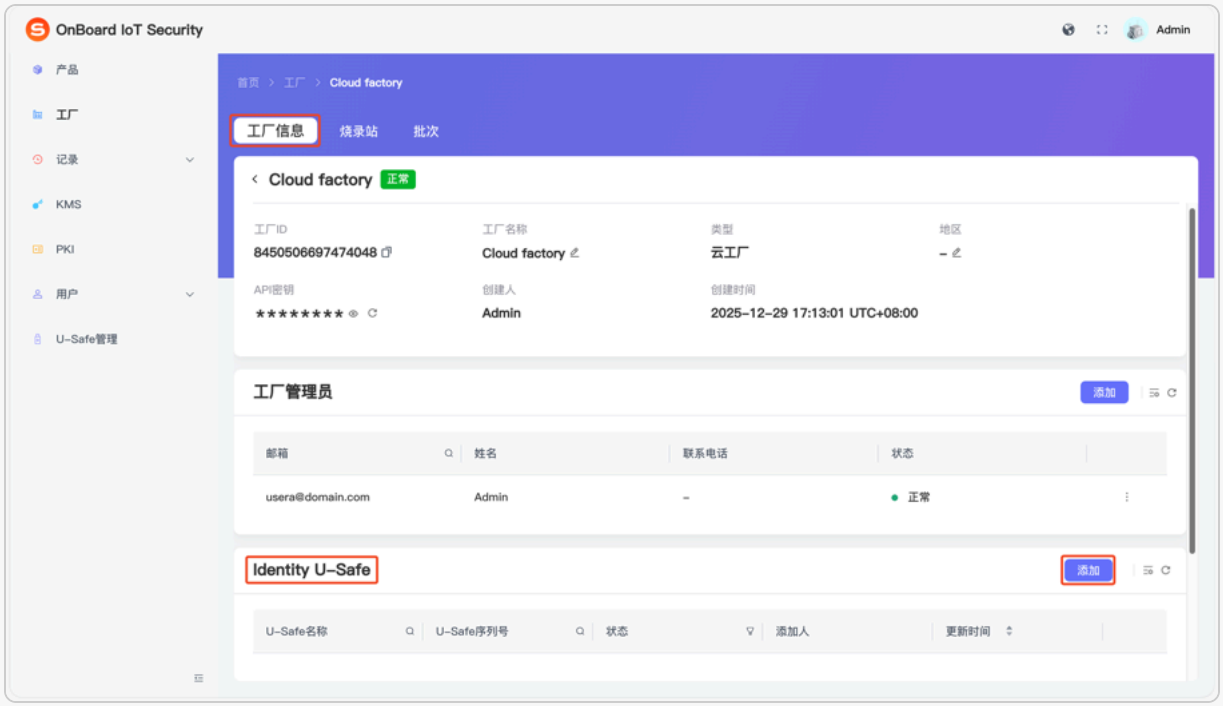
4. 在**确认删除**窗口中点击**确认**，即可删除该工厂管理员。



添加 Identity U-Safe

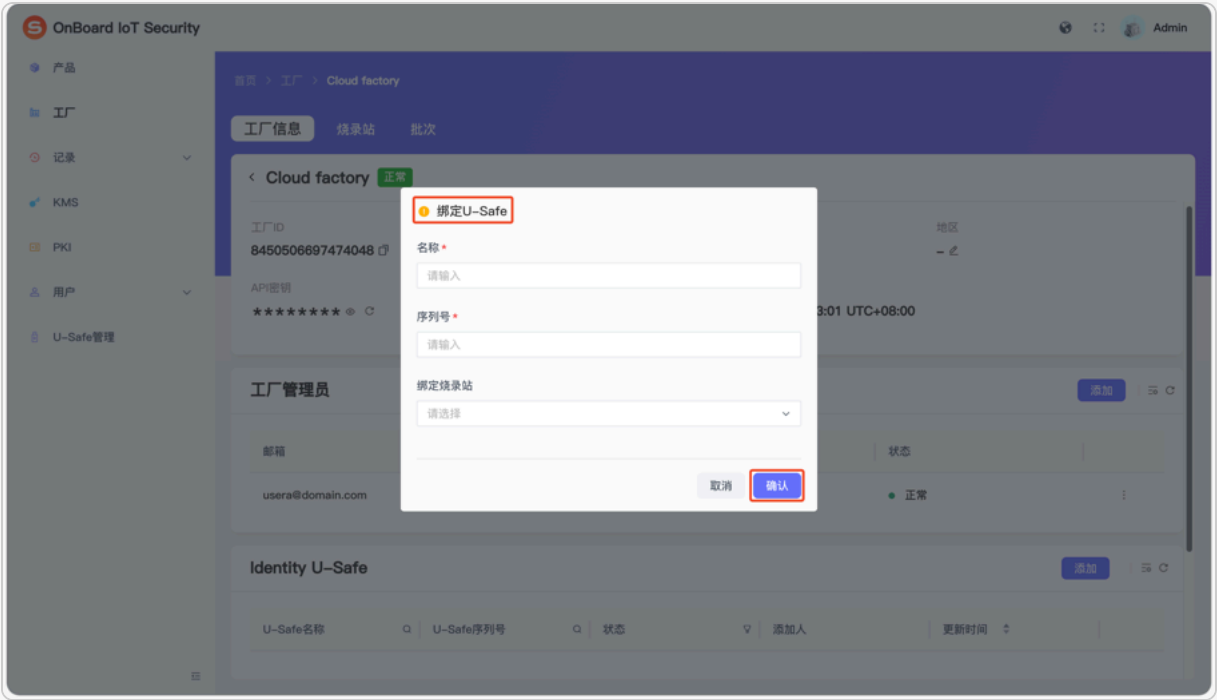
> **提示**
>
> 添加 Identity U-Safe 的目的是为了将其与对应的工厂进行绑定，将其与烧录站进行绑定后，可用于登录 Programming Station。同时，通过管理 Identity U-safe 从而实现烧录站的登录认证。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，找到 **Identity U-Safe** 区域，点击**添加**。



4. 在**绑定 U-Safe** 窗口中，输入需要绑定的 Identity U-Safe 的**名称**（可自定义）和**序列号**（可在设备本体上查看）。

> **提示**
>
> - 如果已添加烧录站，可以选择并绑定对应的烧录站。
> - 如果还未添加烧录站，你可以在添加烧录站时进行绑定。

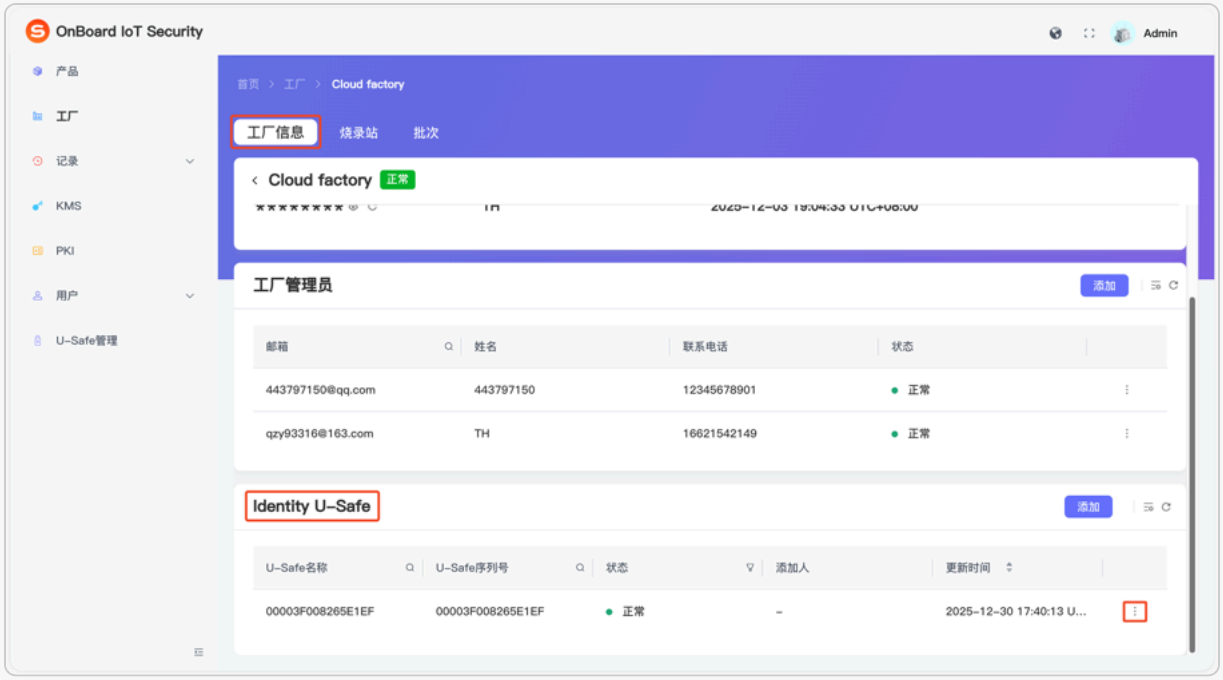


5. 点击**确认**，即可绑定 Identity U-Safe。

解绑，锁定或解锁 Identity U-Safe

> **提示**
>
> 锁定 Identity U-Safe 以禁止其使用和解绑。锁定后，状态会变更为**锁定**。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，找到 **Identity U-Safe** 区域，点击对应 Identity U-Safe 的选项 ⋮ 图标并选择**解绑**，**锁定**或**解锁**。



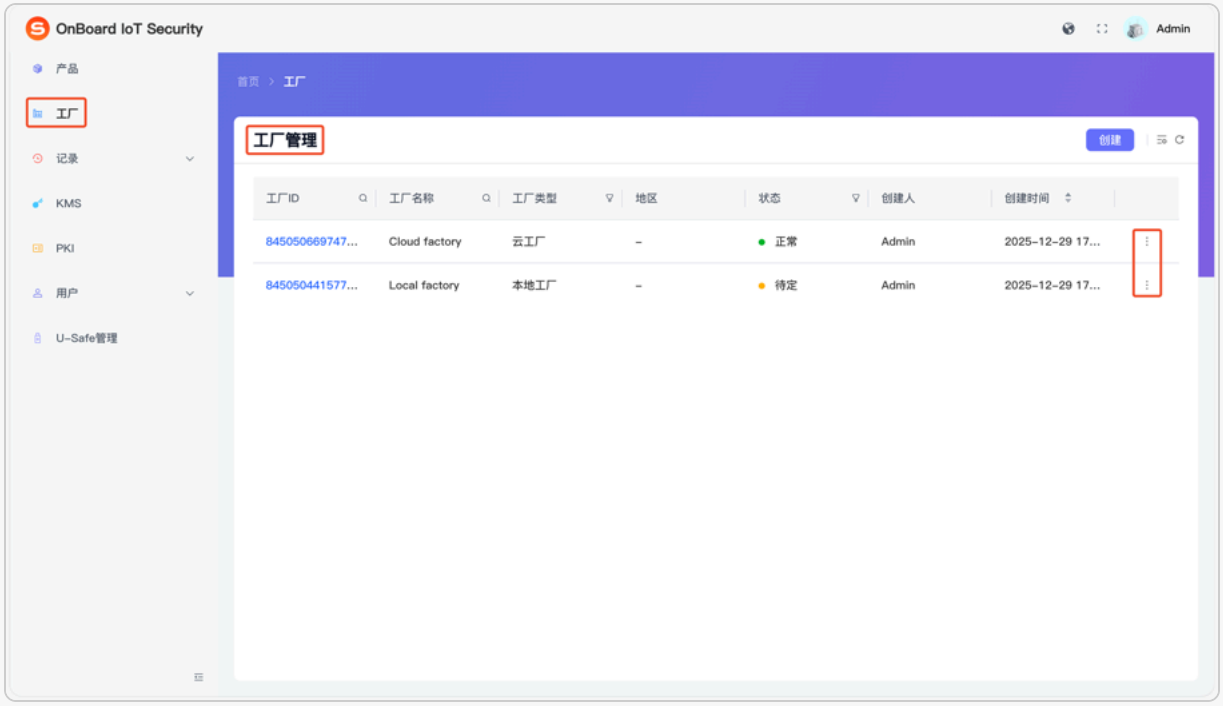
4. 在弹出的窗口中，点击**确认**，即可解绑，锁定或解锁该 Identity U-Safe。

锁定或解锁工厂

> **提示**
>
> 锁定工厂以禁止其使用。锁定后，状态会变更为**锁定**。

1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，进入**工厂管理**页面，点击对应工厂的选项 ⋮ 图标并选择**锁定**或**解锁**。



3. 在弹出的窗口中，点击**确认**，即可锁定或解锁该工厂。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中配置和管理烧录站

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

- 创建烧录站
- 为烧录站绑定 Identity U-Safe
- 为烧录站解绑 Identity U-Safe
- 删除烧录站
- 查看烧录站密码

前提条件

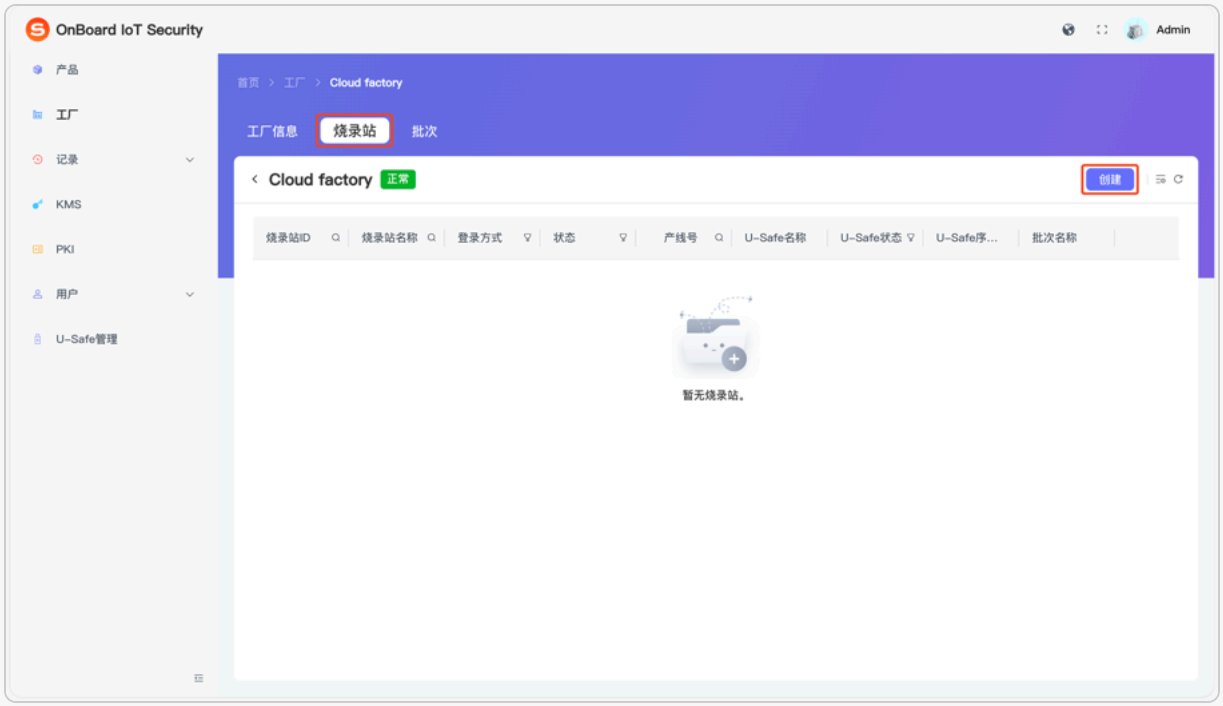
在开始之前，请确保：

- 已完成工厂的配置。
- 你已拥有对应工厂的 Factory Manager 角色权限。
- 已从雪球获取用于身份鉴权的 Identity U-Safe。

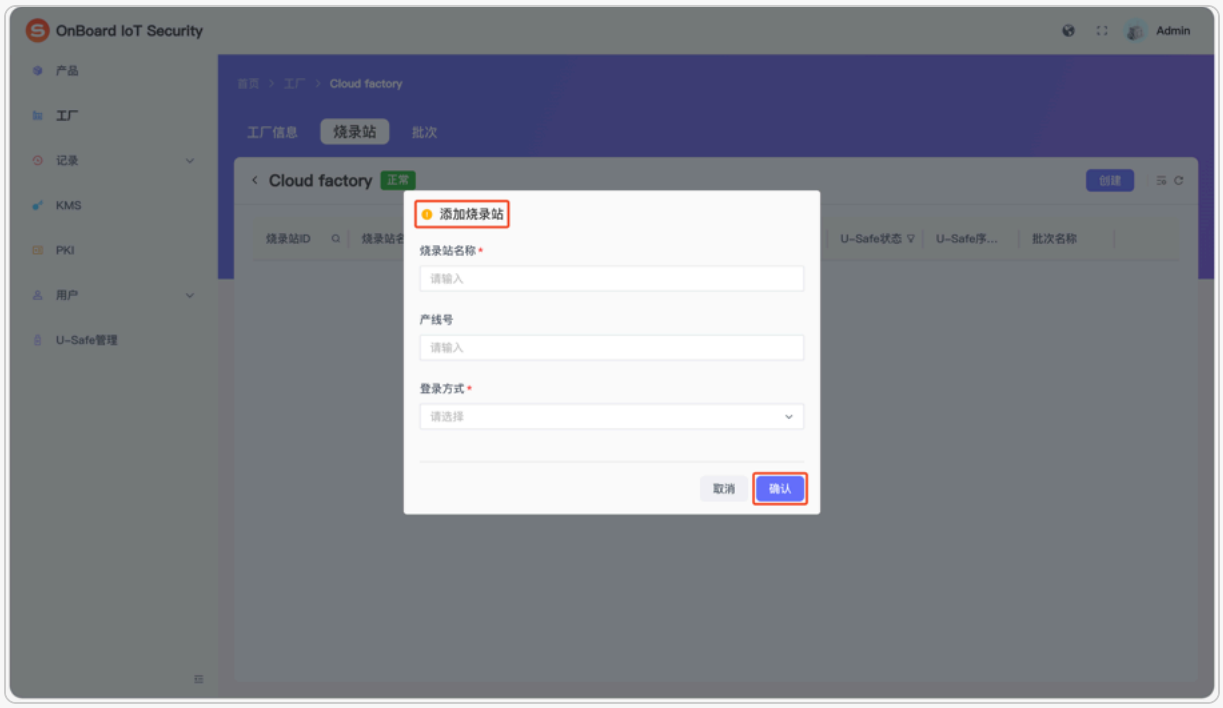
创建烧录站

> **提示**
>
> 为工厂创建烧录站后，你可以根据实际需求为批次分配具体的烧录站。执行设备烧录前，需要根据烧录站 ID 查询并获取对应的批次信息和烧录数据。
>
> 支持两种烧录方式：
>
> - 使用烧录客户端（Programming Station）
> - 使用命令行工具（OBIS Programming CLI）

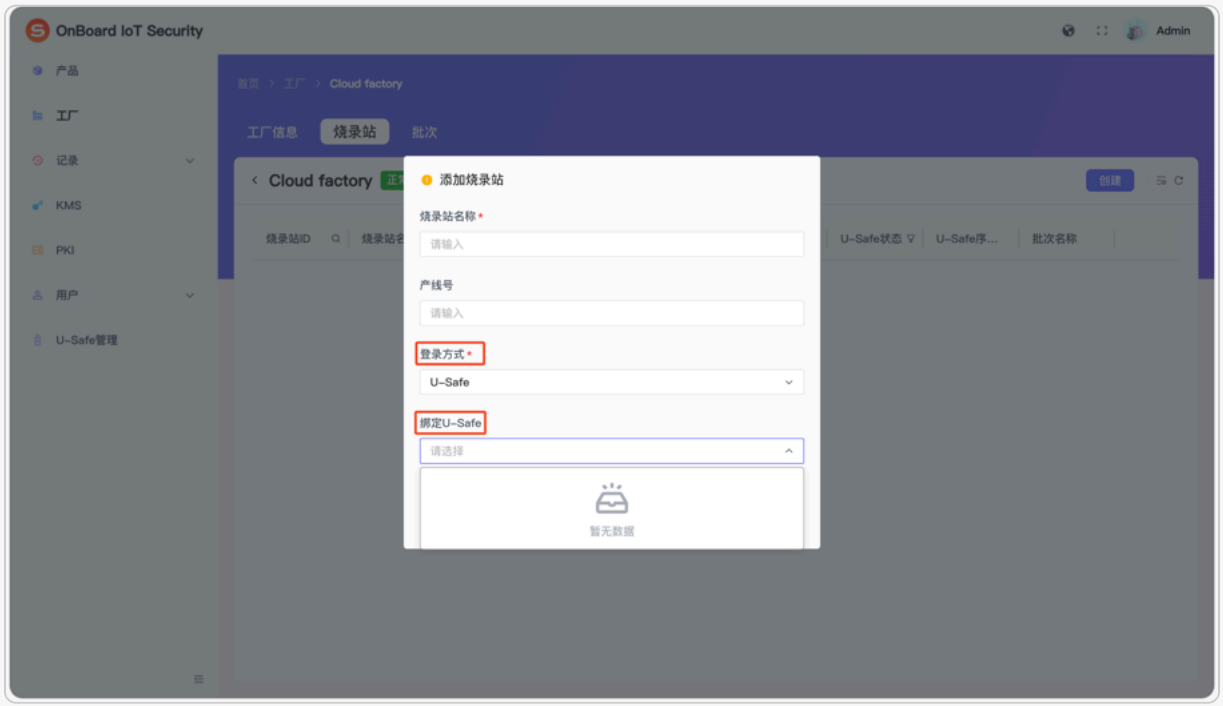
1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击**创建**。



4. 在**添加烧录站**窗口，输入**烧录站名称**和**产线号**，并选择**登录方式**。**登录方式**可选**密码**或**U-Safe**。



> **提示**
>
> 如果登录方式为 **U-Safe**，你可以在**绑定 U-Safe** 下拉框中选择已添加的 Identity U-Safe 进行绑定，也可以在烧录站添加后再进行绑定。



5. 点击**确认**，即可完成烧录站的创建。

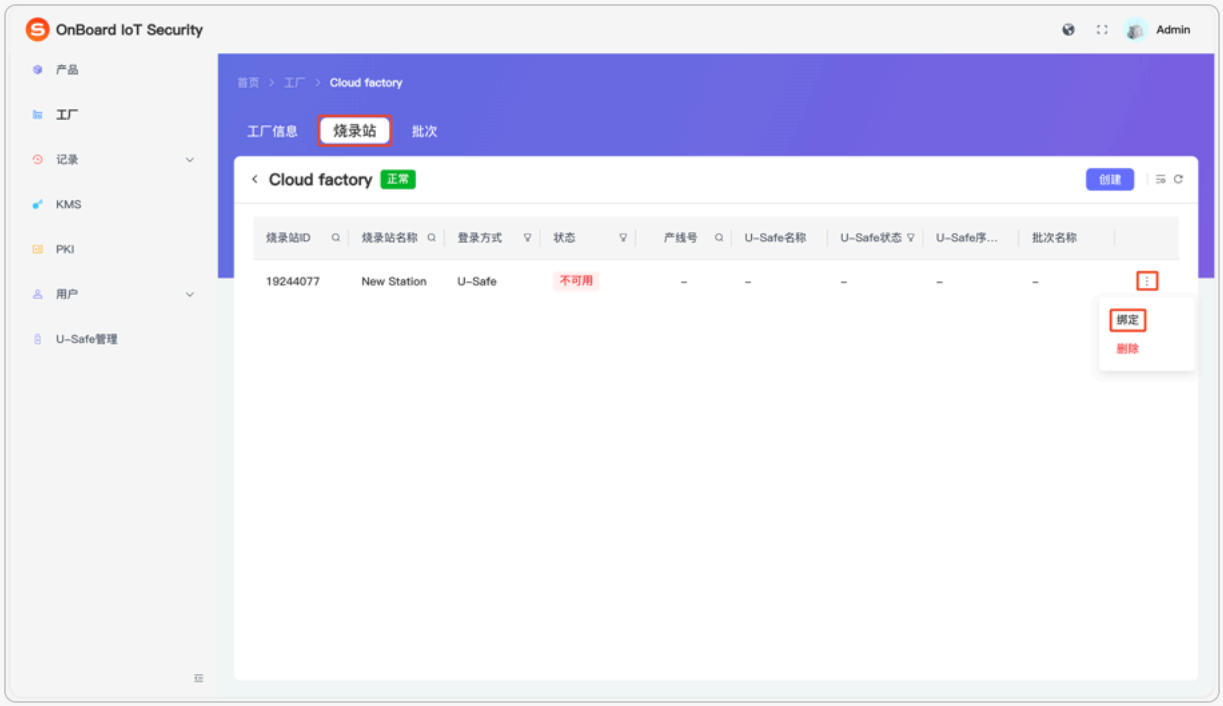
为烧录站绑定 Identity U-Safe

> **提示**
>
> 绑定后，可使用 Identity U-Safe 登录 Programming Station 客户端。

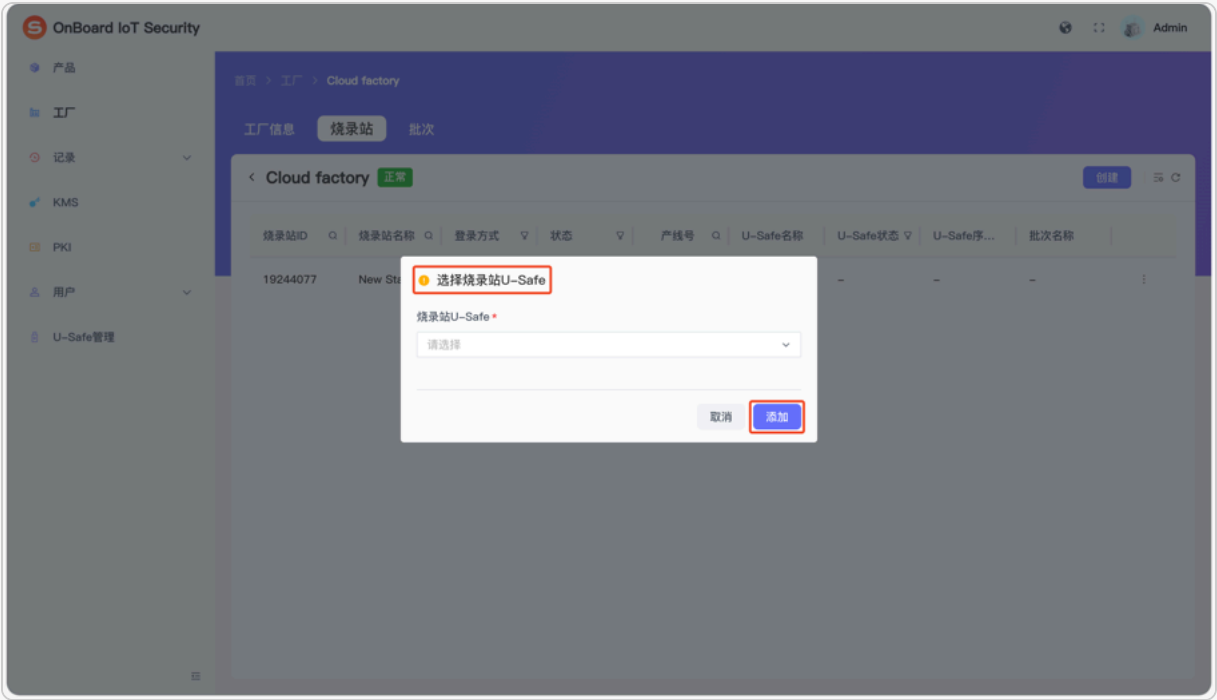
1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**绑定**。



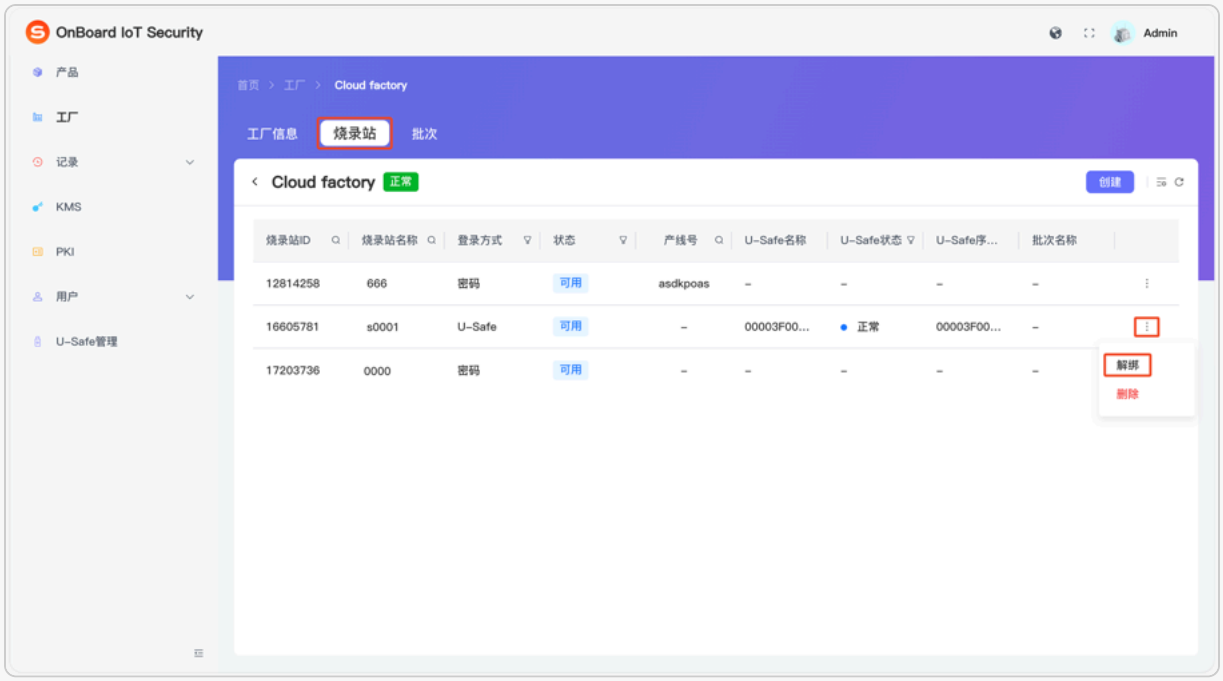
4. 在**选择烧录站 U-Safe** 窗口中，选择要绑定的 Identity U-Safe 并点击**添加**，即可为该烧录站绑定 Identity U-Safe。



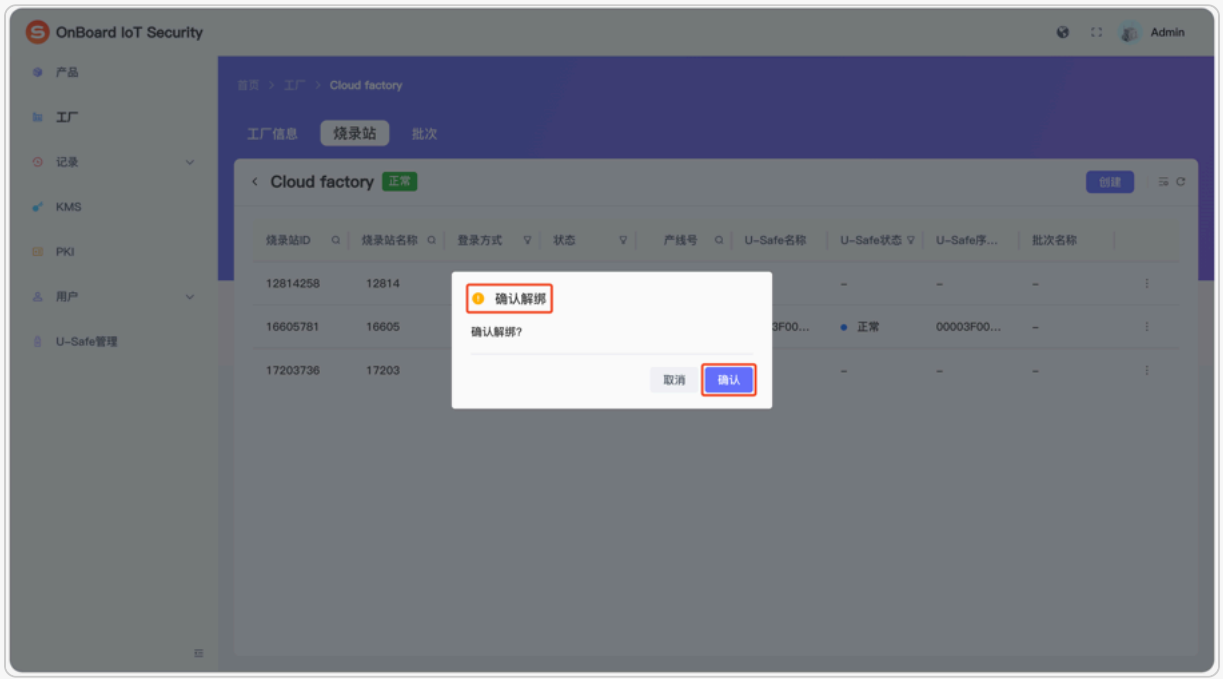
为烧录站解绑 Identity U-Safe

> **注意**
>
> 解绑后，将无法使用 Identity U-Safe 登录 Programming Station 客户端。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**解绑**。

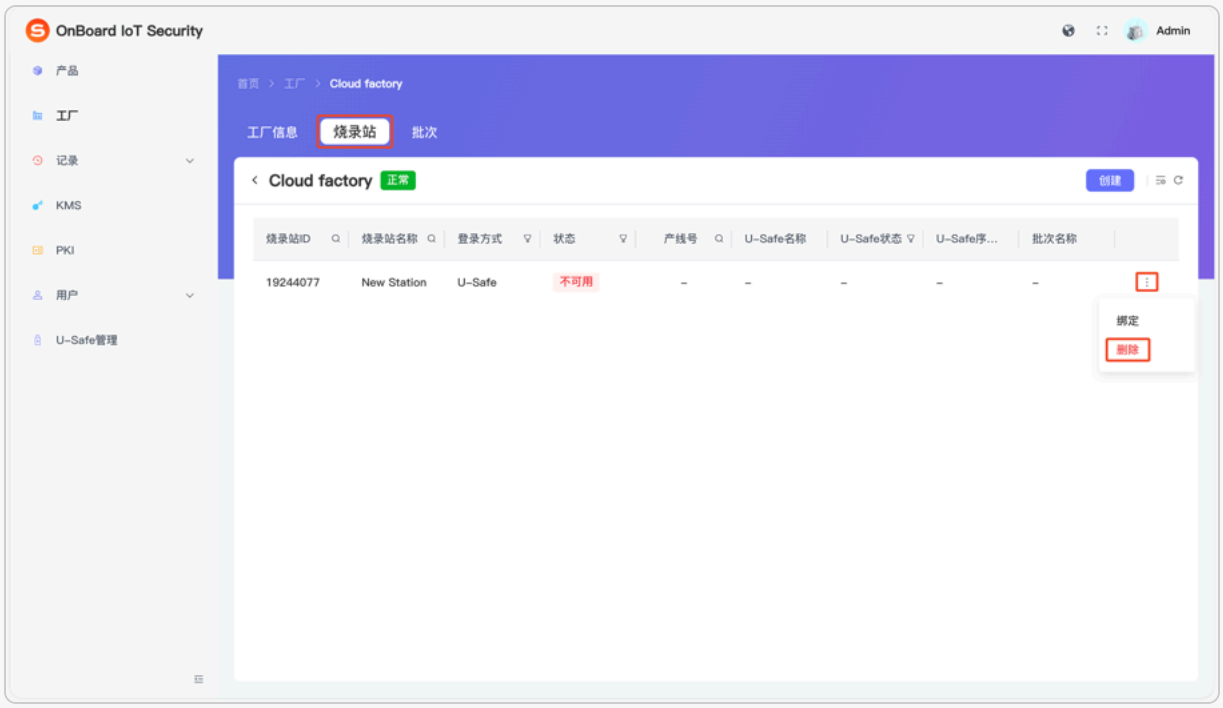


4. 在**确认解绑**窗口中点击**确认**，即可为该烧录站解绑 Identity U-Safe。

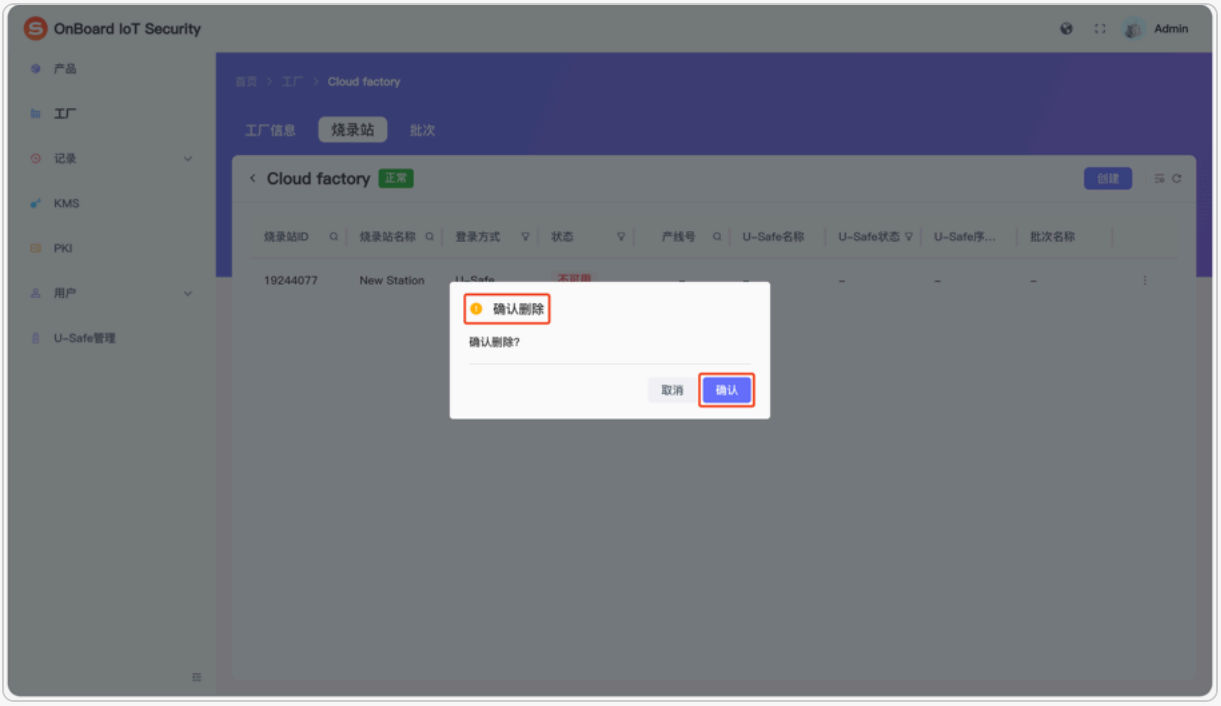


删除烧录站

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**删除**。



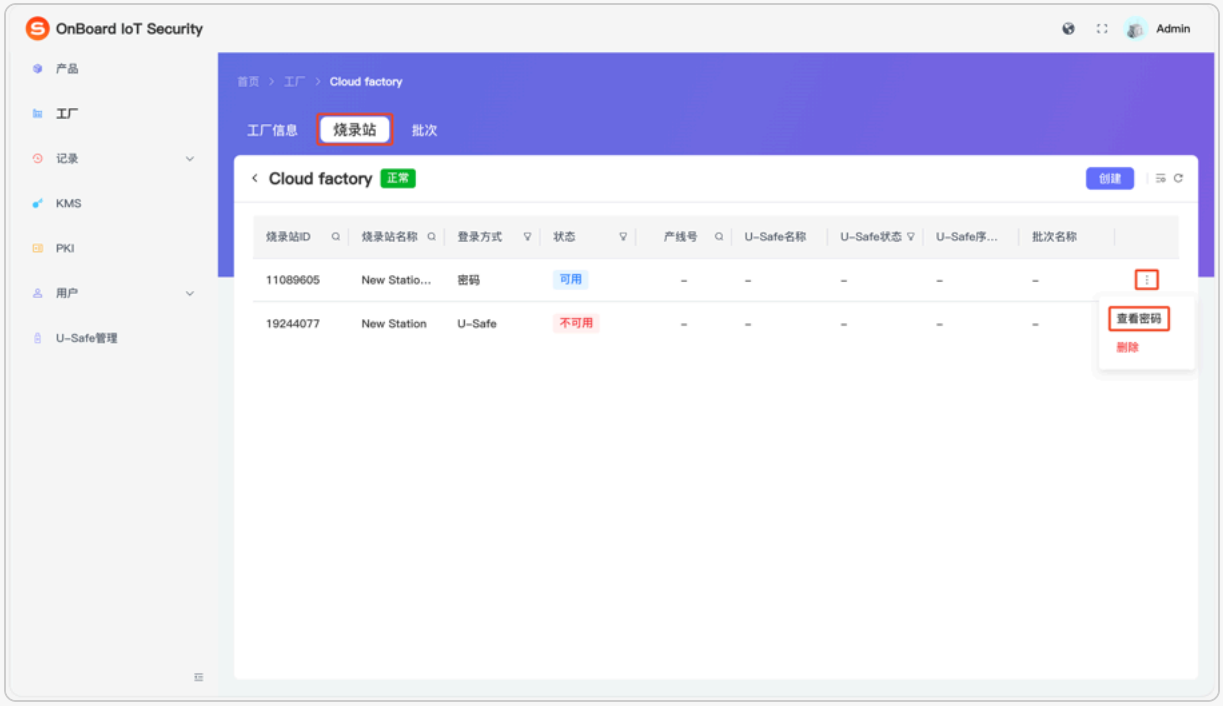
4. 在**确认删除**窗口中点击**确认**，即可删除该烧录站。



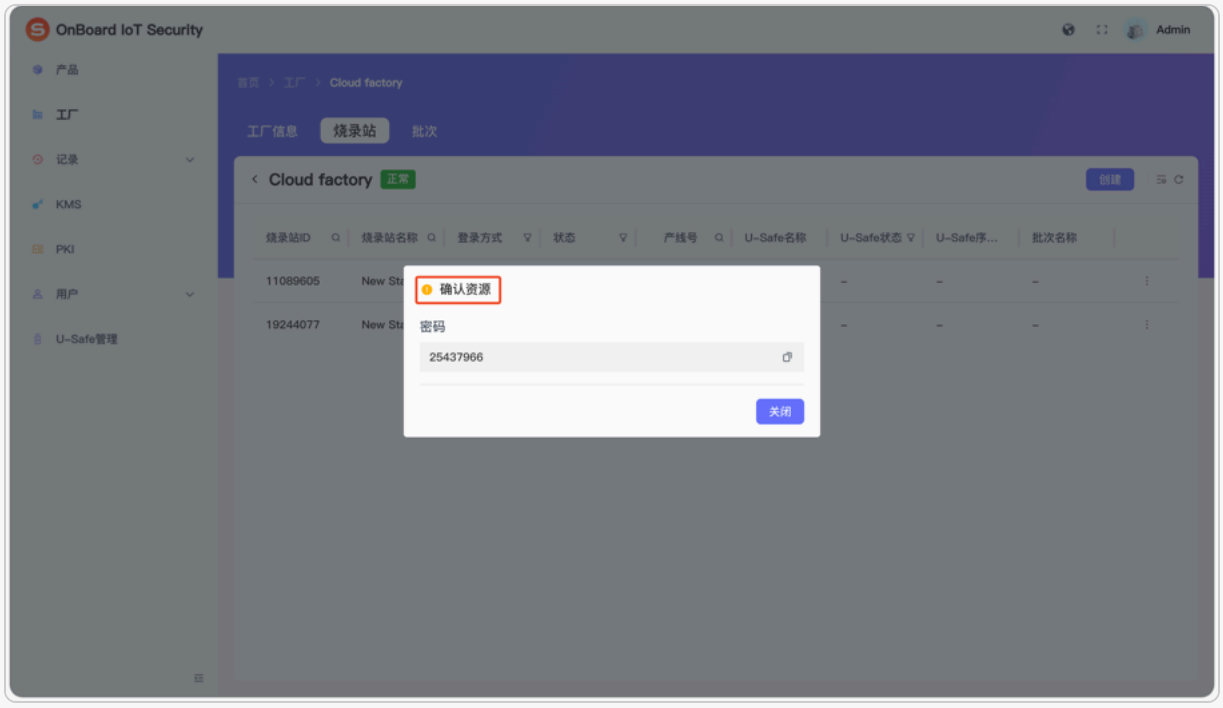
查看烧录站密码

> **提示**
>
> 当烧录站的登录方式为密码登录，系统会生成一个随机密码，该密码用于登录 Programming Station 客户端。该随机密码目前暂不支持修改。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，在**烧录站**标签页，点击对应烧录站的选项 ⋮ 图标并选择**查看密码**。



4. 在**确认资源**窗口，即可查看并复制烧录站的登录密码。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理批次与分配烧录站

本文将指导你在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

- 查看批次状态
- 为批次分配烧录站
- 为批次取消分配烧录站
- 关闭与删除批次

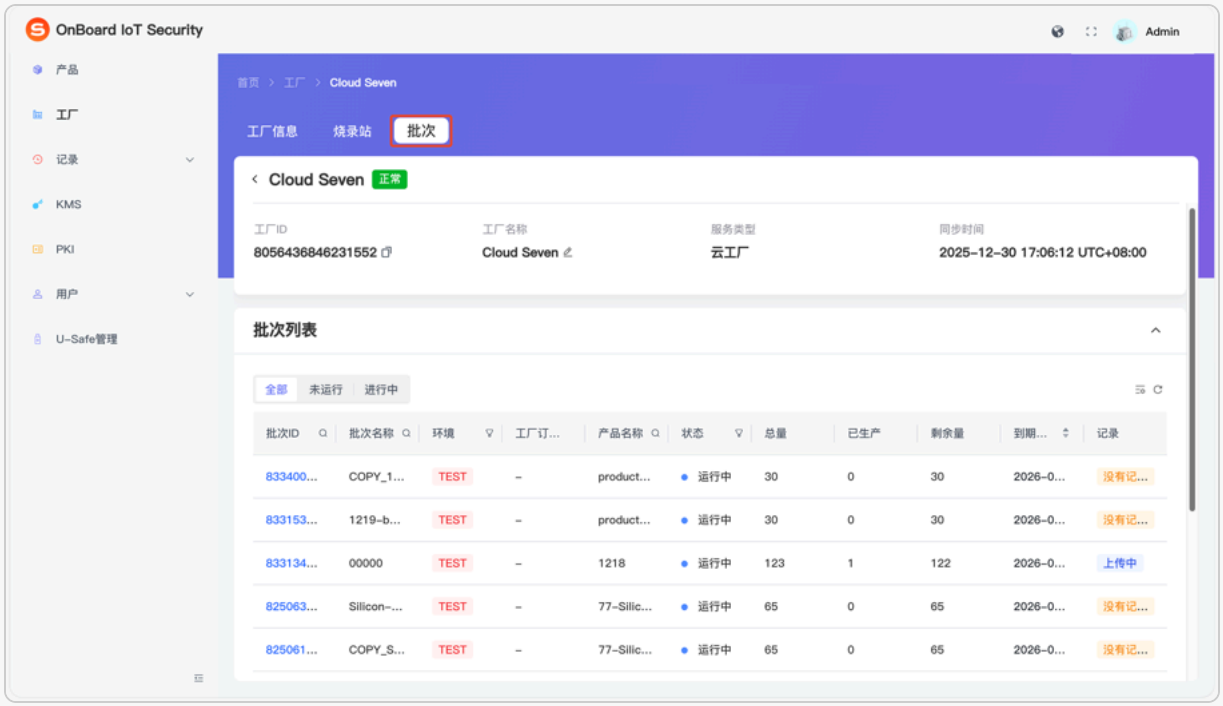
前提条件

在开始之前，请确保：

- 已完成工厂的配置。
- 你已拥有对应工厂的 Factory Manager 角色权限。

查看批次状态

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**批次**标签页，进入**批次**标签页面，即可查看批次状态。



批次状态说明：

批次状态	可否开启	说明
运行	-	创建批次后的默认状态，可绑定烧录站。
停止	可以	批次已被关闭，可重新开启并绑定烧录站。
已完成	不可以	批次烧录完成后的状态。

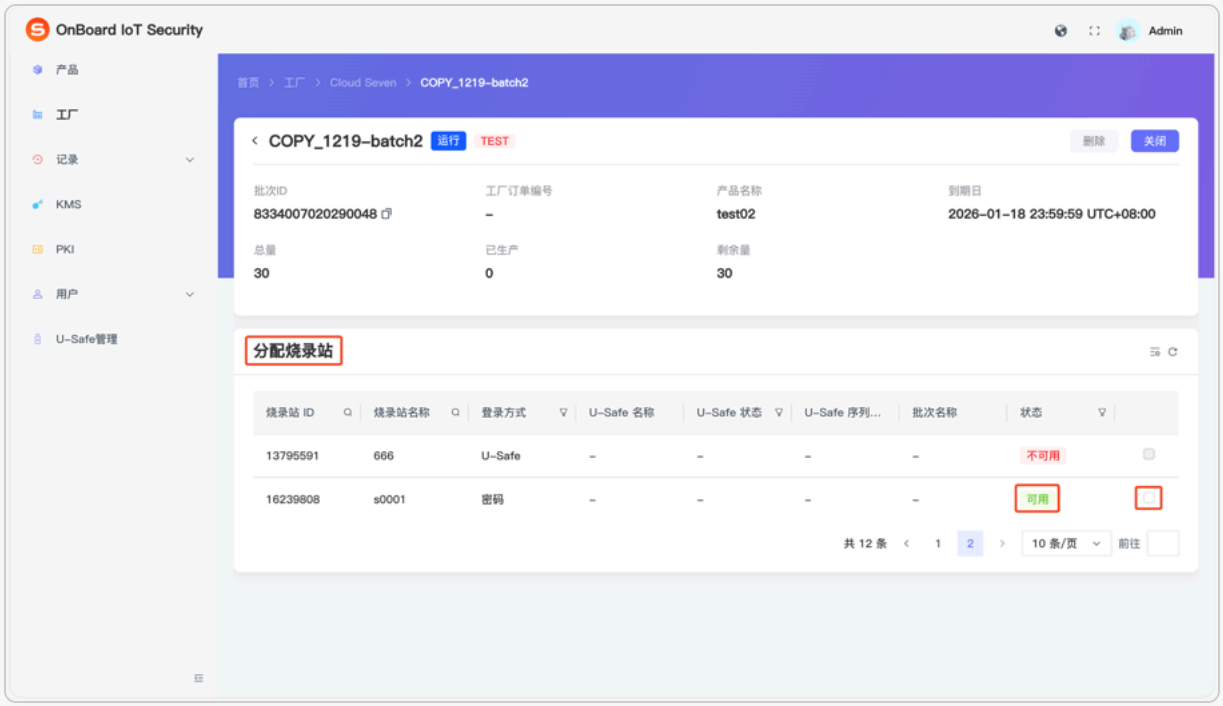
批次状态	可否开启	说明
已过期	不可以	批次在完成烧录前已超过有效期。 **注释**：批次过期后，针对该批次的大部分功能将无法正常操作，包括在Programming Station 客户端进行烧录。建议删除已过期批次。

为批次分配烧录站

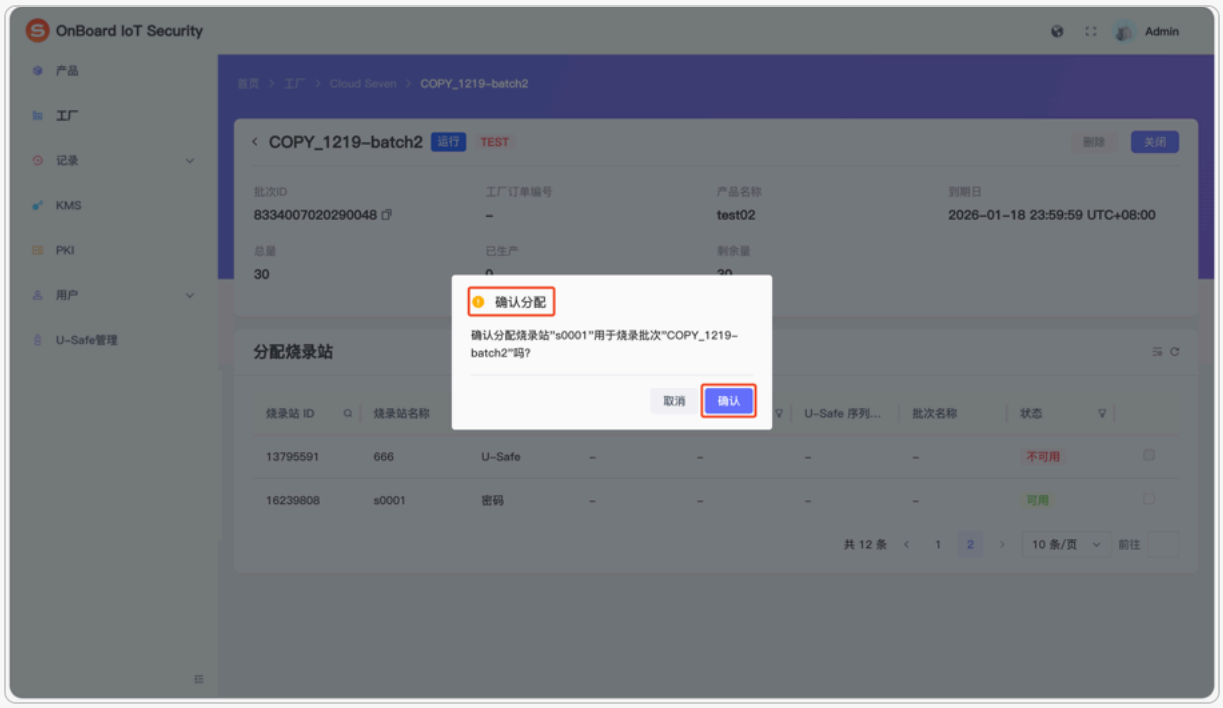
> **提示**
>
> 一个批次同时可以分配给多个烧录站，而一个烧录站同时只可烧录一个批次。

为批次分配状态为**可用**的烧录站用于烧录：

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**批次**标签页，然后点击对应的批次 ID进入批次页面。
4. 在批次页面的**分配烧录站**区域，确认批次的状态为**运行**，并勾选状态为**可用**的烧录站进行分配。



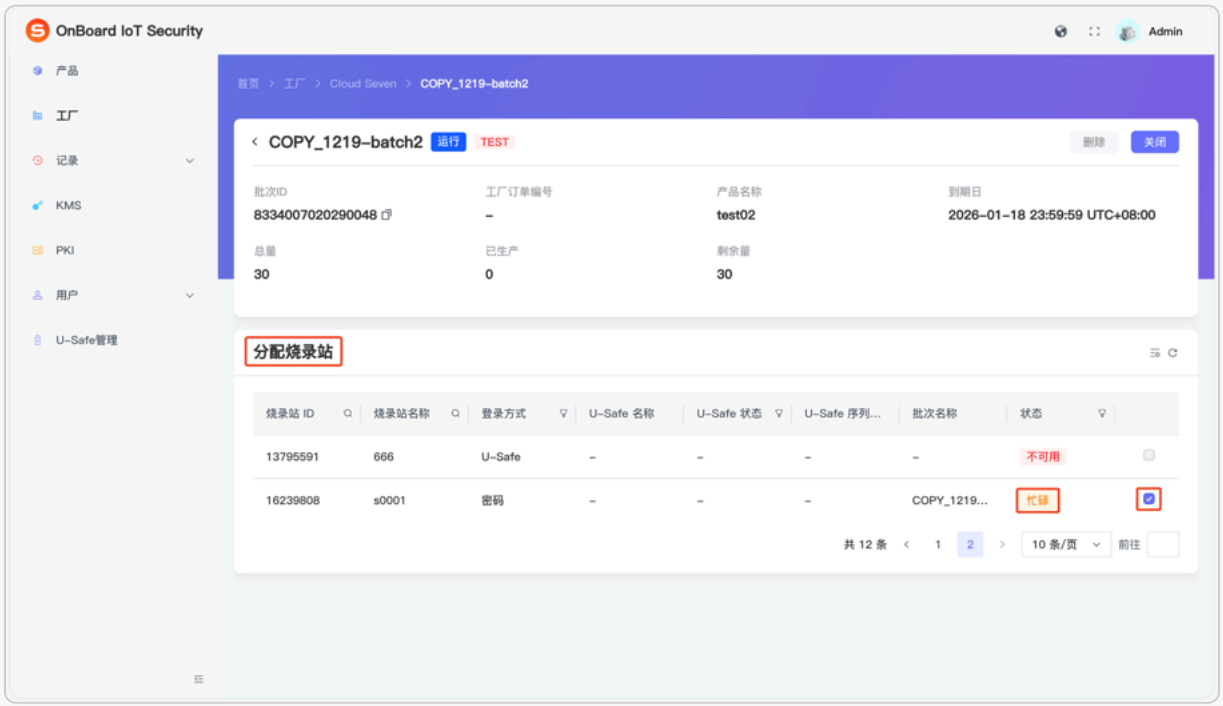
5. 在**确认分配**窗口，点击**确认**，即可完成烧录站的分配。分配后，烧录站状态会变更为**忙碌**。



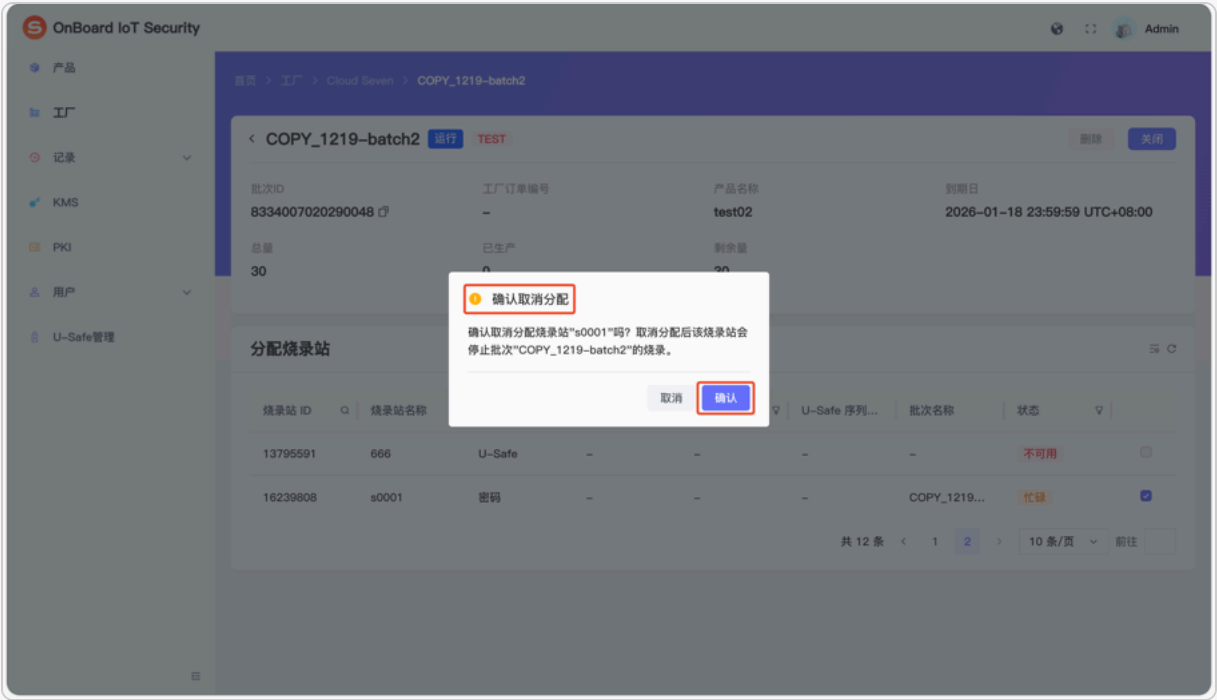
为批次取消分配烧录站

> **注意**
>
> 取消分配后，对应的烧录站无法继续用于该批次的烧录。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**批次**标签页，然后点击对应的批次 ID进入批次页面。
4. 在批次页面的**分配烧录站**区域，找到对应批次，取消勾选状态为**忙碌**的烧录站。



5. 在**确认取消分配**窗口，点击**确认**，即可完成烧录站的取消分配。取消分配后，烧录站状态会变更为**可用**。

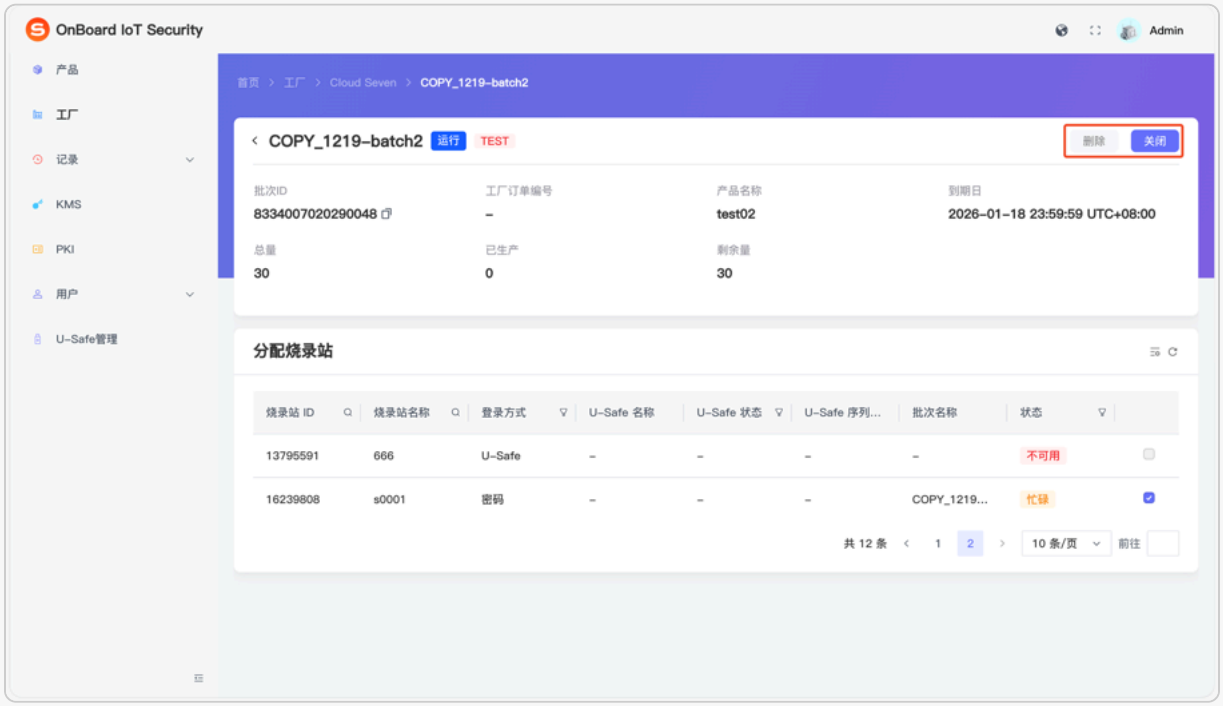


关闭与删除批次

> **信息**
>
> 批次删除后，该批次下已分配的烧录站将自动解除分配。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**批次**标签页，然后点击对应的批次 ID进入批次页面。
4. 在批次页面，点击**关闭**或**删除**。

> **提示**
>
> 若**删除**按钮显示为灰色，表示当前不可用。请点击**关闭**将批次关闭。当批次处于非运行状态时，即可删除批次。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看工厂

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看工厂状态，工厂信息，烧录站信息和批次信息。

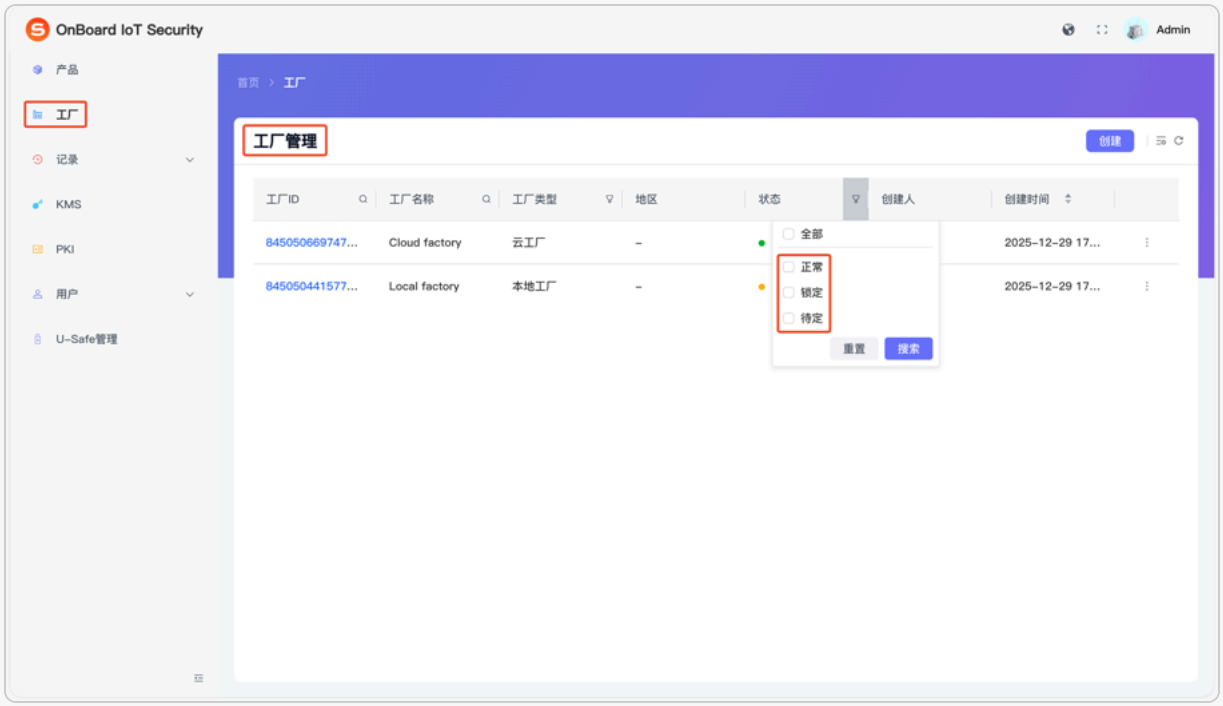
前提条件

在开始之前，请确保你已拥有对应工厂的 Factory Manager 角色权限。

查看工厂状态

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，在**工厂管理**页面，工厂状态列会显示三种状态（**正常**，**锁定**和**待定**）。

> **提示**
>
> 工厂列表包含你参与的及租户内所有已创建的工厂。



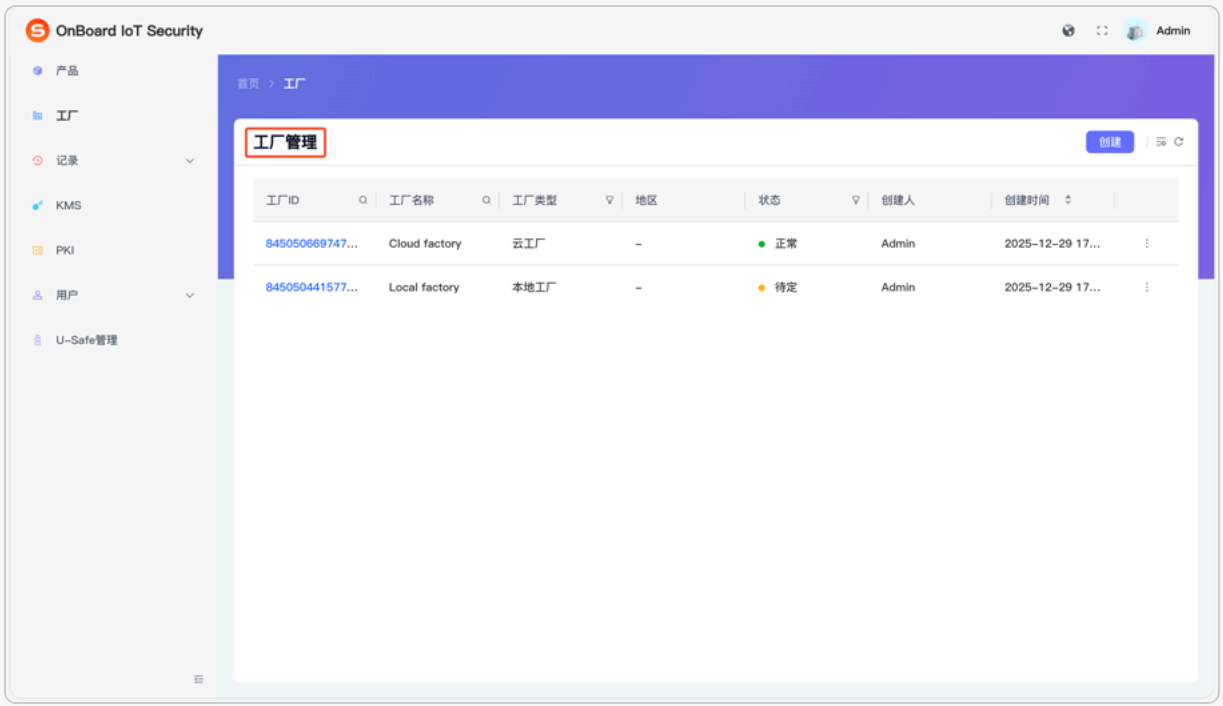
工厂状态说明：

状态	说明
正常	工厂可在创建批次时选择： ◦ 本地工厂：将 HSM U-Safe 添加到本地工厂后，工厂状态将从**待定**变为**正常**。 ◦ 云工厂：云工厂的初始状态为**正常**。
锁定	工厂已被锁定，且无法正常使用。
待定	工厂创建后但尚未添加 HSM U-Safe 时的状态，并无法在创建批次时选择。 **注释**：此状态仅适用于本地工厂。

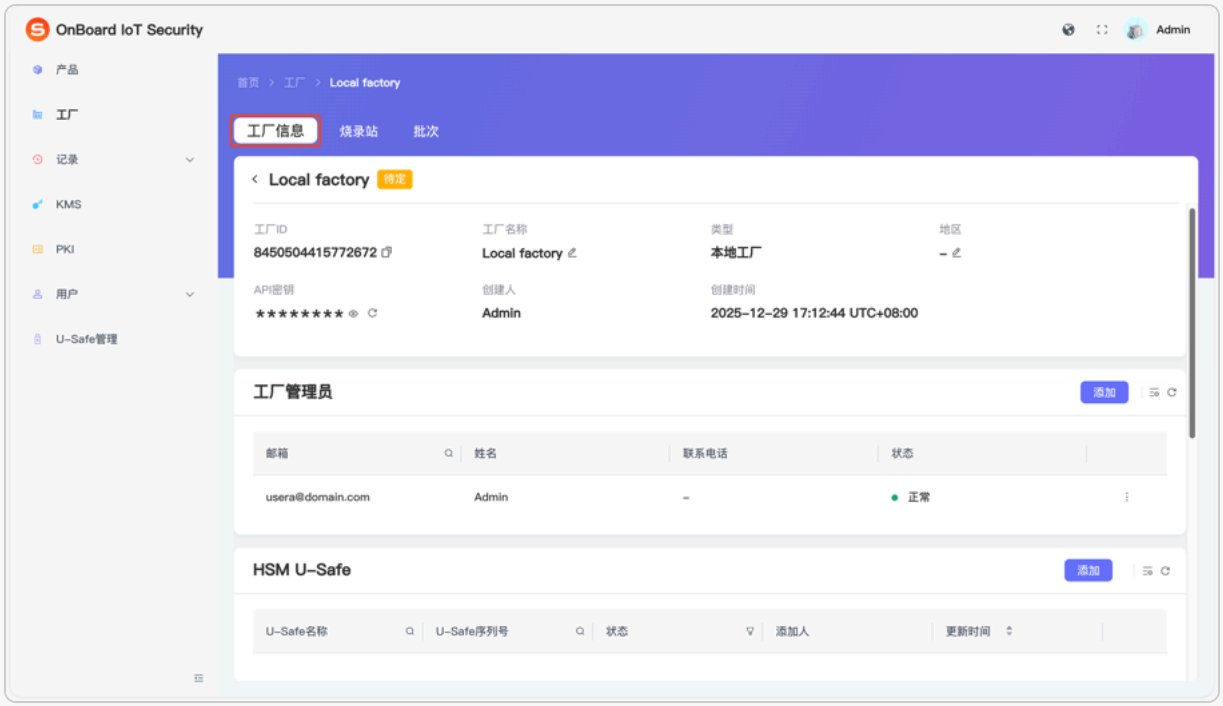
查看工厂信息

1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，在**工厂管理**页面，即可查看工厂的列表。

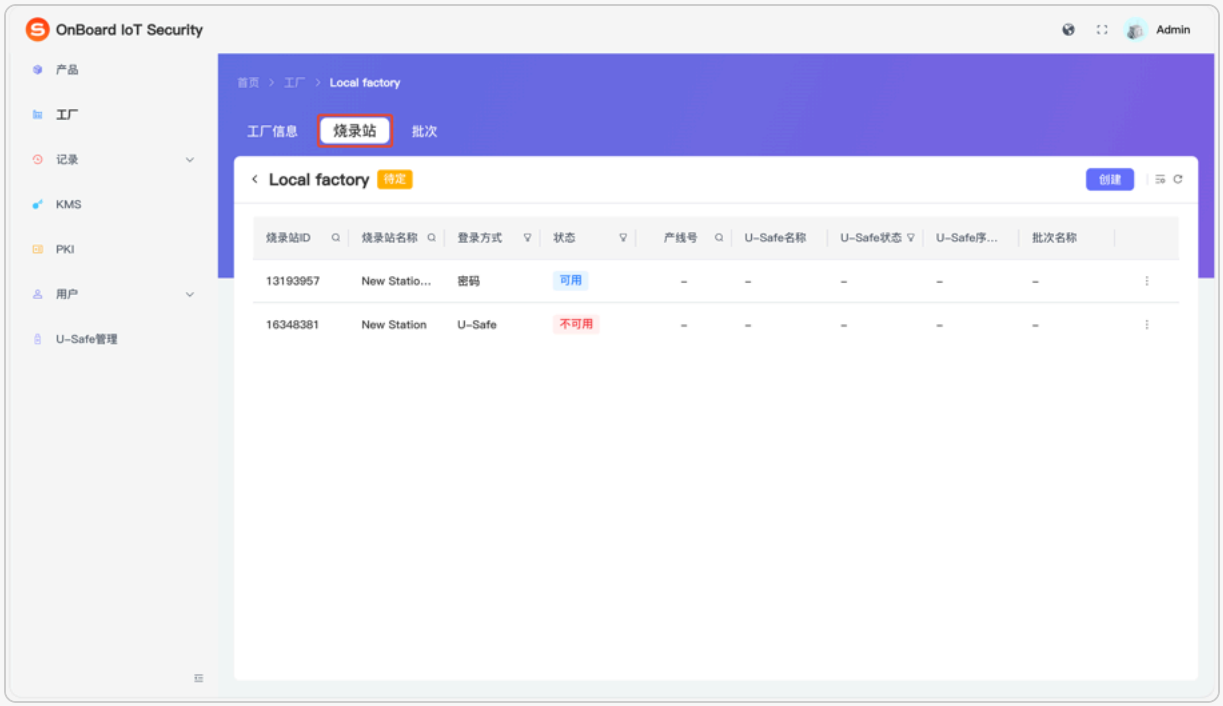


3. 点击对应的工厂 ID，进入工厂页面，即可查看工厂的相关信息（**工厂 ID**，**工厂名称**和**API 密钥**等）。



查看烧录站信息

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**烧录站**标签页，即可查看烧录站的相关信息（**烧录站 ID**，**烧录站名称**和**登录方式**等）。



查看批次信息

本地工厂 | 云工厂

提示

通过查看批次信息，可以获取用于登录 OnBoard Edge Service 和 Programming Station 客户端的服务地址。

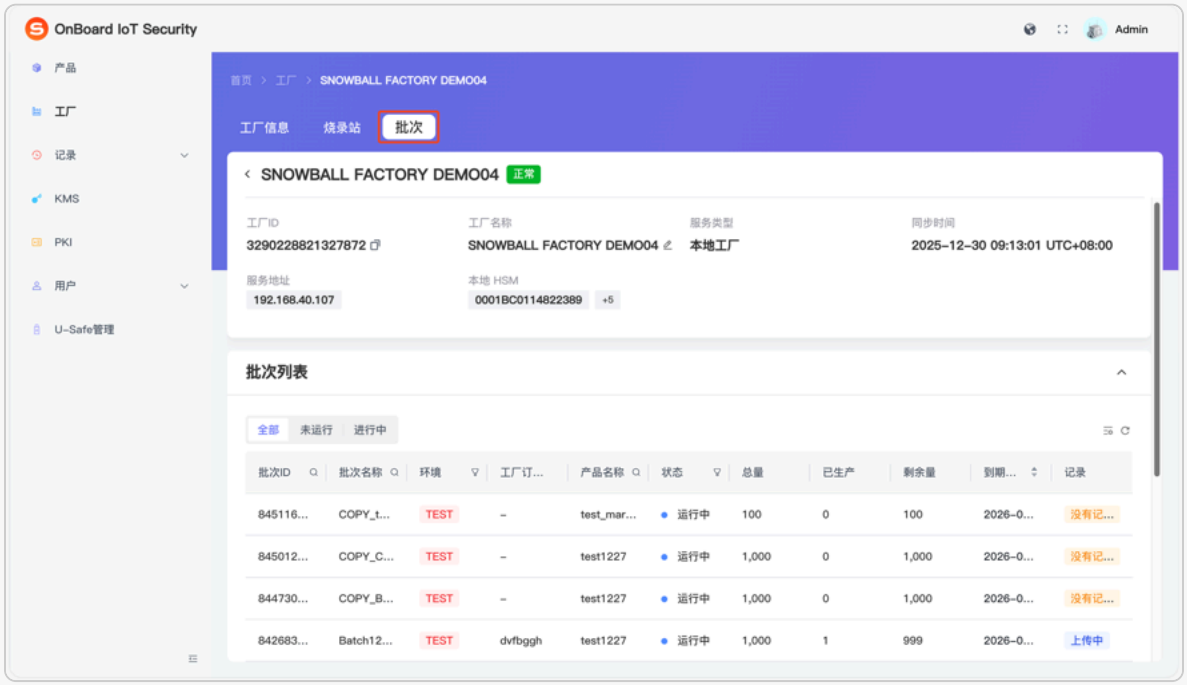
在创建批次时选择对应的工厂后，工厂的**批次**标签页会同步批次数据。

1. 登录 OBIS。
2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。
3. 在工厂页面，选择**批次**标签页，你可以查看以下信息：

- 服务信息：包括**工厂 ID**，**工厂名称**，**服务类型**，**同步时间**，**服务地址**和**本地 HSM**。

> **提示**
>
> 在联系雪球协助完成 OnBoard Edge Service 的部署后，在工厂的**批次**标签页中，找到**服务地址**。此服务地址将用于后续登录 OnBoard Edge Service 和 Programming Station 客户端。

- 批次列表：包括**批次 ID**，**批次名称**和**环境**等。



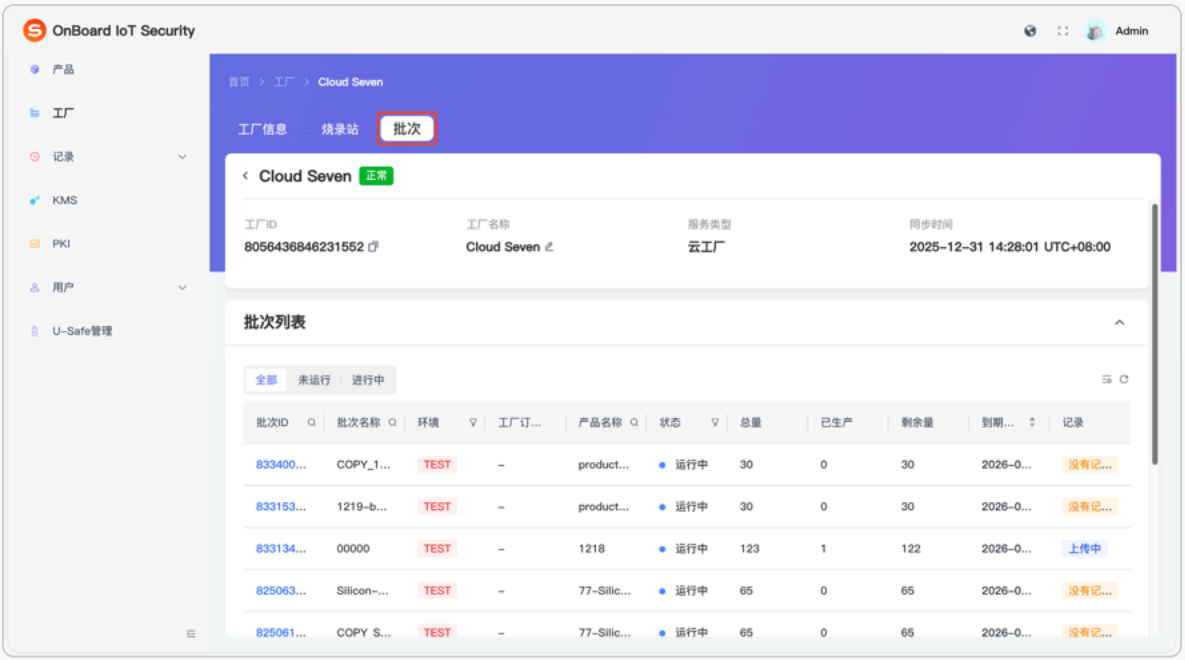
> **提示**
>
> 在创建批次时选择对应的工厂后，工厂的**批次**标签页会同步批次数据。

1. 登录 OBIS。

2. 点击左侧菜单中的**工厂**，然后点击对应的工厂 ID，进入工厂页面。

3. 在工厂页面，选择**批次**标签页，你可以查看以下信息：

    - 服务信息：包括**工厂 ID**，**工厂名称**，**服务类型**和**同步时间**。
    - 批次列表：包括**批次 ID**，**批次名称**和**环境**等。



4. 点击对应的批次 ID，进入批次页面，即可查看批次信息。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看设备历史

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看租户内烧录设备的档案和日志。

前提条件

在开始之前，请确保:

- 已完成设备烧录。
- 你已拥有对应产品的任意产品角色权限或对应工厂的 Factory Manager 角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**记录** > **设备历史**，在**设备历史**页面，即可查看设备历史。
3. 设置搜索条件并点击**搜索**，设备历史列表中将返回所有符合条件的设备历史。

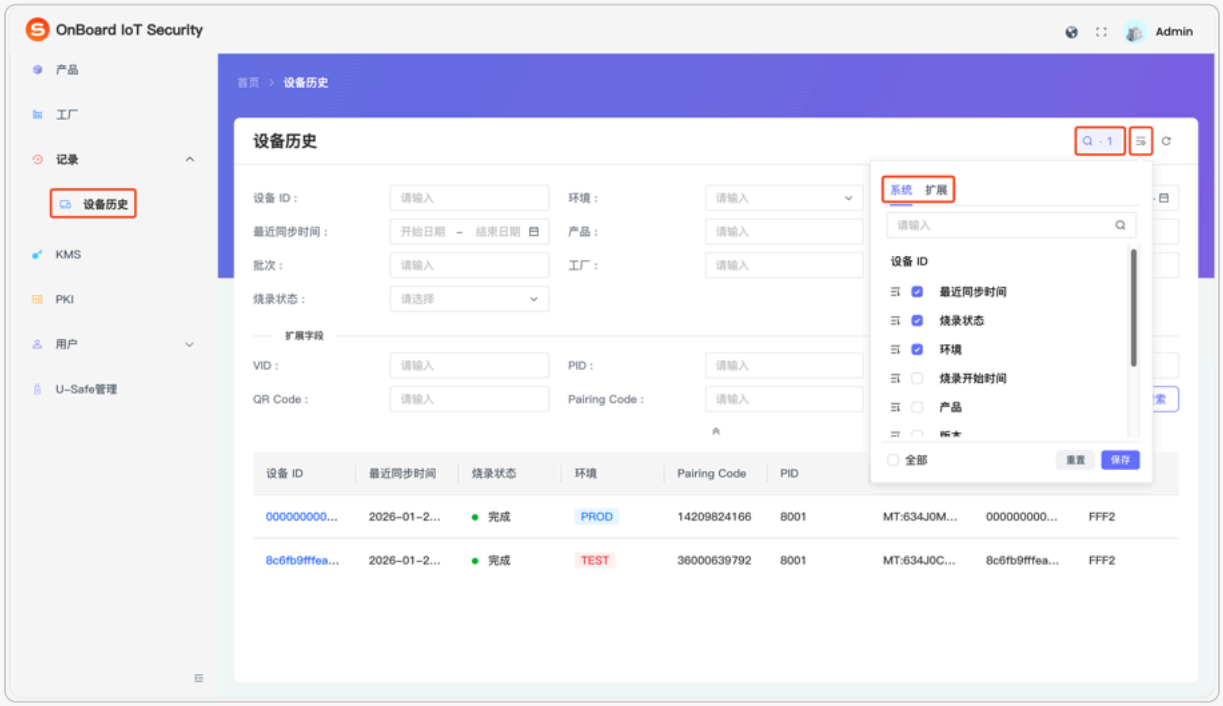
提示

目前支持使用以下条件搜索设备历史记录：

- 系统字段：**设备 ID，环境，烧录开始时间，最近同步时间，产品，版本，批次，工厂，烧录站**和**烧录状态**
- 扩展字段：**VID，PID，Chip UID，QR Code** 和 **Pairing Code**

点击搜索 图标可以隐藏或显示搜索条件。图标右侧的数字表示当前已设置的搜索条件个数。

同时，点击列选项 图标可以自定义显示哪些列以及列的排列顺序。



设备历史信息说明：

信息	说明
设备 ID	设备标识符。
环境	版本对应的环境，例如，**TEST** 环境和 **PROD** 环境。
烧录开始时间	设备开始烧录的日期时间，格式为 **YYYY-MM-DD HH:MM:SS**。
最近同步时间	设备档案最近同步的日期时间，格式为 **YYYY-MM-DD HH:MM:SS**。
产品	设备对应的产品名称。
版本	设备对应的版本名称。
批次	设备对应的批次名称。
工厂	设备对应的工厂名称。
烧录站	设备对应的烧录站名称。
烧录状态	设备结果的状态，例如，**未上报**，**完成**和**失败**。
VID	供应商标识符。
PID	产品标识符。
Chip UID	设备内部芯片的唯一标识符。
QR Code	设备二维码，用于实现安全快速的配网。

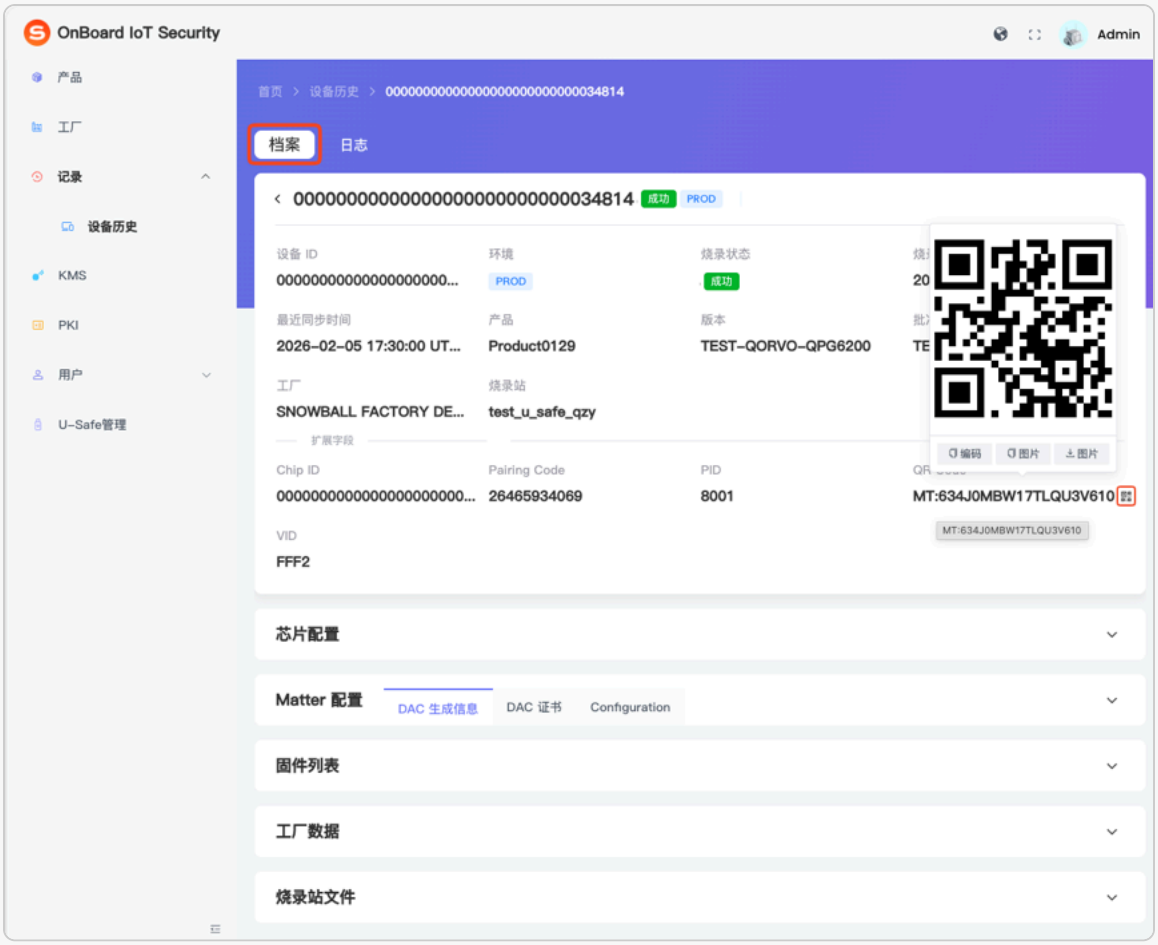
信息	说明
Pairing Code	用户手动输入以完成设备配对的临时验证码。

4. 点击对应的设备 ID，进入设备页面，即可查看设备相关信息。

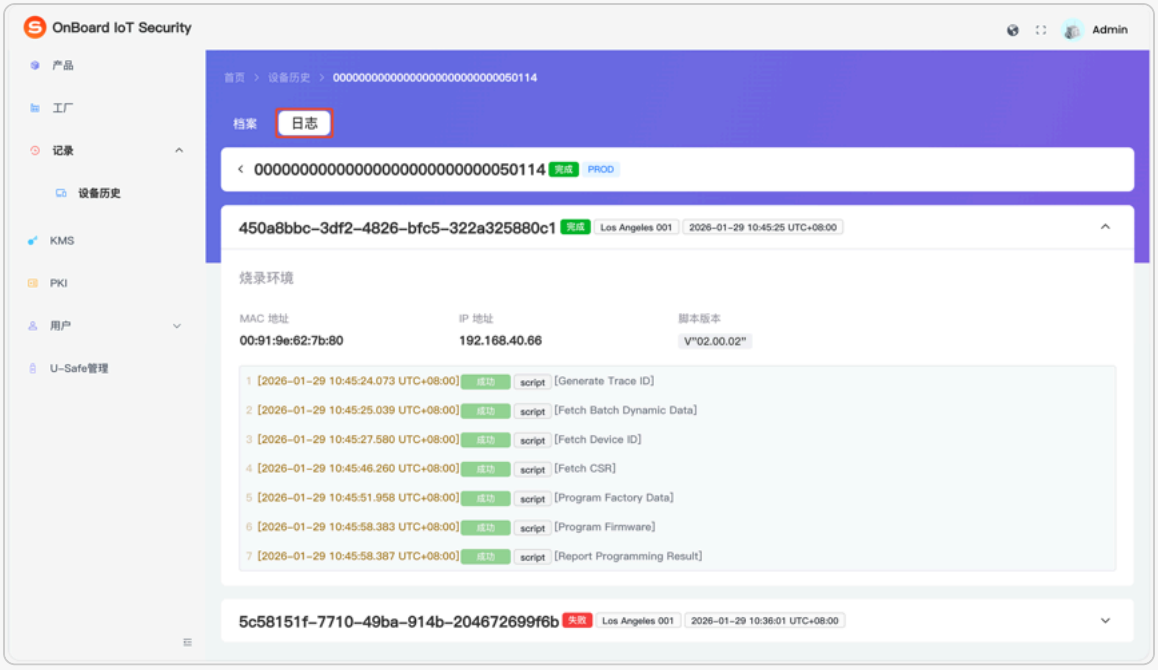
    - **档案**标签页显示了设备的档案数据，可以查看设备的完整信息。

        > **提示**
        >
        > 将鼠标悬停在二维码 图标上，即可显示设备二维码，该二维码用于实现安全快速的配网。
        >
        > - **DAC 生成信息**区域显示了 DAC 相关信息（**DAC 序列号**，**生效日期**和**失效日期**等）。**DAC 序列号**可用于在证书中吊销对应设备的认证状态。
        > - **DAC 证书**区域显示了证书的编码。可通过解析编码提取出证书的颁发者信息，有效期，公钥等关键证书属性。

        - 系统字段和扩展字段
        - 芯片配置
        - Matter 配置（DAC 生成信息，DAC 证书和 Configuration）
        - 固件列表
        - 工厂数据
        - 烧录站文件



- **日志**标签页显示了设备的的烧录日志（每次烧录的日志都会被保存），可以排查故障和定位烧录问题。


OnBoard IoT Security
Admin
产品
工厂
记录
设备历史
KMS
PKI
用户
U-Safe管理
首页 > 设备历史 > 00000000000000000000000000050114
档案
日志
00000000000000000000000000050114 完成 PROD
450a8bbc-3df2-4826-bfc5-322a325880c1 完成 Los Angeles 001 2026-01-29 10:45:25 UTC+08:00
烧录环境
MAC 地址
00:91:9e:62:7b:80
IP 地址
192.168.40.66
脚本版本
V"02.00.02"
1 [2026-01-29 10:45:24.073 UTC+08:00] 成功 script [Generate Trace ID]
2 [2026-01-29 10:45:25.039 UTC+08:00] 成功 script [Fetch Batch Dynamic Data]
3 [2026-01-29 10:45:27.580 UTC+08:00] 成功 script [Fetch Device ID]
4 [2026-01-29 10:45:46.260 UTC+08:00] 成功 script [Fetch CSR]
5 [2026-01-29 10:45:51.958 UTC+08:00] 成功 script [Program Factory Data]
6 [2026-01-29 10:45:58.383 UTC+08:00] 成功 script [Program Firmware]
7 [2026-01-29 10:45:58.387 UTC+08:00] 成功 script [Report Programming Result]
5c58151f-7710-49ba-914b-204672699f6b 失败 Los Angeles 001 2026-01-29 10:36:01 UTC+08:00

版本：v3.2.2

OnBoard™ IoT Security 中 KMS 的概述

OBIS（OnBoard™ IoT Security）中的 KMS（Key Management Service）是负责密钥安全的核心模块，提供密钥的安全管理和访问控制。支持用户创建，导入，集成和管理各类密钥，并基于 HSM（Hardware Security Module）提供高安全等级的密钥生成，存储与使用保障。

- 支持对称密钥与非对称密钥，以及各种密钥算法。
- 支持约束密钥的用途，环境，有效期和访问权限。
- 支持使用多种密钥源创建，导入和集成密钥。
- 支持在多种场景下使用 KMS：
  - 资源加密与解密
  - 固件签名与验证
  - 安全启动（Secure Boot）
  - 安全调试（Secure Debug）
  - HMAC（Hash-based message authentication code）认证

下表总结了 KMS 的核心能力：

核心能力	说明
密钥生命周期管理	• 对称密钥和非对称密钥的生命周期管理
密钥状态与存储	• 密钥状态管理（有效、暂停、吊销） • 硬件支持的安全存储与加密执行

核心能力	说明
企业 BYOK（密钥导入）	• OEM 生成并导入密钥至 HSM • KMS 通过授权策略管理密钥使用
云原生 BYOK（外部 KMS）	• 密钥保留在客户的云 KMS 中（例如 AWS KMS） • KMS 通过安全认证 API 调用操作

KMS 并不规定密钥必须存放的位置，而是规定加密能力的治理、约束和审计方式。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建密钥

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中创建对称密钥和非对称密钥。

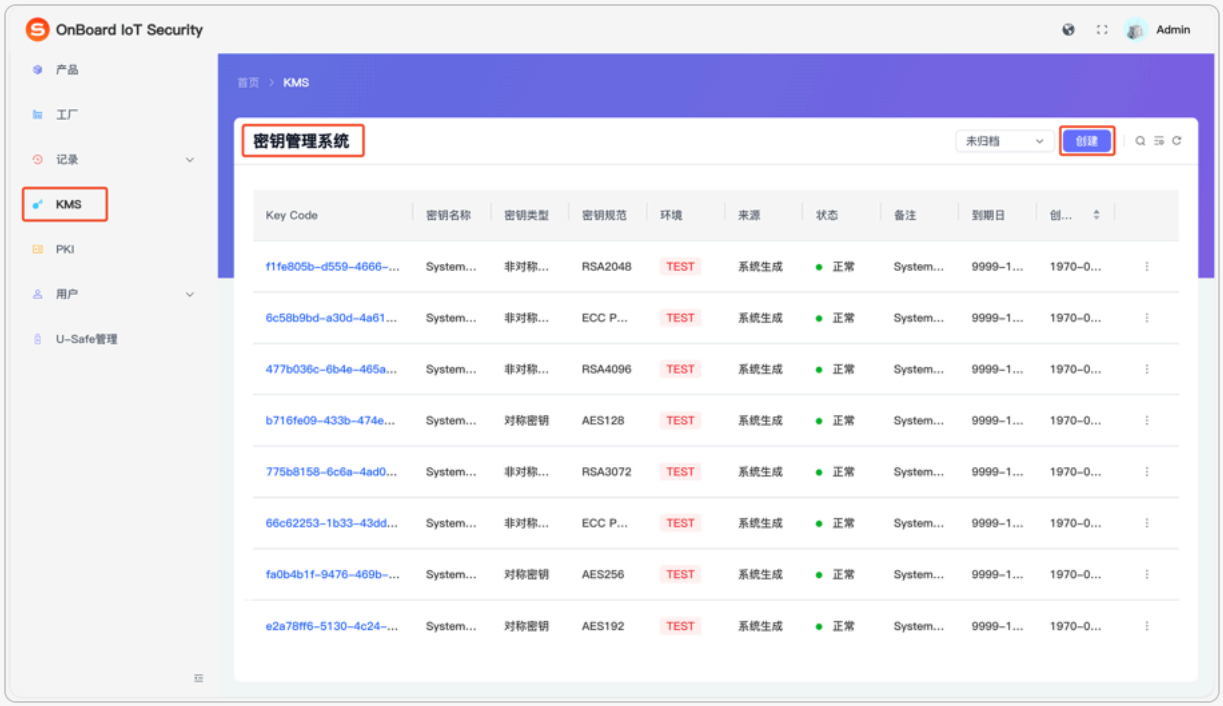
前提条件

在开始之前，请确保你已获得 System Manager 或 Resource Manager 的 OBIS 角色权限。

操作步骤

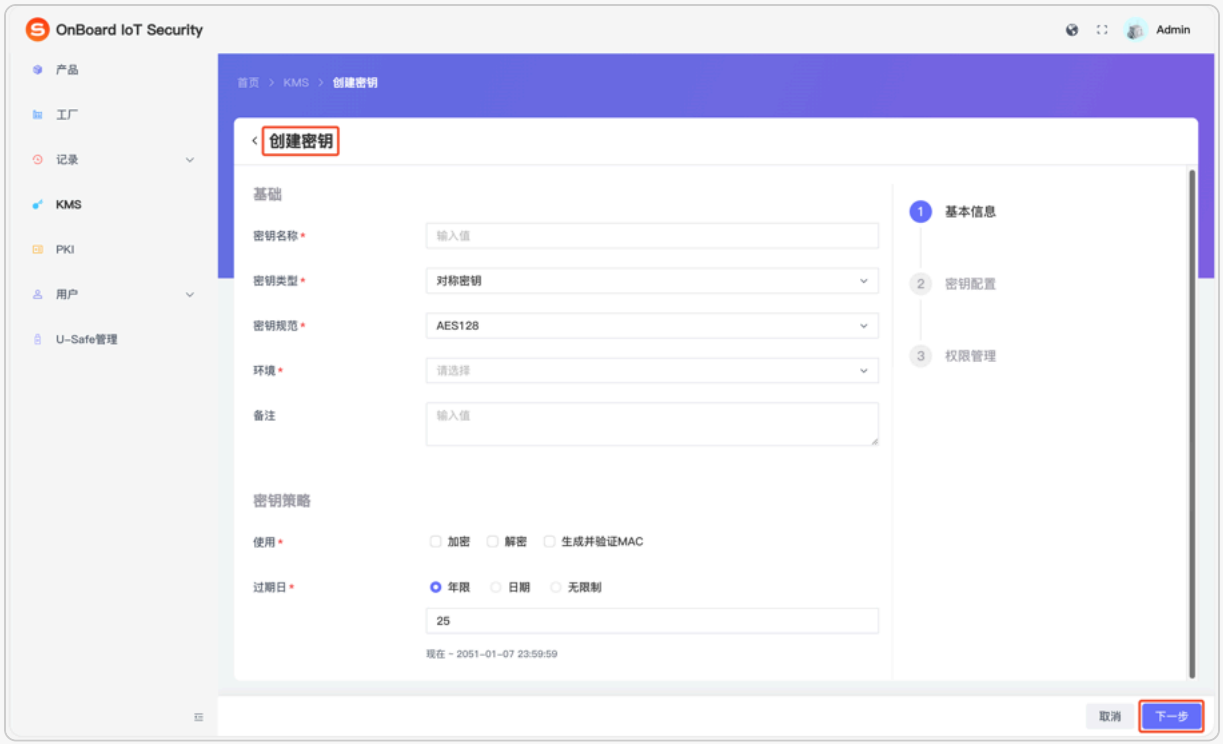
> **提示**
>
> 创建密钥后，可在产品配置 Key 时进行选择。

1. 登录 OBIS。
2. 点击左侧菜单中的 **KMS**，在**密钥管理系统**页面，点击**创建**。



第一步：配置基本信息

1. 在**创建密钥**页面，选择和输入密钥的基本信息。



信息	是否必填	说明
密钥名称	是	名称必须唯一。
密钥类型	是	有两种类型可以选择，**对称密钥**和**非对称密钥**。
密钥规范	是	选择对称密钥和非对称密钥的加密算法： ◦ 对称密钥：**AES128，AES192，AES256** ◦ 非对称密钥：**RSA2048，RSA3072，RSA4096，ECC P256，ECC P384**
环境	是	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。

信息	是否必填	说明
备注	否	用于补充说明密钥相关的额外信息，如特殊要求和注意事项等。
使用	是	限定本密钥所允许执行的操作（可多选）： ◦ **签名**：允许使用非对称密钥生成数字签名。 ◦ **验证**：允许使用非对称密钥验证数字签名。 ◦ **加密**：允许使用对称密钥或非对称密钥对数据进行加密。 ◦ **解密**：允许使用对称密钥或非对称密钥对数据进行解密。 ◦ **生成并验证 MAC**：允许使用对称密钥生成和验证消息认证码，用于保障数据完整性。 **注释**：对于非对称密钥，选择**签名**或**解密**后，**密钥源**区域中的导入公钥功能将不可用。
过期日	是	设定密钥的过期日。

2. 确认基本信息无误后，点击**下一步**，页面会跳转至**密钥配置**页面。

第二步：设定密钥配置

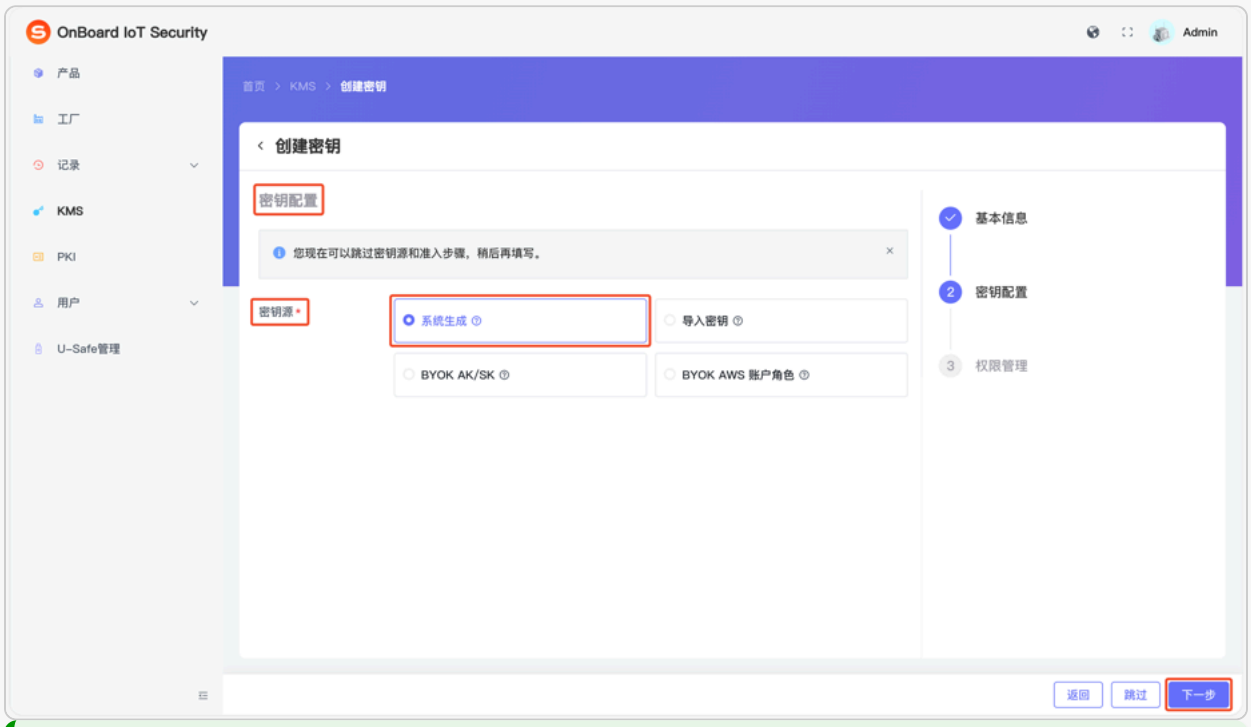
> **提示**
>
> 你可以在**密钥配置**页面中点击**跳过**，暂时不配置密钥源，直接为密钥配置权限。在完成密钥配置前，该密钥状态将处于**待完成**，且该密钥无法正常使用。
>
> 如果你选择跳过密钥配置，后续可在**密钥管理系统**页面中点击对应的密钥。在密钥页面，点击提示中关于**这些最后的步骤**的链接进行配置。

对称密钥

系统生成 | 导入密钥 | BYOK AK/SK | BYOK AWS 账户角色

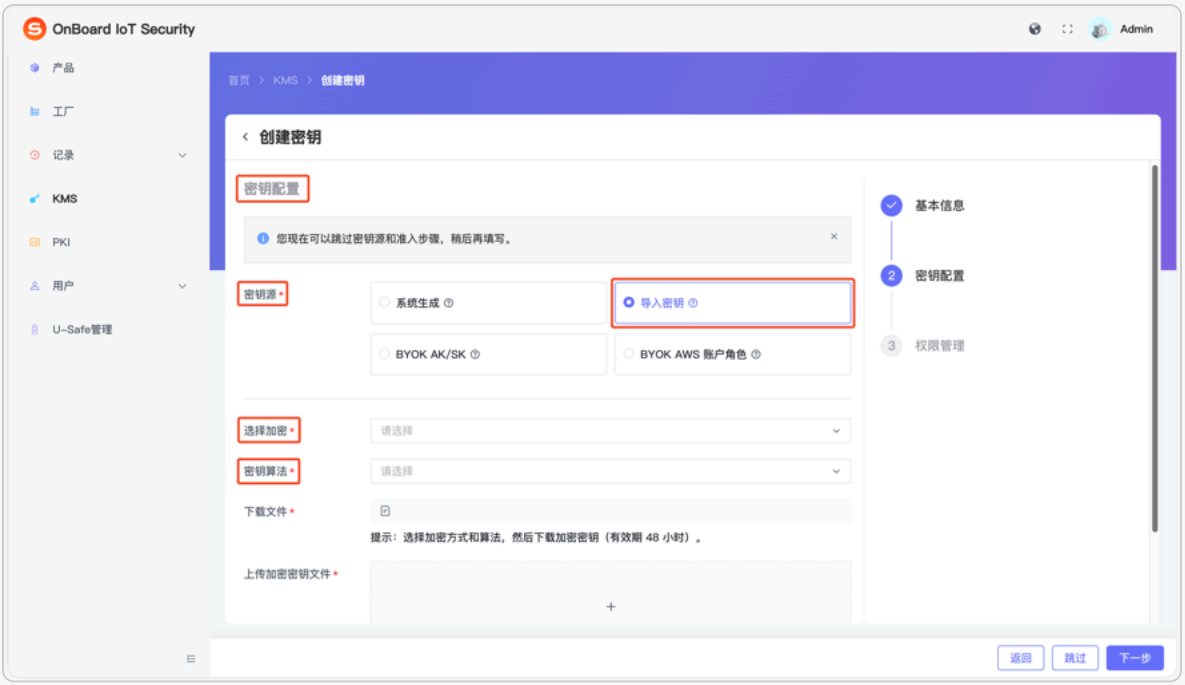
> **提示**
>
> 系统会在硬件安全模块（HSM）内安全地生成一对随机密钥。

在**密钥配置**页面，找到**密钥源**区域，选择**系统生成**，点击**下一步**，页面会跳转至**权限管理**页面。



> **提示**
>
> 手动上传在系统外部生成的对称密钥。

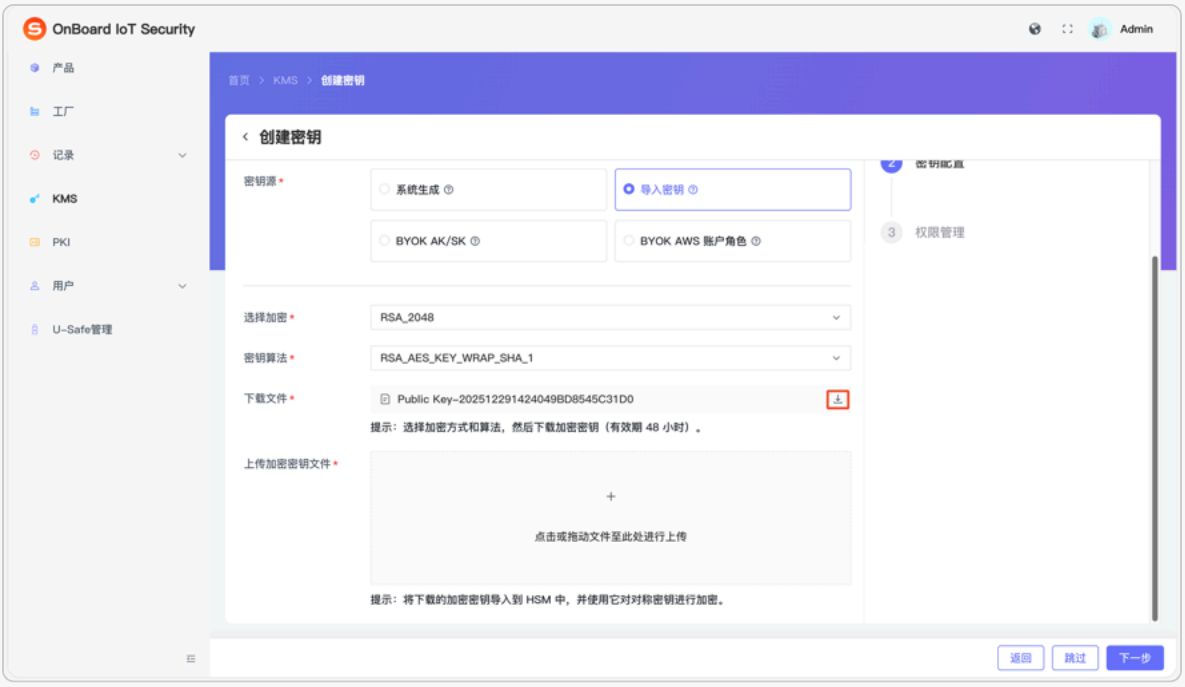
1. 在**密钥配置**页面，找到**密钥源**区域，选择**导入密钥**。选择加密方式和密钥算法。



导入密钥信息说明：

信息	是否必填	说明
选择加密	是	选择加密算法。目前仅支持 **RSA_2048**，提供基本安全强度，适用于密钥加密及数字签名等场景。
密钥算法	是	选择密钥算法: ◦ **RSA_AES_KEY_WRAP_SHA_1** ◦ **RSA_AES_KEY_WRAP_SHA_256**

2. 点击下载 图标，下载加密密钥文件。



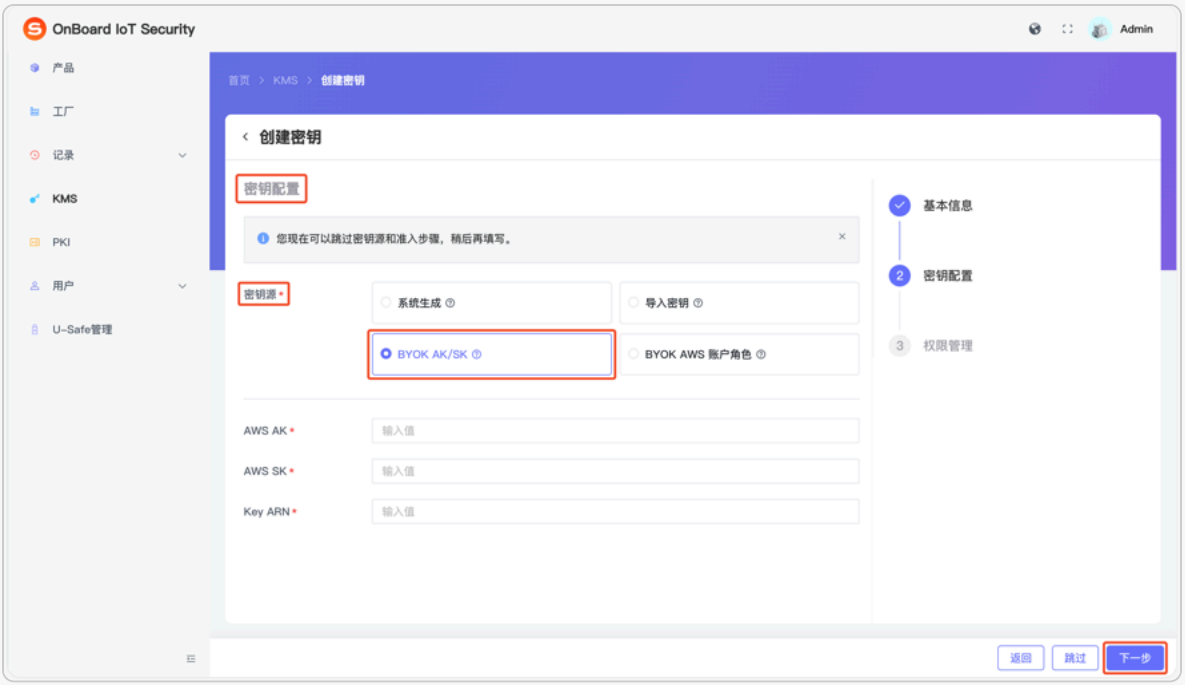
3. 将下载的加密密钥导入到 HSM 中，并使用它对对称密钥进行加密。

4. 对称密钥加密完成后，在**上传加密密钥文件**区域中，选择并上传格式为 **.json** 的加密密钥文件。

5. 确认信息无误后，点击**下一步**，页面会跳转至**权限管理**页面。

> **提示**
>
> 使用访问密钥和密钥凭证从第三方 KMS 访问外部密钥。

1. 在**密钥配置**页面，找到**密钥源**区域，选择 **BYOK AK/SK** ，输入相关的信息。



AWS AK/SK 信息说明：

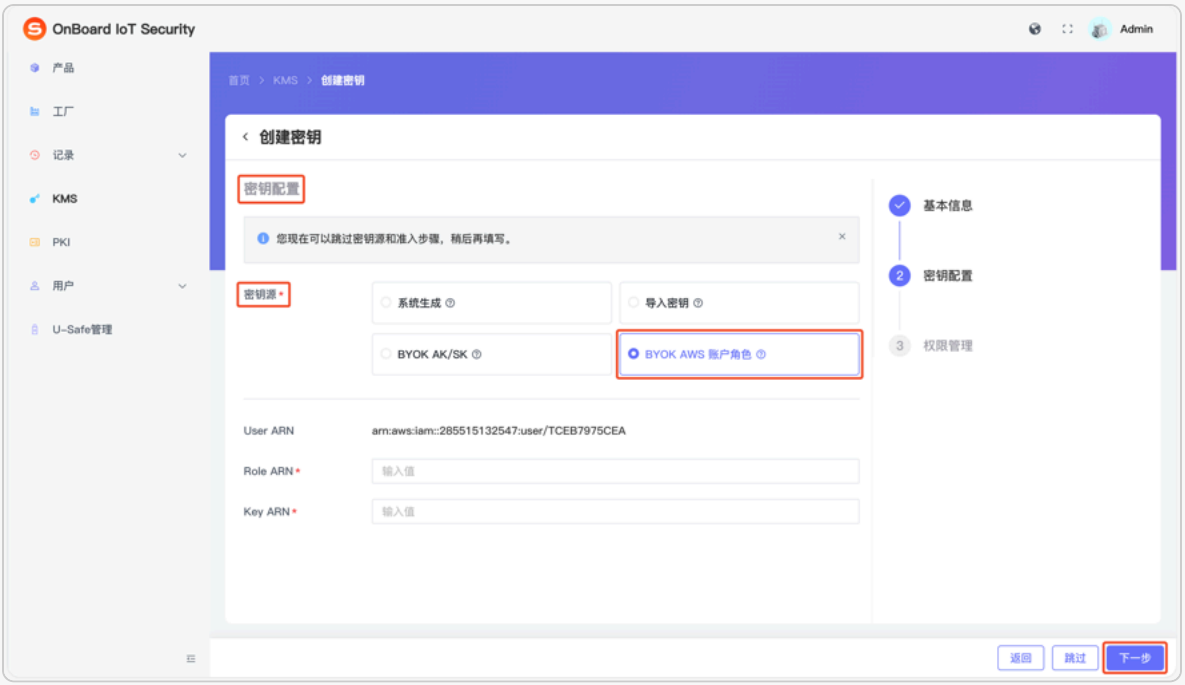
信息	是否必填	说明
AWS AK	是	AK（Access Key ID）： ◦ 身份标识符：唯一标识 AWS IAM (Identity and Access Management) 中的用户，类似于用户名。 ◦ 公开属性：可安全分享（如配置在代码或 CLI 中），但需配合 SK 使用才能完成认证。

信息	是否必填	说明
AWS SK	是	SK（Secret Access Key）： ◦ 私密凭证：用于生成请求签名，验证身份合法性，类似于密码。 ◦ 严格保密：泄露会导致账户被完全控制，需像密码一样妥善保管。
Key ARN	是	ARN (Amazon Resource Name)：唯一标识 AWS KMS 中的加密密钥，用于数据加密与解密。

2. 确认信息无误后，点击**下一步**，页面会跳转至**权限管理**页面。

> **提示**
>
> 使用基于 IAM 角色的访问权限连接到 AWS KMS - 安全可靠，是云原生部署的理想选择。

1. 在**密钥配置**页面，找到**密钥源**区域，选择 **BYOK AWS 账户角色**，输入相关的信息。



AWS 账户角色信息说明：

信息	是否必填	说明
User ARN	否	唯一标识 AWS IAM 中的用户（如管理员或开发者等），用于权限分配和资源访问控制。
Role ARN	是	唯一标识 AWS IAM 中的角色，用于临时权限分配（如 EC2 实例或 Lambda 函数等跨账户访问）。
Key ARN	是	唯一标识 AWS KMS 中的加密密钥，用于数据加密与解密。

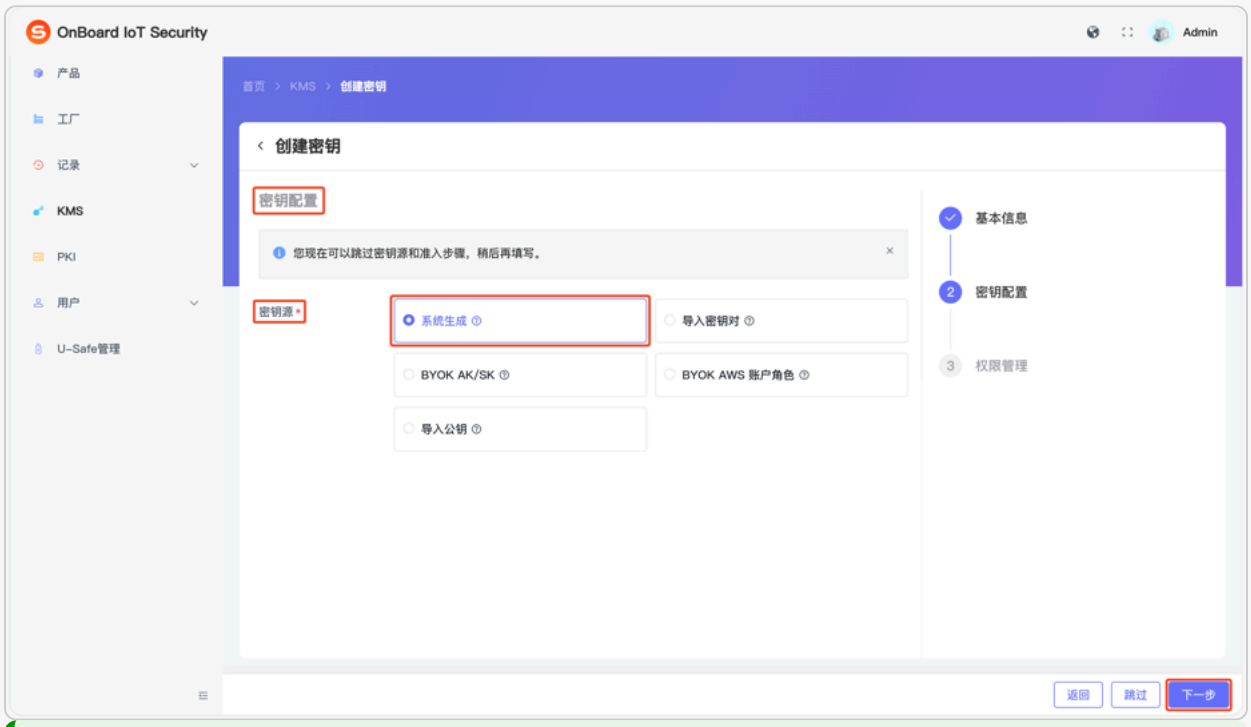
2. 确认信息无误后，点击**下一步**，页面会跳转至**权限管理**页面。

非对称密钥

系统生成 | 导入密钥对 | BYOK AK/SK | BYOK AWS 账户角色 | 导入公钥

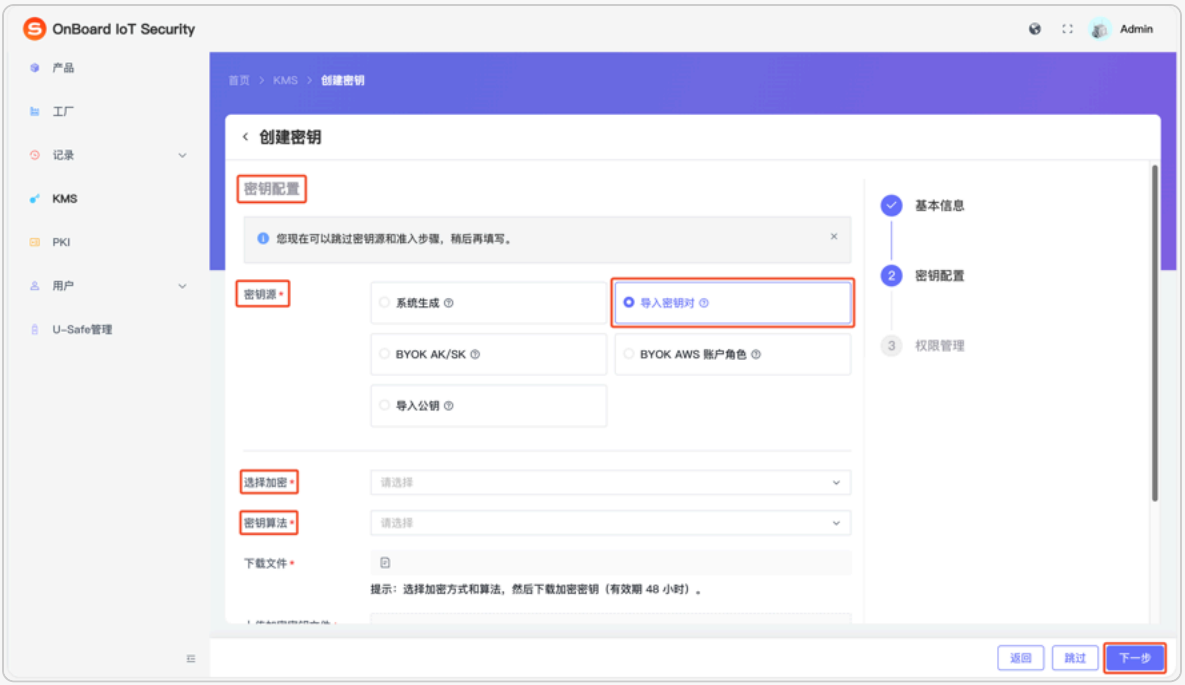
> **提示**
>
> 系统会在硬件安全模块（HSM）内安全地生成一对随机密钥。

在**密钥配置**页面，找到**密钥源**区域，选择**系统生成**，点击**下一步**，页面会跳转至**权限管理**页面。



> **提示**
>
> 手动上传在系统外部生成的非对称密钥对。

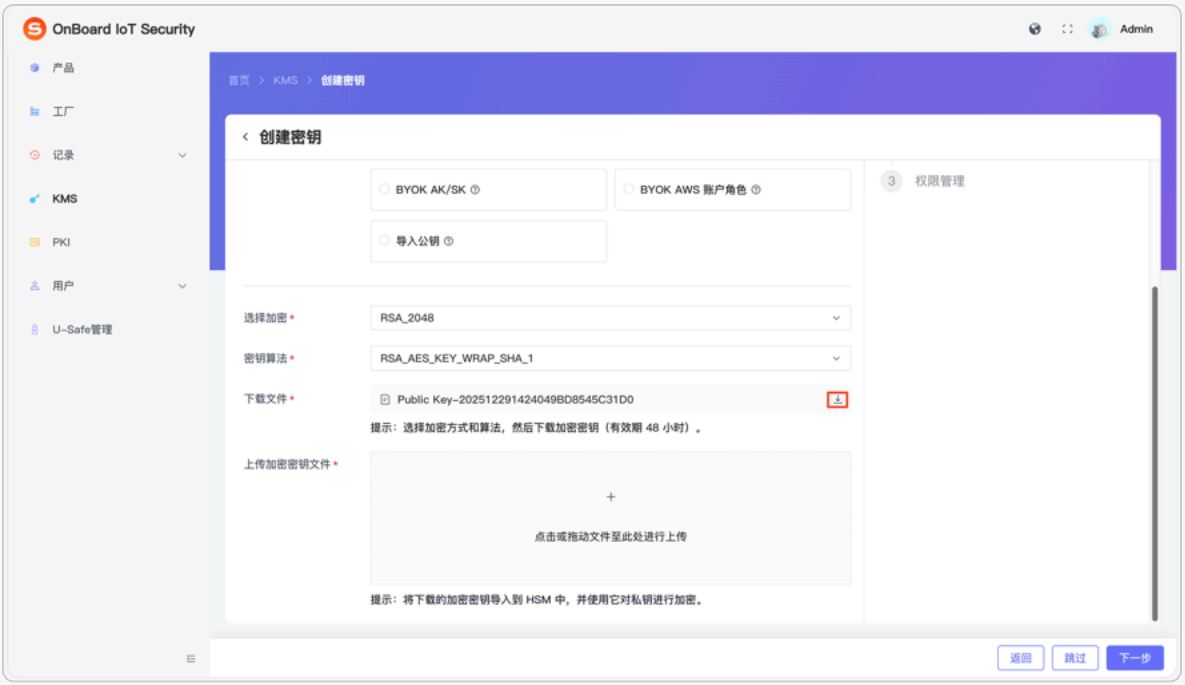
1. 在**密钥配置**页面，找到**密钥源**区域，选择**导入密钥**，选择加密方式和密钥算法。



导入密钥对信息说明：

信息	是否必填	说明
选择加密	是	选择加密算法。目前仅支持 **RSA_2048**，提供基本安全强度，适用于密钥加密及数字签名等场景。
密钥算法	是	选择密钥算法： ◦ **RSA_AES_KEY_WRAP_SHA_1** ◦ **RSA_AES_KEY_WRAP_SHA_256**

2. 点击下载 图标，下载加密密钥文件。



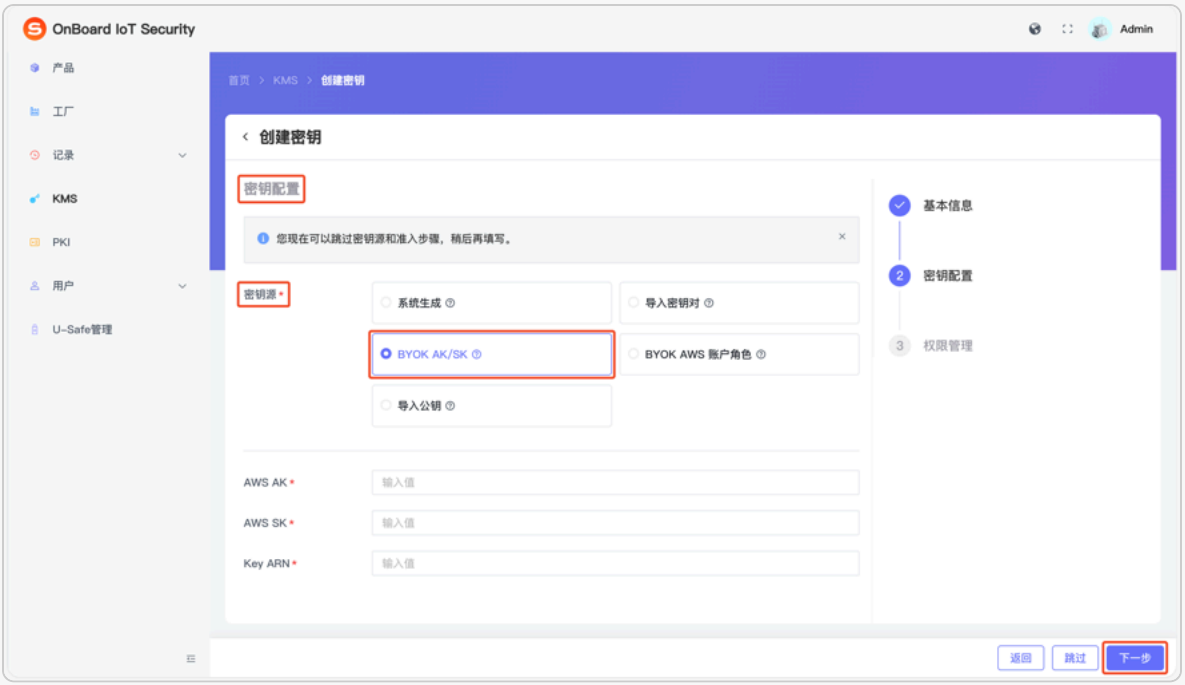
3. 将下载的加密密钥导入到 HSM 中，并使用它对私钥进行加密。

4. 私钥加密完成后，在**上传加密密钥文件**区域中，上传格式为 **.json** 的加密密钥文件。

5. 确认信息无误后，点击**下一步**，页面会跳转至**权限管理**页面。

> **提示**
>
> 使用访问密钥和密钥凭证从第三方 KMS 访问外部密钥。

1. 在**密钥配置**页面，找到**密钥源**区域，选择 **BYOK AK/SK**，选择并输入相关的信息。



AWS AK/SK 信息说明：

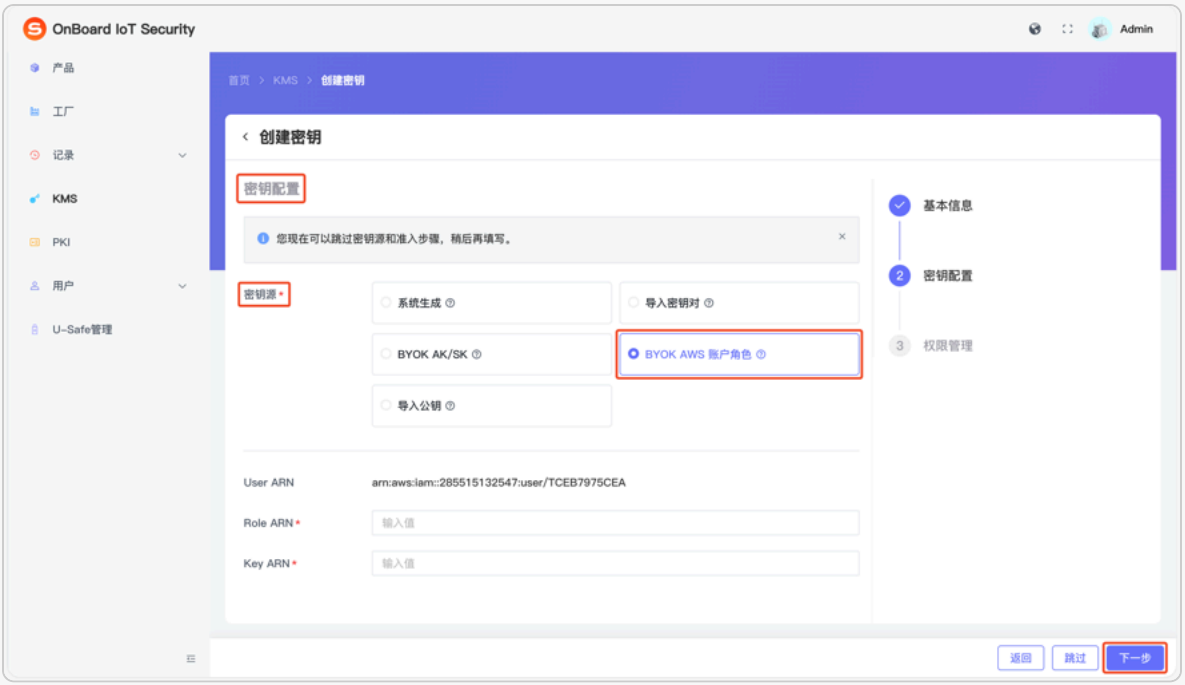
信息	是否必填	说明
AWS AK	是	AK（Access Key ID）： ◦ 身份标识符：唯一标识 AWS IAM (Identity and Access Management) 中的用户，类似于用户名。 ◦ 公开属性：可安全分享（如配置在代码或 CLI 中），但需配合 SK 使用才能完成认证。

信息	是否必填	说明
AWS SK	是	SK（Secret Access Key）： ◦ 私密凭证：用于生成请求签名，验证身份合法性，类似于密码。 ◦ 严格保密：泄露会导致账户被完全控制，需像密码一样妥善保管。
Key ARN	是	ARN (Amazon Resource Name)：唯一标识 AWS KMS 中的加密密钥，用于数据加密与解密。

2. 确认信息无误后，点击**下一步**，页面会跳转至**权限管理**页面。

> **提示**
>
> 使用基于 IAM 角色的访问权限连接到 AWS KMS - 安全可靠，是云原生部署的理想选择。

1. 在**密钥配置**页面，找到**密钥源**区域，选择 **BYOK AWS 账户角色**，输入相关的信息。



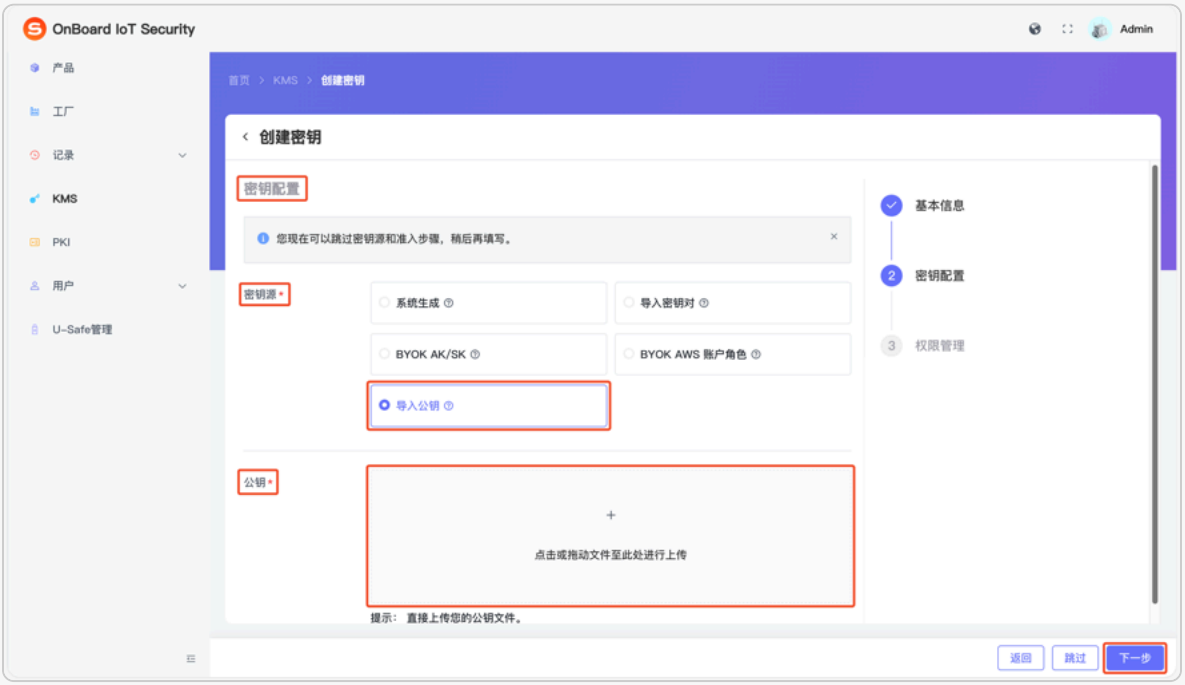
AWS 账户角色信息说明：

信息	是否必填	说明
User ARN	否	唯一标识 AWS IAM 中的用户（如管理员或开发者等），用于权限分配和资源访问控制。
Role ARN	是	唯一标识 AWS IAM 中的角色，用于临时权限分配（如 EC2 实例或 Lambda 函数等跨账户访问）。
Key ARN	是	唯一标识 AWS KMS 中的加密密钥，用于数据加密与解密。

2. 确认信息无误后，点击**下一步**，页面会跳转至**权限管理**页面。

> **提示**
>
> 仅导入用于验证签名或设备身份的公钥 - 无需私钥。

1. 在**密钥配置**页面，找到**密钥源**区域，选择**导入公钥**，在**公钥**区域中，上传格式为 **.json** 的公钥文件。

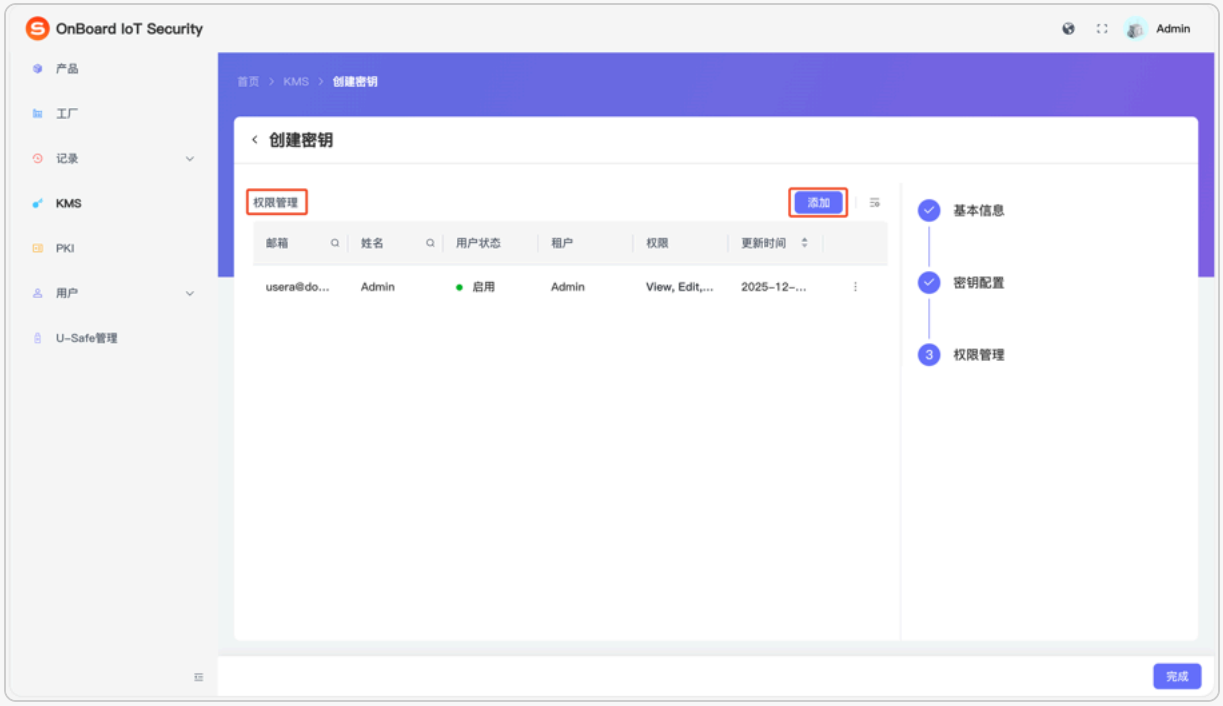


2. 确认文件无误后，点击**下一步**，跳转至**权限管理**页面。

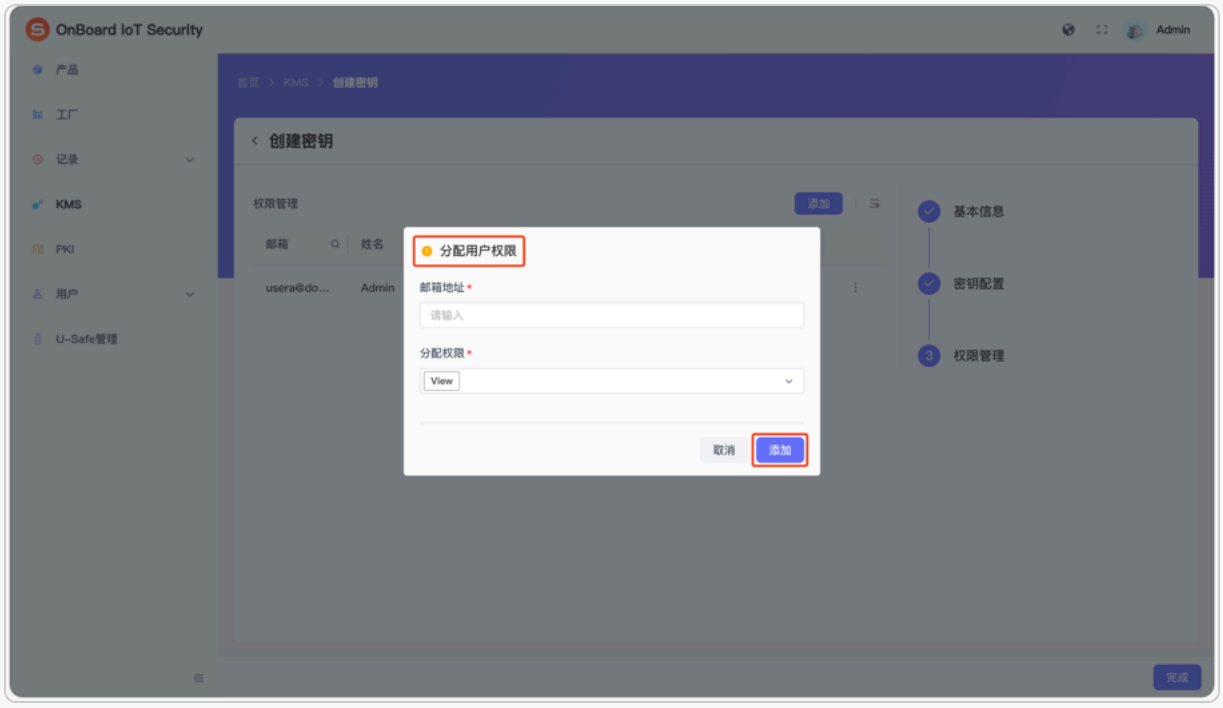
第三步：配置密钥权限

密钥源配置完成后，你可以通过以下步骤，调整并设定用户对密钥的权限（View，Edit，Use，Archive，Lock， Grant 和 Delete）：

1. 在**权限管理**页面，点击**添加**。



2. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限。然后，点击**添加**。



3. 点击**完成**，密钥创建成功后可在**密钥管理系统**页面中查看。

版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看密钥

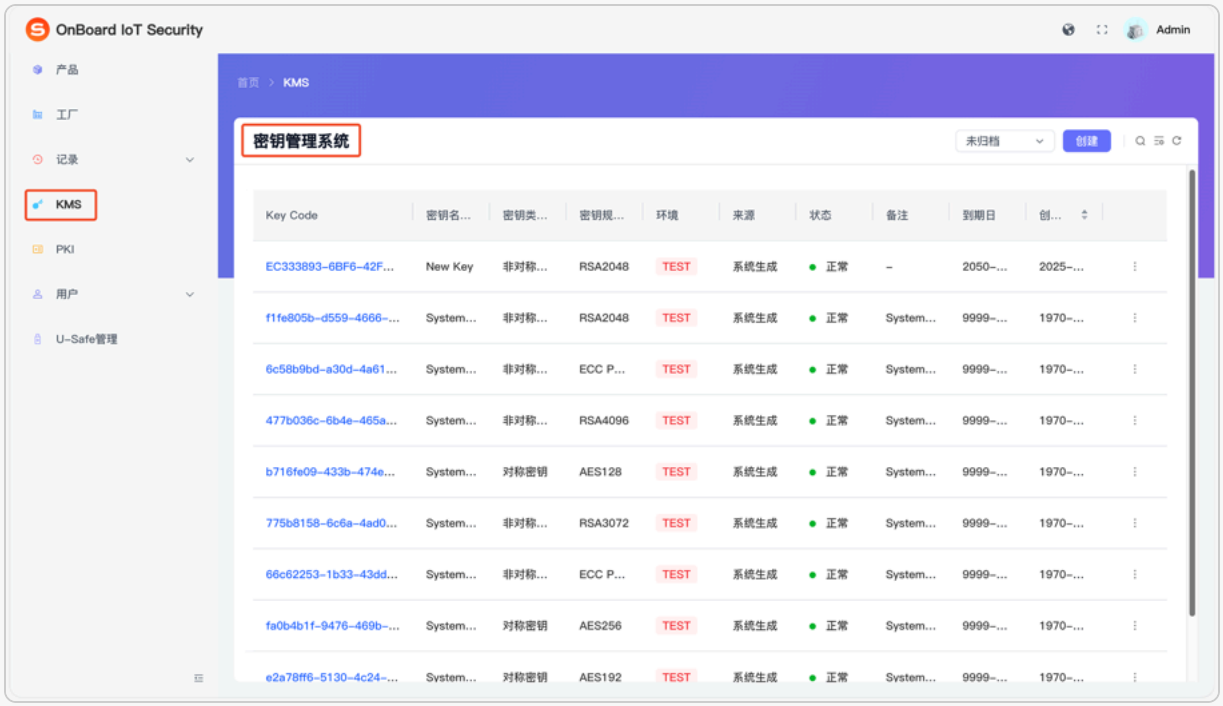
本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看密钥的信息。

操作步骤

1. 登录 OBIS。

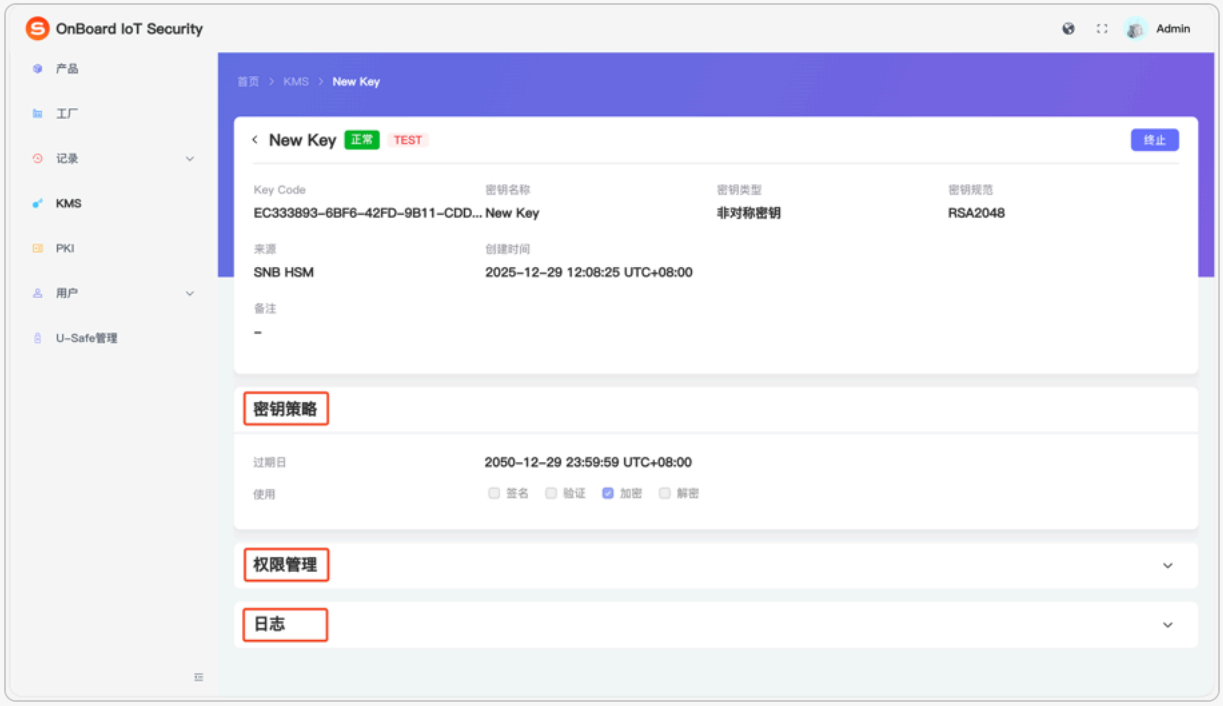
2. 点击左侧菜单中的 **KMS**，进入**密钥管理系统**页面，即可查看密钥信息（**Key Code**，**密钥名称**和**密钥类型**等）。同时，可以根据密钥信息搜索相应密钥，并筛选显示已归档和未归档的密钥。

   > **提示**
   >
   > 目前支持使用以下条件搜索密钥：**Key Code**，**密钥名称**，**密钥类型**，**密钥规范**，**环境**，**来源**，**状态**和**创建时间**。
   >
   > 点击搜索图标可以显示或隐藏搜索条件。



3. 在**密钥管理系统**页面，点击对应的 Key Code，在密钥页面，即可查看密钥相关信息（基础信息，密钥策略，权限管理和日志）。

> **提示**
>
> 在**权限管理**区域，可以管理该密钥的权限。
>
> 在**日志**区域，可以查看密钥的历史操作记录。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > KMS > New Key
New Key 正常 TEST
终止
Key Code
EC333893-6BF6-42FD-9B11-CDD...
密钥名称
New Key
密钥类型
非对称密钥
密钥规范
RSA2048
来源
SNB HSM
创建时间
2025-12-29 12:08:25 UTC+08:00
备注
-
密钥策略
过期日
2050-12-29 23:59:59 UTC+08:00
使用
签名
验证
加密
解密
权限管理
日志

版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理密钥

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

- 管理密钥权限
- 管理密钥归档
- 完成密钥
- 下载密钥
- 锁定或解锁密钥
- 删除密钥

前提条件

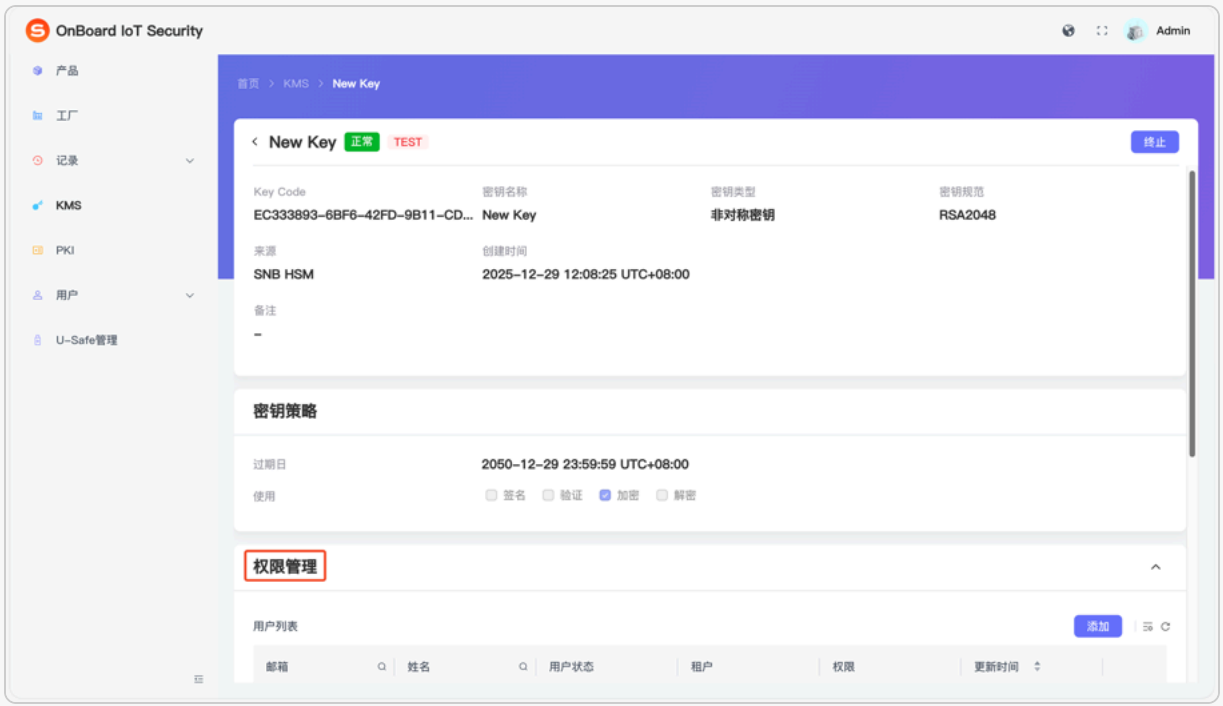
在开始之前，请确保你已获得 System Manager 或 Resource Manager 的 OBIS 角色权限，或拥有对应密钥的操作权限。

管理密钥权限

你可以通过以下步骤调整并设定用户对密钥的权限（View，Edit，Use，Archive，Lock，Grant 和 Delete）：

1. 登录 OBIS。
2. 点击左侧菜单中的 **KMS**，然后点击对应密钥的 Key Code 进入密钥页面。

3. 在密钥页面的**权限管理**区域，可以管理该密钥的权限。

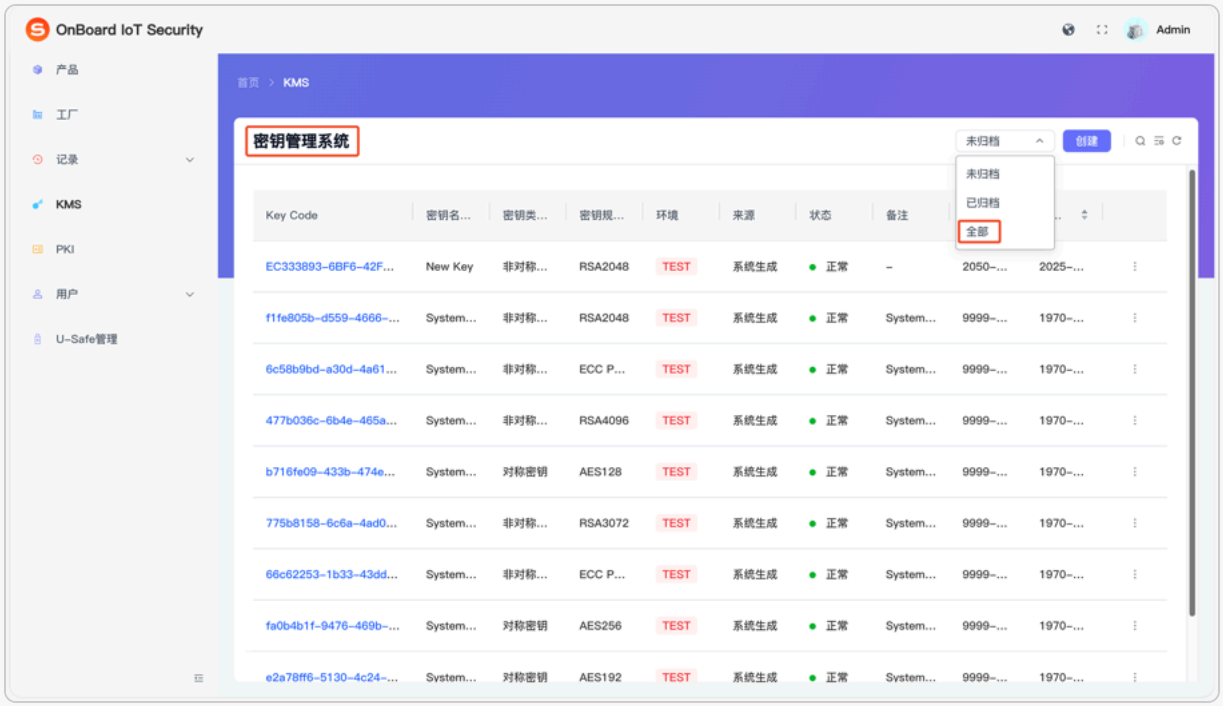


管理密钥归档

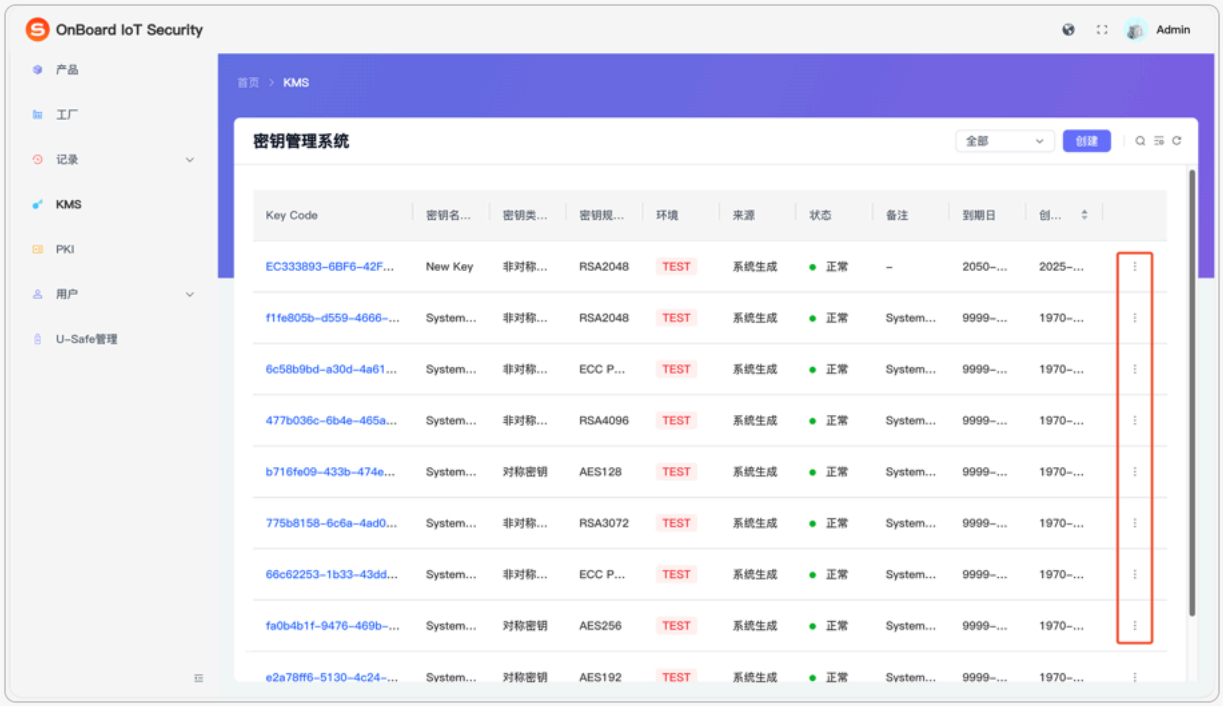
> **提示**
>
> 密钥列表默认显示未归档的密钥。密钥归档后，在配置产品 Key 时，该密钥将显示在密钥列表底部。

1. 登录 OBIS。

2. 点击左侧菜单中的 **KMS**，并在**密钥管理系统**页面的右上角选择**全部**。



3. 点击对应密钥的选项 ⋮ 图标并选择**归档**或**取消归档**，即可归档或取消归档该密钥。

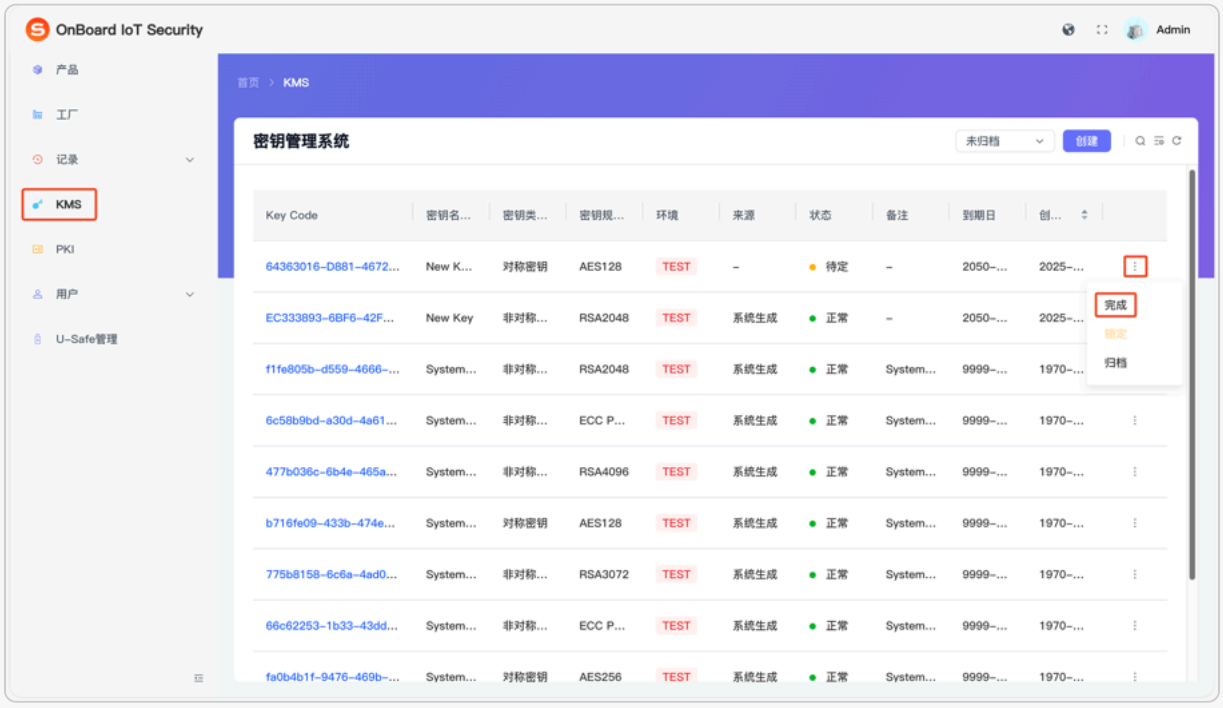


完成密钥

> **提示**
>
> 完成密钥创建后，可在产品配置 Key 时进行选择。

1. 登录 OBIS。

2. 点击左侧菜单中的 **KMS**，然后点击对应密钥的选项 ⋮ 图标并选择**完成**，然后参考创建密钥步骤完成密钥的创建。

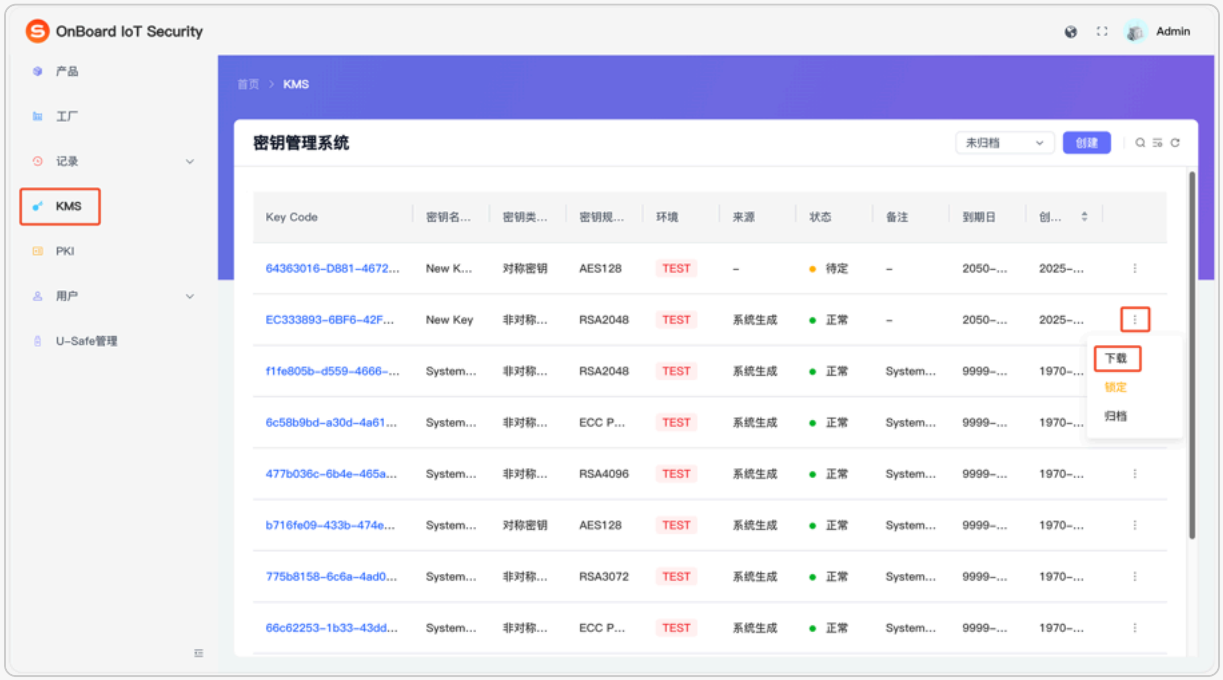


下载密钥

> **提示**
>
> 下载非对称密钥的公钥用于加密与验证。

1. 登录 OBIS。

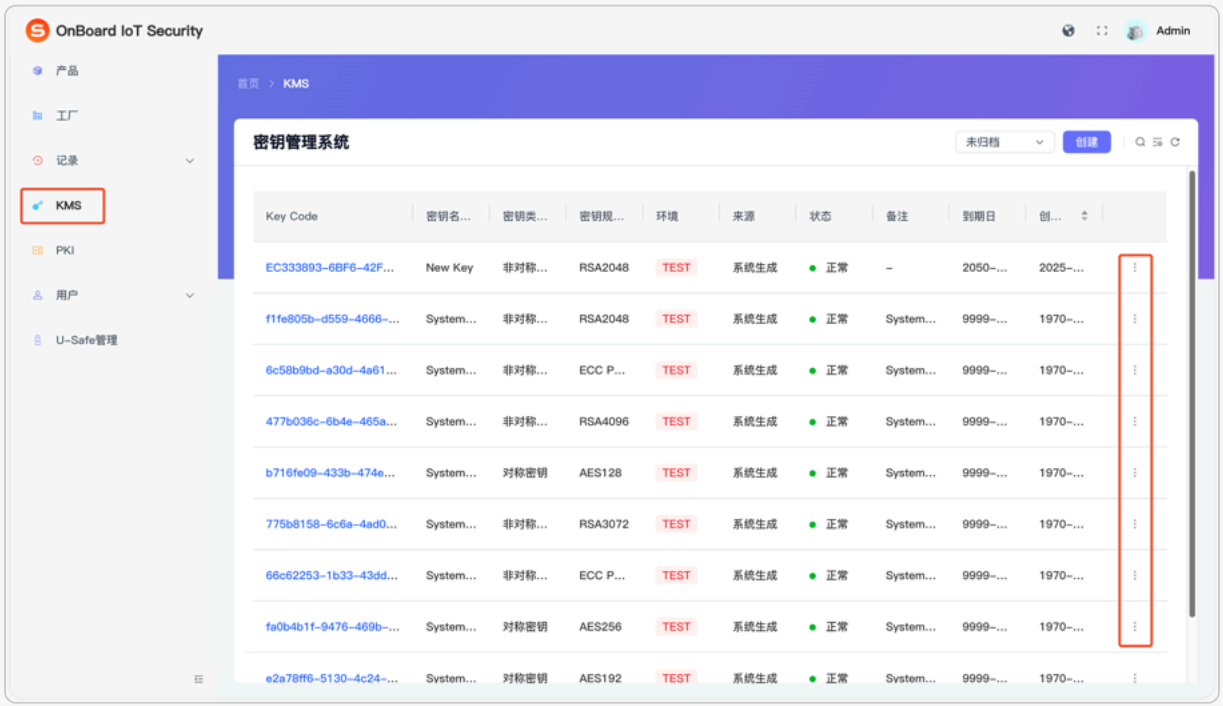
2. 点击左侧菜单中的 **KMS**，然后点击对应非对称密钥的选项 ⋮ 图标并选择**下载**，即可下载该非对称密钥的公钥。



锁定或解锁密钥

> **提示**
>
> 在锁定或解锁密钥前，请确保该密钥没有被归档。密钥锁定后，配置产品 Key 时将无法选择该密钥。

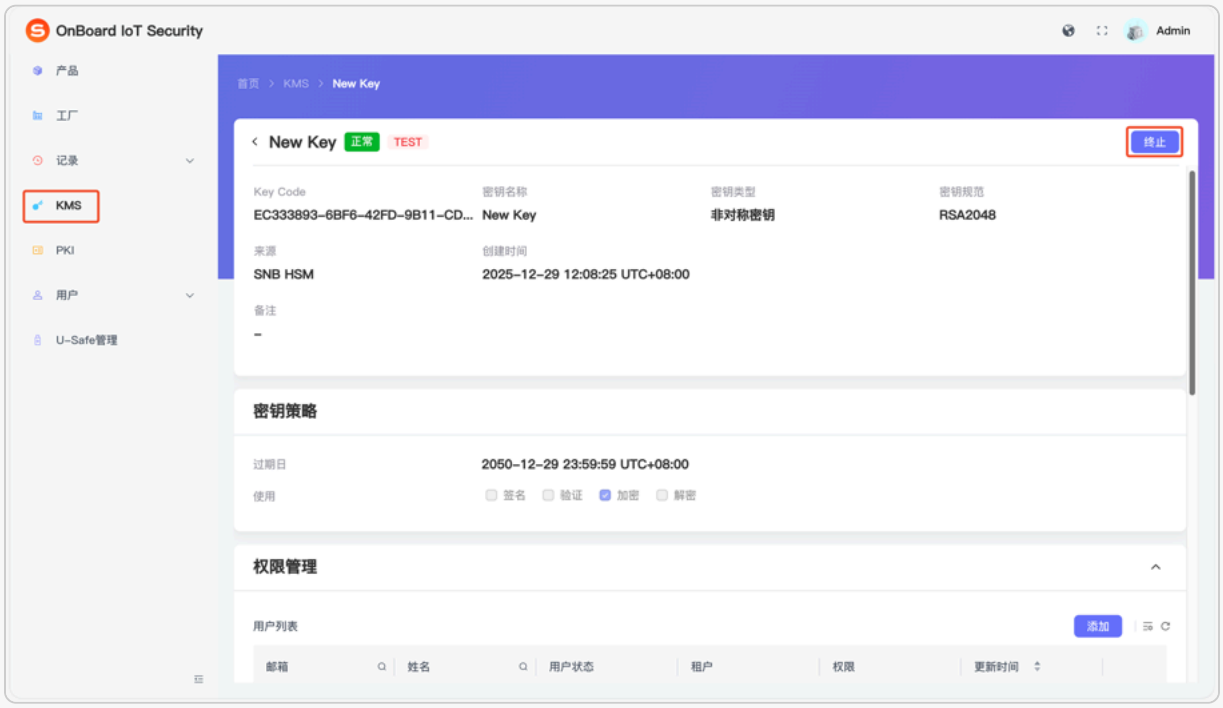
1. 登录 OBIS。
2. 点击左侧菜单中的 **KMS**，然后点击对应密钥的选项 ⋮ 图标并选择**锁定**或**解锁**，即可锁定或解锁该密钥。



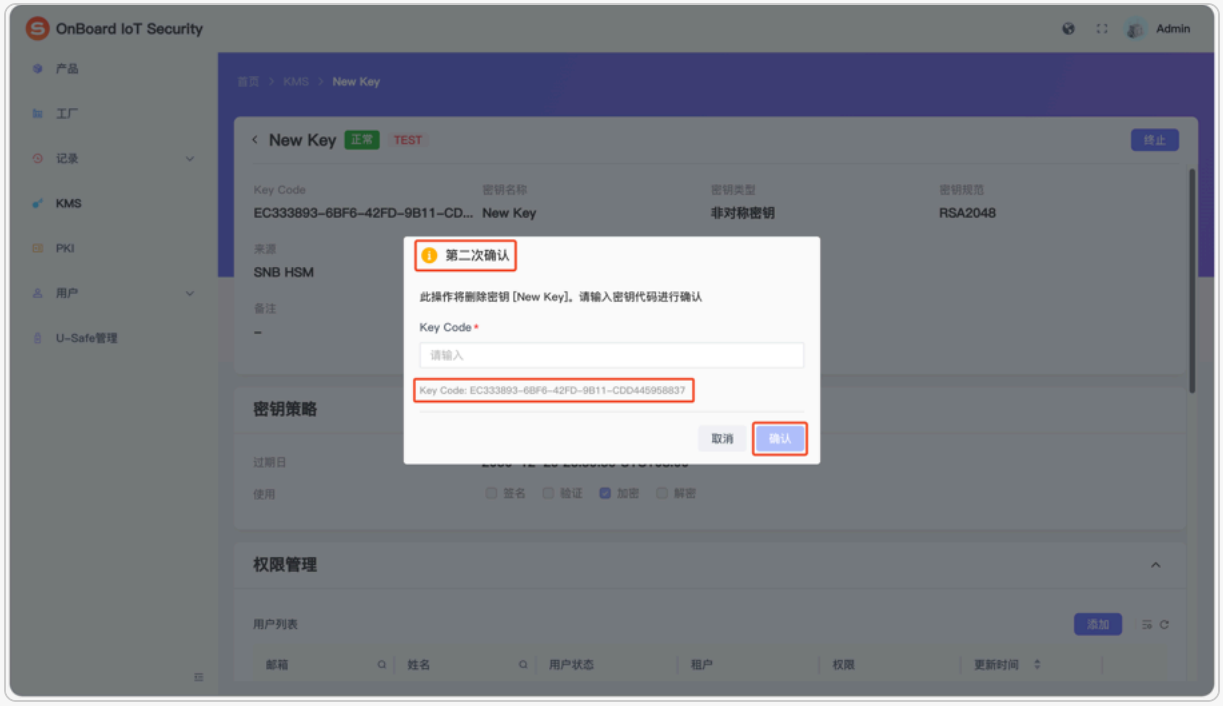
删除密钥

> **信息**
>
> 在删除密钥前，请确保该密钥没有被资源引用。如果已被资源引用，系统将提示"无法删除密钥，密钥已被引用"并列出引用该密钥的资源列表。

1. 登录 OBIS。
2. 点击左侧菜单中的 **KMS**，然后点击对应密钥的 Key Code 进入密钥页面。
3. 在密钥页面，点击**终止**。



4. 在**第二次确认**窗口中，输入窗口下方提示的 Key Code，点击**确认**即可删除该密钥。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > KMS > New Key
New Key 正常 TEST
终止
Key Code
EC333893-6BF6-42FD-9B11-CD...
密钥名称
New Key
密钥类型
非对称密钥
密钥规范
RSA2048
来源
SNB HSM
备注
–
第二次确认
此操作将删除密钥 [New Key]，请输入密钥代码进行确认
Key Code *
请输入
Key Code: EC333893-6BF6-42FD-9B11-CDD445958837
取消
确认
密钥策略
过期日
使用
签名
验证
加密
解密
权限管理
用户列表
添加
邮箱
姓名
用户状态
租户
权限
更新时间

版本：v3.2.2

OnBoard™ IoT Security 中 PKI 的概述

OBIS（OnBoard™ IoT Security）中的 PKI（Public Key Infrastructure）是负责证书安全的核心模块，提供证书的生成，签发和吊销服务，为物联网设备保障通信安全。主要应用于：

- 批量为设备预置证书模板。
- 使用命令行工具基于模板签发终端实体模板。
- 使用命令行工具验证证书及证书链有效性。

此外，OBIS 还提供 Matter 协议的证书模板，可快速生成对应的证书链。下表总结了 PKI 的核心能力：

核心能力	说明
CA 层级管理	• 根 CA 与中级 CA 层级管理 • 证书链验证与信任路径校验
证书生命周期管理	• 证书签发、续期与吊销 • 证书模板与策略管理 • 证书有效性与状态监控
外部 PKI 集成（根 CA 委托）	• OEM 在企业 PKI 中保留根 CA 所有权 • PKI 在委托授权下签发 CA

版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建证书

本文将指导你在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

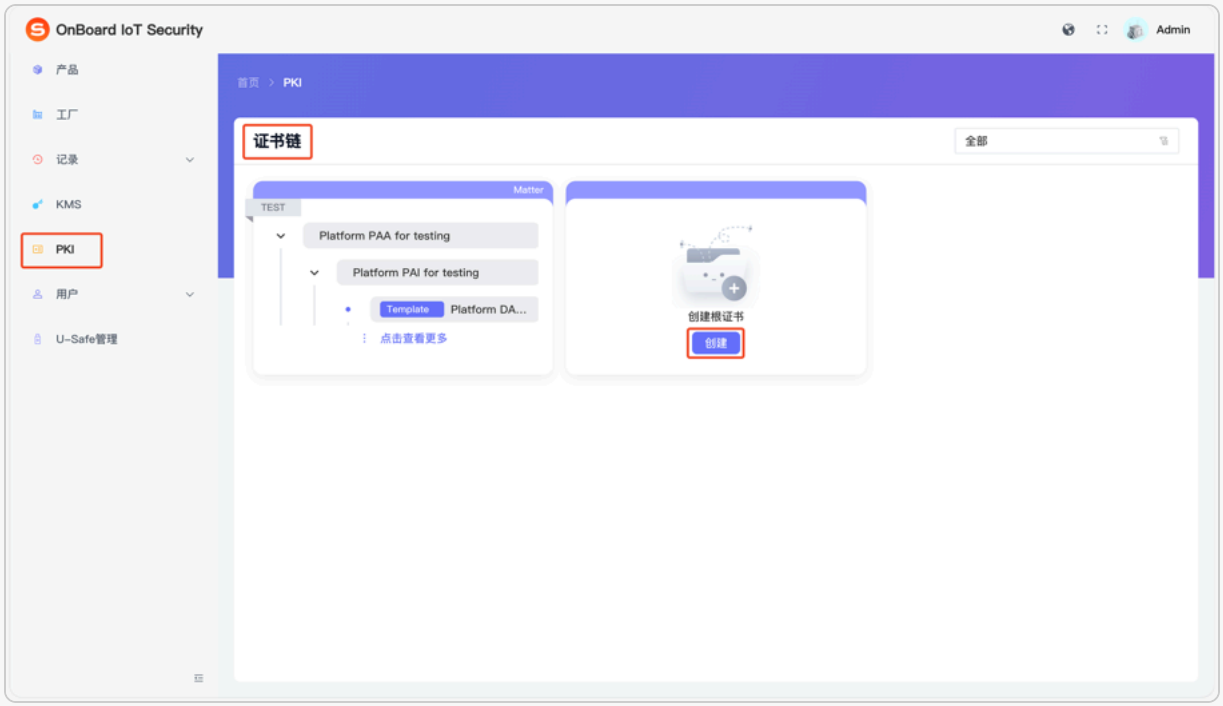
- 创建证书链
- 创建自定义证书
- 创建基于 Matter 模板的证书
- 创建基于 Google Cast 模板的证书

前提条件

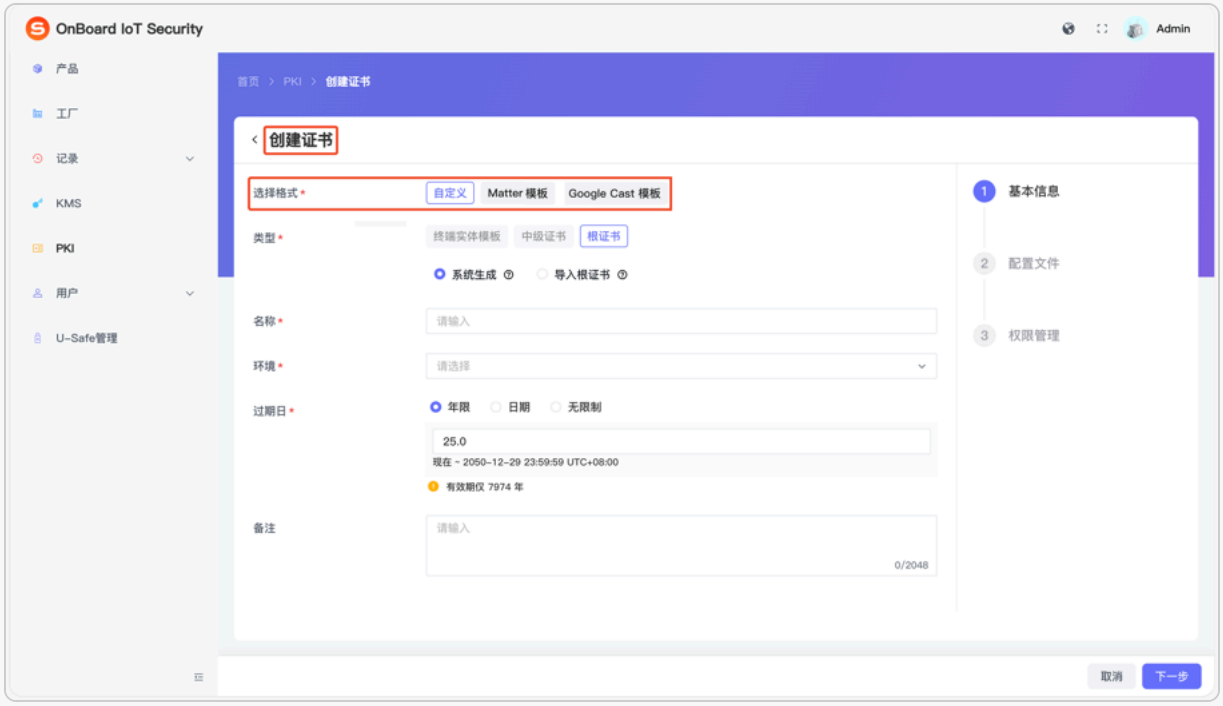
在开始之前，请确保你已获得 System Manager 或 Resource Manager 的 OBIS 角色权限。

创建证书链

1. 登录 OBIS。
2. 点击左侧菜单中的 **PKI**，然后在**证书链**页面点击**创建**。



3. 在**创建证书**页面，你可以根据实际需求创建**自定义**，**Matter 模板**或 **Google Cast 模板**格式的证书。



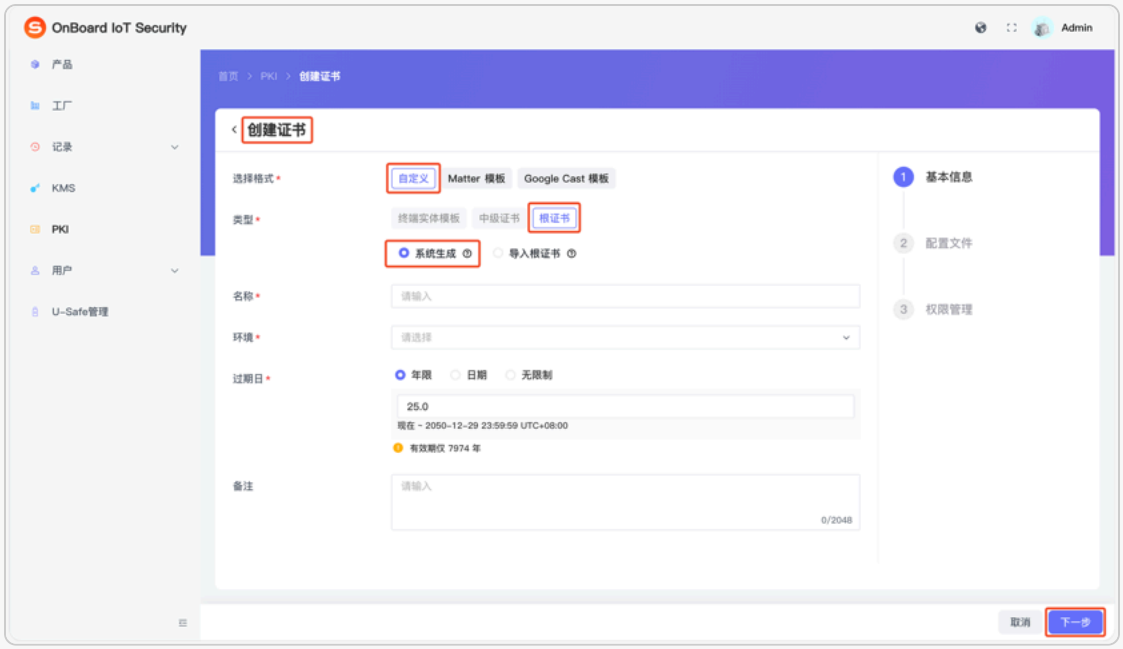
创建自定义证书

> **提示**
>
> 创建自定义证书或证书模板后，可在配置产品 Certificate 时进行选择。

创建根证书 | 创建中级证书 | 创建终端实体模板

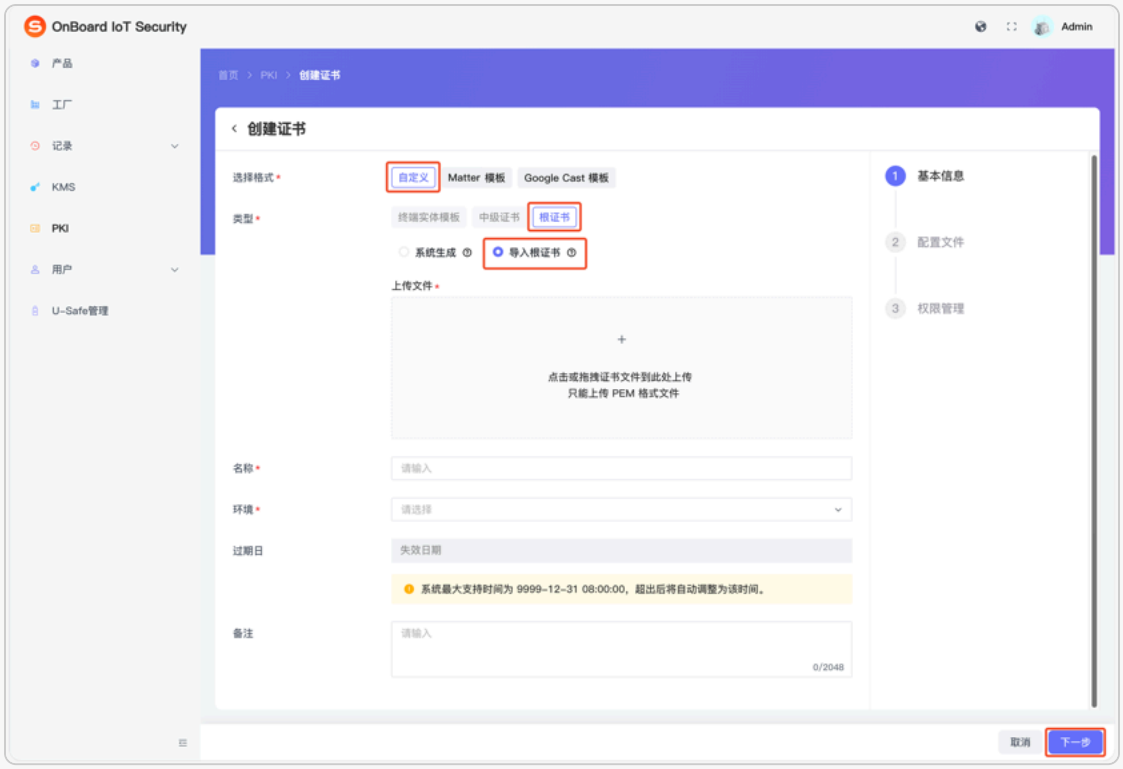
1. 在**创建证书**页面，格式选择**自定义**，类型选择**根证书**。
2. 根据你的需求选择**系统生成**或**导入根证书**。
    - 如果不使用自有根证书，请选择**系统生成**并输入与选择基础信息。

> **提示**
>
> 系统会在硬件安全模块（HSM）内安全地生成一对随机密钥对，并完成根证书的自签发。证书公钥可导出。



- 如果你使用自有根证书，请选择**导入根证书**并输入与选择基础信息。

> **提示**
>
> 手动上传在系统外部生成的根证书（仅包含公钥），用于签名验证。



根证书基础信息说明：

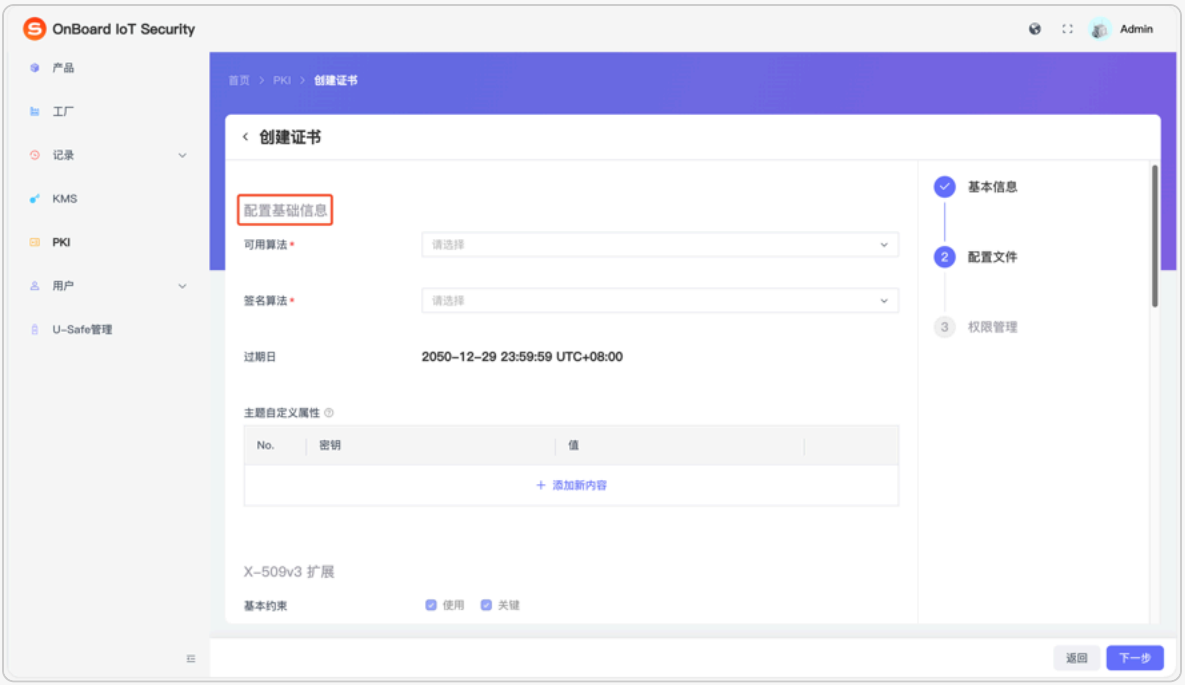
信息	系统生成	导入根证书
上传文件	不需要上传文件。	在**上传文件**区域中点击添加 ➕ 图标，选择并上传格式为 **.pem** 的根证书，或者直接拖曳自有根证书到上传区域。
名称	同一环境中，根证书名称必须唯一。	

信息	系统生成	导入根证书
环境	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。	
过期日	设定根证书的过期日。	根据证书内容自动解析，且无法修改。
备注	用于补充说明根证书相关的额外信息，如特殊要求和注意事项等。	

3. 确认信息无误后，点击**下一步**并参考以下信息进行配置。

> **提示**
>
> 如果使用自有的根证书，相关配置会根据证书内容自动解析生成。

配置基础信息说明

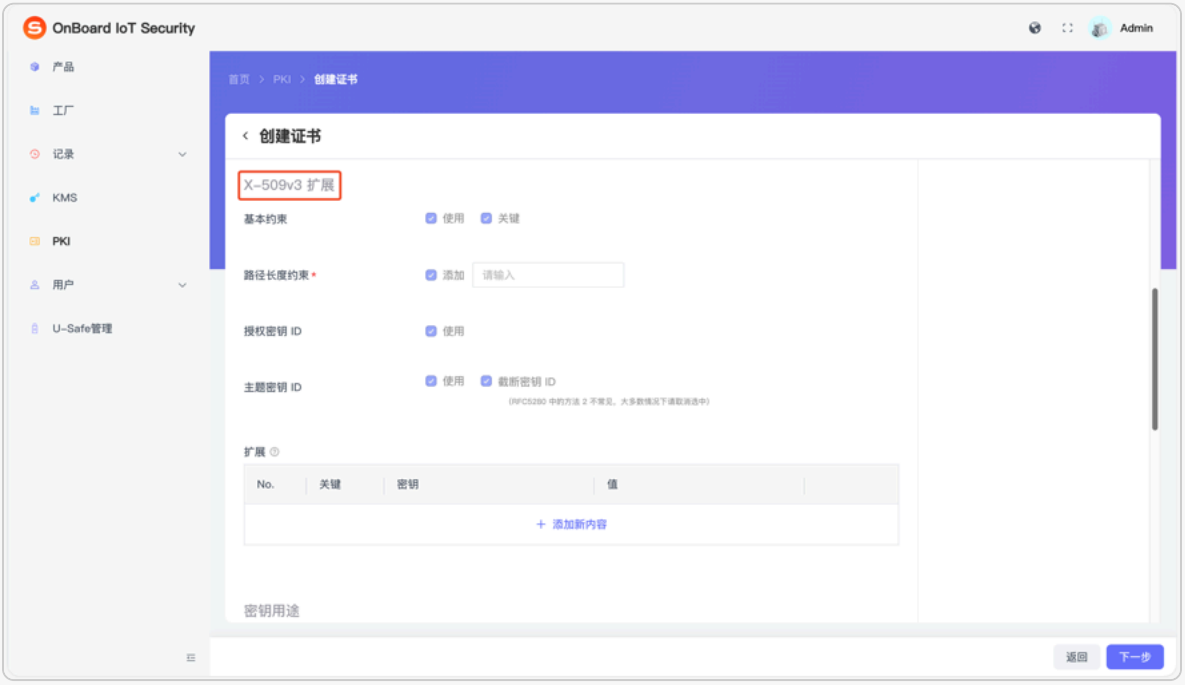


信息	是否必填	说明
可用算法	是	证书相关的可用算法。可选类型包括： ◦ **RSA_2048** ◦ **RSA_3072** ◦ **RSA_4096** ◦ **EC_secp384r1** ◦ **EC_prime256v1**
签名算法	是	证书相关的签名算法。可选类型包括： ◦ **SHA256WITHRSA** ◦ **SHA384WITHRSA** ◦ **SHA512WITHRSA**

信息	是否必填	说明
过期日	否	继承上一步在基础信息设置的过期日，且无法修改。
主题自定义属性	否	为证书配置主题自定义属性，可添加特定的密钥和值。

X-509v3 扩展信息说明

> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。

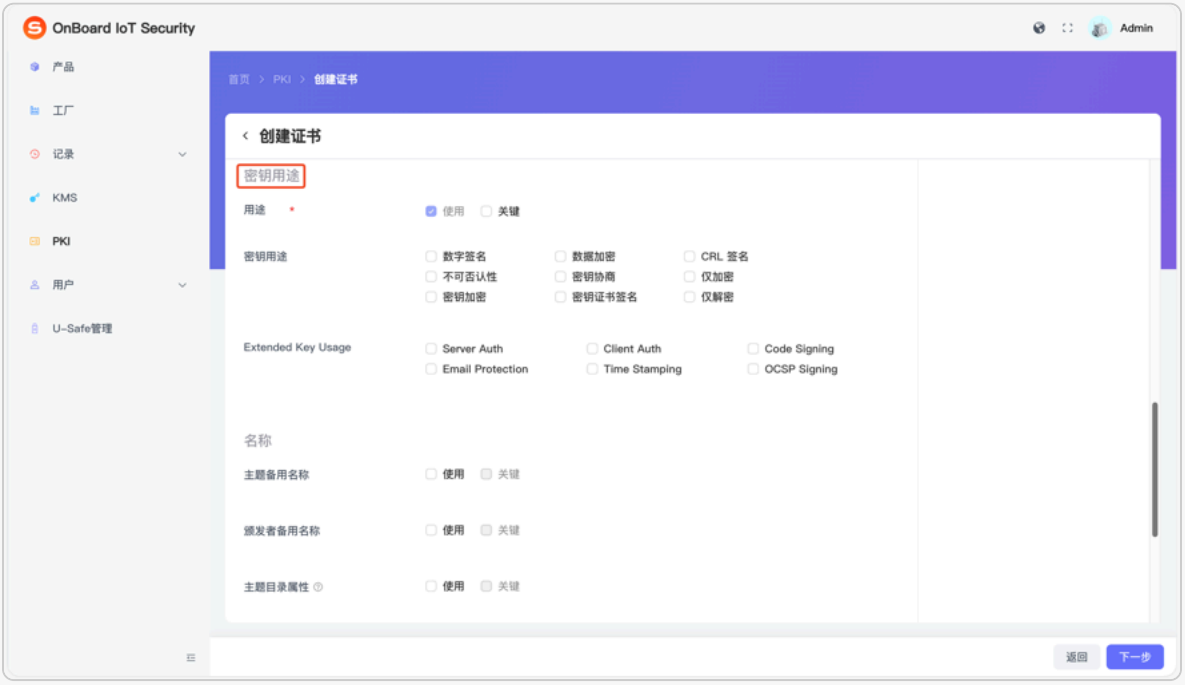


信息	是否必填	说明
基本约束	否	根据证书内容自动解析，且无法修改。
路径长度约束	是	限定子级证书的数量。
授权密钥 ID	否	根据证书内容自动解析，且无法修改。

信息	是否必填	说明
主题密钥 ID	否	根据证书内容自动解析，且无法修改。
扩展	否	为证书配置 X-509v3 的扩展属性，可添加特定的密钥和值。

密钥用途信息说明

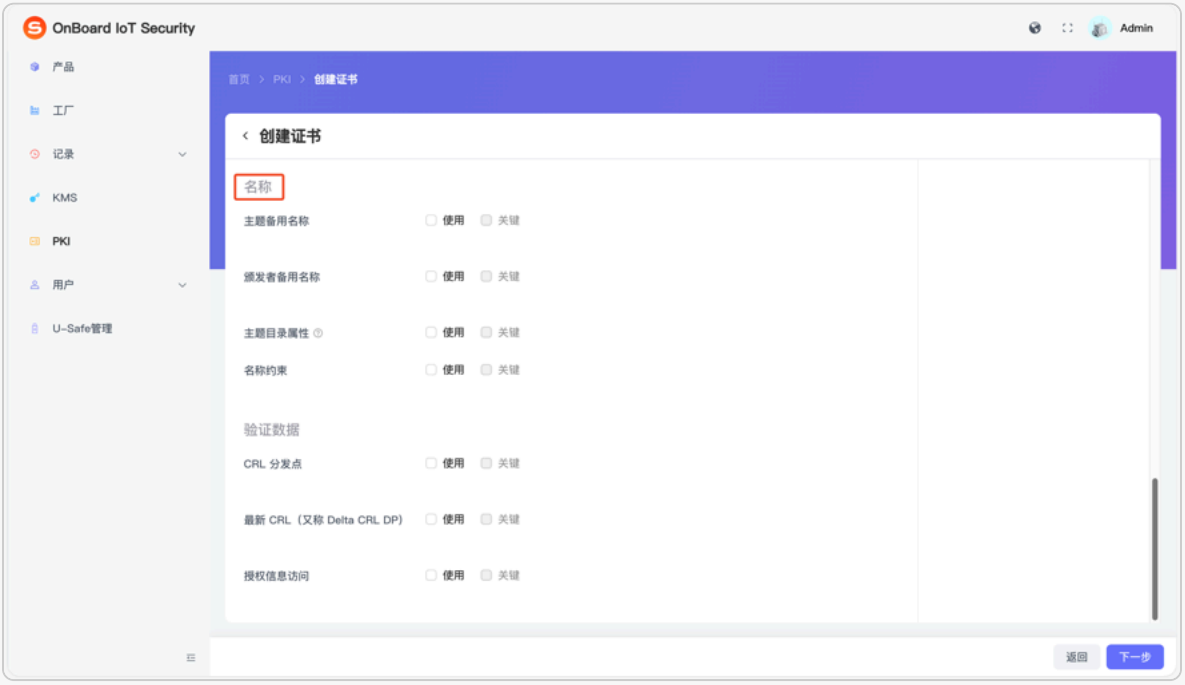
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



信息	是否必填	说明
用途	是	默认已勾选**使用**，且无法修改。
密钥用途	是	可选**数字签名**，**数据加密**和 **CRL 签名**等密钥用途。
Extended Key Usage	否	可选 **Server Auth**，**Client Auth** 和 **Code Signing** 等扩展密钥用途。

名称信息说明

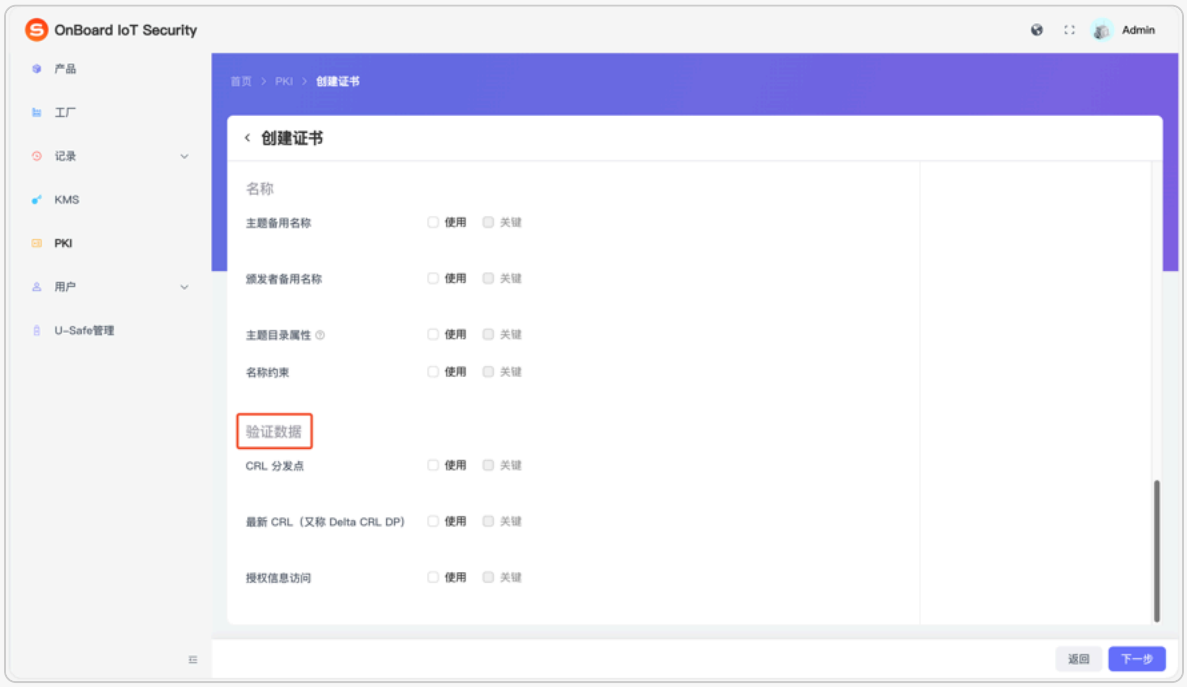
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



信息	是否必填	说明
主题备用名称	否	勾选**使用**后，需添加主题备用名称。
颁发者备用名称	否	勾选**使用**后，需添加颁发者备用名称。
主题目录属性	否	勾选**使用**后，需添加特定的密钥和值。
名称约束	否	勾选**使用**后，需在黑名单中添加特定的值。

验证数据信息说明

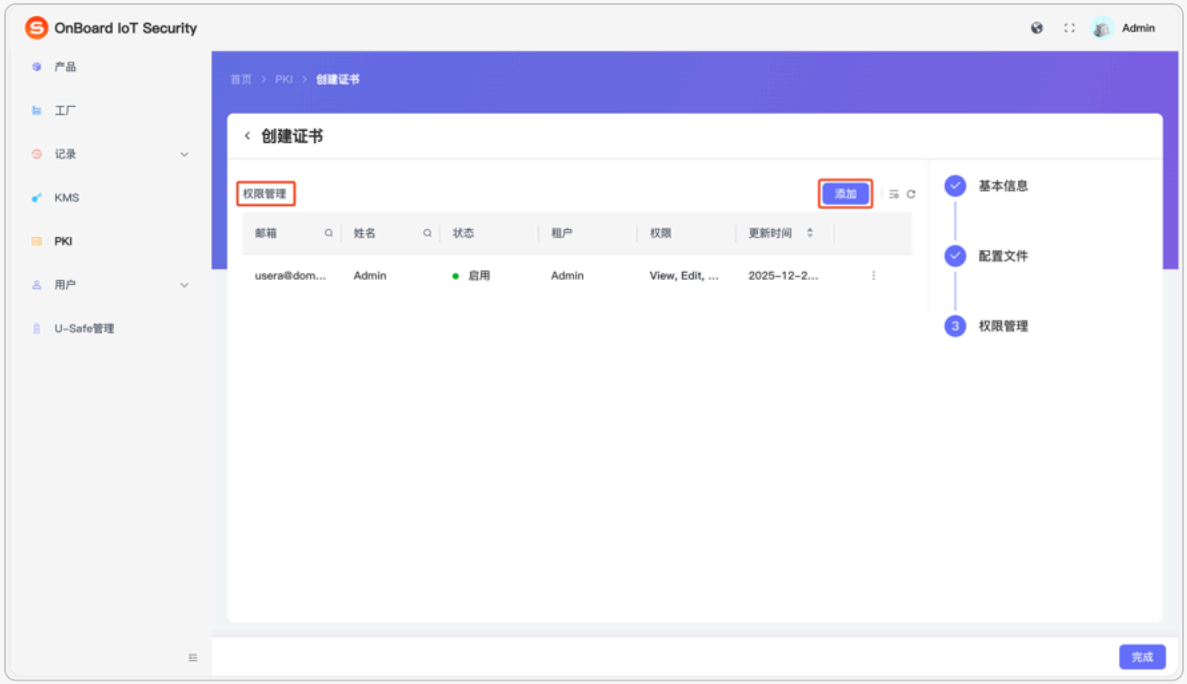
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



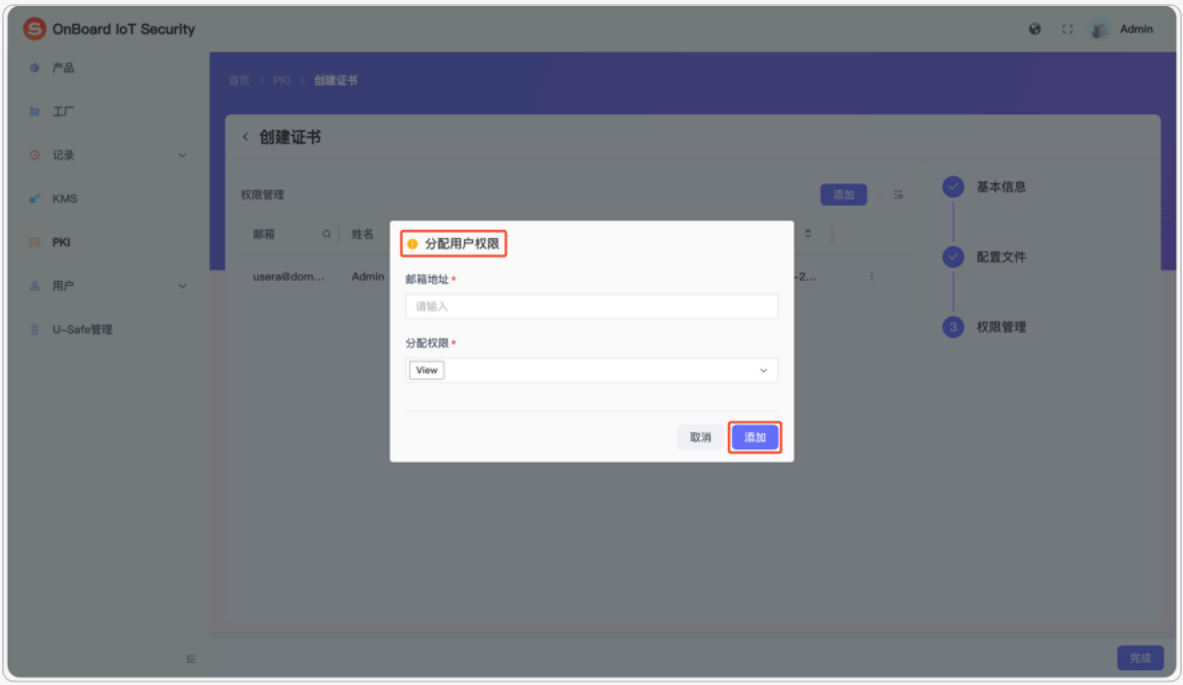
信息	是否必填	说明
CRL 分发点	否	勾选**使用**后，需添加 CRL 分发点 URI 和 CRL 颁发者。
最新 CRL（又称 Delta CRL DP）	否	勾选**使用**后，需添加最新 CRL URI。

信息	是否必填	说明
授权信息访问	否	勾选**使用**后，需添加 OCSP 服务定位器 URI 并在 CA 颁发者 URI 中添加特定的值。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Revoke，Grant 和 Delete）。然后，点击**添加**。

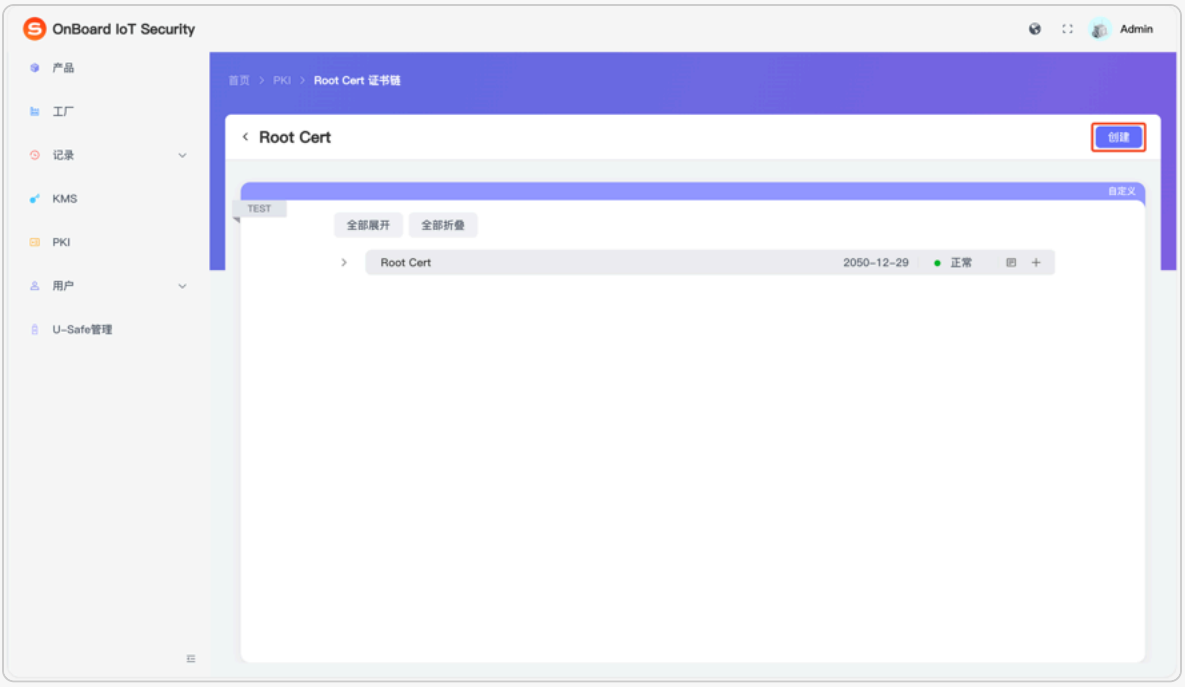


6. 点击**完成**，即可完成根证书的创建。

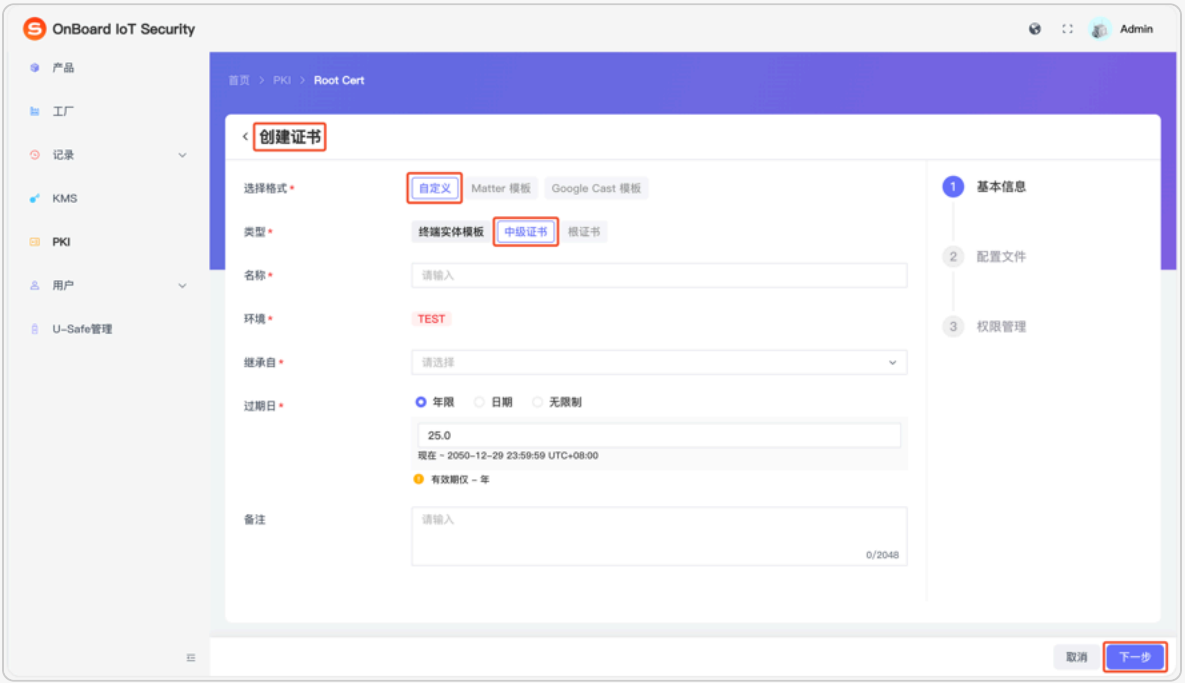
> **信息**
>
> 在创建中级证书之前，请确保已创建至少一个根证书，以便在配置时进行选择并继承。

1. 在**证书链**页面，点击对应的证书链，然后点击**创建**。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



2. 在**创建证书**页面，格式选择**自定义**，类型选择**中级证书**并选择与输入基础信息。



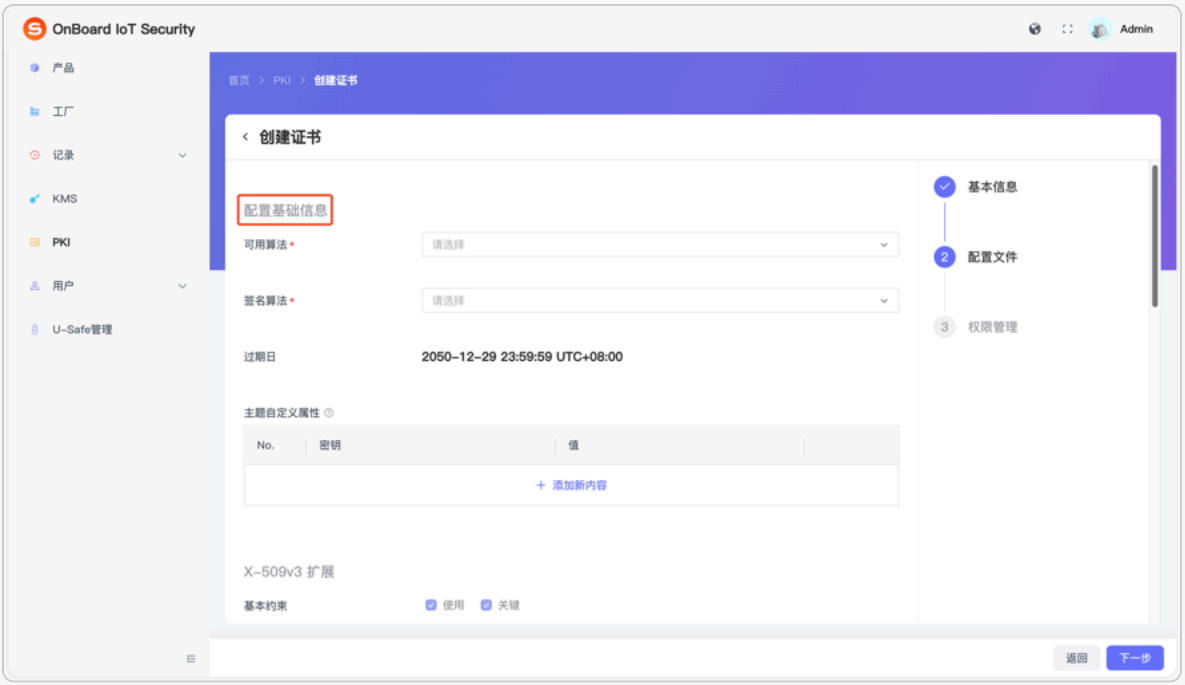
中级证书基础信息说明：

信息	说明
名称	同一证书链中，同一层级的证书名称必须唯一。
环境	继承自根证书的环境。
继承自	选择继承的根证书或中级证书。
过期日	设定中级证书的过期日。不可超过继承的证书的过期日。 **注释**：如果继承自自有根证书，过期日会继承自根证书。
备注	用于补充说明中级证书相关的额外信息，如特殊要求和注意事项等。

3. 确认信息无误后，点击**下一步**并参考以下信息进行配置。

> **提示**
>
> 如果中级证书继承自自有根证书，页面只显示**配置基础信息**区域。

配置基础信息说明

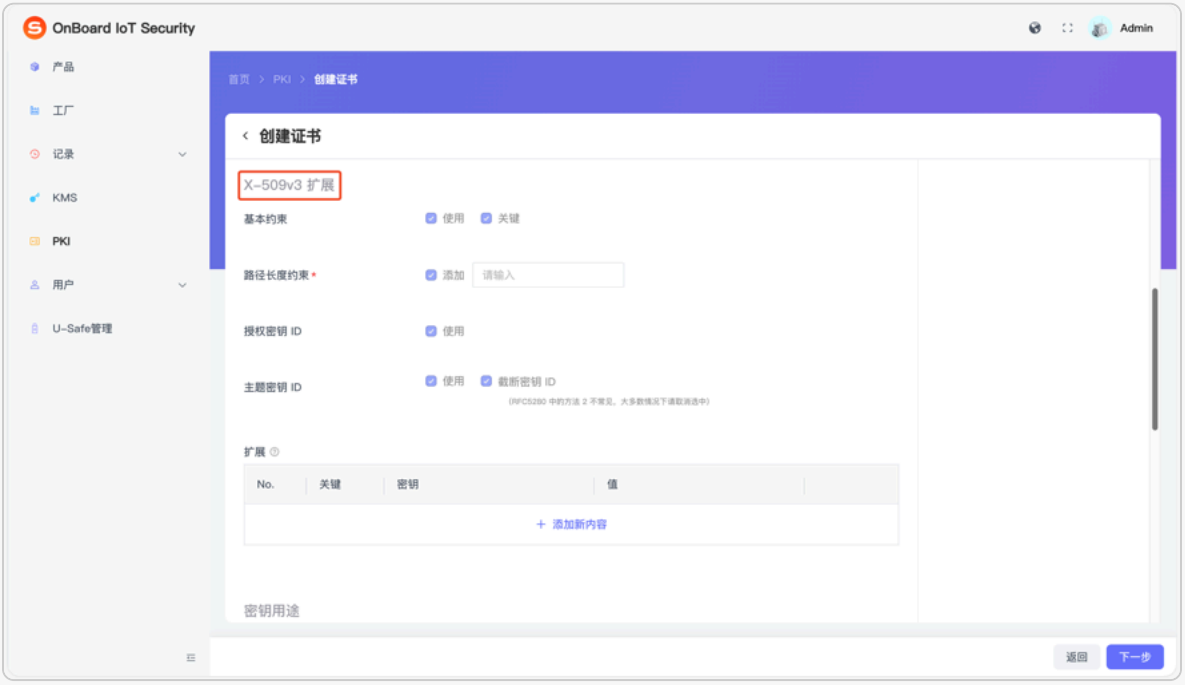


信息	是否必填	说明
可用算法	是	证书相关的可用算法。可选类型包括： ◦ **RSA_2048** ◦ **RSA_3072** ◦ **RSA_4096** ◦ **EC_secp384r1** ◦ **EC_prime256v1**

信息	是否必填	说明
签名算法	是	证书相关的签名算法。可选类型包括： ◦ **SHA256WITHRSA** ◦ **SHA384WITHRSA** ◦ **SHA512WITHRSA**
过期日	否	继承上一步在基础信息设置的过期日，且无法修改。
主题自定义属性	否	为证书配置主题自定义属性，可添加特定的密钥和值。

X-509v3 扩展信息说明

> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。

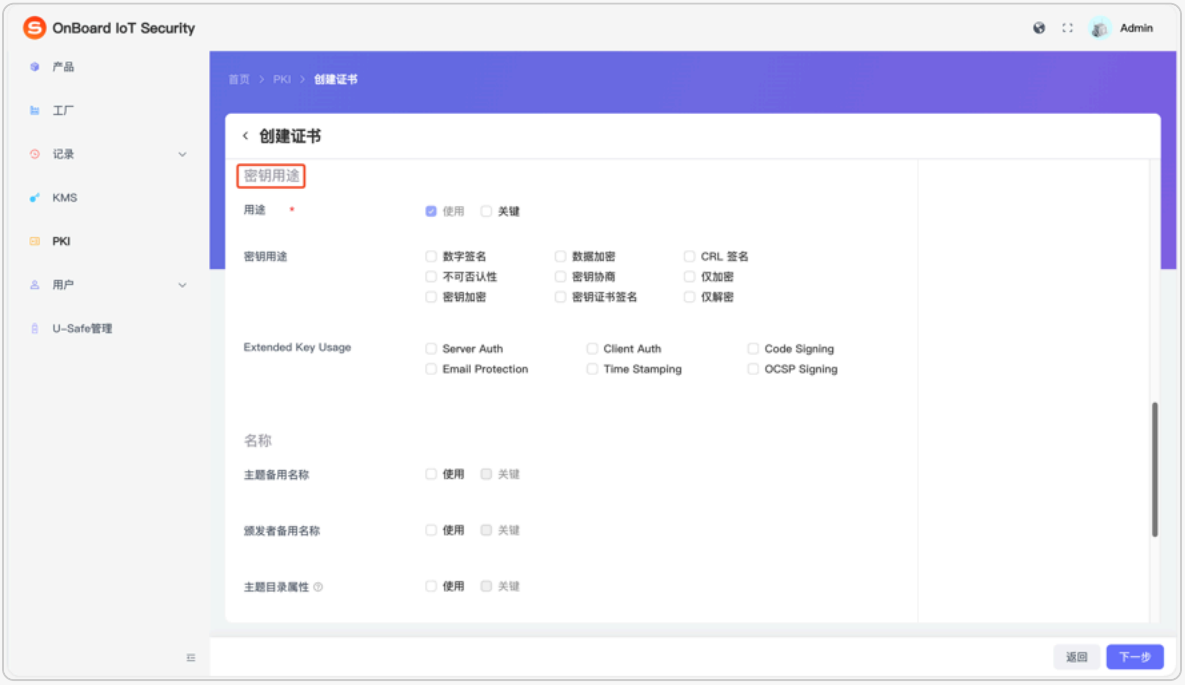


信息	是否必填	说明
基本约束	否	根据证书内容自动解析，且无法修改。
路径长度约束	是	限定子级证书的数量。

信息	是否必填	说明
授权密钥 ID	否	根据证书内容自动解析，且无法修改。
主题密钥 ID	否	根据证书内容自动解析，且无法修改。
扩展	否	为证书配置 X-509v3 的扩展属性，可添加特定的密钥和值。

密钥用途信息说明

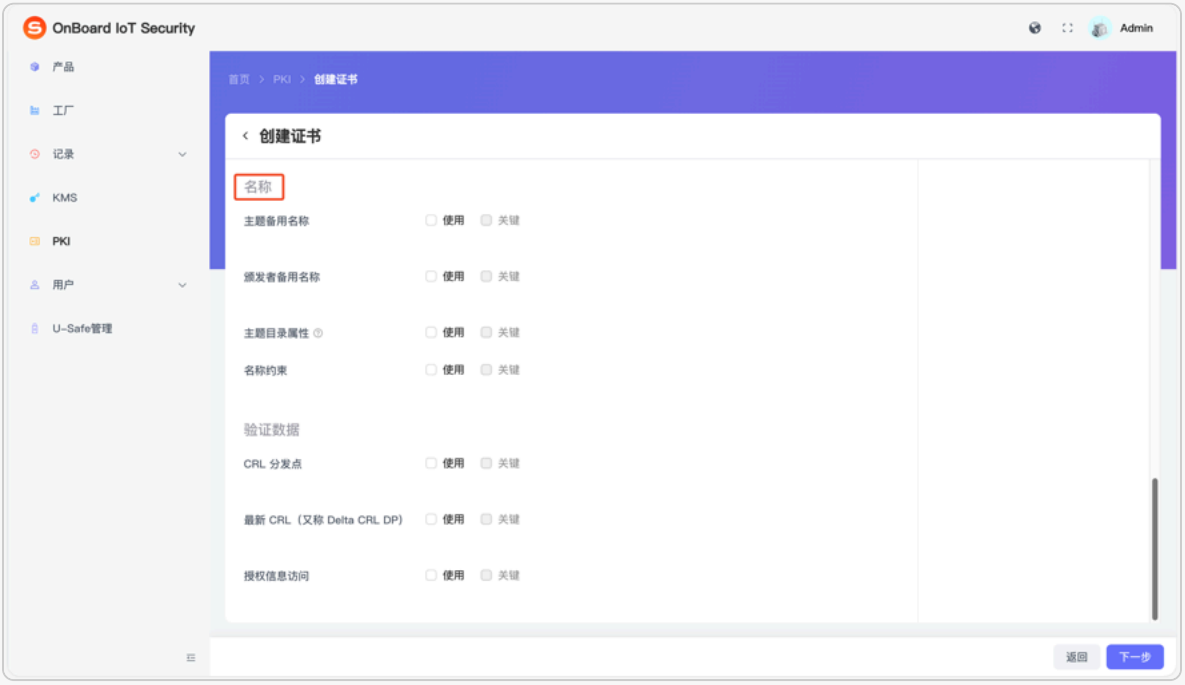
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



信息	是否必填	说明
用途	是	默认已勾选**使用**，且无法修改。
密钥用途	是	可选**数字签名**，**数据加密**和 **CRL 签名**等密钥用途。
Extended Key Usage	否	可选 **Server Auth**，**Client Auth** 和 **Code Signing** 等扩展密钥用途。

名称信息说明

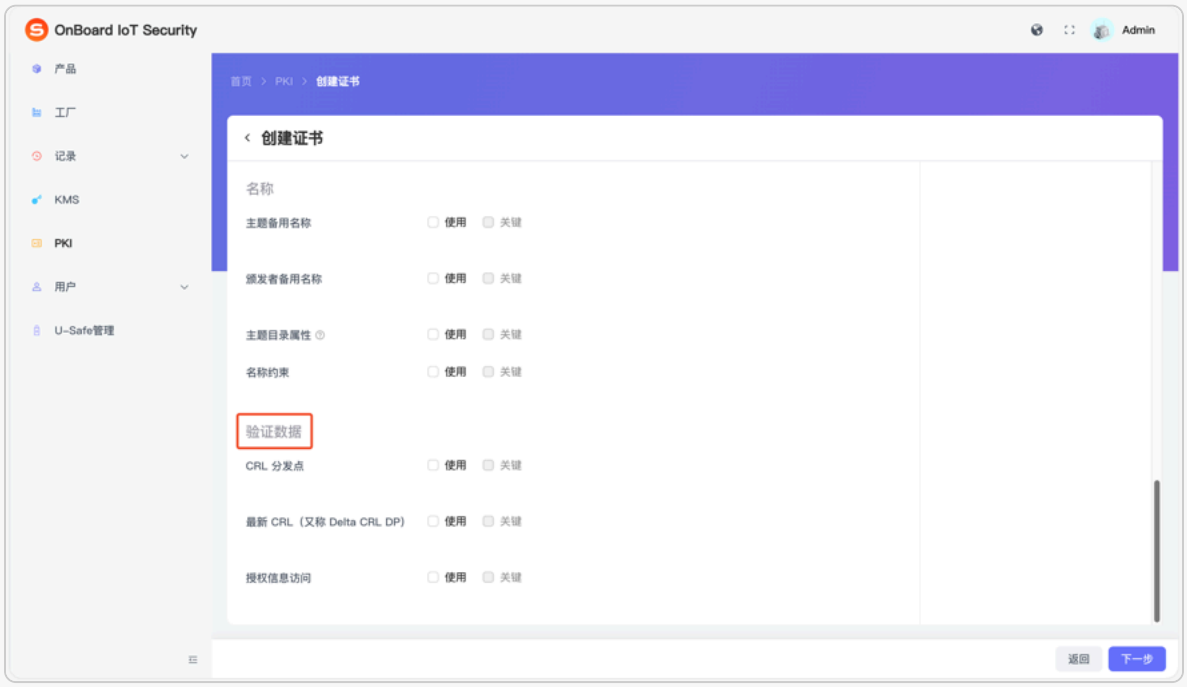
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



信息	是否必填	说明
主题备用名称	否	勾选**使用**后，需添加主题备用名称。
颁发者备用名称	否	勾选**使用**后，需添加颁发者备用名称。
主题目录属性	否	勾选**使用**后，需添加特定的密钥和值。
名称约束	否	勾选**使用**后，需在黑名单中添加特定的值。

验证数据信息说明

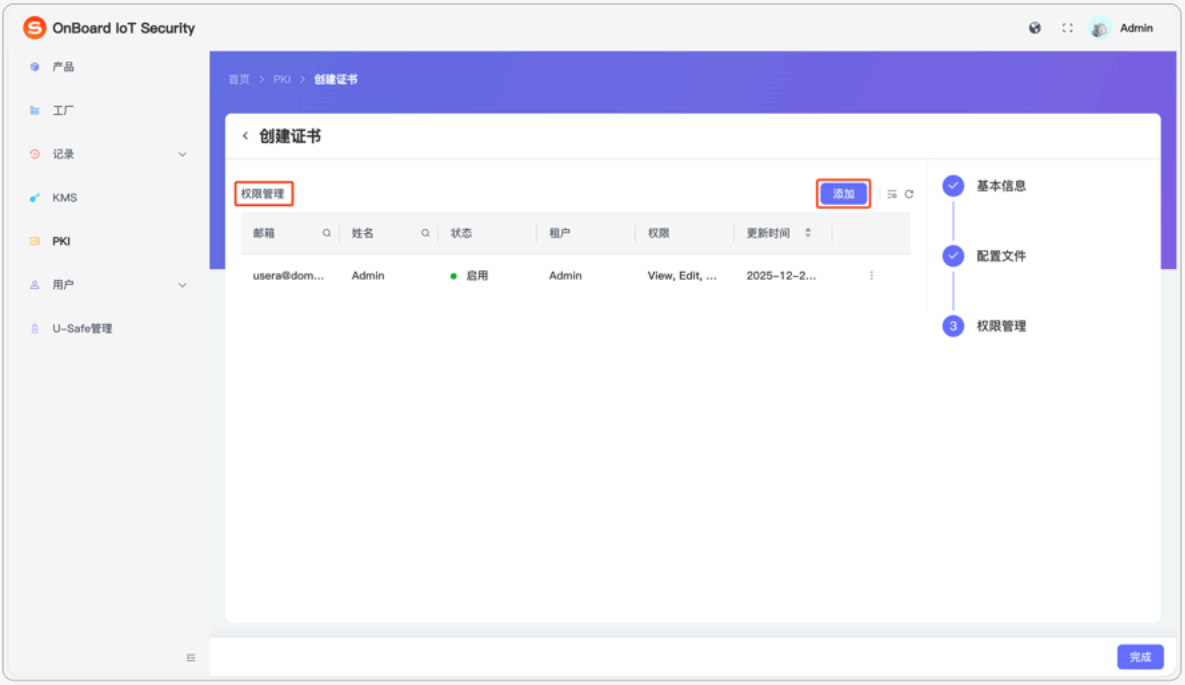
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



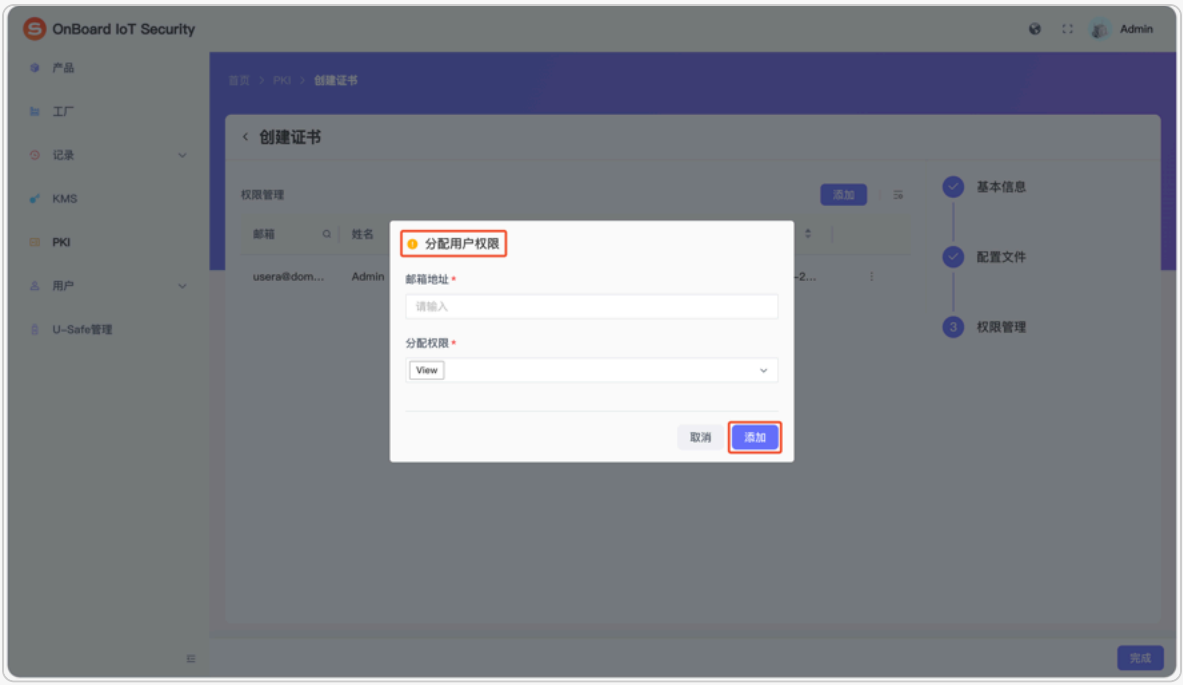
信息	是否必填	说明
CRL 分发点	否	勾选**使用**后，需添加 CRL 分发点 URI 和 CRL 颁发者。

信息	是否必填	说明
最新 CRL（又称 Delta CRL DP）	否	勾选**使用**后，需添加最新 CRL URI。
授权信息访问	否	勾选**使用**后，需添加 OCSP 服务定位器 URI 并在 CA 颁发者 URI 中添加特定的值。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Revoke，Grant 和 Delete）。然后，点击**添加**。



6. 点击**完成**，即可完成中级证书的创建。

> **提示**
>
> 创建中级证书时，如果其继承自自有根证书，你需要完成 CSR （证书签名请求）配置。在 CSR 配置完成之前，该证书将处于**待完成**状态，且无法正常使用。

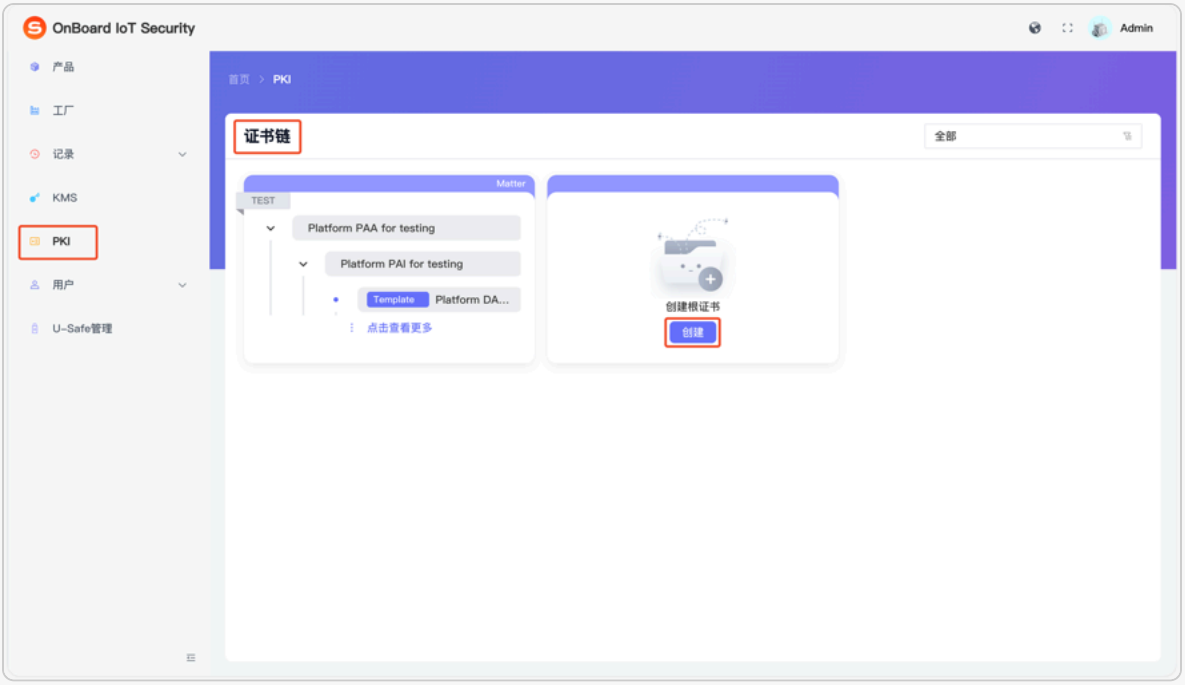
中级证书的 CSR 配置步骤

> **提示**
>
> 创建中级证书时，如果其继承自自有根证书，参考以下步骤完成 CSR 配置。

1. 在**证书链**页面，点击对应的证书链，然后点击对应的中级证书，进入证书页面。

2. 在证书页面，点击提示中的链接以完成 CSR 的配置。

3. 在 **CSR 配置**页面，点击下载 图标来下载 CSR 文件。然后，使用你的 Root CA（Certificate Authority）对该文件进行签署，从而生成由 Root CA 签名的中级证书。



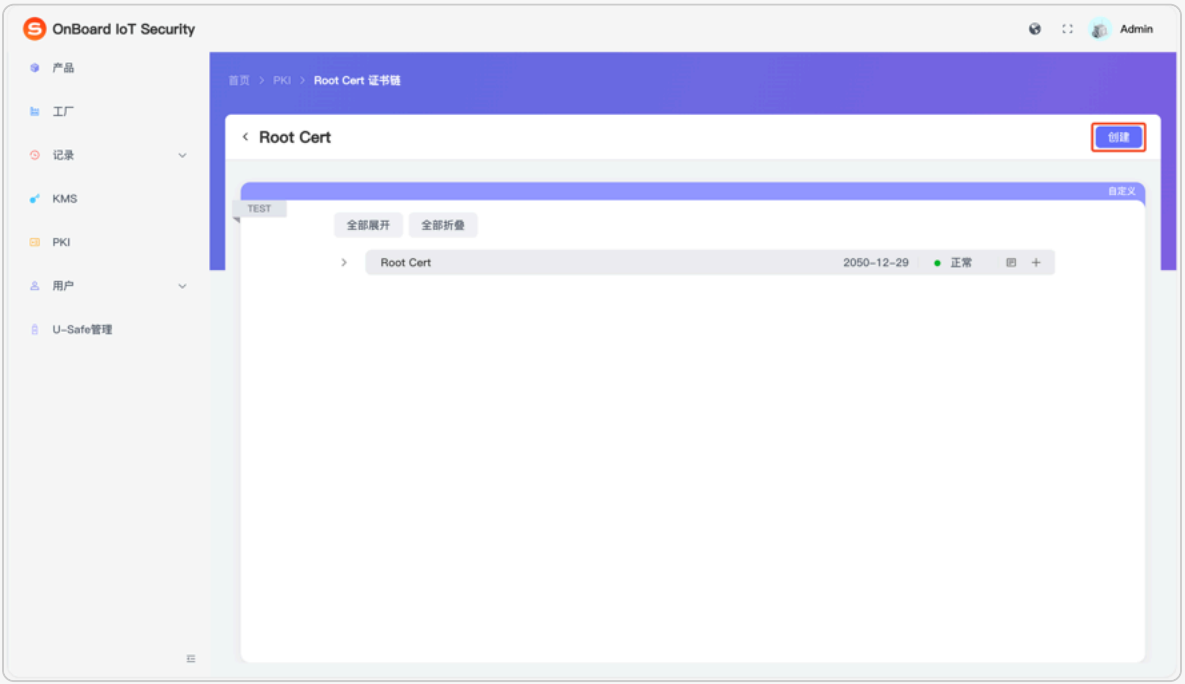
4. 签署完成后，在上传区域中点击添加 图标，选择并上传格式为 **.pem** 的中级证书，或者直接拖曳你的中级证书到上传区域。

5. 点击**提交**，配置完成后，中级证书状态会变更为**正常**。

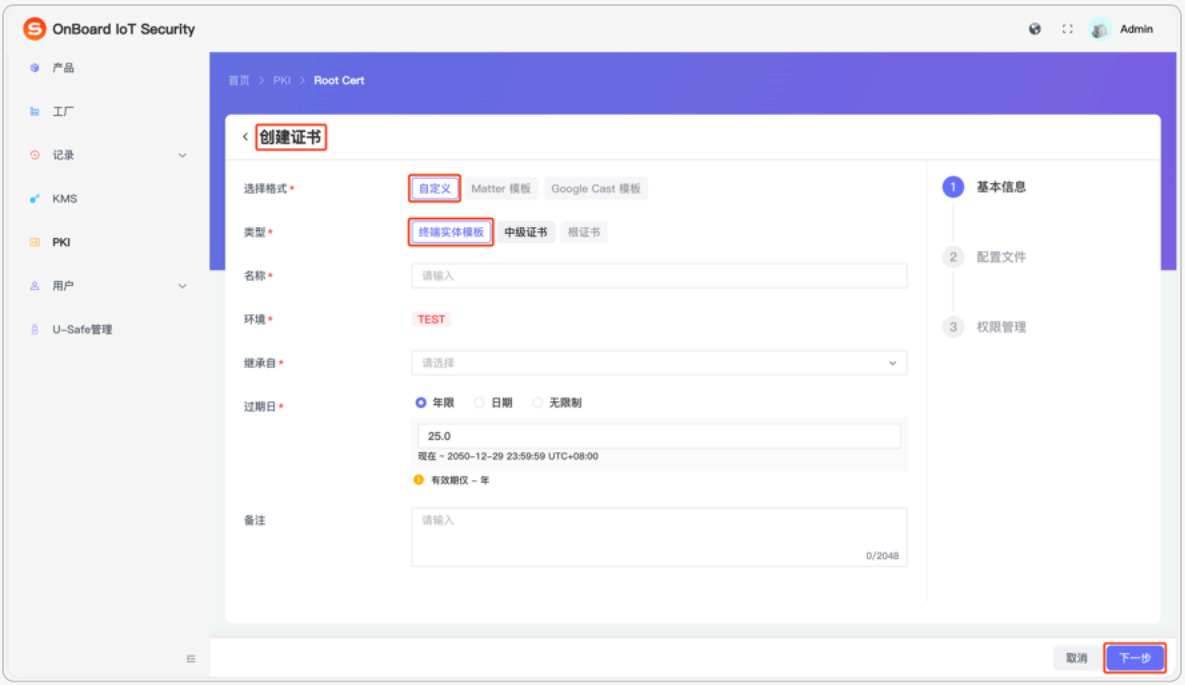
> **信息**
>
> 在创建终端实体模板之前，请确保已创建至少一个根证书或中级证书，以便在配置时进行选择并继承。

1. 在**证书链**页面，点击对应的证书链，然后点击**创建**。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



2. 在**创建证书**页面，格式选择**自定义**，类型选择**终端实体模板**并选择与输入基础信息。



终端实体模板基础信息说明：

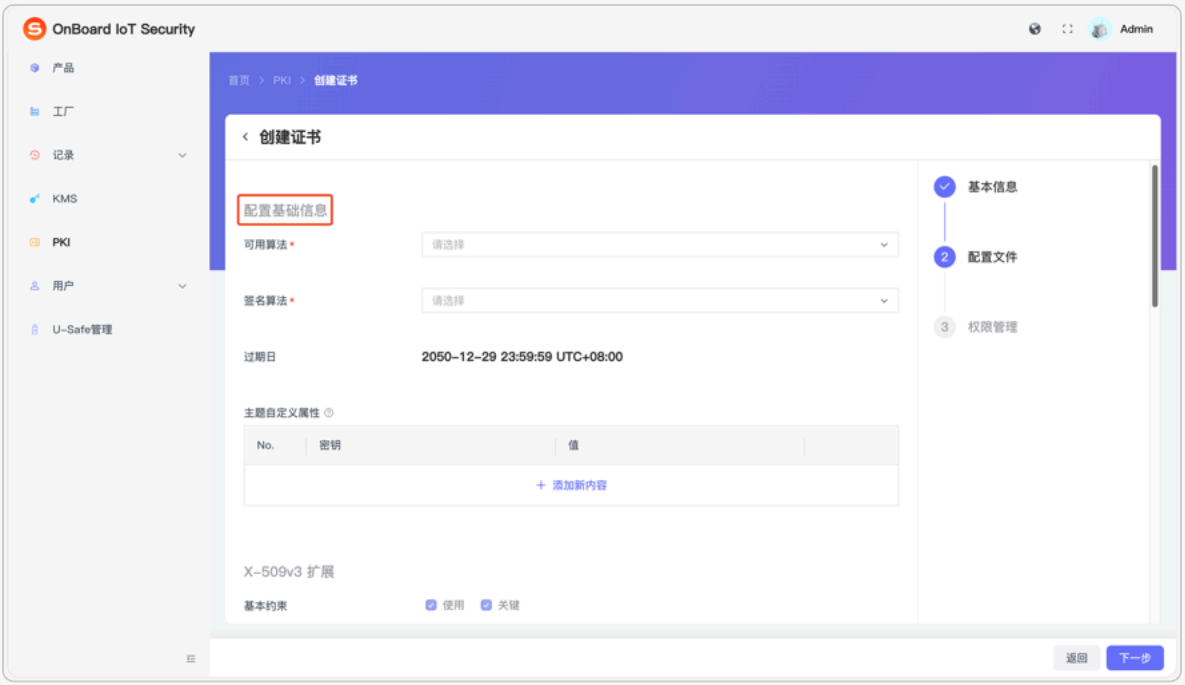
信息	说明
名称	同一证书链中，同一层级的证书名称必须唯一。
环境	继承自根证书的环境。
继承自	选择继承的根证书或中级证书。
过期日	设定终端实体模板的过期日。不可超过继承的证书的过期日。 **注释**：如果继承自自有根证书，过期日会继承自根证书。

信息	说明
备注	用于补充说明终端实体模板相关的额外信息，如特殊要求和注意事项等。

3. 确认信息无误后，点击**下一步**并参考以下信息进行配置。

> **提示**
>
> 如果终端实体模板继承自自有根证书，页面只显示**配置基础信息**区域。配置基础信息后，可以直接继续下一步。

配置基础信息说明

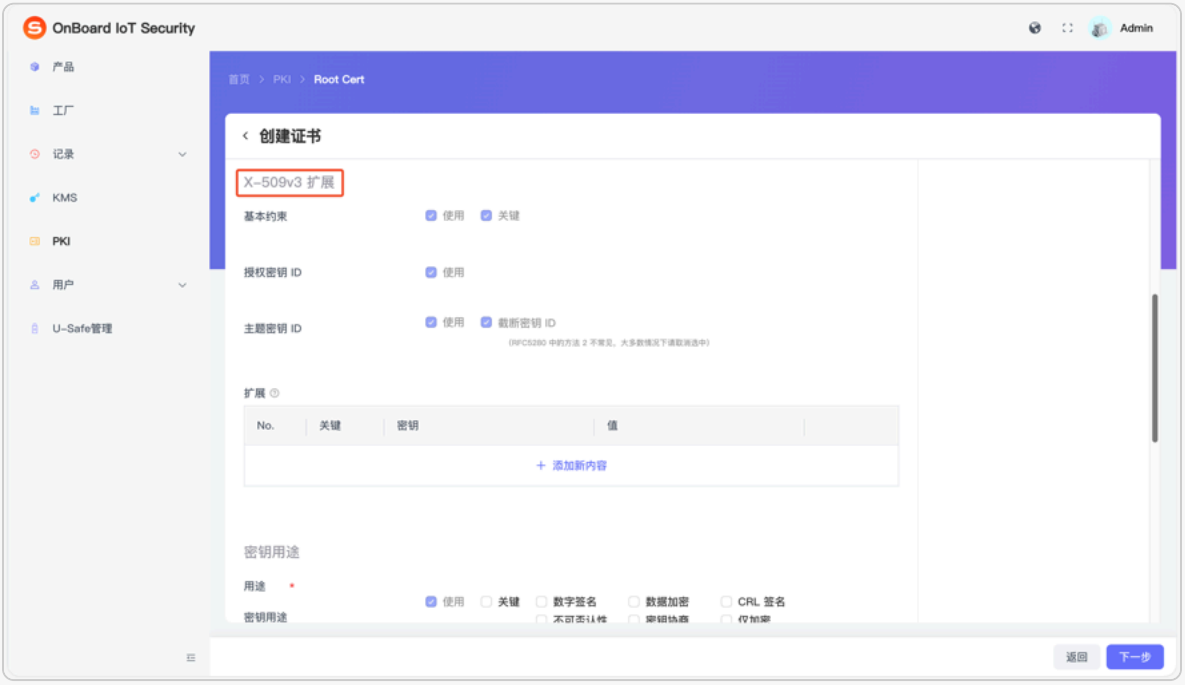


信息	是否必填	说明
可用算法	是	证书相关的可用算法。可选类型包括： ◦ **RSA_2048** ◦ **RSA_3072** ◦ **RSA_4096** ◦ **EC_secp384r1** ◦ **EC_prime256v1**

信息	是否必填	说明
签名算法	是	证书相关的签名算法。可选类型包括： ◦ **SHA256WITHRSA** ◦ **SHA384WITHRSA** ◦ **SHA512WITHRSA**
过期日	否	继承上一步在基础信息设置的过期日，且无法修改。
主题自定义属性	否	为证书配置主题自定义属性，可添加特定的密钥和值。

X-509v3 扩展信息说明

> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。

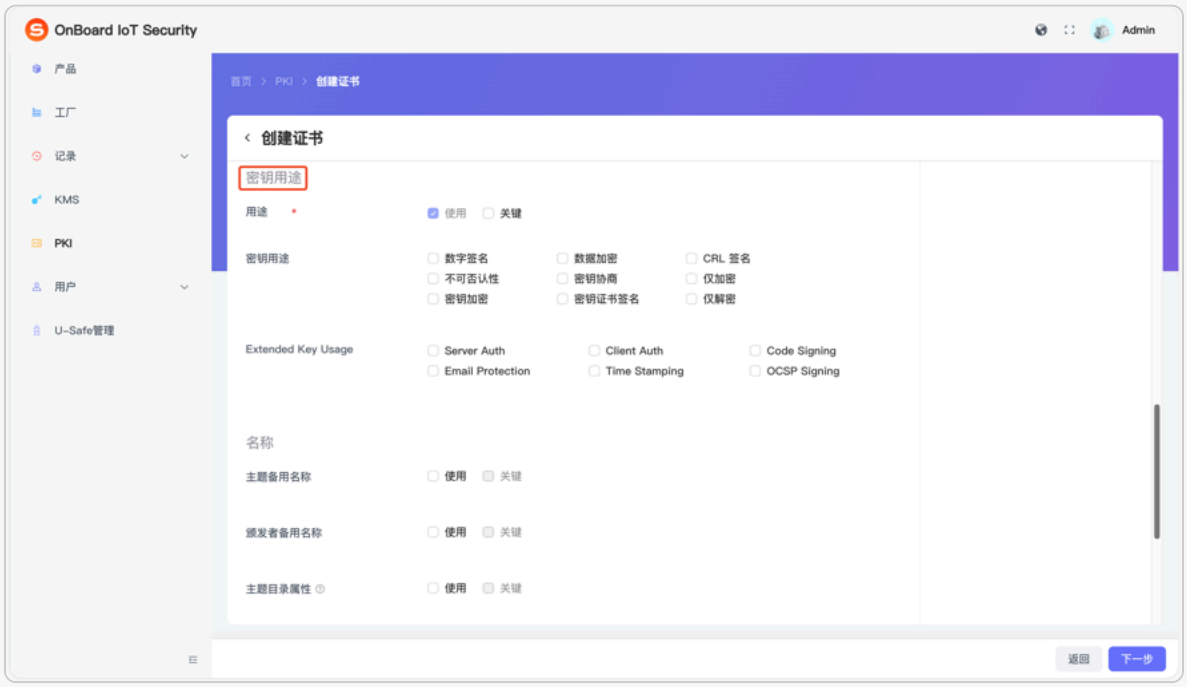


信息	是否必填	说明
基本约束	否	根据证书内容自动解析，且无法修改。
授权密钥 ID	否	根据证书内容自动解析，且无法修改。

信息	是否必填	说明
主题密钥 ID	否	根据证书内容自动解析，且无法修改。
扩展	否	为证书配置 X-509v3 的扩展属性，可添加特定的密钥和值。

密钥用途信息说明

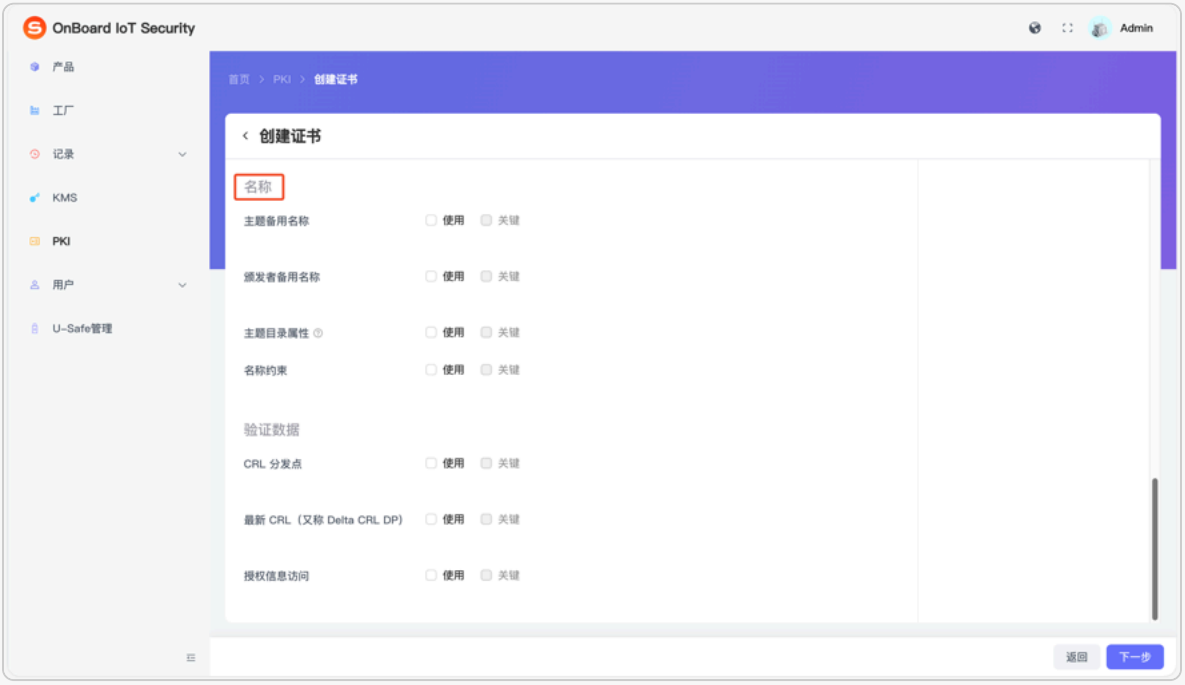
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



信息	是否必填	说明
用途	是	默认已勾选**使用**，且无法修改。
密钥用途	是	可选**数字签名**，**数据加密**和 **CRL 签名**等密钥用途。
Extended Key Usage	否	可选 **Server Auth**，**Client Auth** 和 **Code Signing** 等扩展密钥用途。

名称信息说明

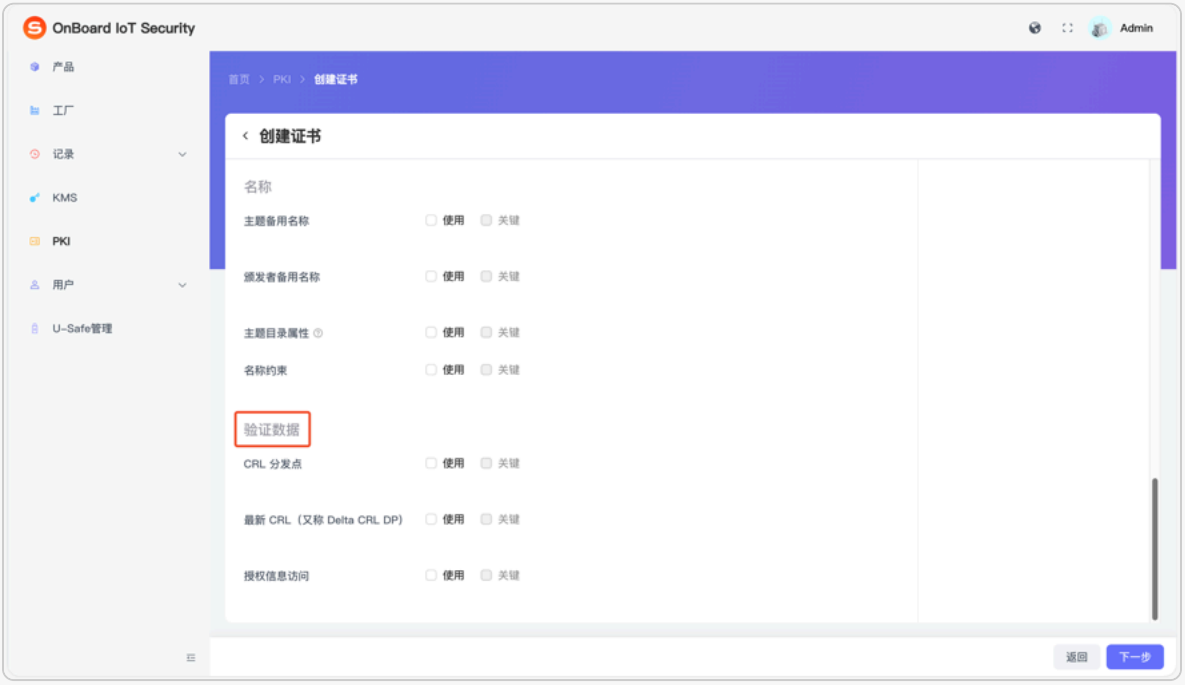
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



信息	是否必填	说明
主题备用名称	否	勾选**使用**后，需添加主题备用名称。
颁发者备用名称	否	勾选**使用**后，需添加颁发者备用名称。
主题目录属性	否	勾选**使用**后，需添加特定的密钥和值。
名称约束	否	勾选**使用**后，需在黑名单中添加特定的值。

验证数据信息说明

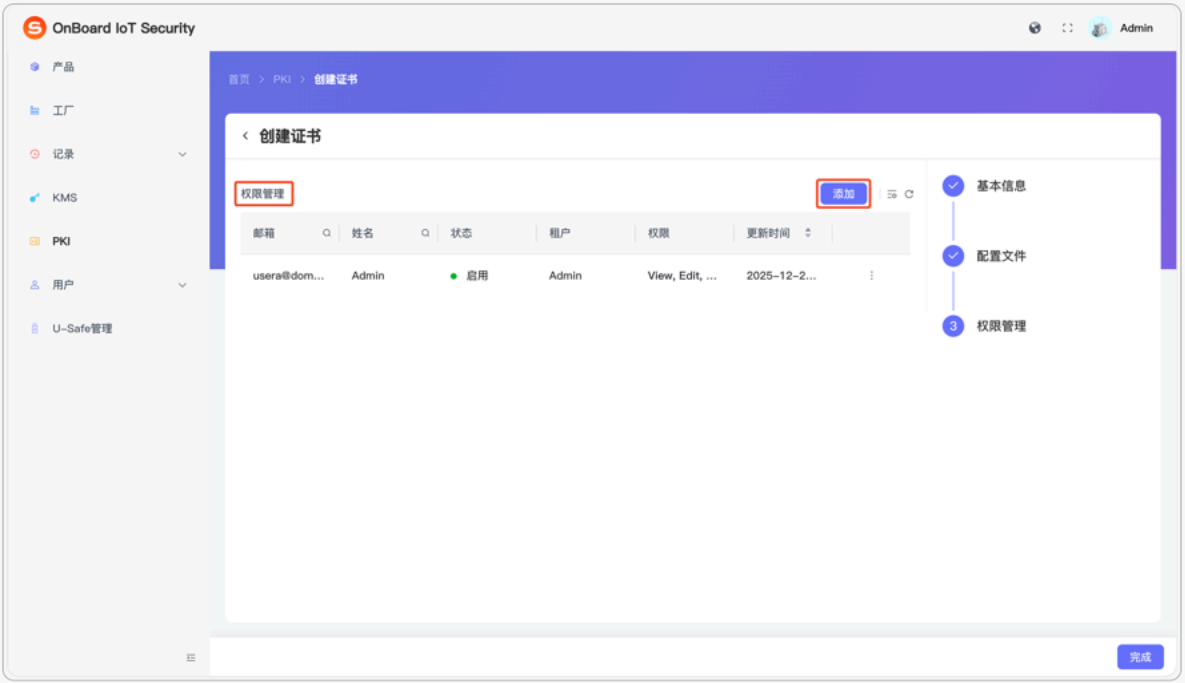
> **提示**
>
> 如果勾选某个信息的**关键**复选框，系统在处理证书时必须识别并支持该信息；否则，该证书将被拒绝。



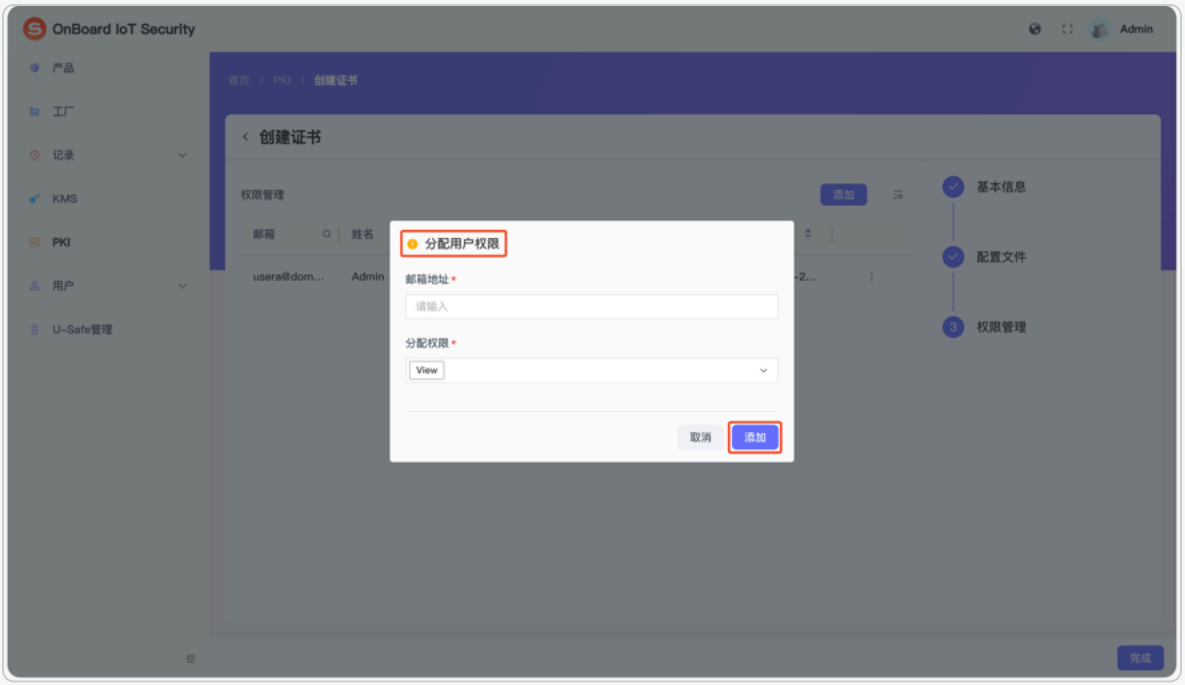
信息	是否必填	说明
CRL 分发点	否	勾选**使用**后，需添加 CRL 分发点 URI 和 CRL 颁发者。

信息	是否必填	说明
最新 CRL（又称 Delta CRL DP）	否	勾选**使用**后，需添加最新 CRL URI。
授权信息访问	否	勾选**使用**后，需添加 OCSP 服务定位器 URI 并在 CA 颁发者 URI 中添加特定的值。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Grant 和 Delete）。然后，点击**添加**。



6. 点击**完成**，即可完成终端实体模板的创建。

> **提示**
>
> 创建终端实体模板时，如果其继承自自有根证书，你需要完成 CSR （证书签名请求）配置。在 CSR 配置完成之前，该模板将处于**待完成**状态，且无法正常使用。

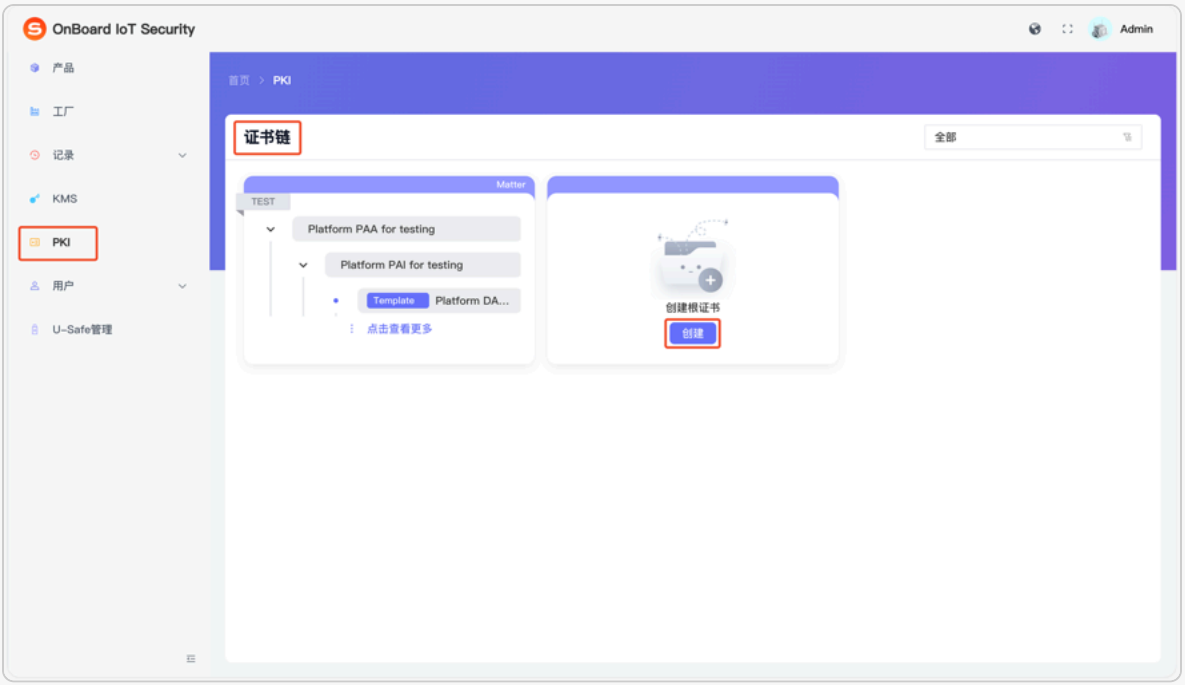
终端实体模板的 CSR 配置步骤

> **提示**
>
> 创建终端实体模板时，如果其继承自自有根证书，参考以下步骤完成 CSR 配置。

1. 在**证书链**页面，点击对应的证书链，然后点击对应的终端实体模板，进入证书页面。

2. 在证书页面，点击提示中的链接以完成 CSR 的配置。

3. 在 **CSR 配置**页面，点击下载 图标来下载 CSR 文件。然后，使用你的 Root CA（Certificate Authority）对该文件进行签署，从而生成由 Root CA 签名的终端实体模板。



4. 签署完成后，在上传区域中点击添加 图标，选择并上传格式为 **.pem** 的终端实体模板，或者直接拖曳你的终端实体模板到上传区域。

5. 点击**提交**，配置完成后，终端实体模板状态会变更为**正常**。

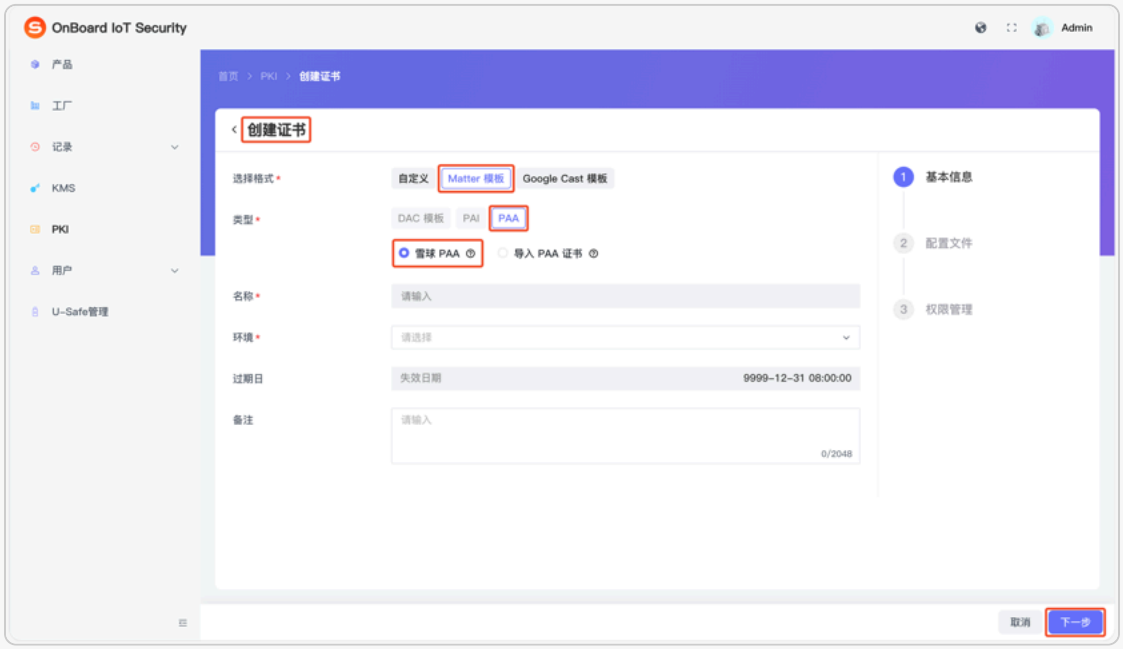
创建基于 Matter 模板的证书

> **提示**
>
> 创建基于 Matter 模板的证书或证书模板后，可在配置产品 Certificate 时进行选择。

创建基于 Matter 模板的 PAA 证书 | 创建基于 Matter 模板的 PAI 证书 | 创建基于 Matter 模板的 DAC 模板

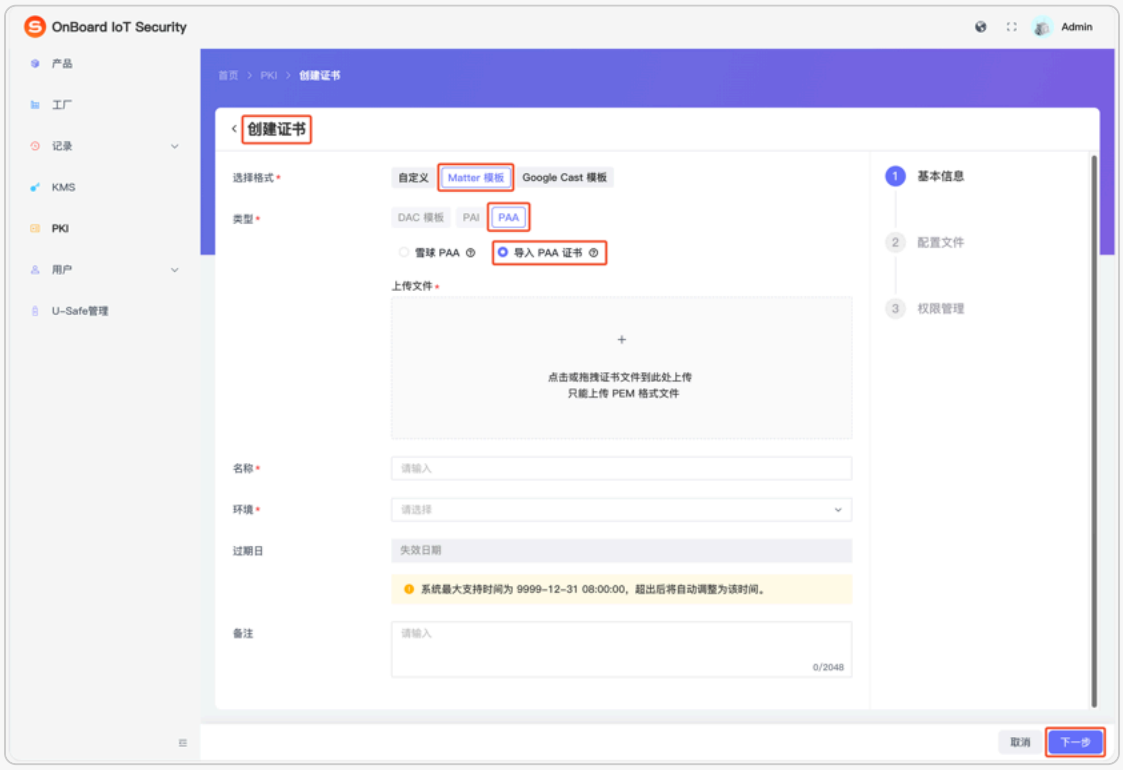
1. 在**创建证书**页面，格式选择 **Matter 模板**，类型选择 **PAA**。
2. 根据你的需求选择**雪球 PAA** 或 **Import your own PAA to serve as the trust anchor** （自有的 PAA 证书）。
    - 如果不使用自有基于 Matter 模板的 PAA 证书，请选择**雪球 PAA** 并输入基础信息。

> **提示**
>
> 使用雪球提供的 PAA。



- 如果使用自有基于 Matter 模板的 PAA 证书，请选择 **Import your own PAA to serve as the trust anchor** 并选择与输入基础信息。

> **提示**
>
> 手动上传你在系统外部生成的 PAA 证书（仅包含公钥），用于签名验证。



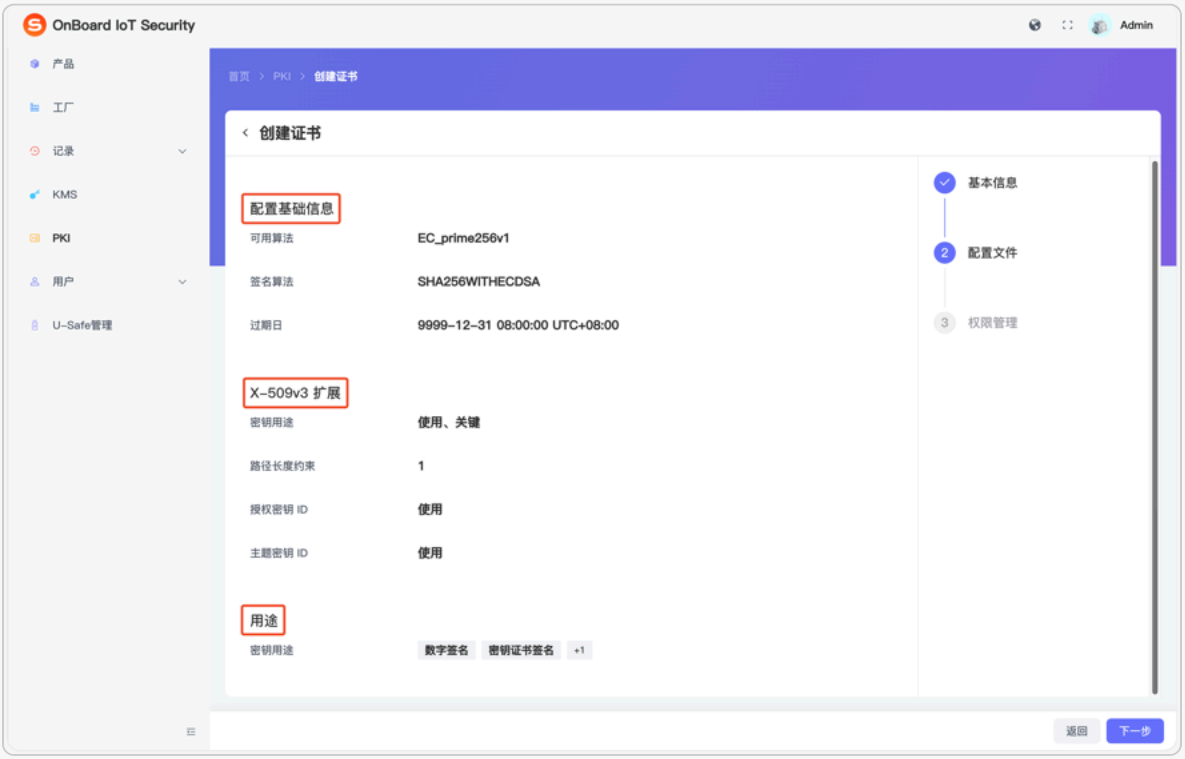
PAA 证书基础信息说明：

信息	雪球 PAA	Import your own PAA to serve as the trust anchor
上传文件	不需要上传文件。	在**上传文件**区域中点击添加 ➕ 图标，选择并上传格式为 **.pem** 的 PAA 证书，或者直接拖曳自有基于 Matter 模板的 PAA 证书到上传区域。
名称	名称固定且无法修改。 ▪ TEST 环境：Snowball Matter UAT PAA 01 ▪ PROD 环境：Snowball Matter PROD PAA 01	同一环境中，名称必须唯一。
环境	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。	
过期日	雪球 PAA 证书过期日，且无法修改。	根据证书内容自动解析，且无法修改。

信息	**雪球 PAA**	**Import your own PAA to serve as the trust anchor**
备注	用于补充说明 PAA 证书相关的额外信息，如特殊要求和注意事项等。	

3. 确认信息无误后，点击**下一步**并查看以下信息。

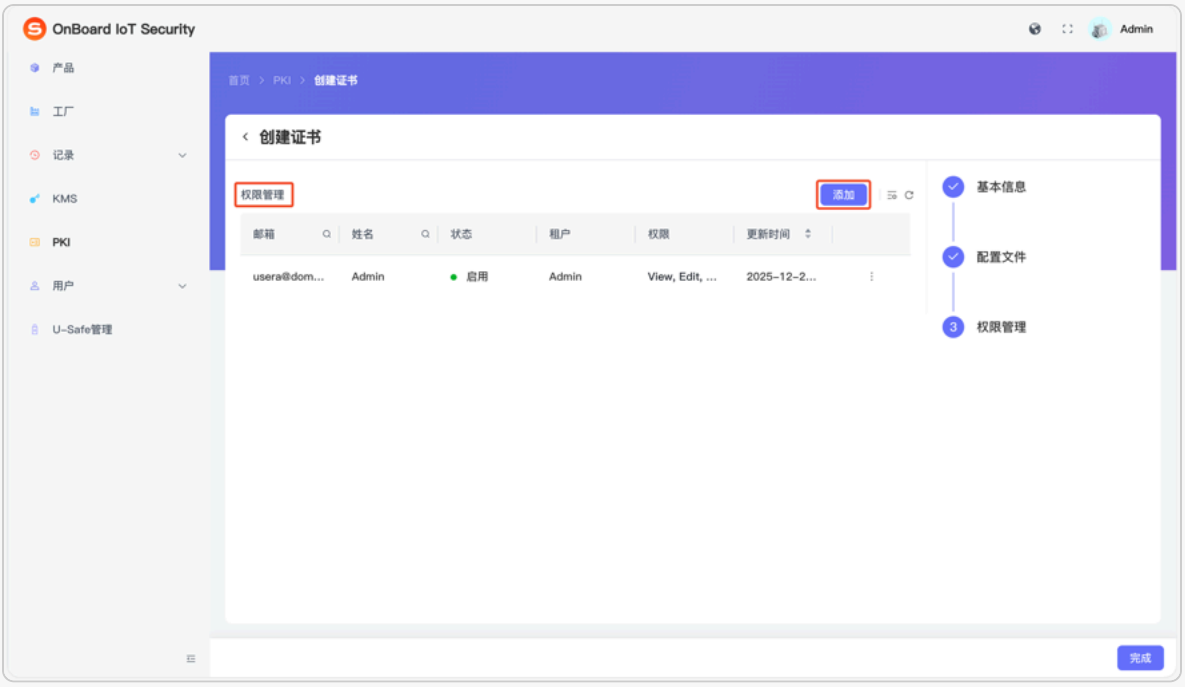
> **提示**
>
> 如果使用自有 PAA 证书，页面只显示**配置基础信息**区域。



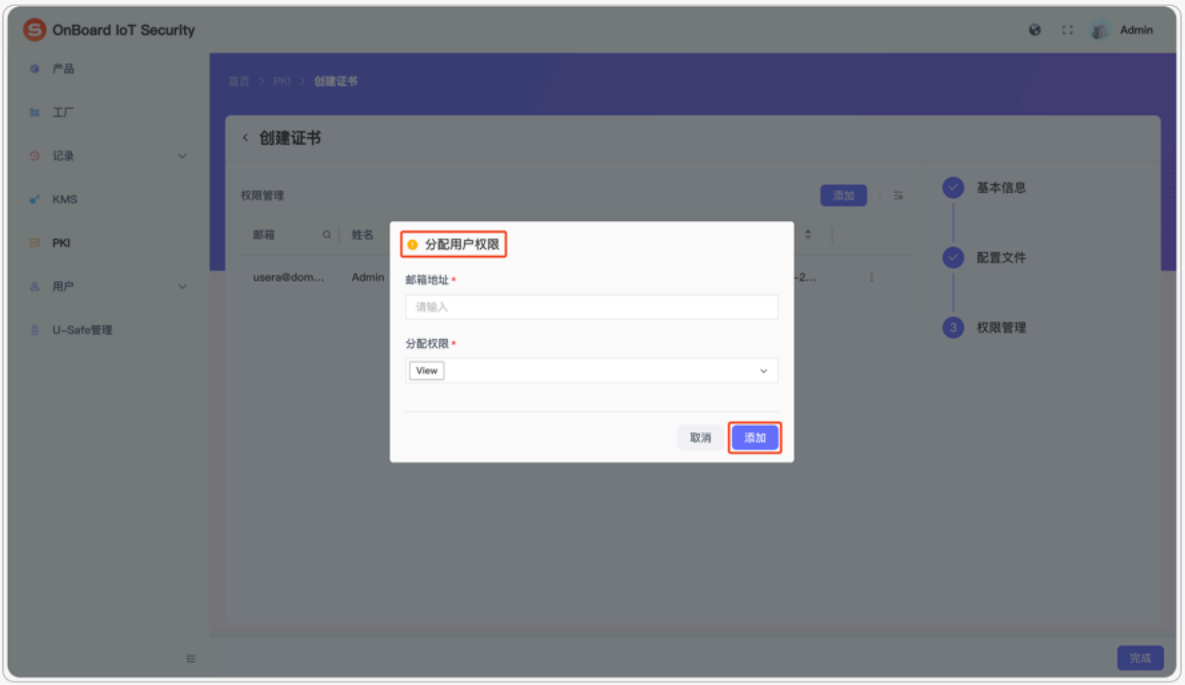
PAA 证书配置信息说明：

区域	信息	说明
配置基础信息	可用算法	根据证书内容自动解析，且无法修改。
	签名算法	根据证书内容自动解析，且无法修改。
	过期日	根据证书内容自动解析，且无法修改。
X-509v3 扩展	密钥用途	根据证书内容自动解析，且无法修改。
	路径长度约束	根据证书内容自动解析，且无法修改。
	授权密钥 ID	根据证书内容自动解析，且无法修改。
	主题密钥 ID	根据证书内容自动解析，且无法修改。
用途	密钥用途	**数字签名**，**密钥证书签名**和 **CRL 签名**等密钥用途。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Revoke，Grant 和 Delete）。然后，点击**添加**。

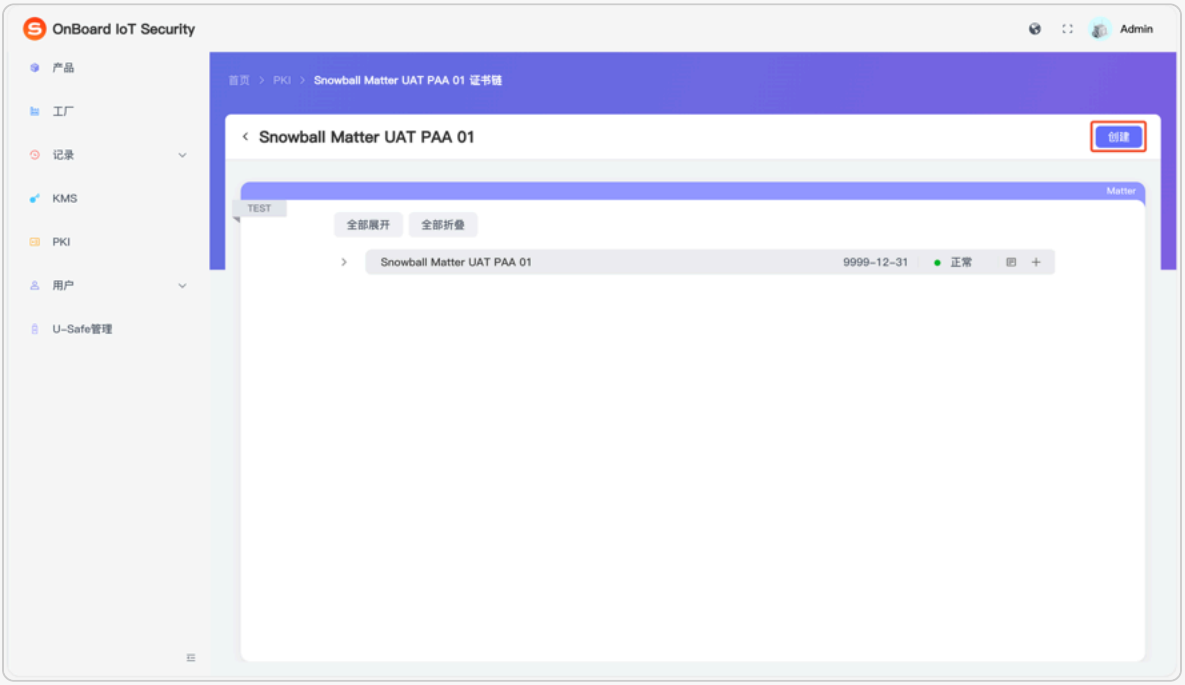


6. 点击**完成**，即可完成基于 Matter 模板的 PAA 证书的创建。

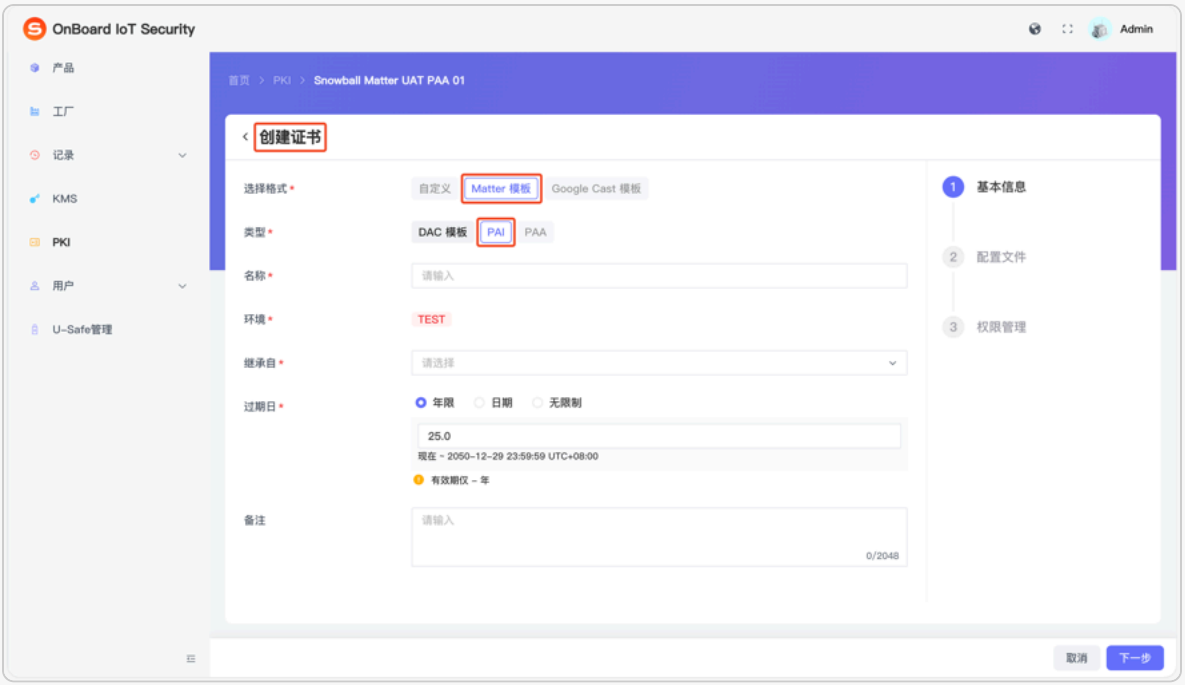
> **信息**
>
> 在创建基于 Matter 模板的 PAI 证书之前，请确保已创建至少一个基于 Matter 模板的 PAA 证书，以便在配置时进行选择并继承。

1. 在**证书链**页面，点击对应的证书链，然后点击**创建**。

> **提示**
>
> 密钥图标表示该证书包含私钥。



2. 在**创建证书**页面，格式选择 **Matter 模板**，类型选择 **PAI** 并选择与输入基础信息。

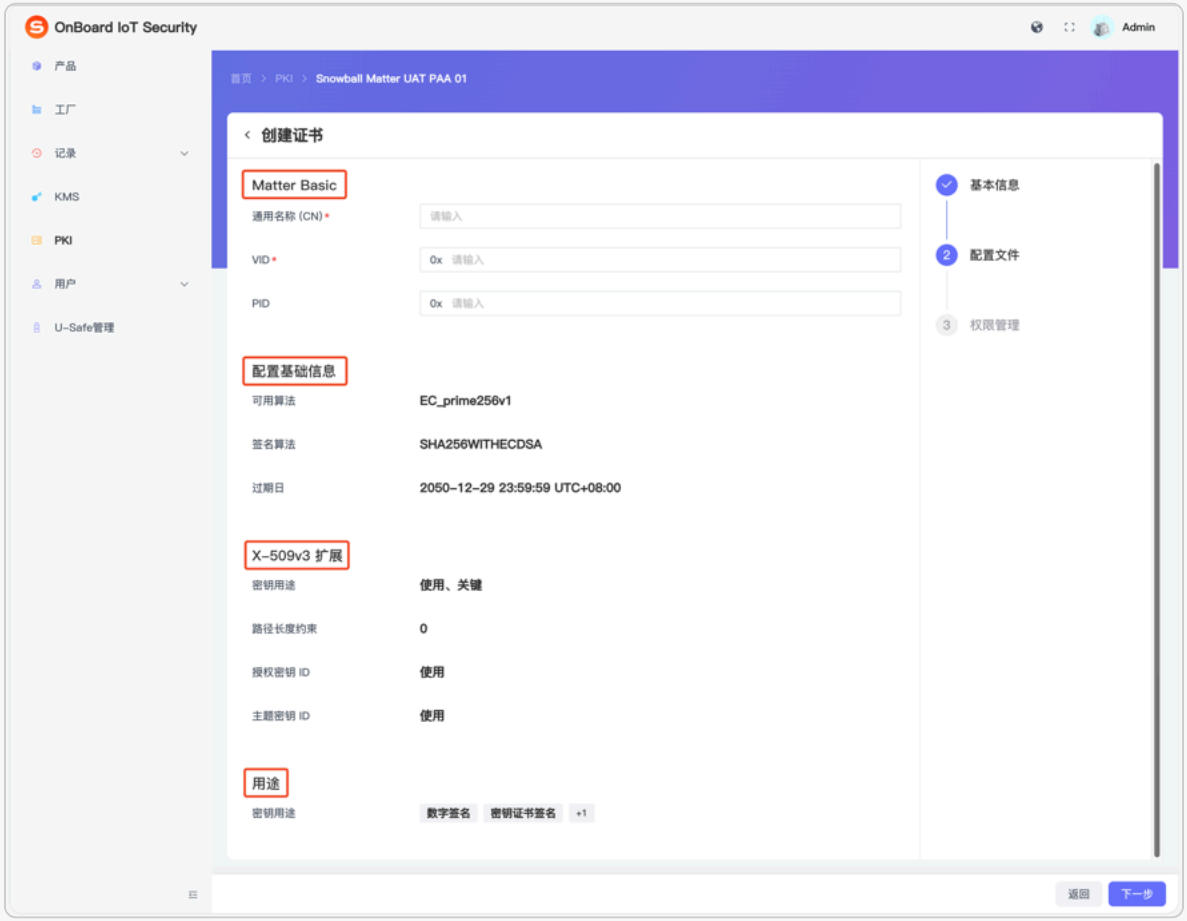


PAI 证书基础信息说明：

信息	说明
名称	同一证书链中，同一层级的证书名称必须唯一。
环境	继承自 PAA 证书的环境。
继承自	选择继承的 PAA 证书。
过期日	设定 PAI 证书的过期日。不可超过继承的 PAA 证书的过期日。 **注释**：如果继承自自有 PAA 证书，过期日会继承自 PAA 证书。
备注	用于补充说明 PAI 证书相关的额外信息，如特殊要求和注意事项等。

3. 确认信息无误后，点击**下一步**并参考以下信息进行配置。

> **提示**
>
> 如果 PAI 证书继承自自有 PAA 证书，页面只显示 **Matter Basic** 和**配置基础信息**区域。

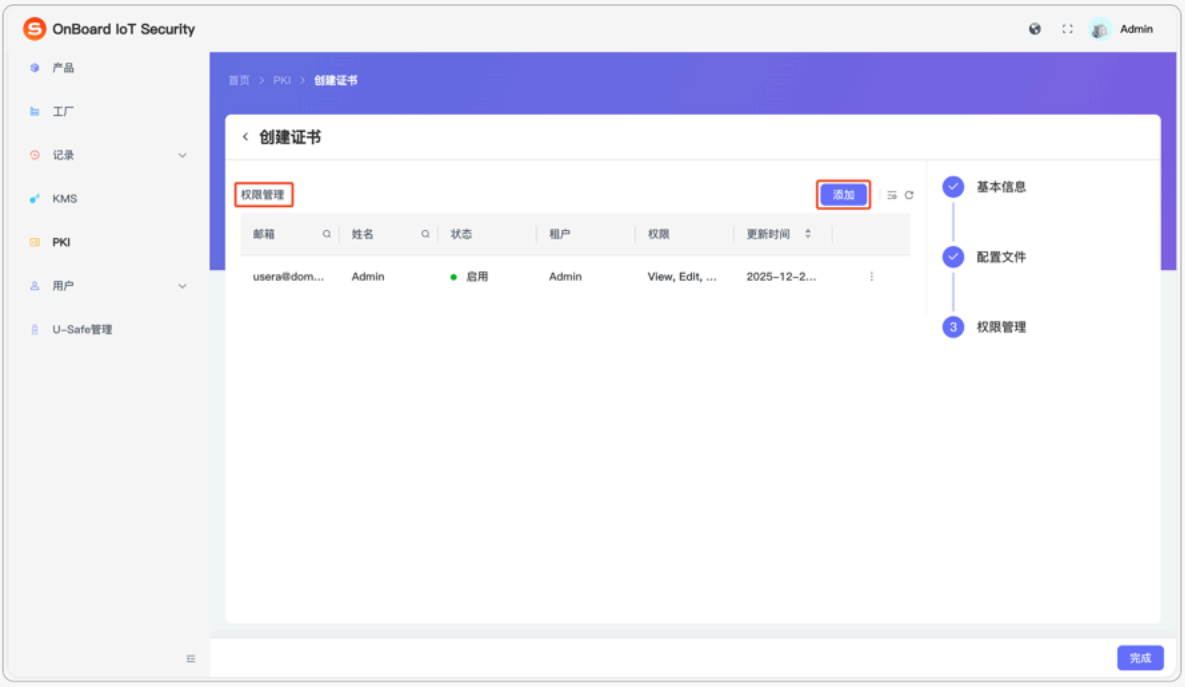


PAI 证书配置信息说明：

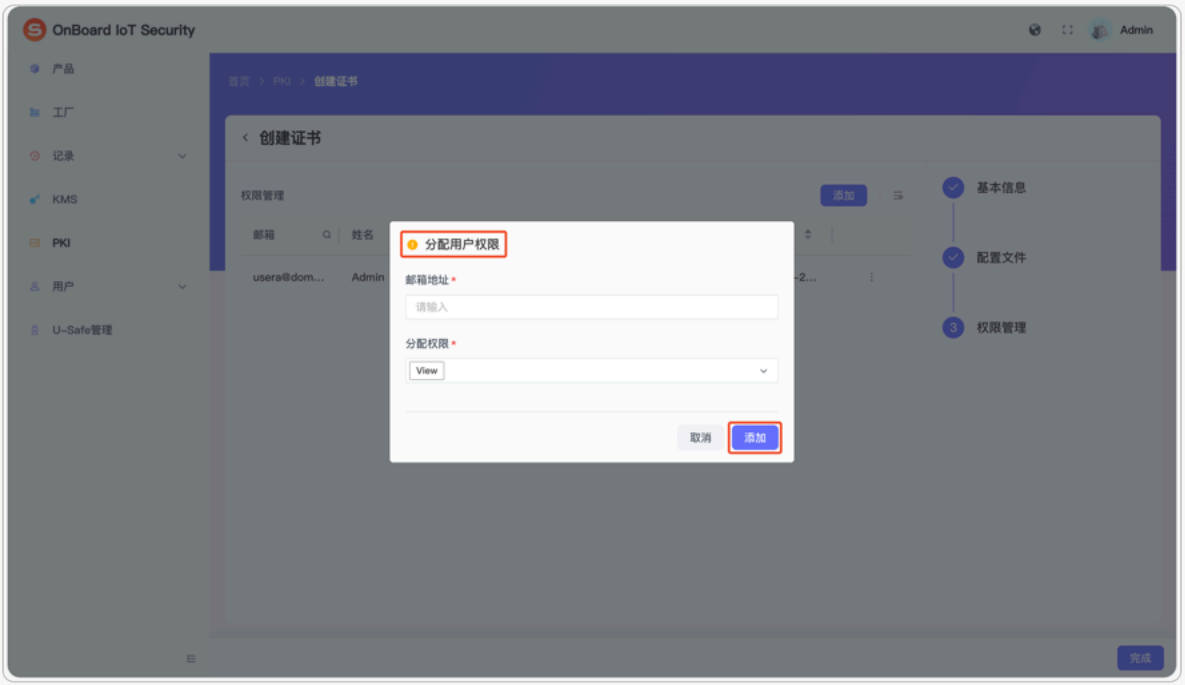
区域	信息	说明
Matter Basic	通用名称（CN）（必选）	证书的通用名称。

区域	信息	说明
	VID（必选）	供应商 ID（Vendor ID）。
	PID（可选）	产品 ID（Product ID）。
配置基础信息	可用算法	根据证书内容自动解析，且无法修改。
	签名算法	根据证书内容自动解析，且无法修改。
	过期日	继承上一步在基础信息设置的过期日，且无法修改。
X-509v3 扩展	密钥用途	根据证书内容自动解析，且无法修改。
	路径长度约束	根据证书内容自动解析，且无法修改。
	授权密钥 ID	根据证书内容自动解析，且无法修改。
	主题密钥 ID	根据证书内容自动解析，且无法修改。
用途	密钥用途	**数字签名**，**密钥证书签名**和 **CRL 签名**等密钥用途。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Revoke，Grant 和 Delete）。然后，点击**添加**。



6. 点击**完成**，即可完成 PAI 证书的创建。

> **提示**
>
> 创建基于 Matter 模板的 PAI 证书时，如果其继承自自有 PAA 证书，你需要完成 CSR （证书签名请求）的配置。在CSR 配置完成之前，该证书将处于**待完成**状态，且无法正常使用。

基于 Matter 模板的 PAI 证书的 CSR 配置步骤

> **提示**
>
> 创建基于 Matter 模板的 PAI 证书时，如果其继承自自有的 PAA 证书，参考以下步骤完成 CSR 配置。

1. 在**证书链**页面，点击对应的证书链，然后点击对应的 PAI 证书，进入证书页面。
2. 在证书页面，点击提示中的链接以完成 CSR 的配置。
3. 在 **CSR 配置**页面，点击下载 图标来下载 CSR 文件。然后，使用你的 Root CA 对该文件进行签署，从而生成由 Root CA 签名的 PAI 证书。



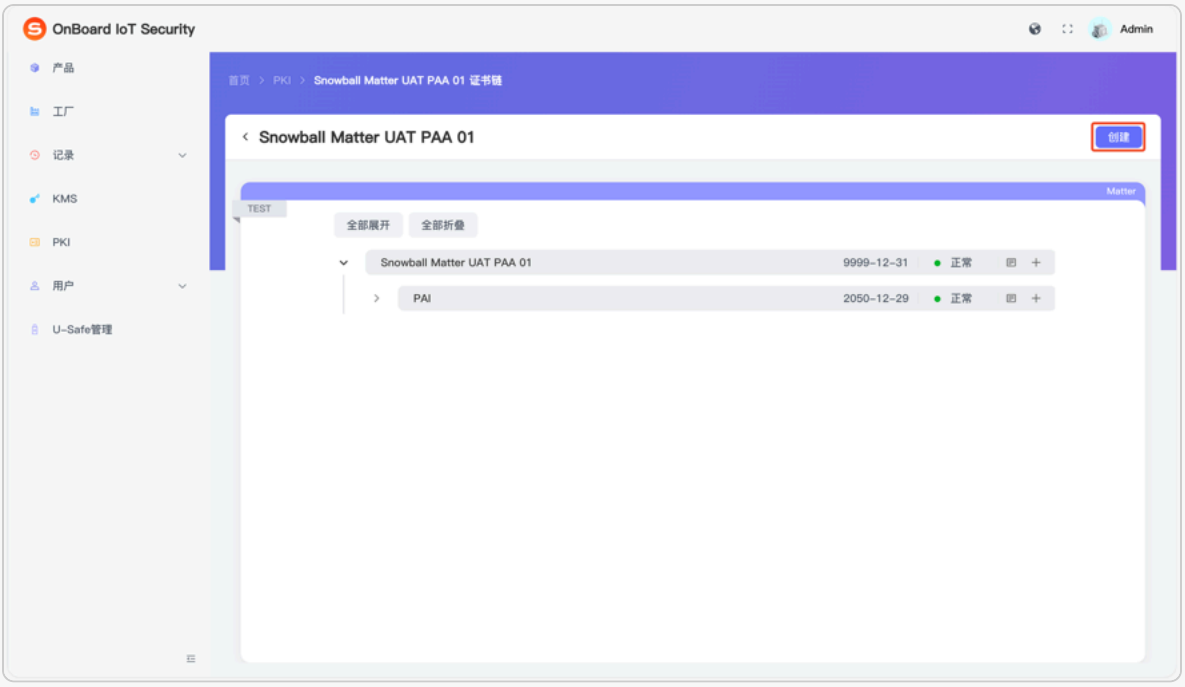
4. 签署完成后，在上传区域中点击添加 ➕ 图标，选择并上传格式为 **.pem** 的 PAI 证书，或者直接拖曳你的 PAI 证书到上传区域。

5. 点击**提交**，配置完成后，PAI 证书会变更为**正常**状态。

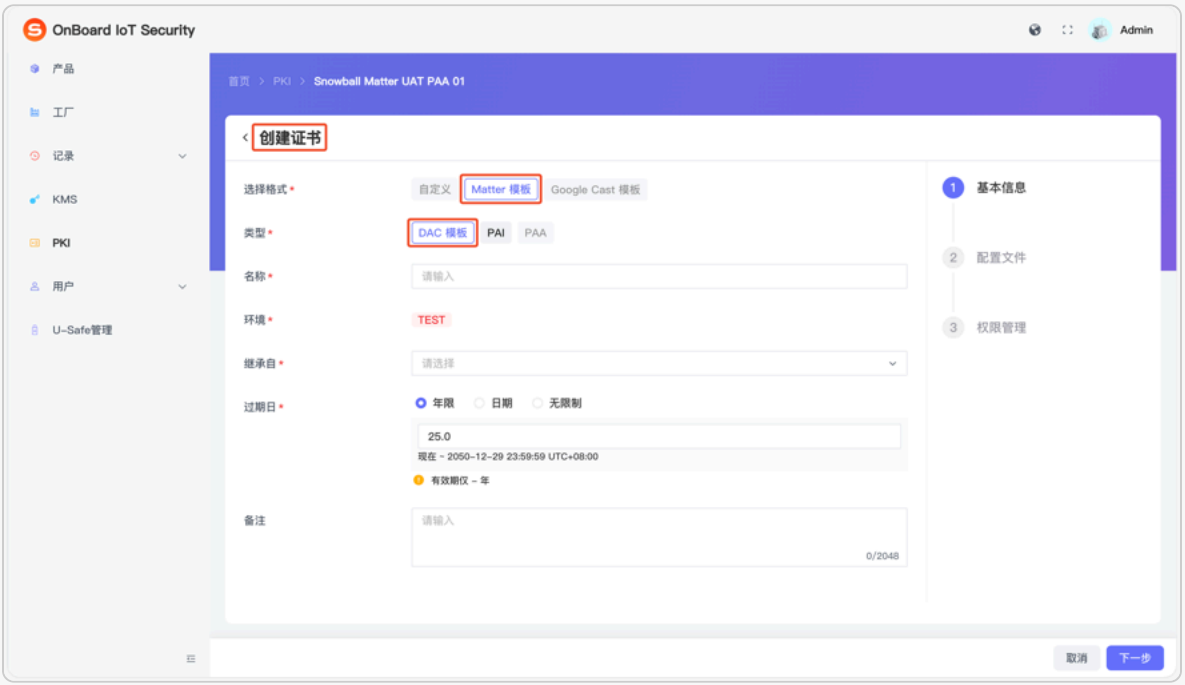
> **信息**
>
> 在创建 DAC 模板之前，请确保已创建至少一个基于 Matter 模板的 PAI 证书，以便在配置时进行选择并继承。

1. 在**证书链**页面，点击对应的证书链，然后点击**创建**。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



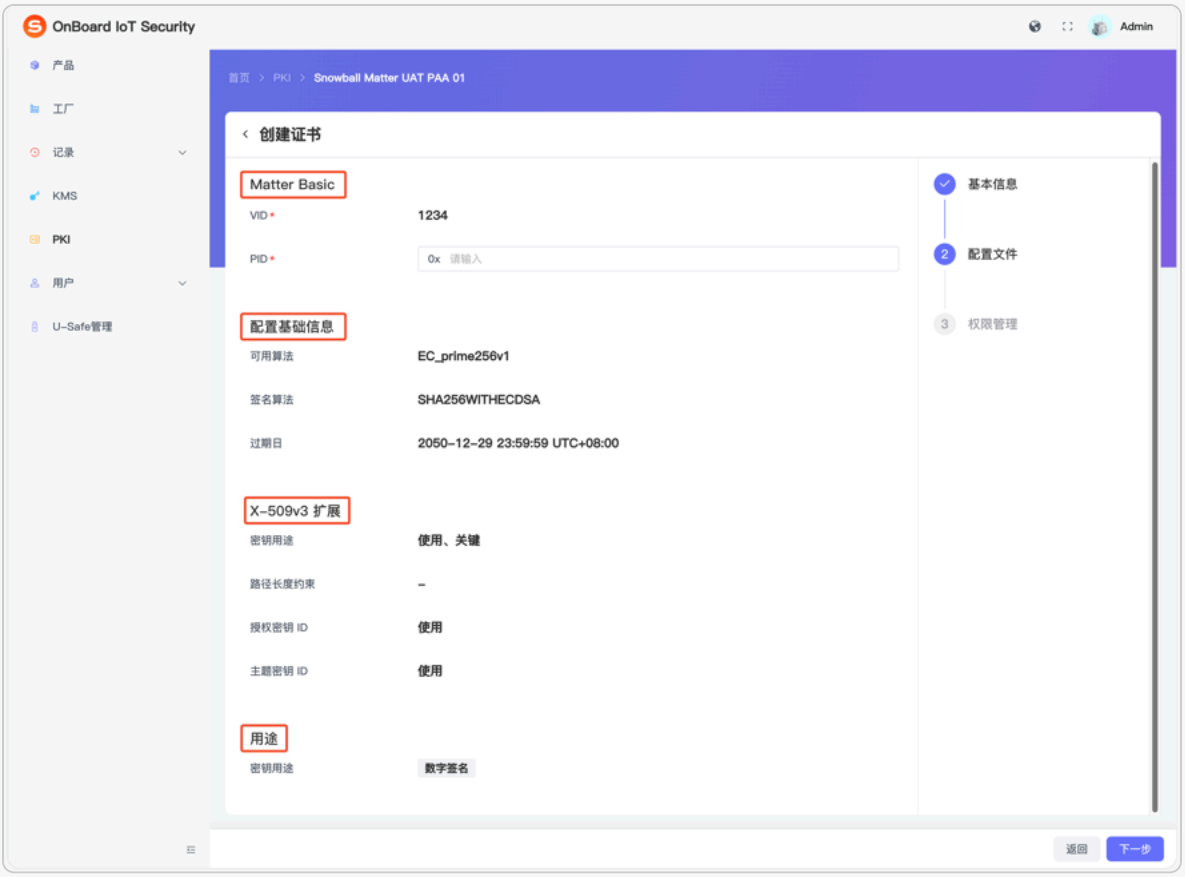
2. 在**创建证书**页面，格式选择 **Matter 模板**，类型选择 **DAC 模板**并参考以下信息进行配置。



DAC 模板基础信息说明：

信息	说明
名称	同一证书链中，同一层级的证书名称必须唯一。
环境	继承自 PAA 证书的环境。
继承自	选择继承的 PAI 证书。
过期日	设定 DAC 模板的过期日，不可超过继承的 PAI 证书的过期日。
备注	用于补充说明 DAC 模板相关的额外信息，如特殊要求和注意事项等。

3. 确认信息无误后，点击**下一步**并查看以下信息。

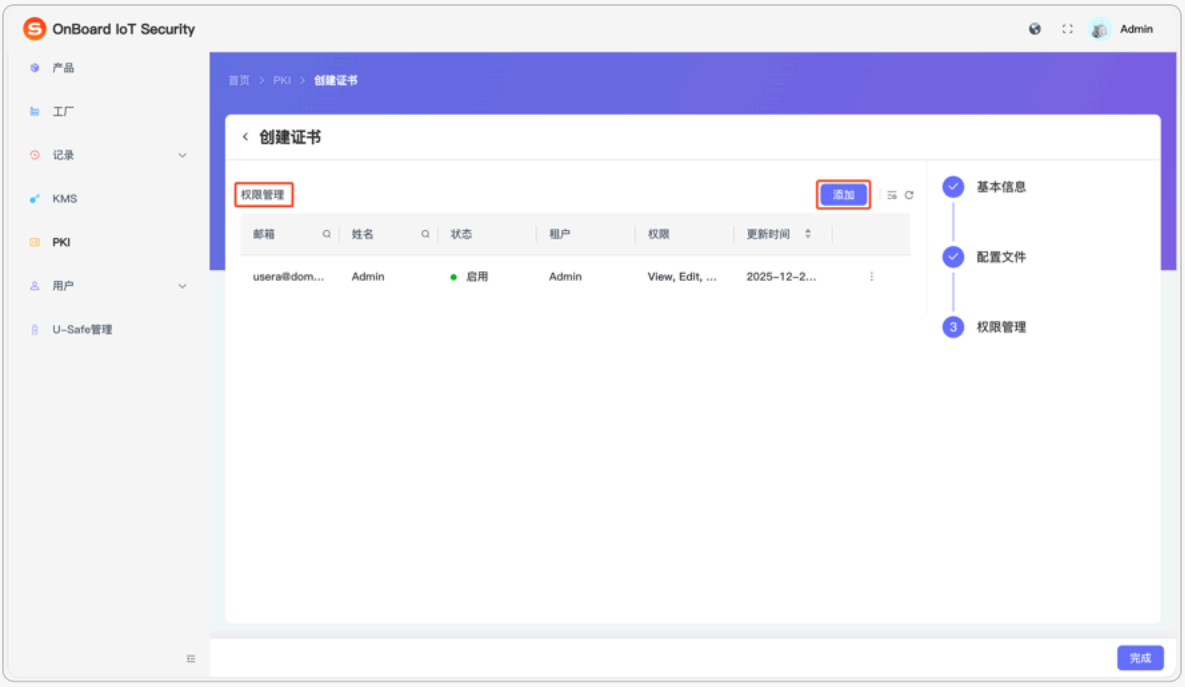


DAC 模板配置信息说明：

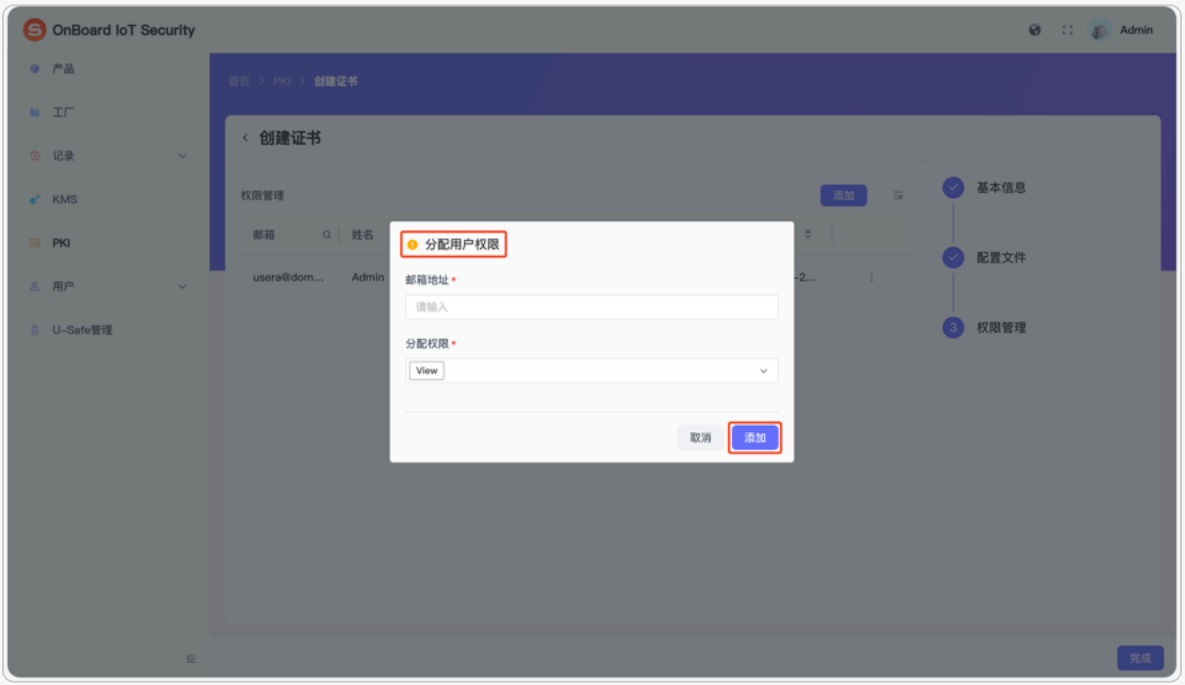
区域	信息	说明
Matter Basic	VID	继承自 PAI 证书的供应商 ID（Vendor ID）。
	PID（必选）	产品 ID（Product ID）。 **注释**：当 DAC 模板继承自一个已配置 PID 的 PAI 证书时，PID会自动继承，且无法修改。
配置基础信息	可用算法	根据证书内容自动解析，且无法修改。

区域	信息	说明
	签名算法	根据证书内容自动解析，且无法修改。
	过期日	继承上一步在基础信息设置的过期日，且无法修改。
X-509v3 扩展	密钥用途	根据证书内容自动解析，且无法修改。
	路径长度约束	根据证书内容自动解析，且无法修改。
	授权密钥 ID	根据证书内容自动解析，且无法修改。
	主题密钥 ID	根据证书内容自动解析，且无法修改。
用途	密钥用途	**数字签名**等密钥用途。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Grant 和 Delete）。然后，点击**添加**。



6. 点击**完成**，即可完成 DAC 模板的创建。

创建基于 Google Cast 模板的证书

> **提示**
>
> 创建基于 Google Cast 模板的证书或证书模板后，可在配置产品 Certificate 时进行选择。

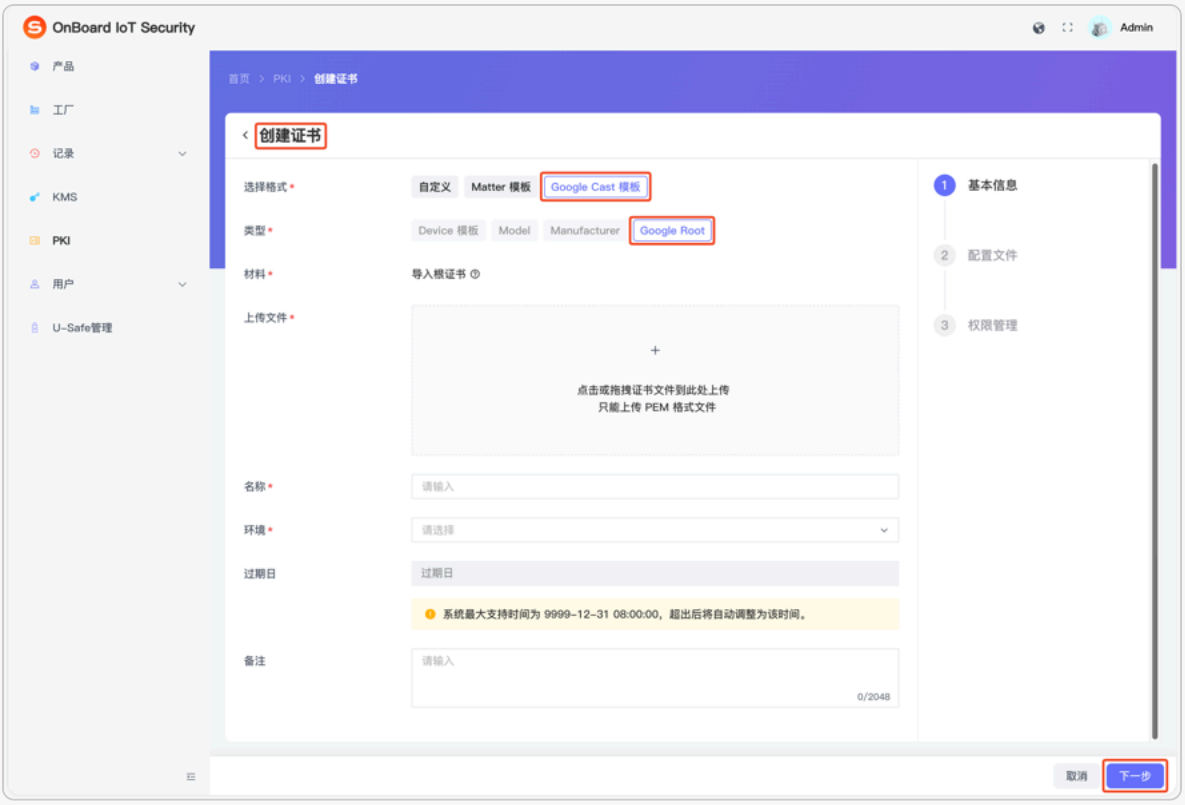
创建基于 Google Cast 模板的 Google Root 证书

创建基于 Google Cast 模板的 Manufacturer 证书

创建基于 Google Cast 模板的 Model 证书

创建基于 Google Cast 模板的 Device 模板

1. 在**创建证书**页面，格式选择 **Google Cast 模板**，类型选择 **Google Root** 并选择与输入基础信息。

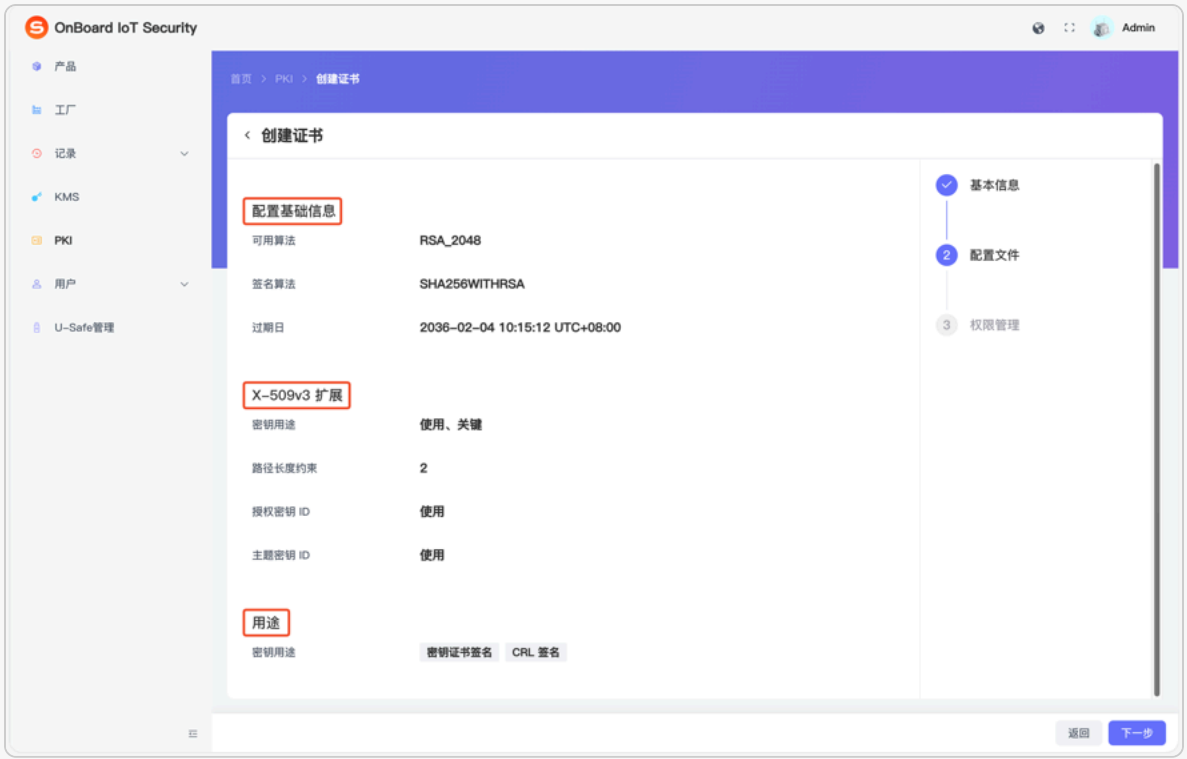


Google Root 证书基础信息说明：

信息	说明
上传文件	在**上传文件**区域中点击添加 ➕ 图标，选择并上传格式为 **.pem** 的 Google Root 证书，或者直接拖曳基于 Google Cast 模板的 Google Root 证书到上传区域。
名称	同一环境中，名称必须唯一。

信息	说明
环境	有两种环境可以选择，分别是 **TEST** 环境和 **PROD** 环境。
过期日	根据证书内容自动解析，且无法修改。
备注	用于补充说明 Google Root 证书相关的额外信息，如特殊要求和注意事项等。

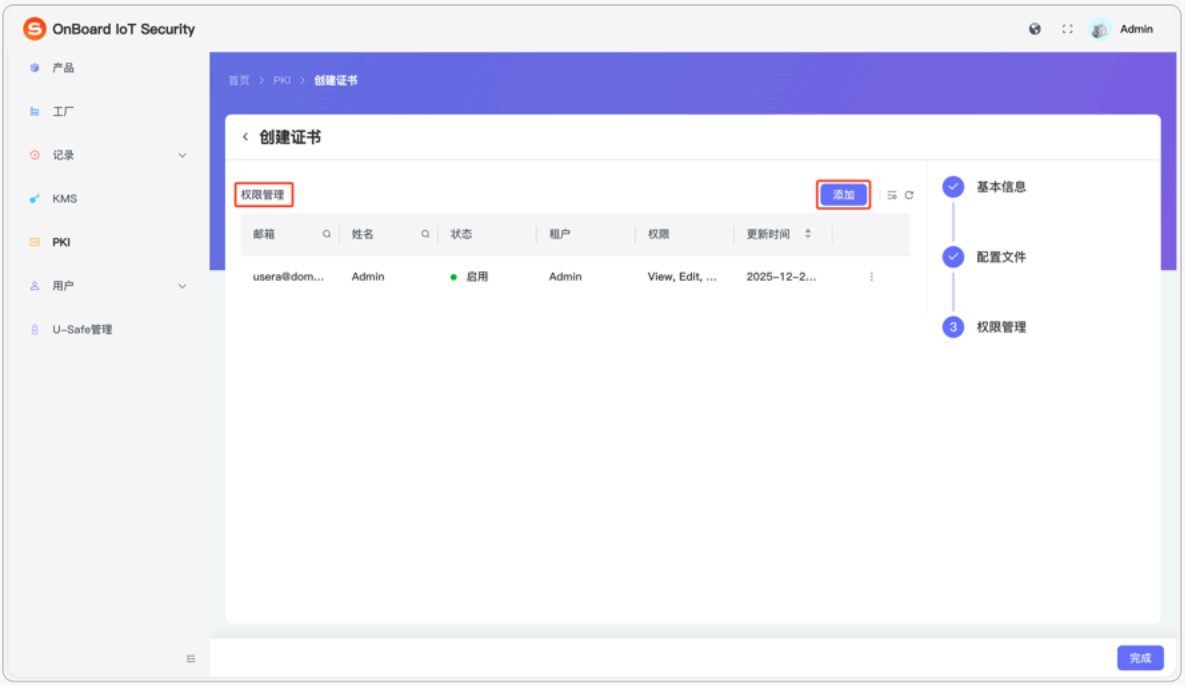
2. 确认信息无误后，点击**下一步**并查看以下信息。



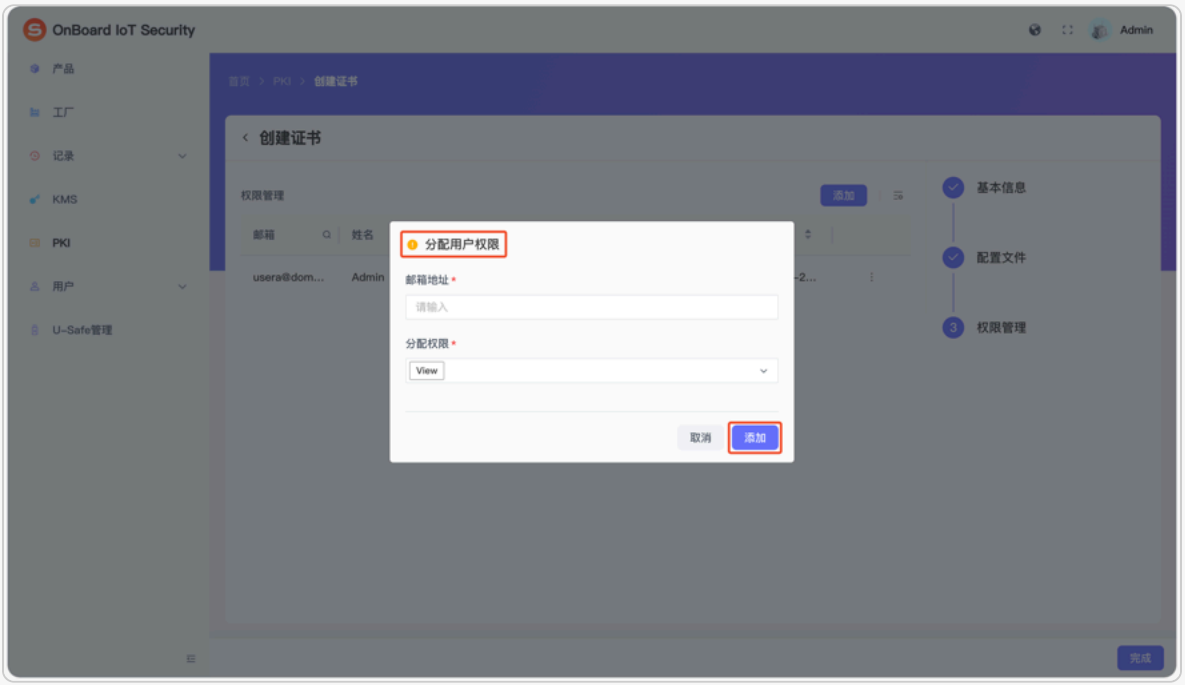
Google Root 证书配置信息说明：

区域	信息	说明
配置基础信息	可用算法	根据证书内容自动解析，且无法修改。
	签名算法	根据证书内容自动解析，且无法修改。
	过期日	根据证书内容自动解析，且无法修改。
X-509v3 扩展	密钥用途	根据证书内容自动解析，且无法修改。
	路径长度约束	根据证书内容自动解析，且无法修改。
	授权密钥 ID	根据证书内容自动解析，且无法修改。
	主题密钥 ID	根据证书内容自动解析，且无法修改。
用途	密钥用途	**数字签名**，**密钥证书签名**和 **CRL 签名**等密钥用途。

3. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



4. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Revoke，Grant 和 Delete）。然后，点击**添加**。

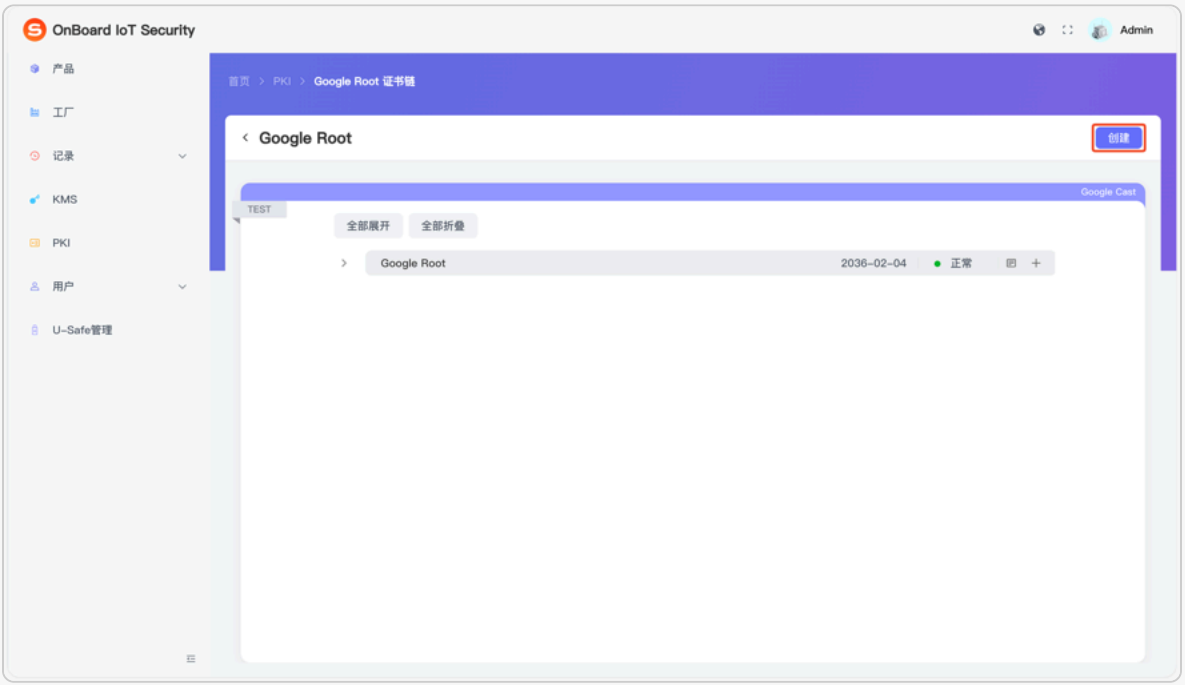


5. 点击**完成**，即可完成基于 Google Cast 模板的 Google Root 证书的创建。

> **信息**
>
> 在创建基于 Google Cast 模板的 Manufacturer 证书之前，请确保已创建至少一个基于 Google Cast 模板的 Google Root 证书，以便在配置时进行选择并继承。

1. 在**证书链**页面，点击对应的证书链，然后点击**创建**。

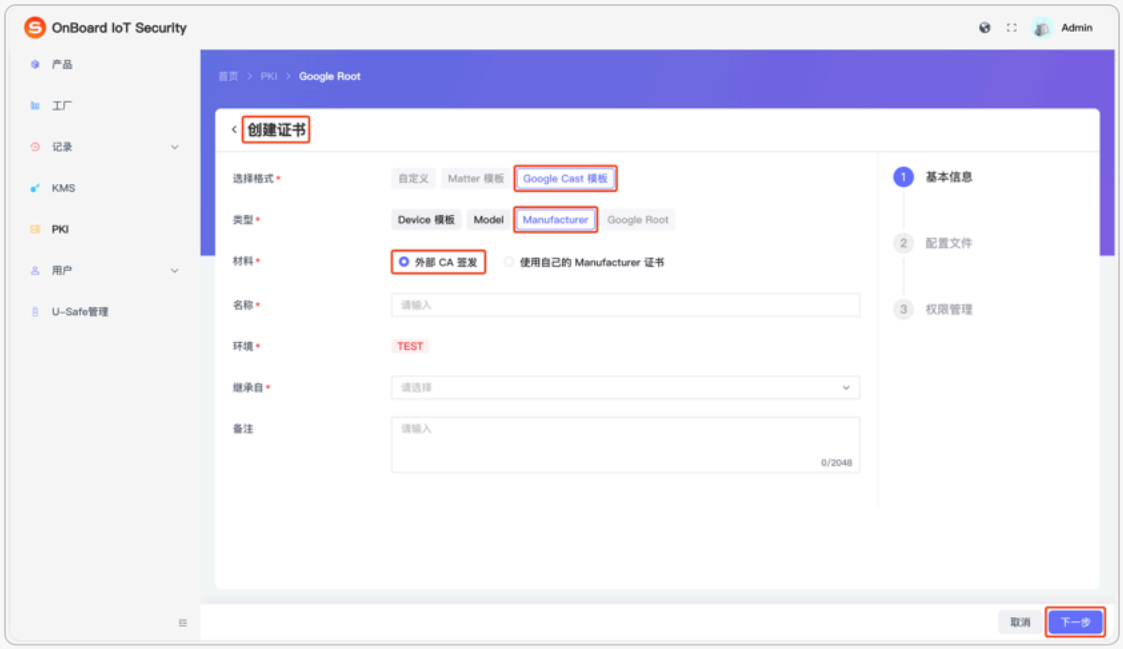
   > **提示**
   >
   > 密钥图标表示该证书包含私钥。

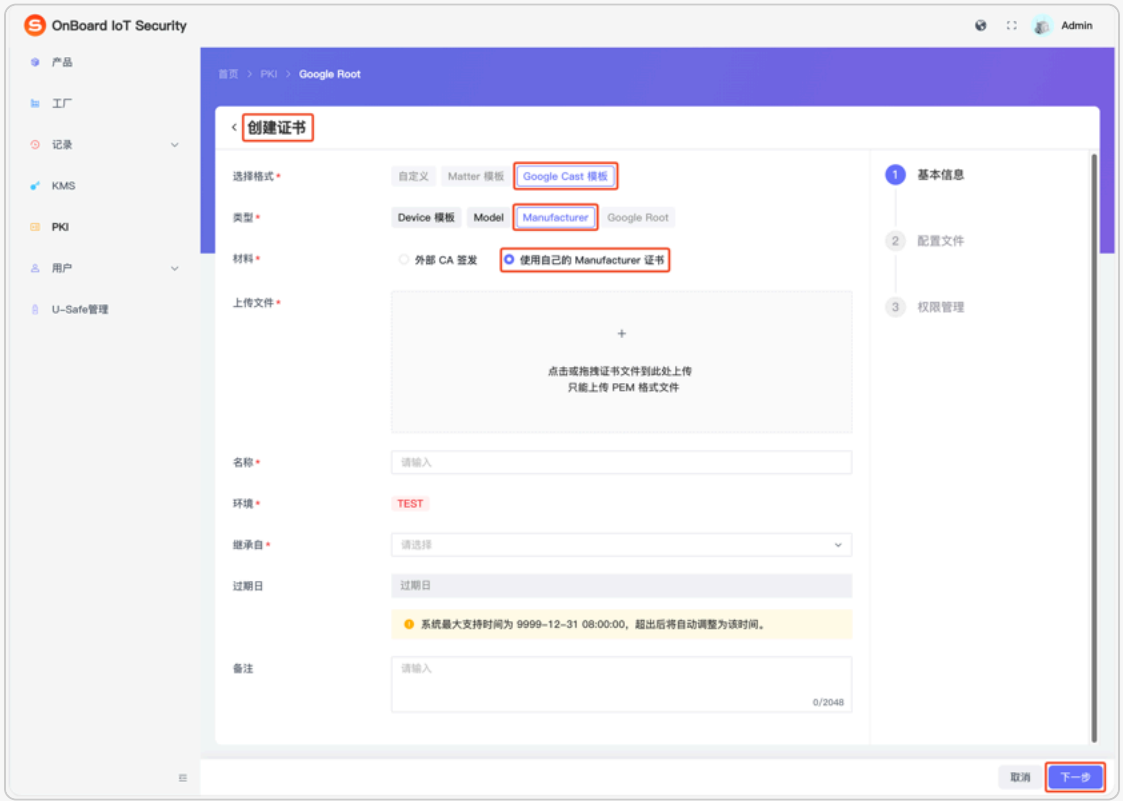
2. 在**创建证书**页面，格式选择 **Google Cast 模板**，类型选择 **Manufacturer**。

3. 根据你的需求选择**外部 CA 签发**或**使用自己的 Manufacturer 证书**。

   - 如果不使用自有基于 Google Cast 模板的 Manufacturer 证书，请选择**外部 CA 签发**并输入基础信息。



- 如果使用自有基于 Google Cast 模板的 Manufacturer 证书，请选择**使用自己的 Manufacturer** 并选择与输入基础信息。

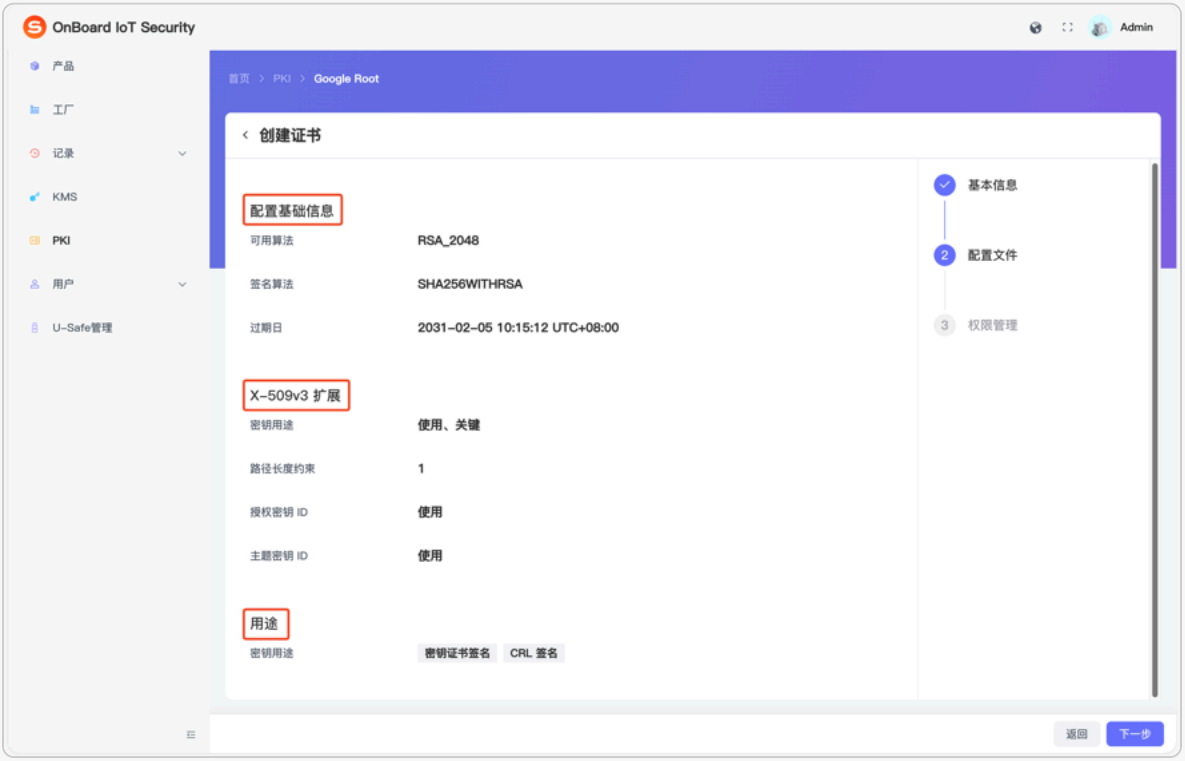


Manufacturer 证书基础信息说明：

信息	外部 CA 签发	使用自己的 Manufacturer 证书
上传文件	不需要上传文件。	在**上传文件**区域中点击添加 ➕ 图标，选择并上传格式为 **.pem** 的 Manufacturer 证书，或者直接拖曳自有基于 Google Cast 模板的 Manufacturer 证书到上传区域。
名称	同一证书链中，同一层级的证书名称必须唯一。	
环境	继承自 Google Root 证书的环境。	
继承自	选择继承的 Google Root 证书。	
过期日	不需要设置过期日。	根据证书内容自动解析，且无法修改。
备注	用于补充说明 Manufacturer 证书相关的额外信息，如特殊要求和注意事项等。	

4. 确认信息无误后，点击**下一步**并参考以下信息进行配置。

> **提示**
>
> 如果使用自有的 Manufacturer 证书，页面只显示**配置基础信息**区域。

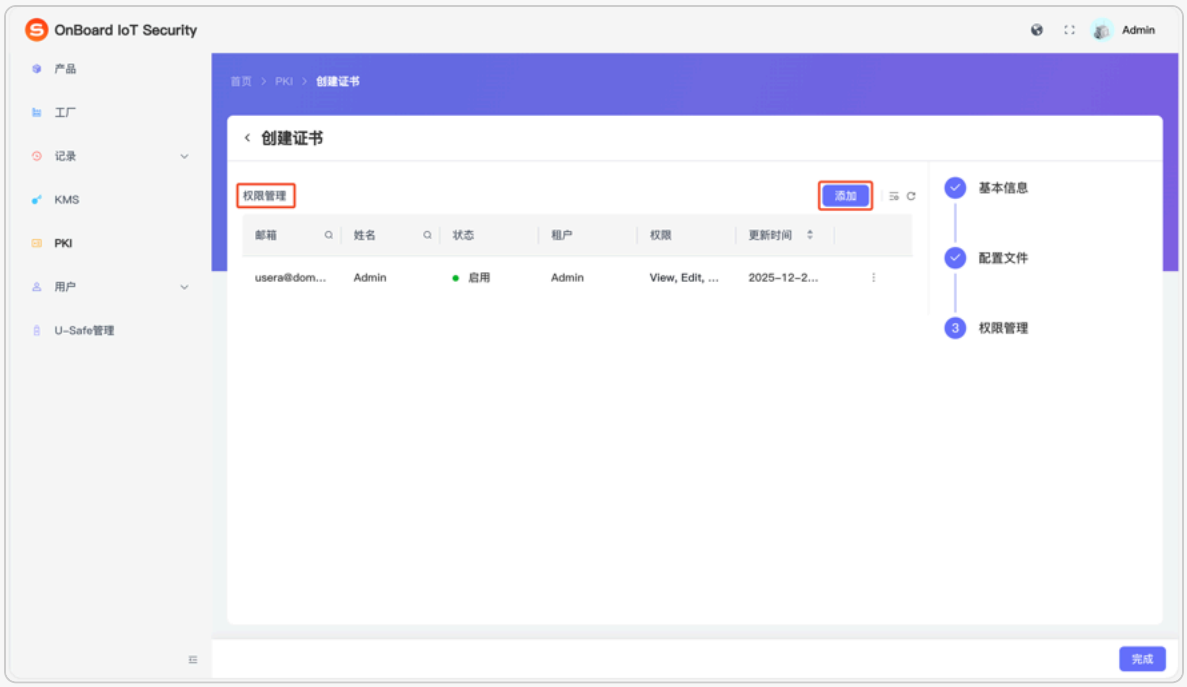


Manufacturer 证书配置信息说明：

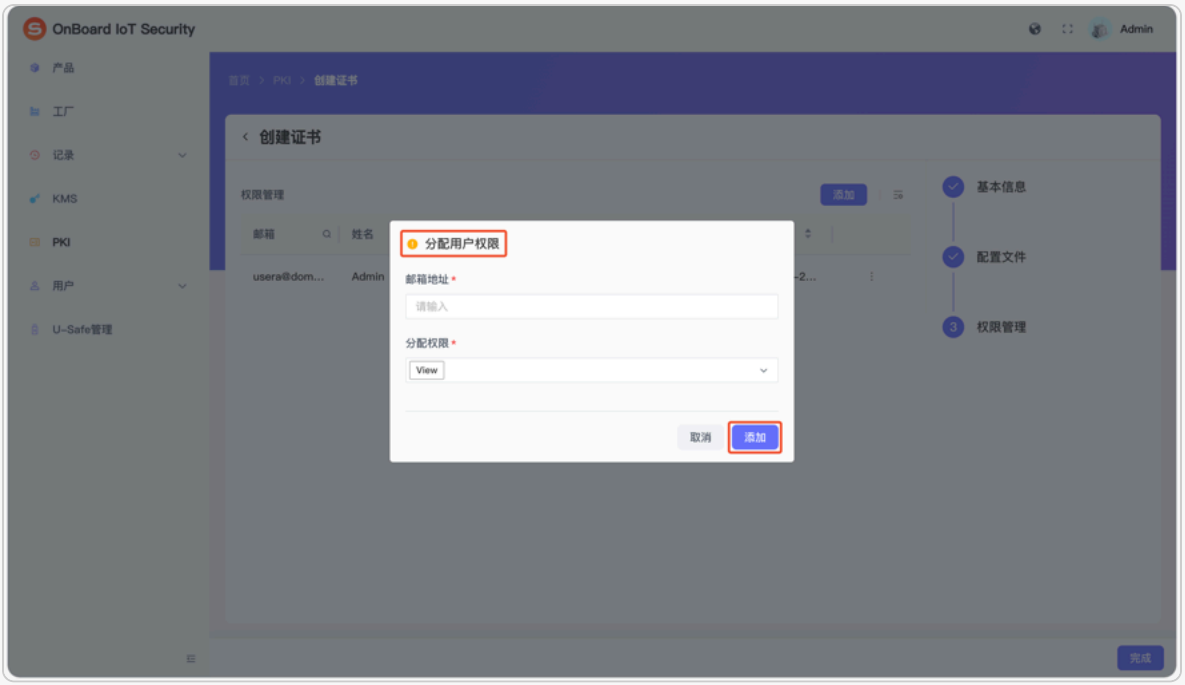
区域	信息	说明
配置基础信息	可用算法	根据证书内容自动解析，且无法修改。
	签名算法	根据证书内容自动解析，且无法修改。
	过期日	根据证书内容自动解析，且无法修改。

区域	信息	说明
X-509v3 扩展	密钥用途	根据证书内容自动解析，且无法修改。
	路径长度约束	根据证书内容自动解析，且无法修改。
	授权密钥 ID	根据证书内容自动解析，且无法修改。
	主题密钥 ID	根据证书内容自动解析，且无法修改。
用途	密钥用途	**数字签名**，**密钥证书签名**和 **CRL 签名**等密钥用途。

5. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



6. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Revoke，Grant 和 Delete）。然后，点击**添加**。



7. 点击**完成**，即可完成基于 Google Cast 模板的 Manufacturer 证书的创建。

> **提示**
>
> 创建外部 CA 签发的 Manufacturer 证书时，你需要完成 CSR （证书签名请求）的配置。在CSR 配置完成之前，该证书将处于**待完成**状态，且无法正常使用。

外部 CA 签发的 Manufacturer 证书的 CSR 配置步骤

> **提示**
>
> 创建外部 CA 签发的 Manufacturer 证书时，参考以下步骤完成 CSR 配置。

1. 在**证书链**页面，点击对应的证书链，然后点击对应的 Manufacturer 证书，进入证书页面。

2. 在证书页面，点击提示中的链接以完成 CSR 的配置。

3. 在 **CSR 配置**页面，点击下载 图标来下载 CSR 文件。然后，使用你的 Root CA 对该文件进行签署，从而生成由 Root CA 签名的 Manufacturer 证书。



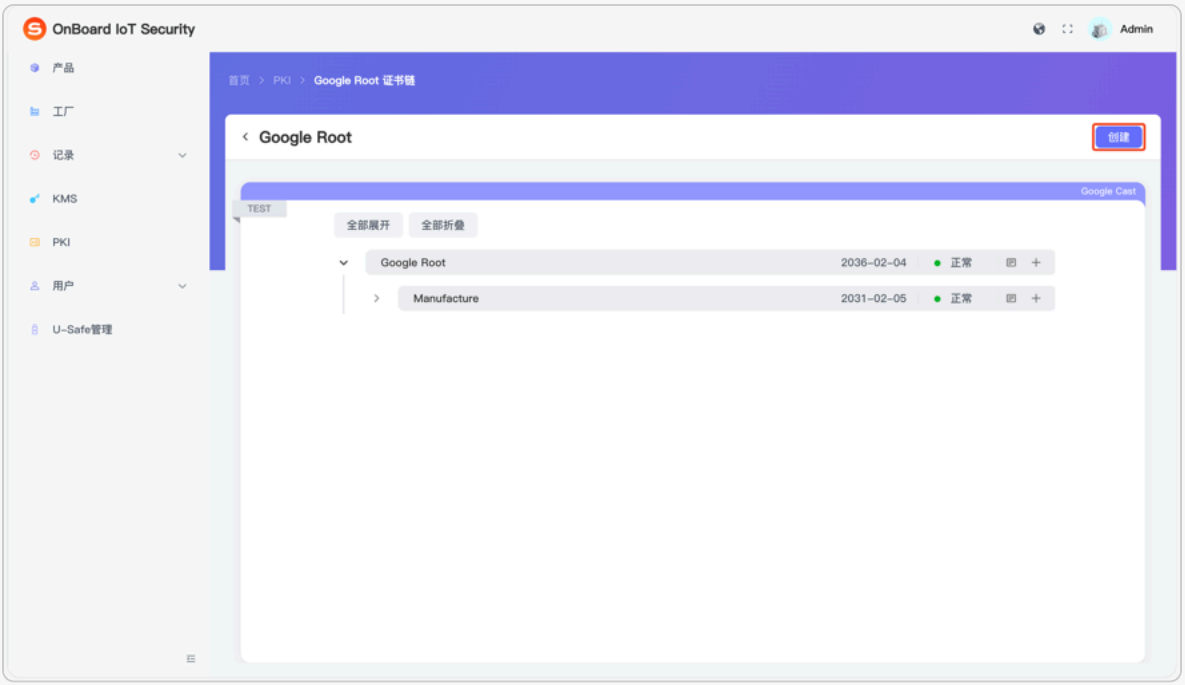
4. 签署完成后，在上传区域中点击添加 图标，选择并上传格式为 **.pem** 的 Manufacturer 证书，或者直接拖曳你的 Manufacturer 证书到上传区域。

5. 点击**提交**，配置完成后，Manufacturer 证书会变更为**正常**状态。

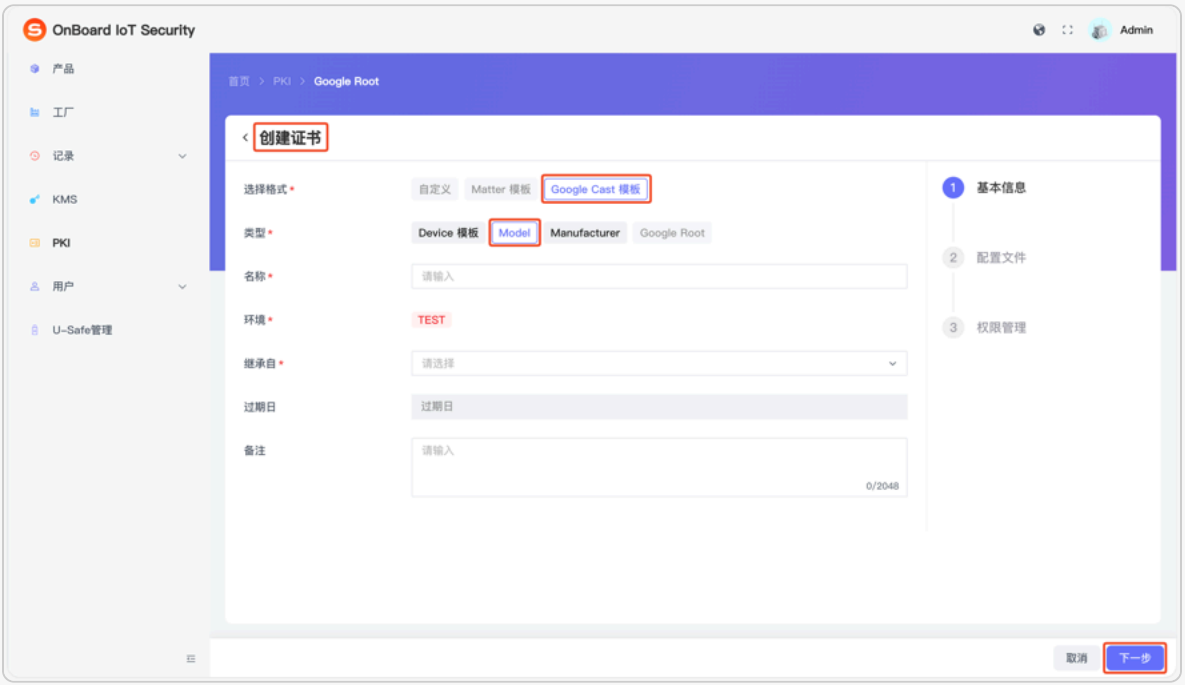
> **信息**
>
> 在创建基于 Google Cast 模板的 Model 证书之前，请确保已创建至少一个基于 Google Cast 模板的 Manufacturer 证书，以便在配置时进行选择并继承。

1. 在**证书链**页面，点击对应的证书链，然后点击**创建**。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



2. 在**创建证书**页面，格式选择 **Google Cast 模板**，类型选择 **Model** 并选择与输入基础信息。

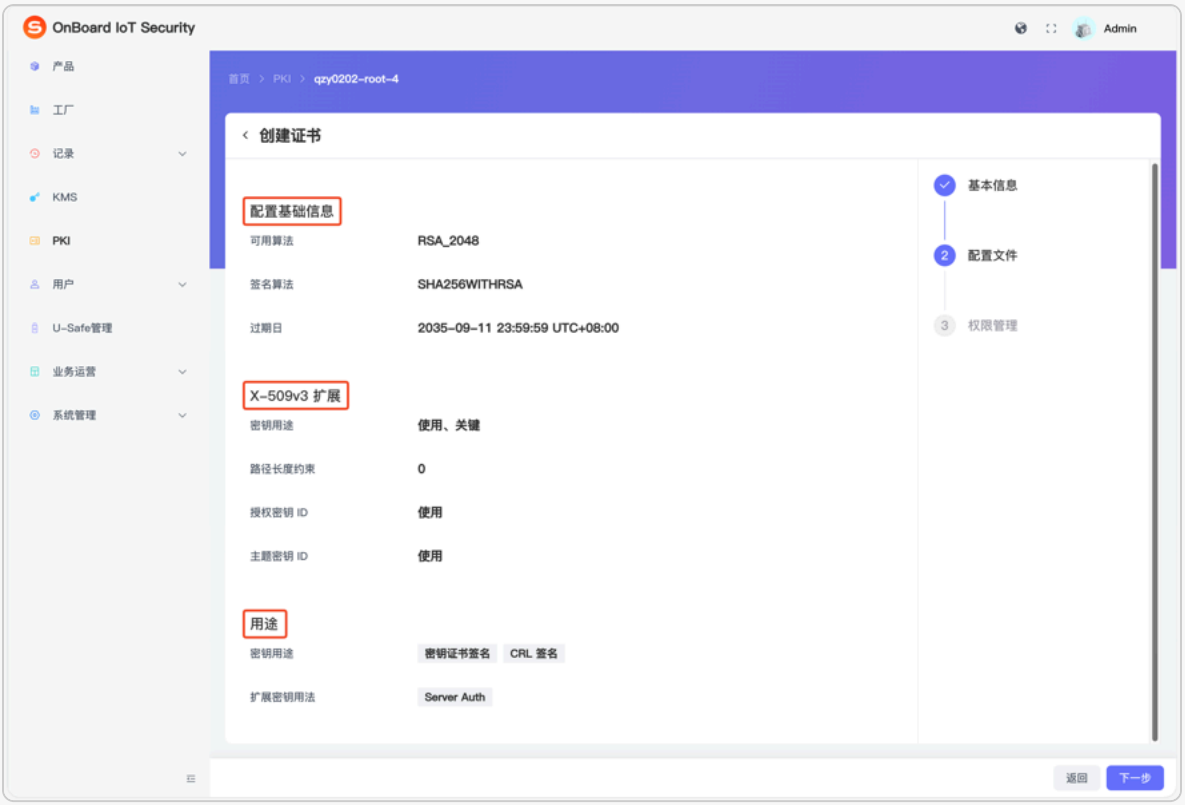


Model 证书基础信息说明：

信息	说明
名称	同一证书链中，同一层级的证书名称必须唯一。
环境	继承自 Google Root 证书的环境。
继承自	选择继承的 Manufacturer 证书。
过期日	设定 Model 证书的过期日。不可超过继承的 Manufacturer 证书的过期日。 **注释**：如果继承自自有 Manufacturer 证书，不需要设置过期日。
备注	用于补充说明 Model 证书相关的额外信息，如特殊要求和注意事项等。

3. 确认信息无误后，点击**下一步**并参考以下信息进行配置。

> **提示**
>
> 如果 Model 证书继承自自有 Manufacturer 证书，页面只显示**配置基础信息**区域。

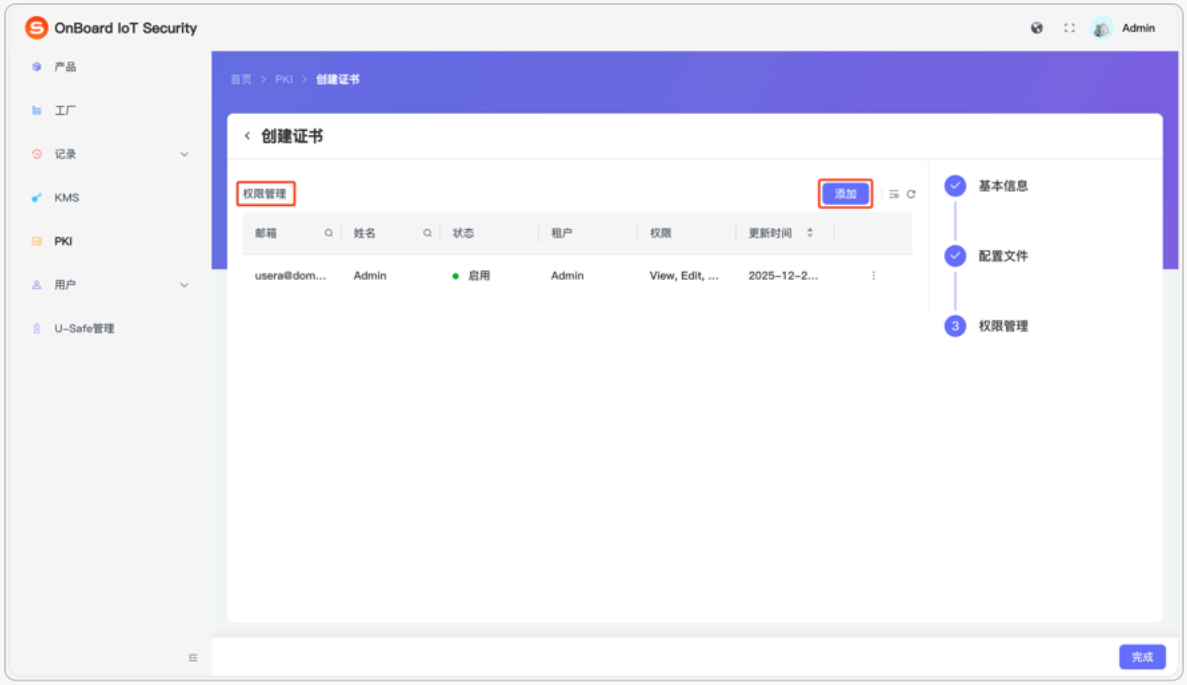


Model 证书配置信息说明：

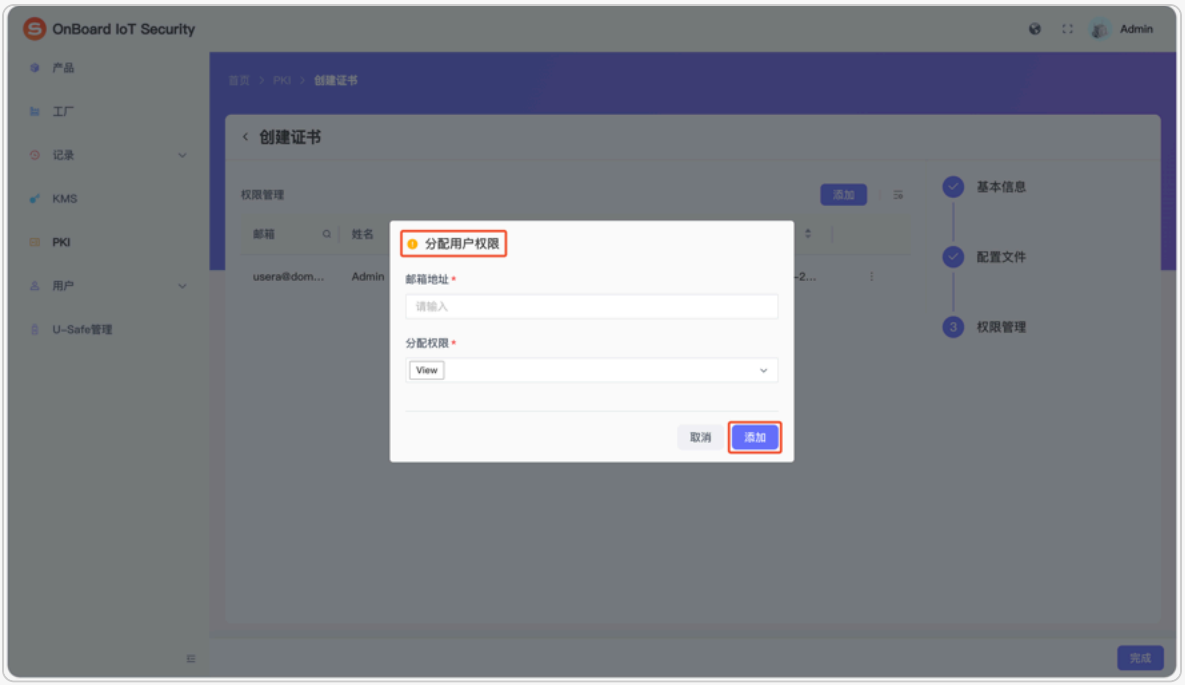
区域	信息	说明
配置基础信息	可用算法	根据证书内容自动解析，且无法修改。
	签名算法	根据证书内容自动解析，且无法修改。
	过期日	继承上一步在基础信息设置的过期日，且无法修改。
X-509v3 扩展	密钥用途	根据证书内容自动解析，且无法修改。
	路径长度约束	根据证书内容自动解析，且无法修改。

区域	信息	说明
	授权密钥 ID	根据证书内容自动解析，且无法修改。
	主题密钥 ID	根据证书内容自动解析，且无法修改。
用途	密钥用途	**密钥证书签名**和 **CRL 签名**等密钥用途。
	扩展密钥用法	**Server Auth**等扩展密钥用法。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。



5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Revoke，Grant 和 Delete）。然后，点击**添加**。



6. 点击**完成**，即可完成 Model 证书的创建。

> **提示**
>
> 创建基于 Google Cast 模板的 Model 证书时，你需要完成 CSR （证书签名请求）的配置。在CSR 配置完成之前，该证书将处于**待完成**状态，且无法正常使用。

基于 Google Cast 模板的 Model 证书的 CSR 配置步骤

> **提示**
>
> 创建基于 Google Cast 模板的 Model 证书时，参考以下步骤完成 CSR 配置。

1. 在**证书链**页面，点击对应的证书链，然后点击对应的 Model 证书，进入证书页面。
2. 在证书页面，点击提示中的链接以完成 CSR 的配置。

3. 在 **CSR 配置**页面，点击下载 图标来下载 CSR 文件。然后，使用你的 Root CA 对该文件进行签署，从而生成由 Root CA 签名的 Model 证书。



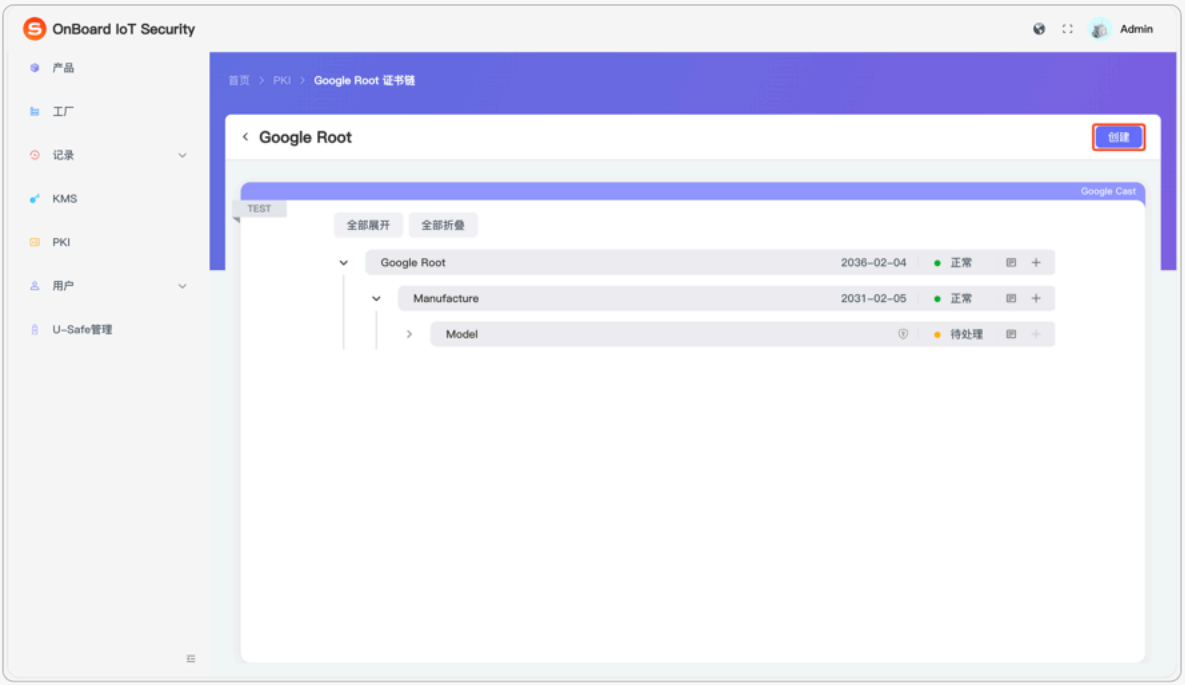
4. 签署完成后，在上传区域中点击添加 图标，选择并上传格式为 **.pem** 的 Model 证书，或者直接拖曳你的 Model 证书到上传区域。

5. 点击**提交**，配置完成后，Model证书会变更为**正常**状态。

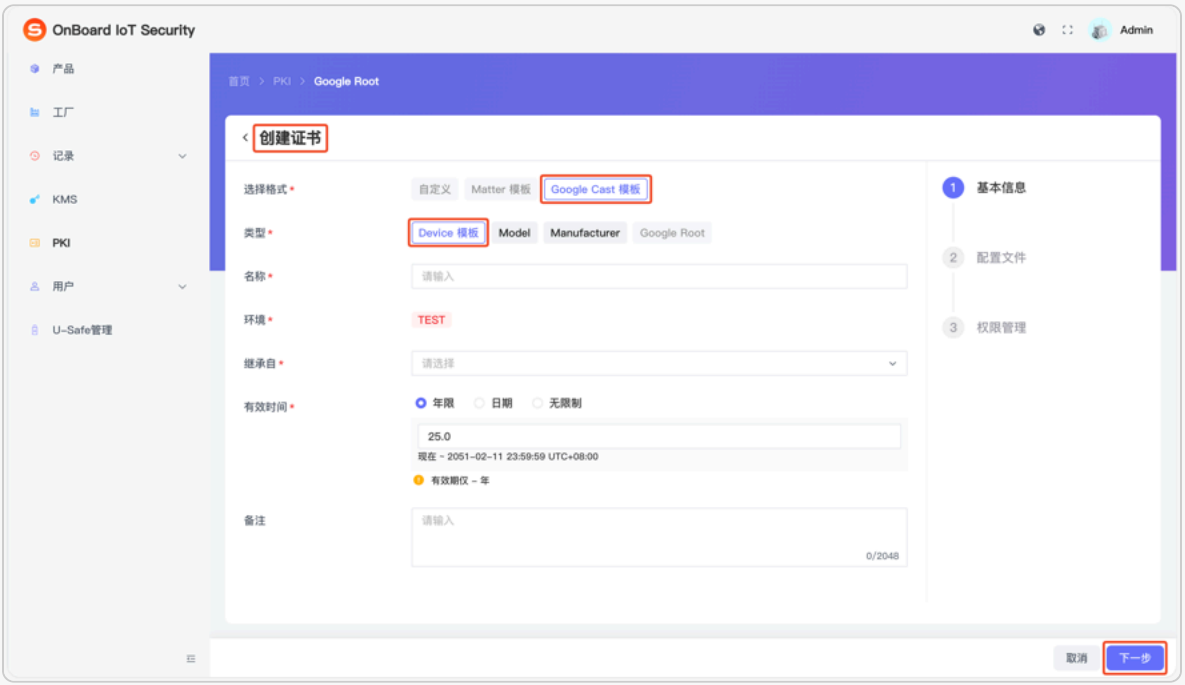
> **信息**
>
> 在创建 Device 模板之前，请确保已创建至少一个基于 Google Cast 模板的 Model 证书，以便在配置时进行选择并继承。

1. 在**证书链**页面，点击对应的证书链，然后点击**创建**。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



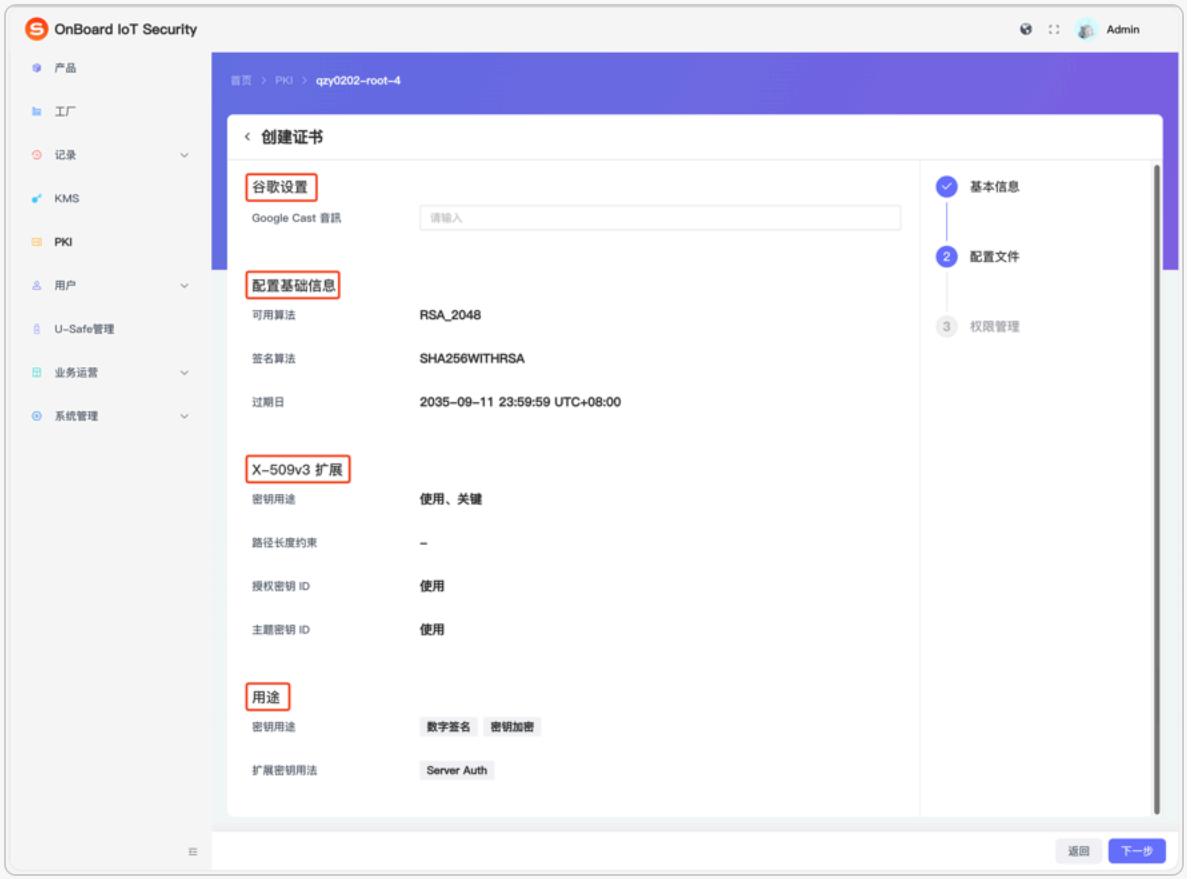
2. 在**创建证书**页面，格式选择 **Google Cast 模板**，类型选择 **Device 模板**并参考以下信息进行配置。



Device 模板基础信息说明：

信息	说明
名称	同一证书链中，同一层级的证书名称必须唯一。
环境	继承自 Google Root 证书的环境。
继承自	选择继承的 Model 证书。
有效时间	设定 Device 模板的有效时间，不可超过继承的 Model 证书的有效时间。
备注	用于补充说明 Device 模板相关的额外信息，如特殊要求和注意事项等。

3. 确认信息无误后，点击**下一步**并查看以下信息。

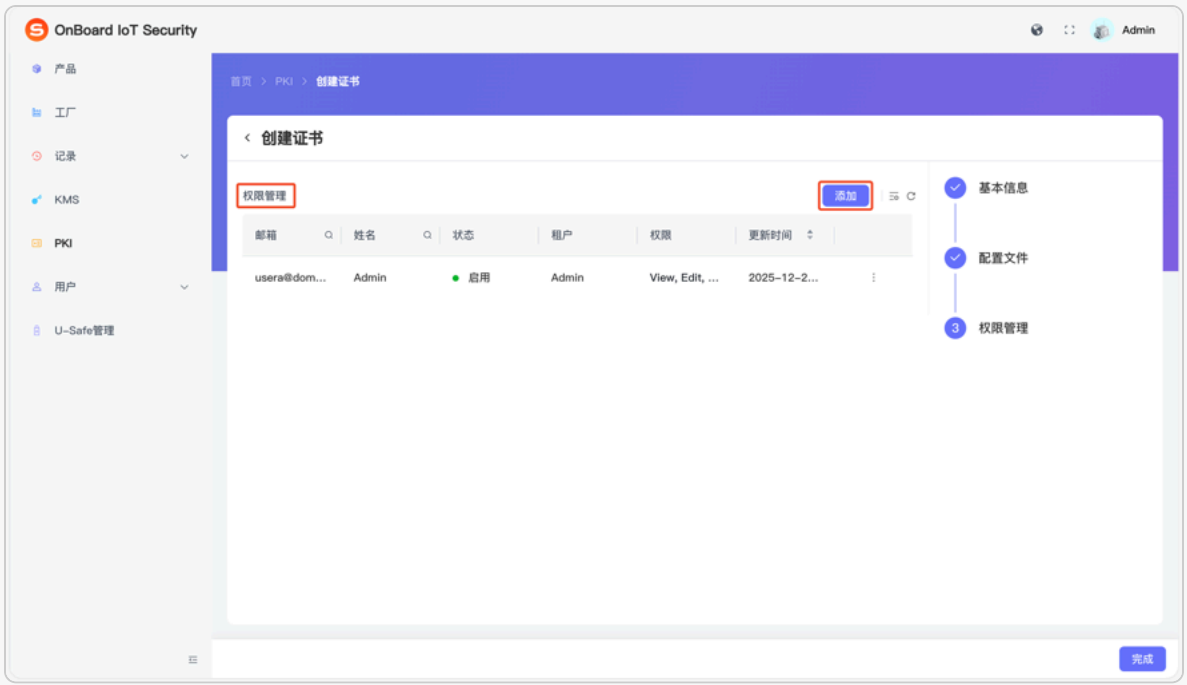


Device 模板配置信息说明：

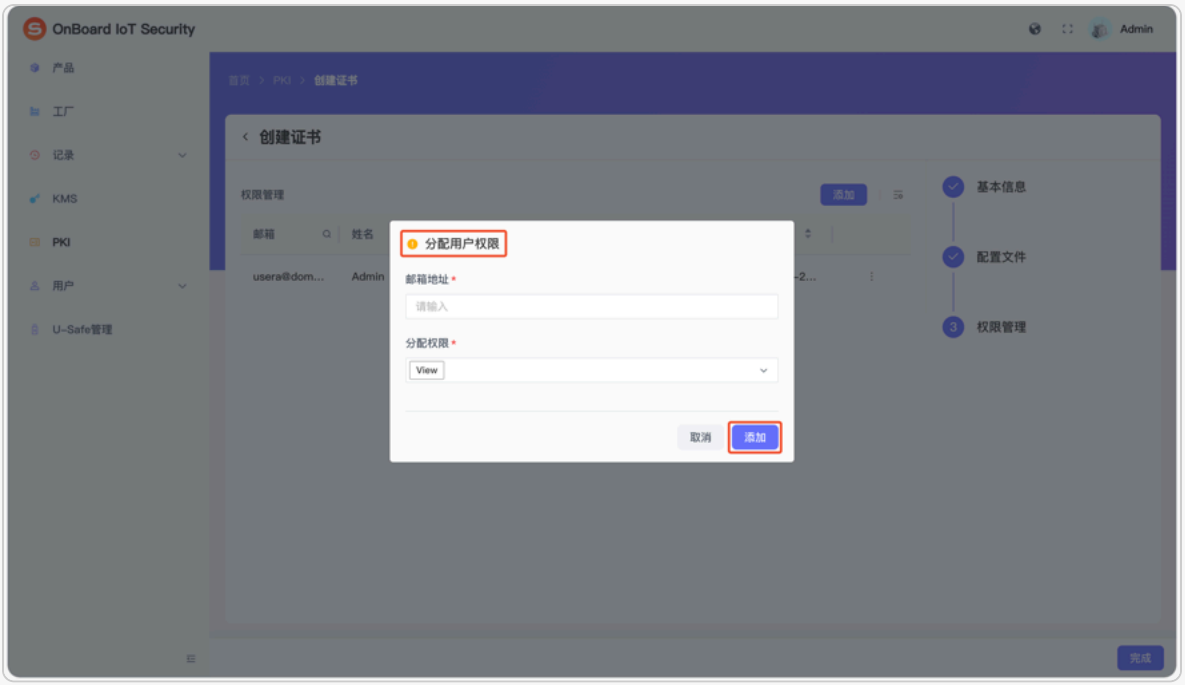
区域	信息	说明
谷歌设置	Google Cast 音讯 (可选)	证书扩展字段，例如：1.3.6.1.4.1.11129.2.5.2。
配置基础信息	可用算法	根据证书内容自动解析，且无法修改。
	签名算法	根据证书内容自动解析，且无法修改。
	过期日	继承上一步在基础信息设置的有效时间，且无法修改。

区域	信息	说明
X-509v3 扩展	密钥用途	根据证书内容自动解析，且无法修改。
	路径长度约束	根据证书内容自动解析，且无法修改。
	授权密钥 ID	根据证书内容自动解析，且无法修改。
	主题密钥 ID	根据证书内容自动解析，且无法修改。
用途	密钥用途	**数字签名**和**密钥加密**等密钥用途。
	扩展密钥用法	**Server Auth**等扩展密钥用法。

4. 确认信息无误后，点击**下一步**，在**权限管理**页面，点击**添加**。

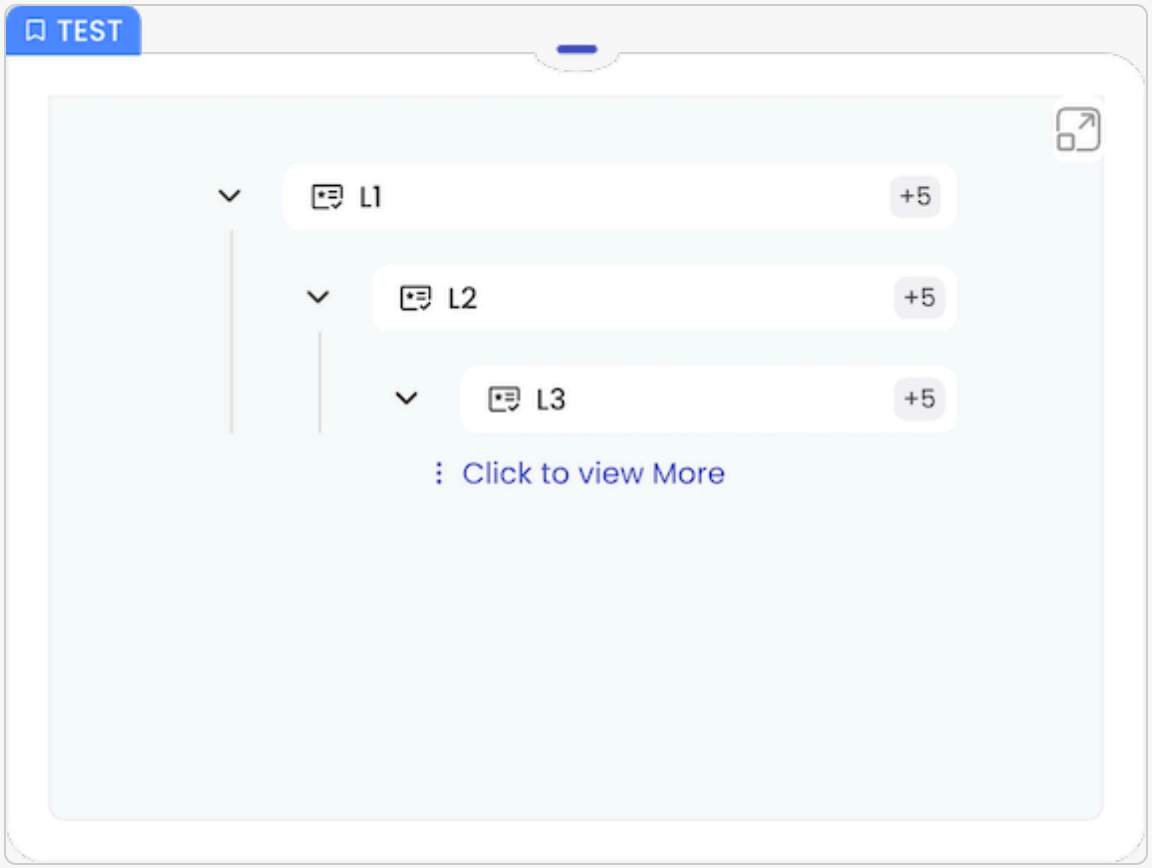


5. 在**分配用户权限**窗口，输入邮箱并分配对应的角色权限（View，Edit，Use，Grant 和 Delete）。然后，点击**添加**。



6. 点击**完成**，即可完成 Device 模板的创建。

证书链层级示例



层级	说明
L1	根证书颁发机构
L2	中间证书颁发机构
L3	终端实体证书

版本：v3.2.2

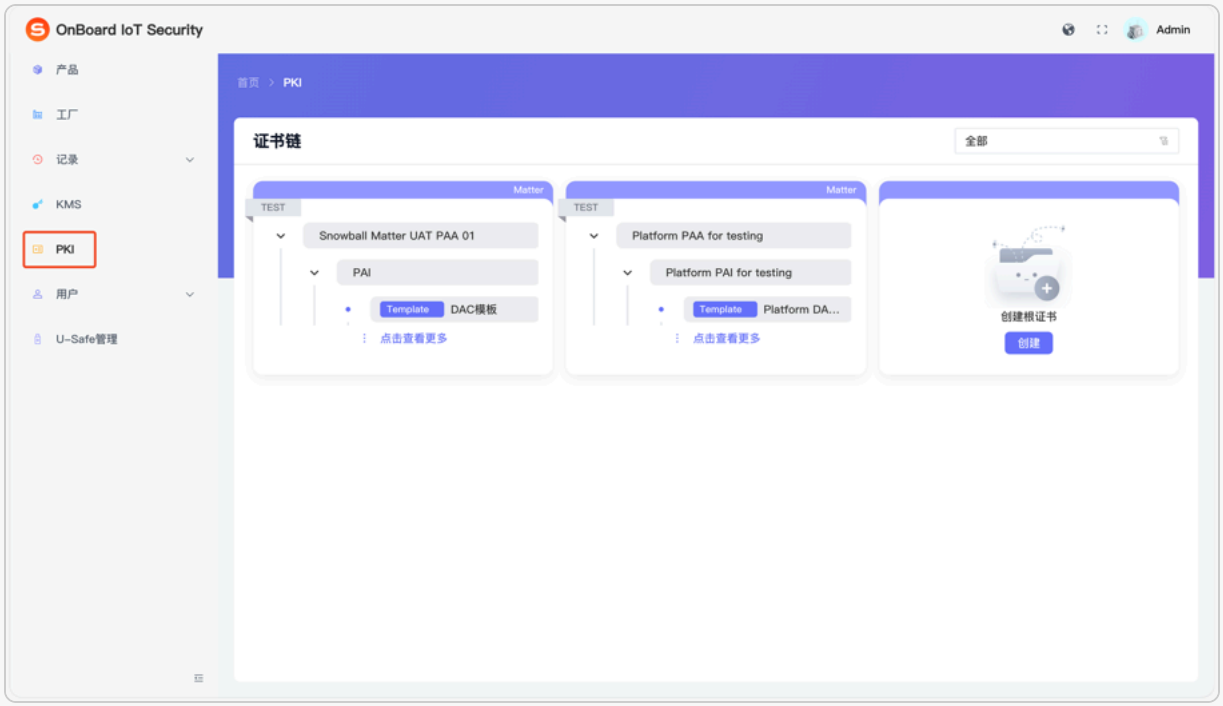
如何在 OnBoard™ IoT Security 中查看证书

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看证书的基础信息，配置文件和日志。

查看基础信息

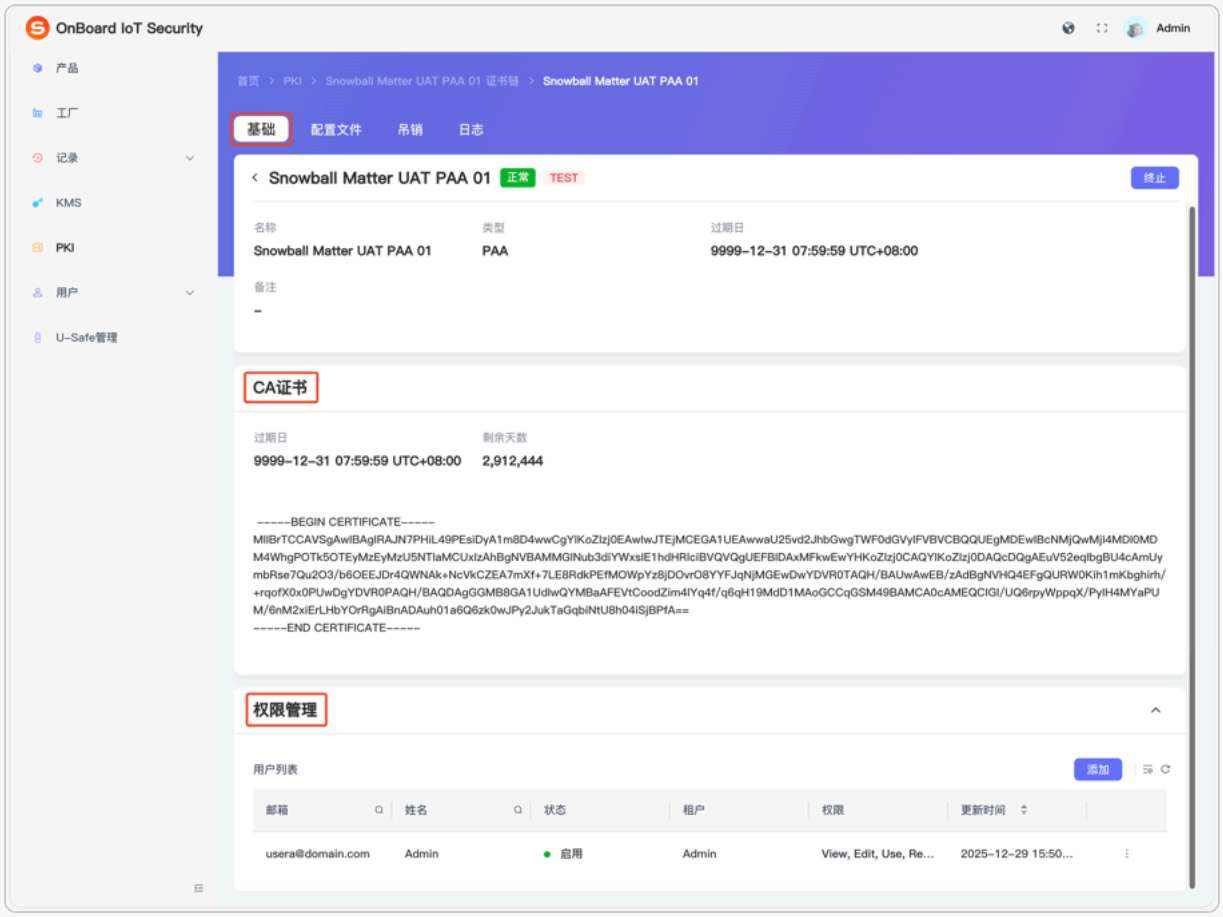
1. 登录 OBIS。
2. 点击左侧菜单中的 **PKI**，然后点击对应的证书链。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



3. 点击对应的证书名称，在证书页面，即可查看证书相关信息（**名称**，**类型**和**过期日**等）。

> **提示**
>
> 在 **CA 证书**区域，可通过解析编码提取出证书的颁发者信息，有效期，公钥等关键证书属性。
>
> 在**权限管理**区域，可以管理该证书的权限。



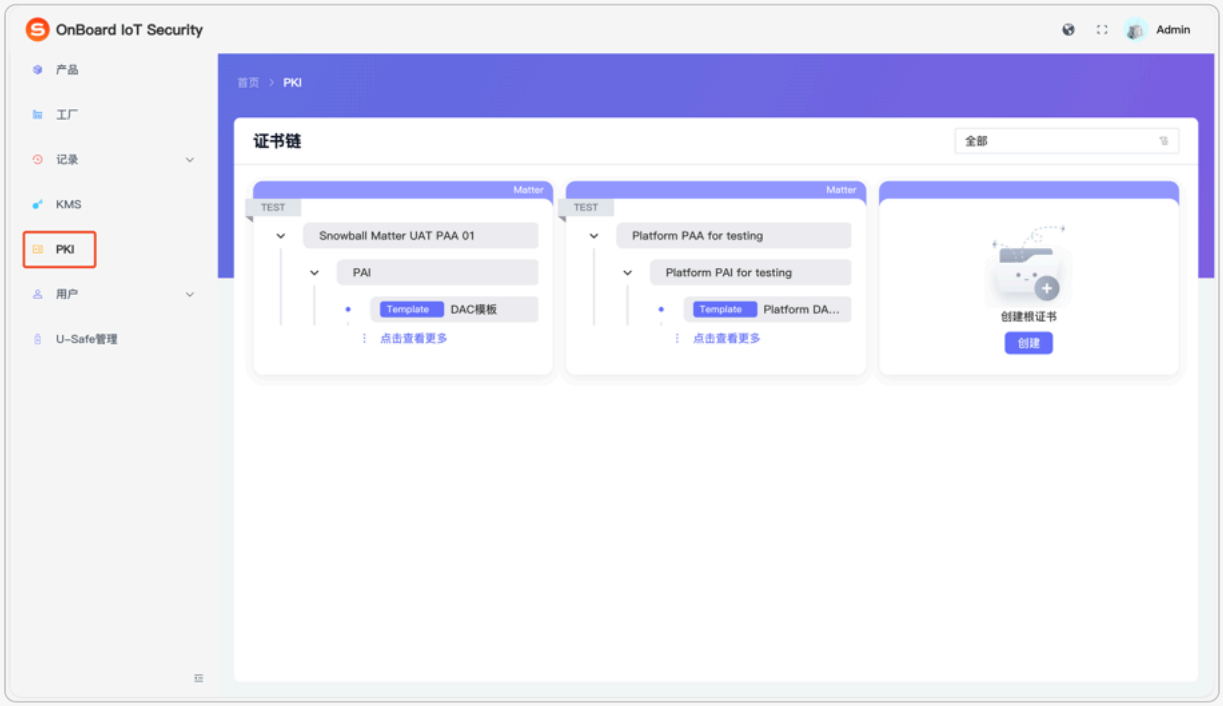
查看配置文件

> **提示**
>
> 通过证书配置文件，可以查看证书的相关配置。

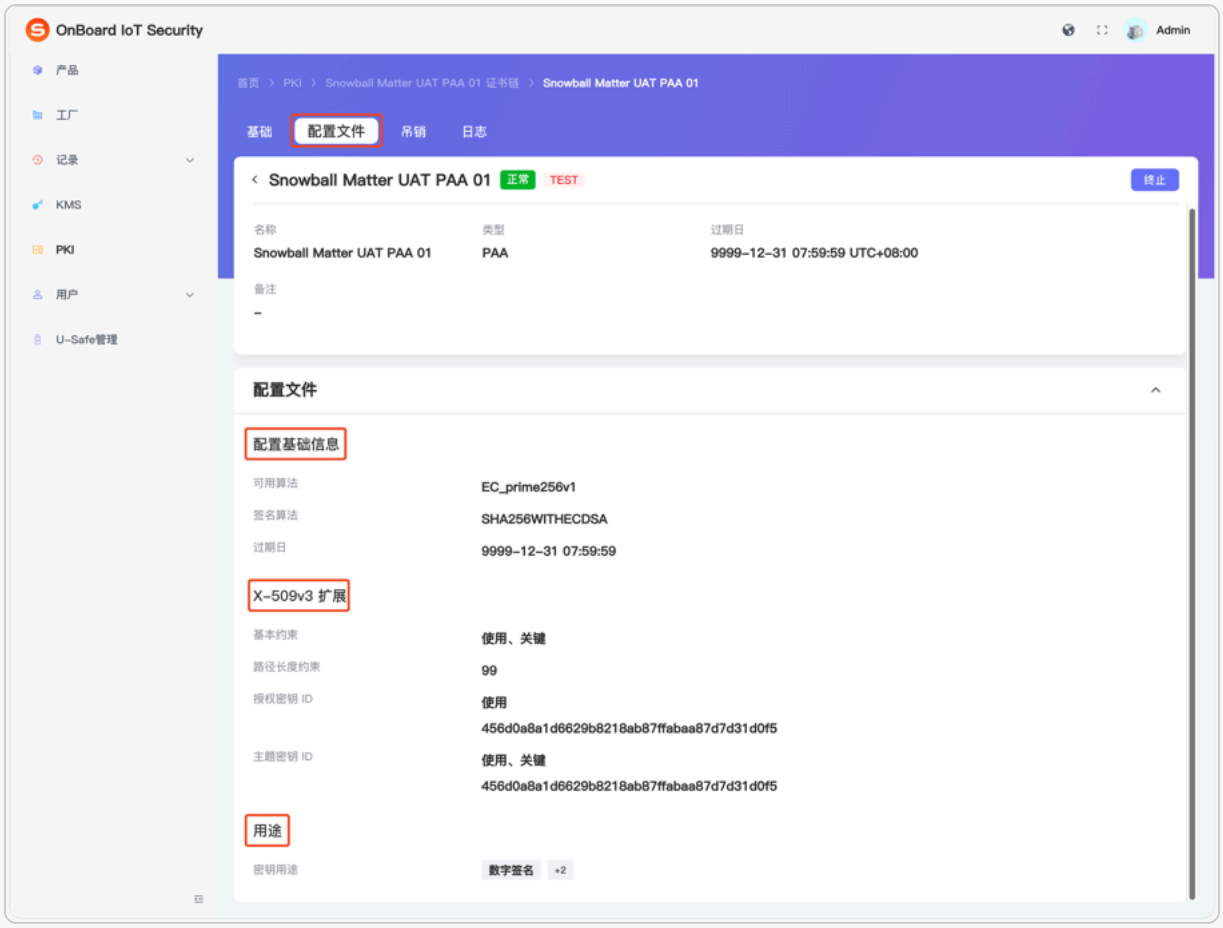
1. 登录 OBIS。

2. 点击左侧菜单中的 **PKI**，然后点击对应的证书链。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



3. 点击对应的证书名称，在证书页面的**配置文件**标签页，即可查看证书的相关配置（**配置基础信息**，**X-509v3 扩展**和**用途**等）。



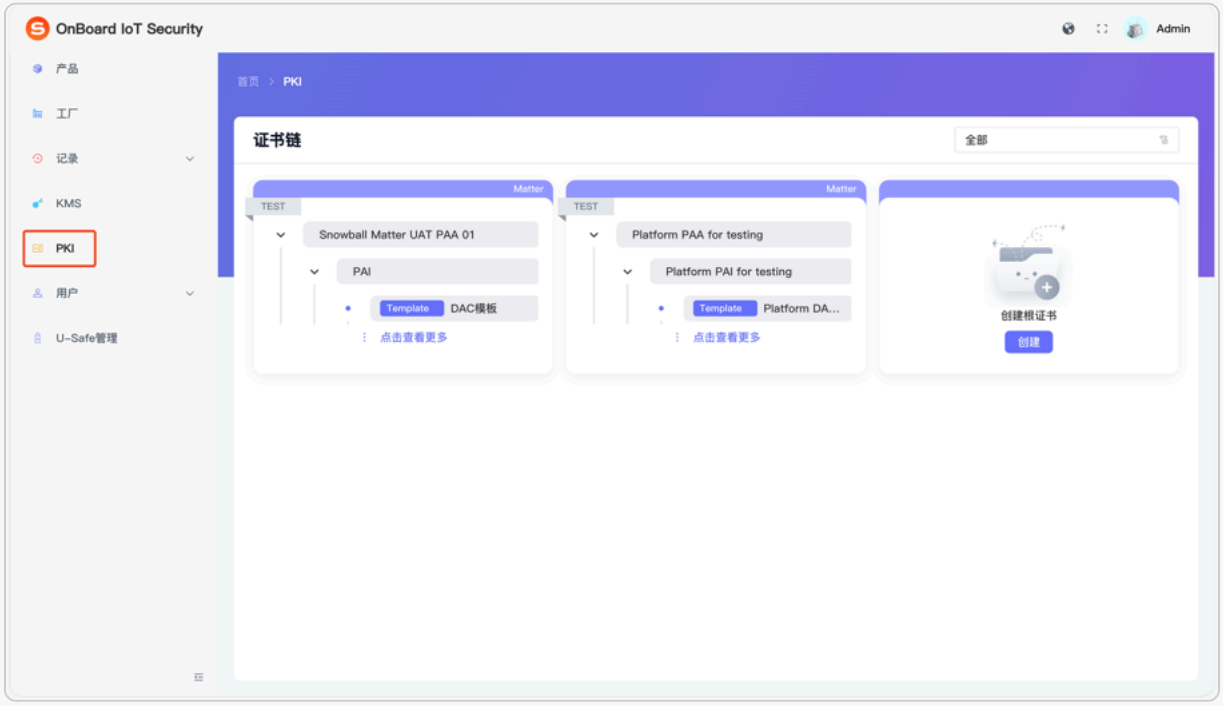
查看日志

> **提示**
>
> 通过证书日志，可以查看证书的历史操作记录。

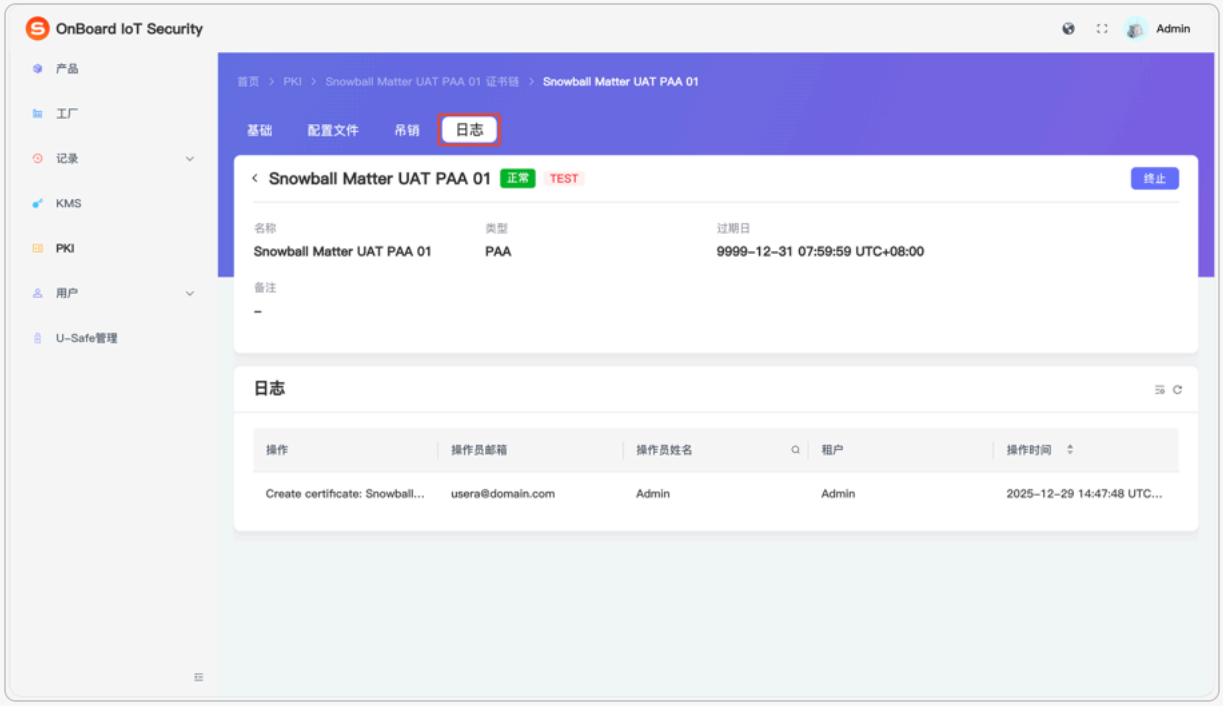
1. 登录 OBIS。

2. 点击左侧菜单中的 **PKI**，然后点击对应的证书链。

> **提示**
>
> 密钥 图标表示该证书包含私钥。



3. 点击对应的证书名称，在证书页面的**日志**标签页，即可查看证书的历史记录（**操作**和**操作员邮箱**等）。


OnBoard IoT Security
Admin
产品
工厂
记录
KMS
PKI
用户
U-Safe管理
首页 > PKI > Snowball Matter UAT PAA 01 证书链 > Snowball Matter UAT PAA 01
基础
配置文件
吊销
日志
Snowball Matter UAT PAA 01
正常
TEST
终止
名称
Snowball Matter UAT PAA 01
类型
PAA
过期日
9999-12-31 07:59:59 UTC+08:00
备注
-
日志
操作
操作员邮箱
操作员姓名
租户
操作时间
Create certificate: Snowball...
usera@domain.com
Admin
Admin
2025-12-29 14:47:48 UTC...

版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理证书

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作：

- 管理证书权限
- 吊销证书
- 删除证书

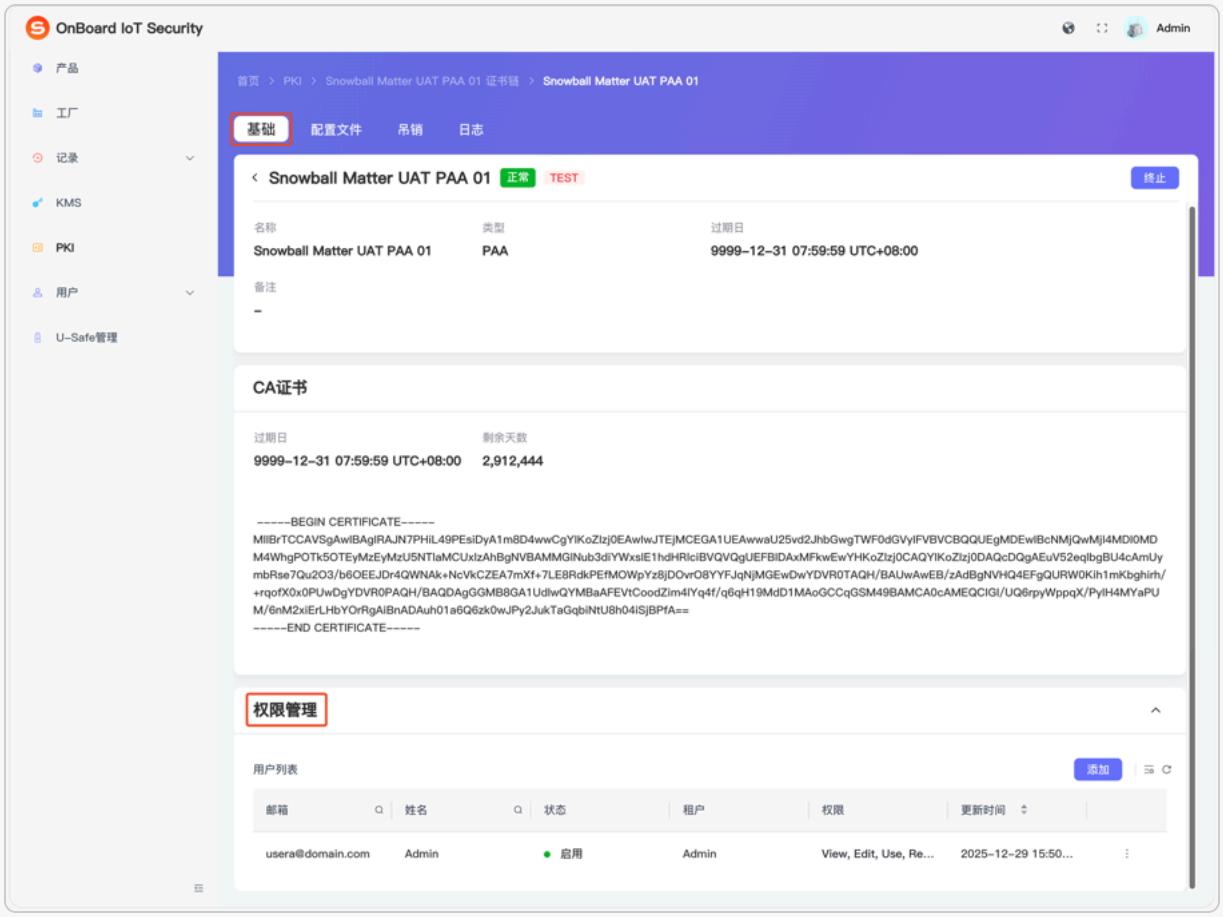
前提条件

在开始之前，请确保你已获得 System Manager 或 Resource Manager 的 OBIS 角色权限，或拥有对应证书的操作权限。

管理证书权限

通过以下步骤，可以调整并设定用户对证书的权限（View，Edit，Use，Revoke，Grant 和 Delete）：

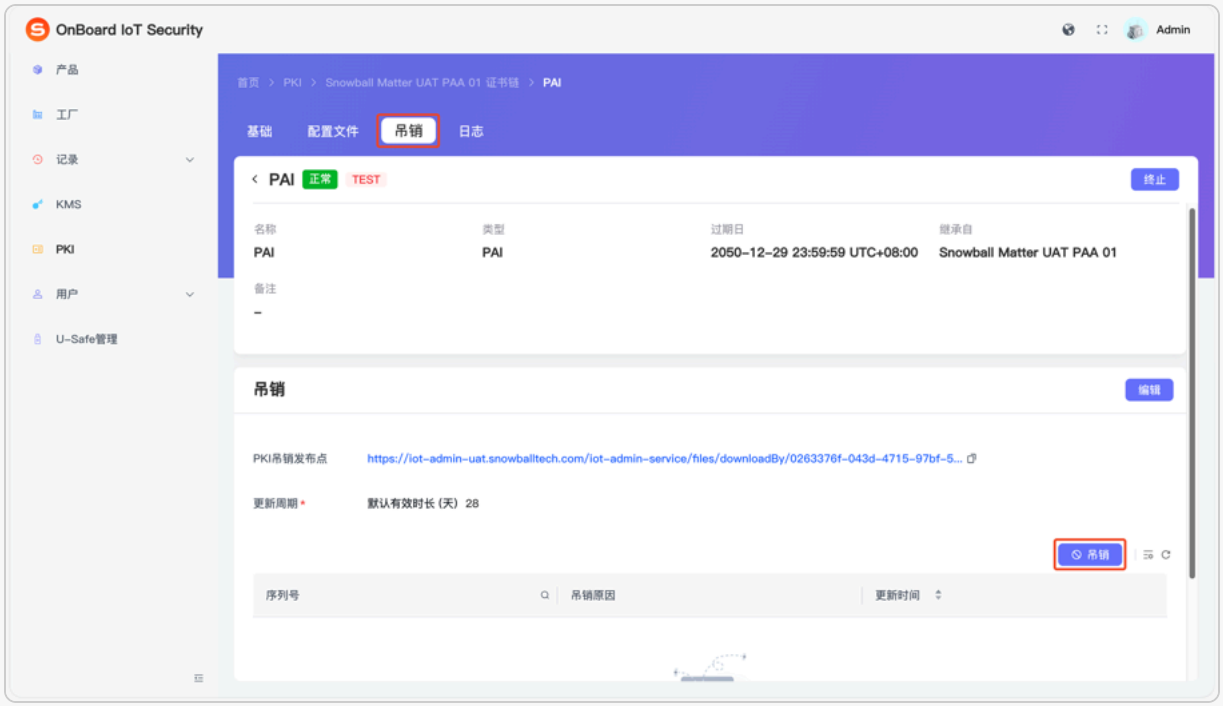
1. 登录 OBIS。
2. 点击左侧菜单中的 **PKI**，然后点击对应的证书链。
3. 点击对应证书名称，在证书页面的**权限管理**区域，可以管理该证书的权限。



吊销证书

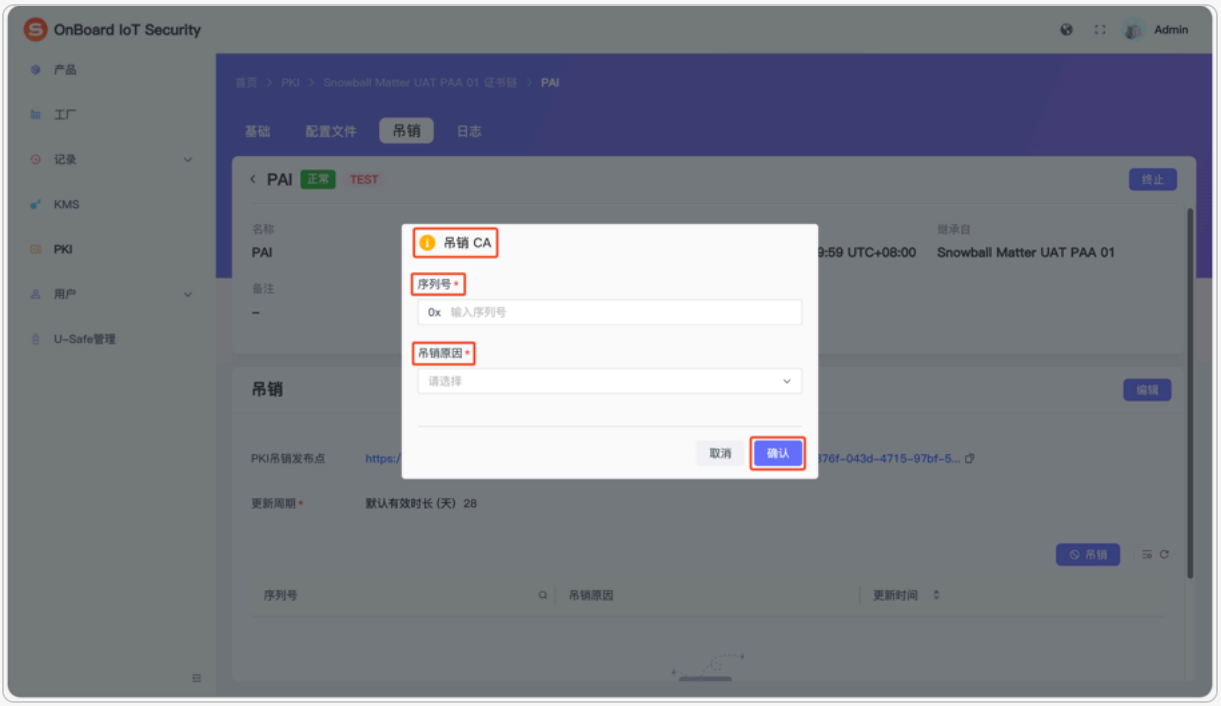
> **注意**
>
> 当证书被吊销后，对应的设备将无法继续执行配网操作。

1. 登录 OBIS。
2. 点击左侧菜单中的 **PKI**，然后点击对应的证书链。
3. 点击对应的证书名称，在证书页面的**吊销**标签页，点击**吊销**。



4. 在**吊销 CA** 窗口，输入**序列号**并选择**吊销原因**，然后点击**确认**，即可吊销该证书。

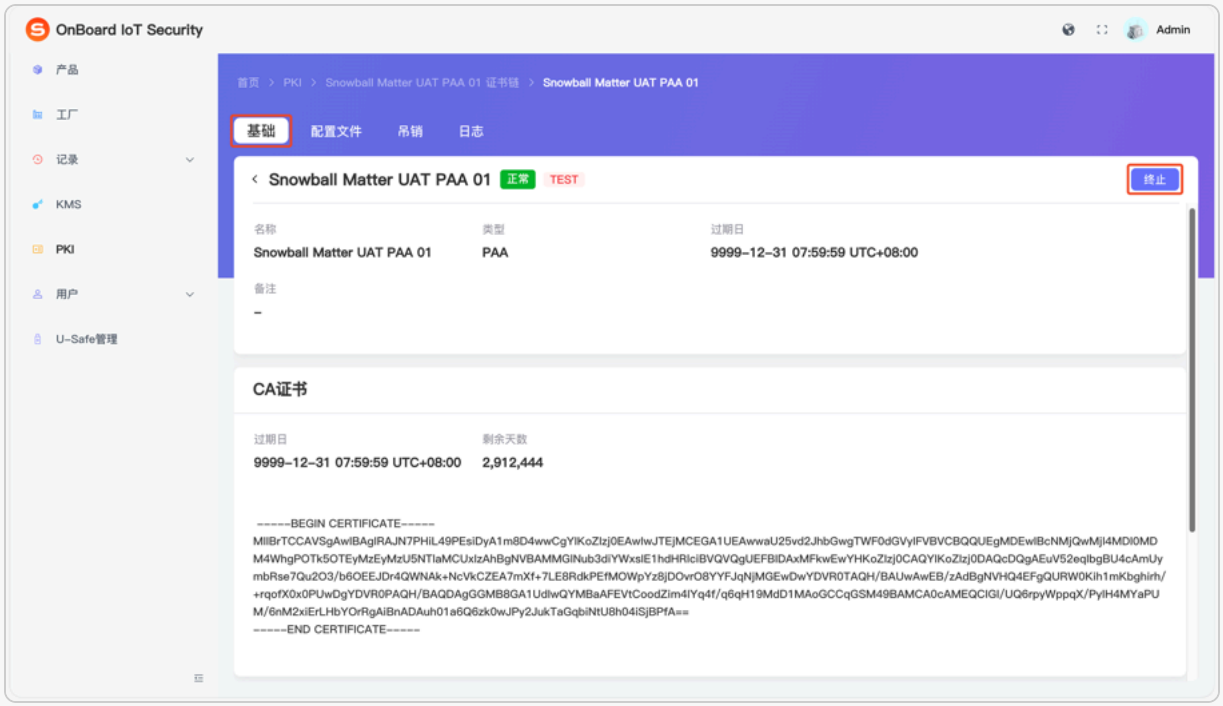
> **提示**
>
> 序列号可以在设备页面中查看。



删除证书

> **信息**
>
> 在删除证书之前，请确保：
>
> - 已删除该证书下的所有子证书。如果该证书为父证书且子证书未删除，系统将提示"证书无法删除。它包含以下子证书"并列出子证书列表。
> - 证书没有被资源引用。如果已被资源引用，系统将提示"无法删除证书，证书已被以下列表引用"并列出引用的资源列表。

1. 登录 OBIS。
2. 点击左侧菜单中的 **PKI**，然后点击对应的证书链。
3. 点击对应证书名称，在证书页面，点击**终止**。



4. 在**第二次确认**窗口，输入窗口下方提示的证书代码，点击**确认**，即可删除该证书。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建账号

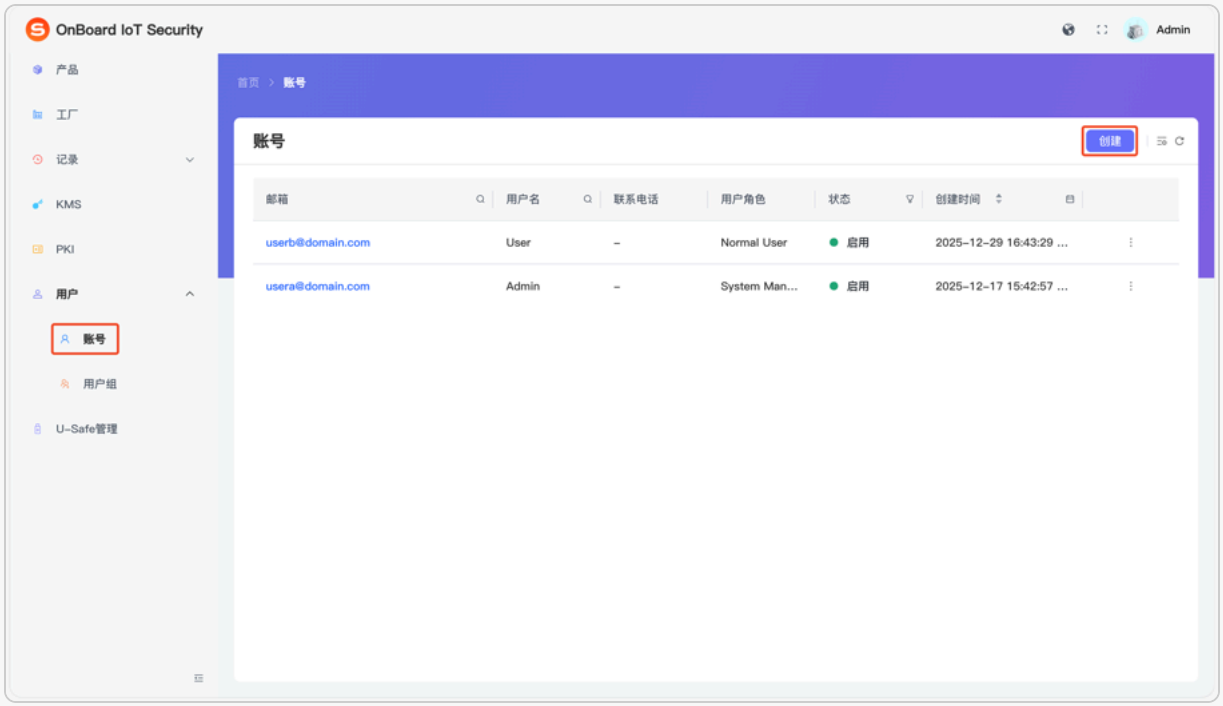
本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中创建账号从而添加用户以访问和操作 OBIS。

前提条件

在开始之前，请确保你已获得 System Manager 的 OBIS 角色权限。

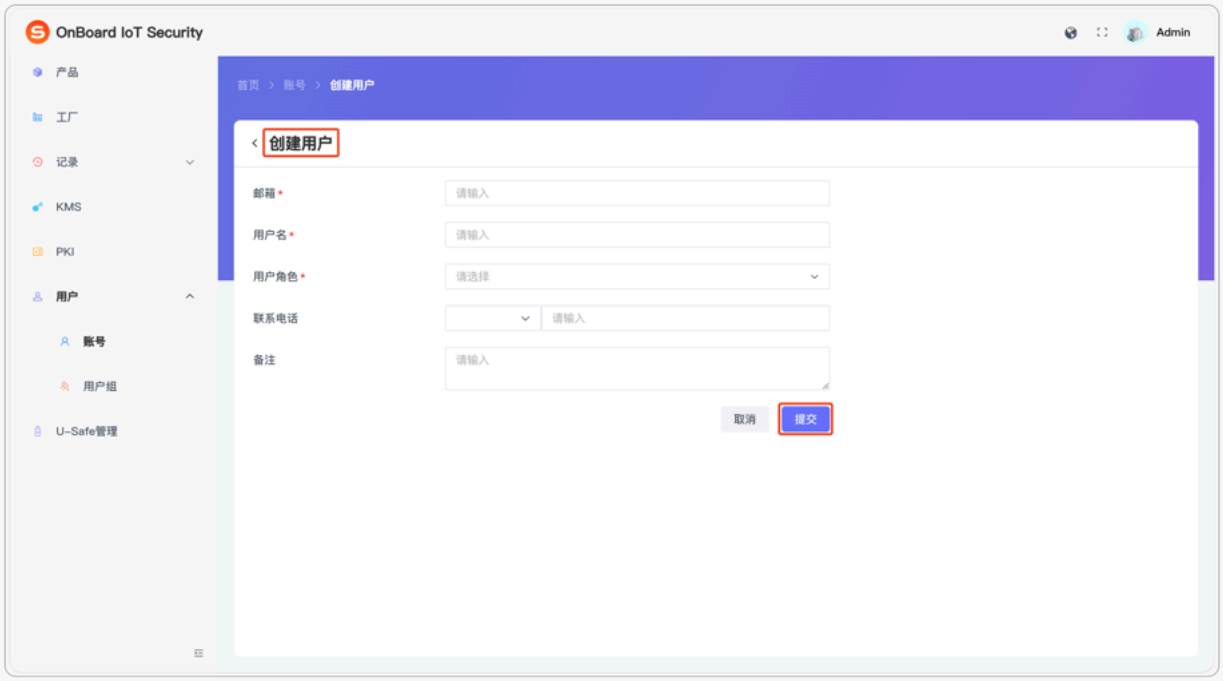
操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **账号**，然后点击**创建**。



3. 在**创建用户**页面，选择并输入用户信息。

> **提示**
>
> 更多信息请参考角色分类与权限说明。



用户信息说明：

信息	是否必填	说明
邮箱	是	用户邮箱，用来接收通知和验证信息，需真实有效且唯一。 **注释**：账号创建后，邮箱不可更改。
用户名	是	用户名可不唯一。

信息	是否必填	说明
用户角色	是	一个用户可以被分配多个角色。用户角色决定了用户的访问和操作权限，更多信息请参考角色分类与权限说明。
联系电话	否	提供除邮箱外的便捷联系途径，方便沟通。
备注	否	用于补充说明用户相关的额外信息，如特殊要求，注意事项等。

4. 点击**提交**，即可完成账号的创建。

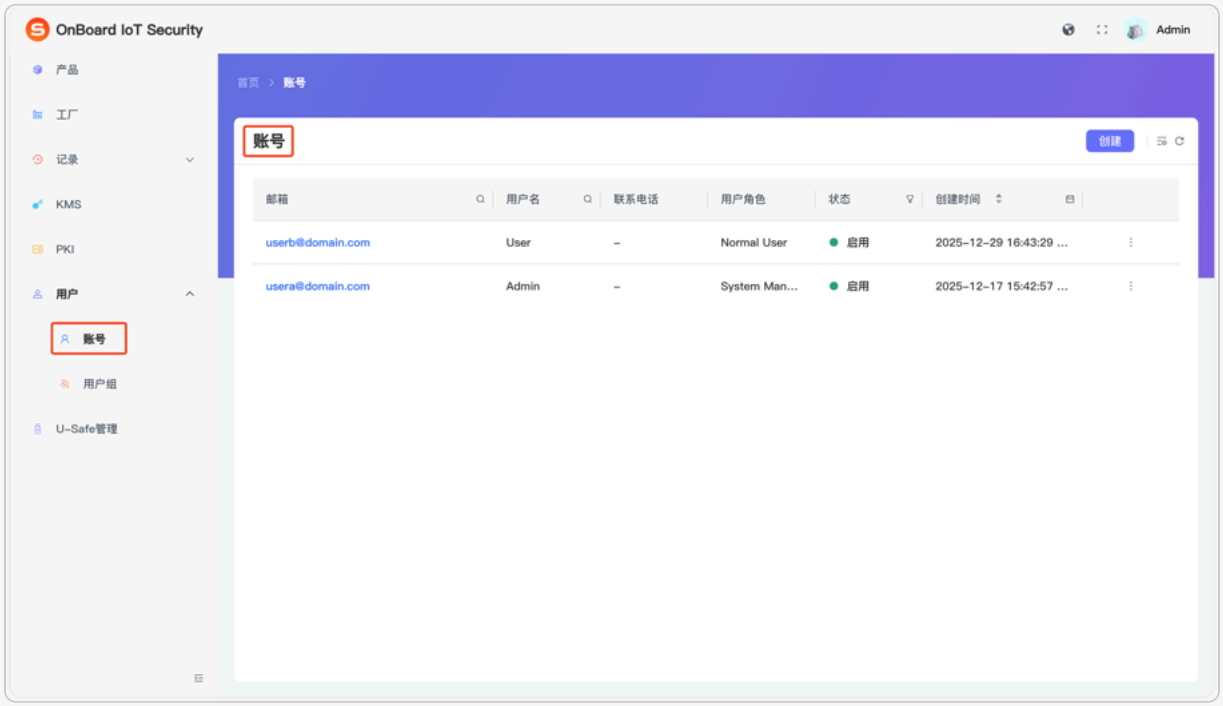
版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看账号

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看账号信息，AK（Access Key）和 SK（Secret Key）。

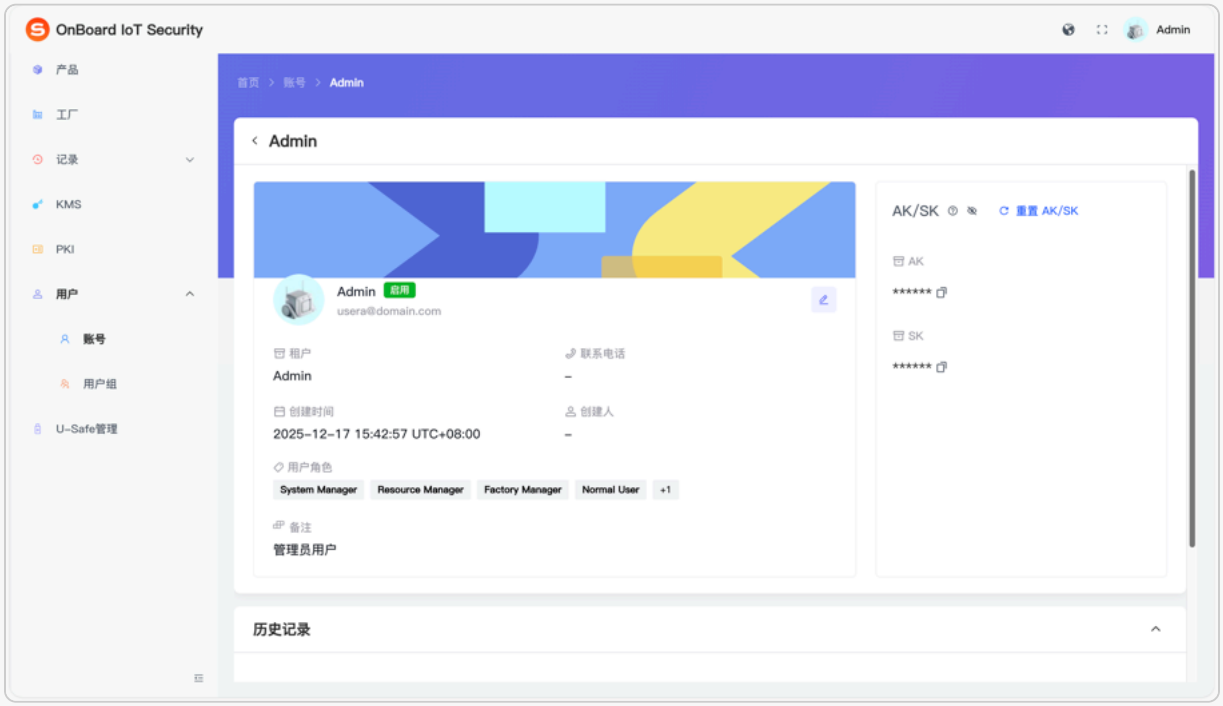
操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **账号**，进入**账号**页面。
3. 在**账号**页面，点击对应的邮箱。



4. 在用户页面，即可查看账号信息（**租户**，**创建时间**，**创建人**和**历史记录**等）。

> **提示**
>
> 在 **AK/SK** 区域可查看或重置 AK 或 SK。 AK 和 SK 可用于身份验证与接口调用。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理账号

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作从而调整用户对 OBIS 的访问与操作权限：

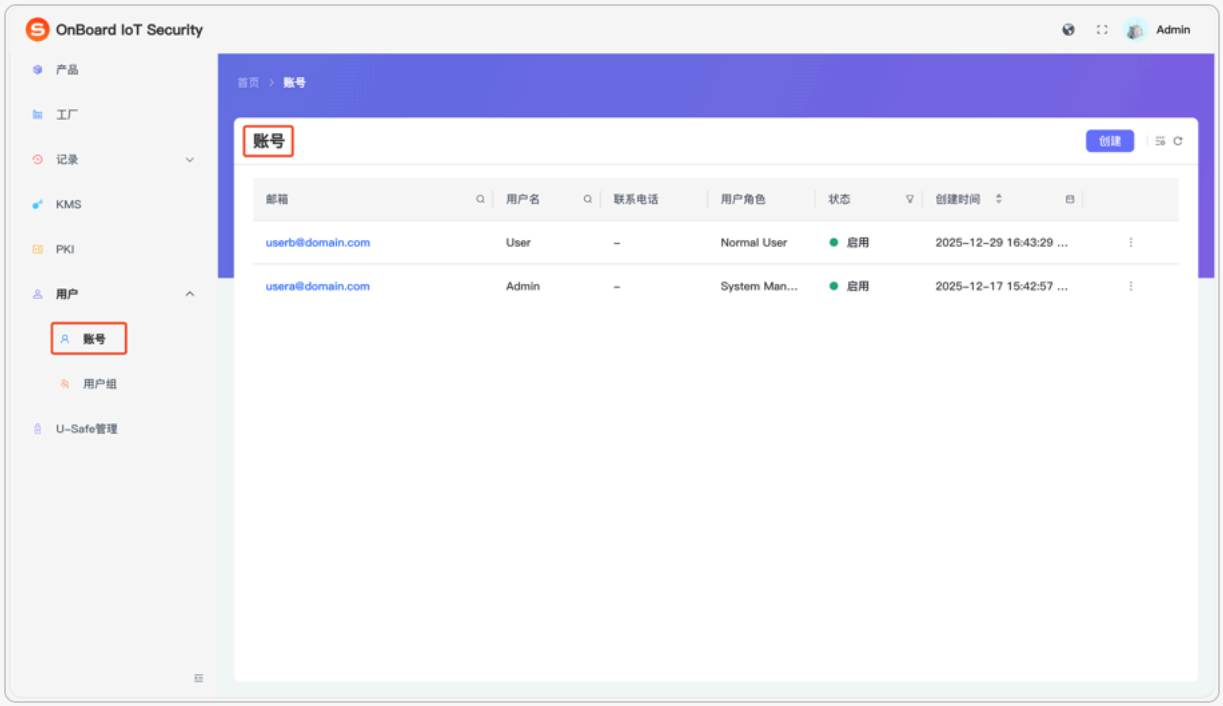
- 编辑账号
- 锁定或解锁账号
- 删除账号

前提条件

在开始之前，请确保你已获得 System Manager 的 OBIS 角色权限。

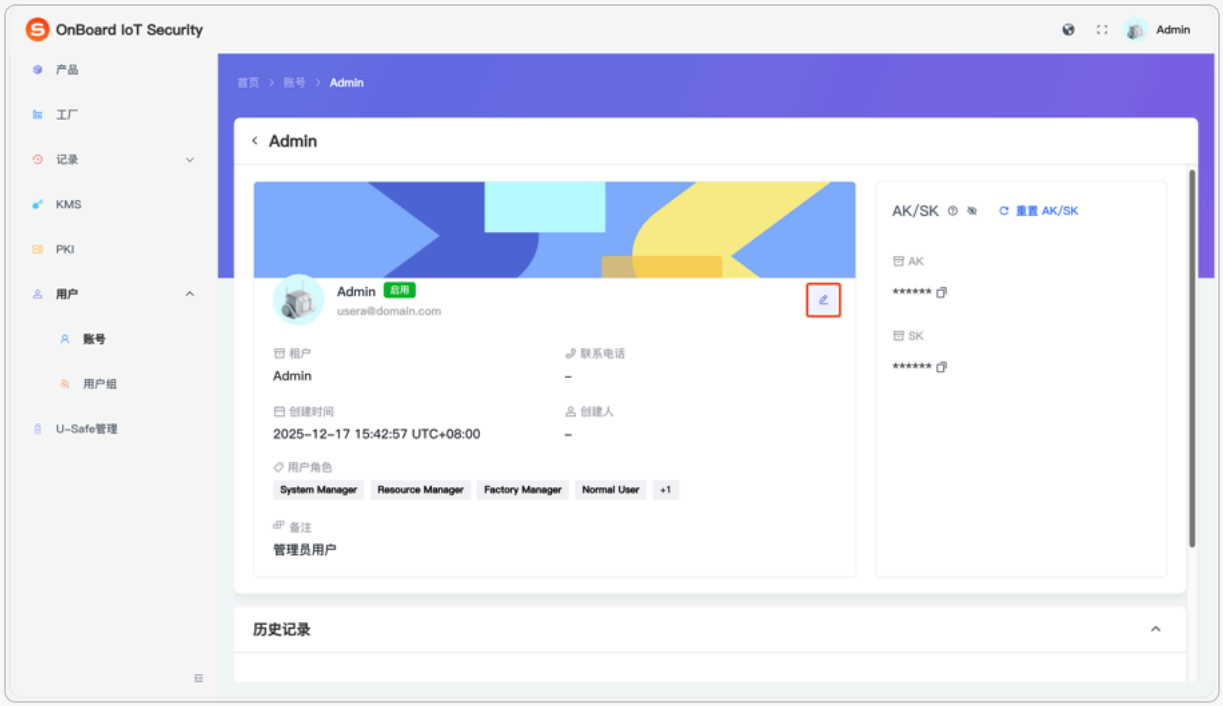
编辑账号

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **账号**，进入**账号**页面。
3. 在**账号**页面，点击对应的邮箱。



4. 在用户页面，点击编辑图标即可更改用户信息（**用户名**，**用户角色**，**联系电话**和**备注**）。

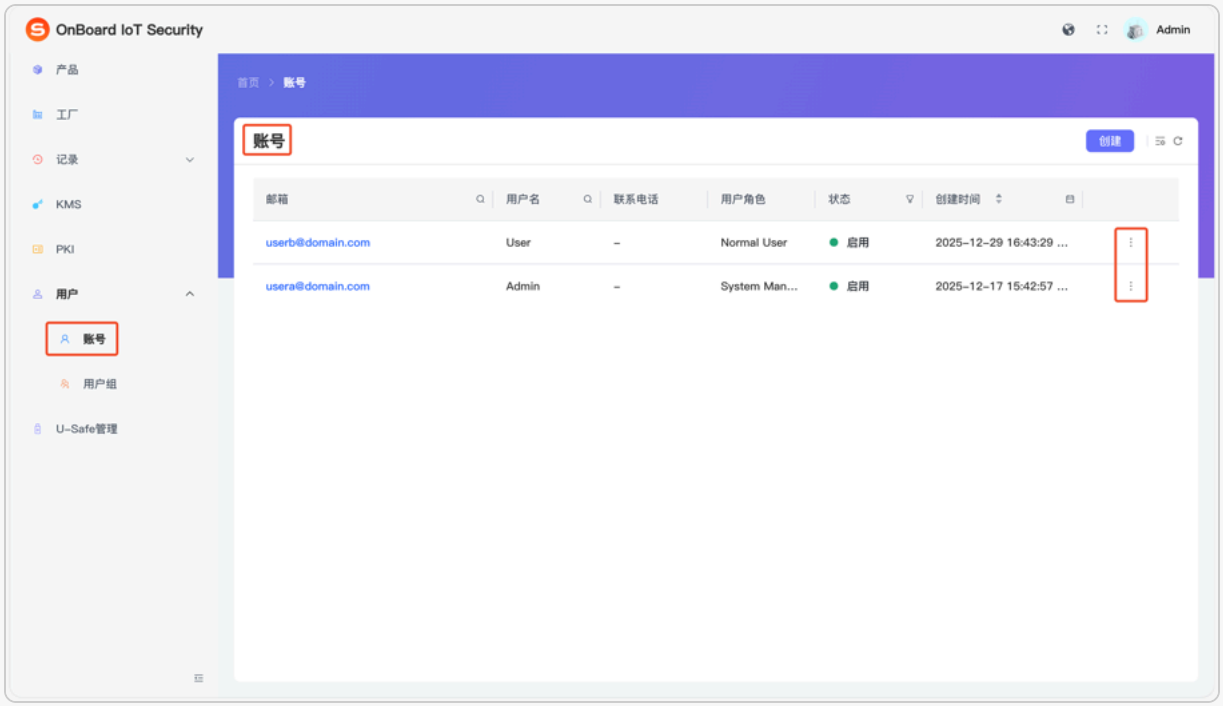
> **提示**
>
> 一个用户可以被分配多个角色。用户角色决定了用户的访问和操作权限，更多信息请参考角色分类与权限说明。



锁定或解锁账号

> **提示**
>
> 锁定账号以禁止其登录和使用。锁定后，状态会变更为**禁用**。

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **账号**，进入**账号**页面。
3. 在**账号**页面，点击对应账号的选项 ⋮ 图标并选择**锁定**或**解锁**。

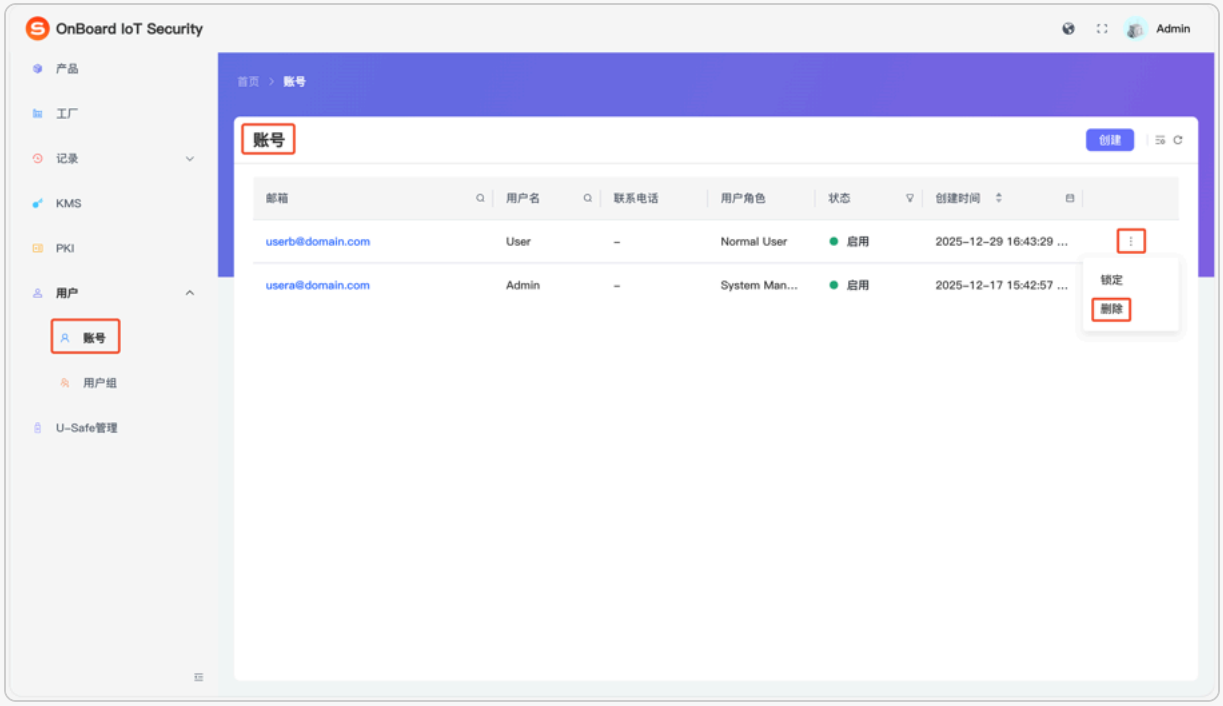


4. 在弹出的窗口中点击**确认**，即可锁定或解锁该账号。

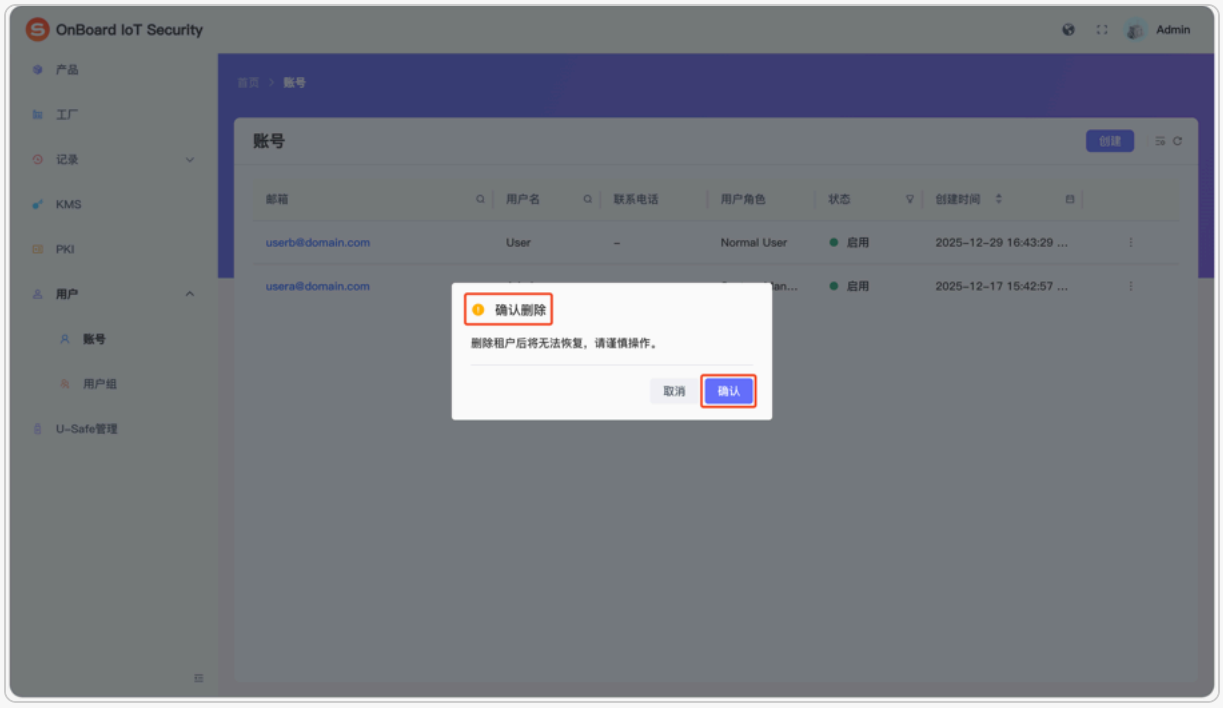
删除账号

> **注意**
>
> 删除账号后将无法恢复，请谨慎操作。

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **账号**，进入**账号**页面。
3. 在**账号**页面，点击对应账号的选项 ⋮ 图标并选择**删除**。



4. 在**确认删除**窗口中点击**确认**，即可删除该账号。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中创建用户组

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中创建用户组从而统一管理用户。

前提条件

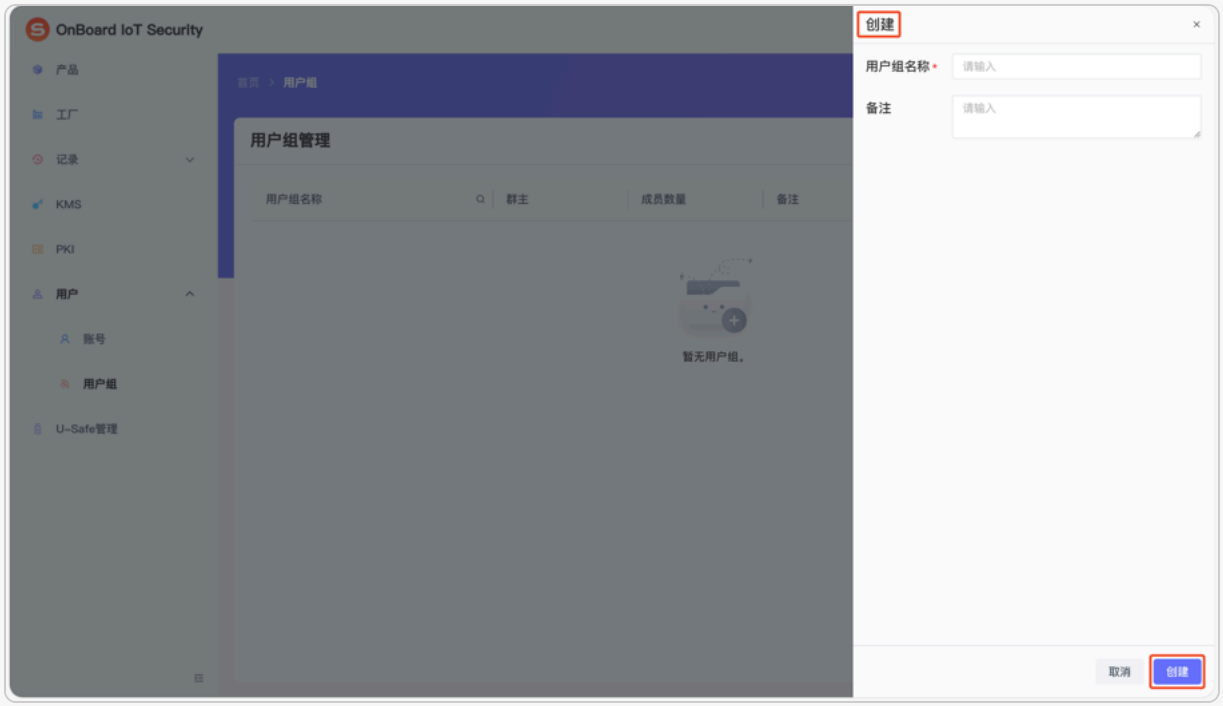
在开始之前，请确保你已获得 System Manager 的 OBIS 角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，然后点击**创建**。



3. 在**创建**面板，输入用户组信息。



用户组信息说明：

信息	是否必填	说明
用户组名称	是	用户组名称必须唯一。
备注	否	用于补充说明用户组相关的额外信息，如特殊要求，注意事项等。

4. 点击**创建**，即可完成用户组的创建。

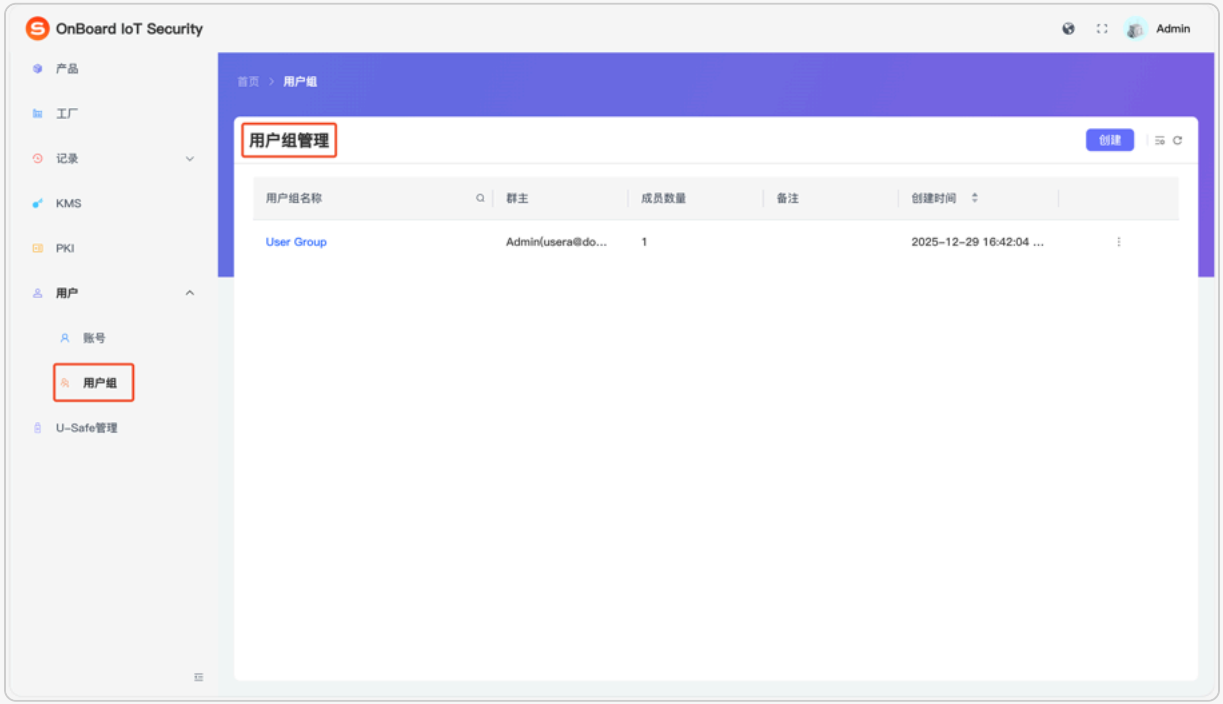
版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看用户组

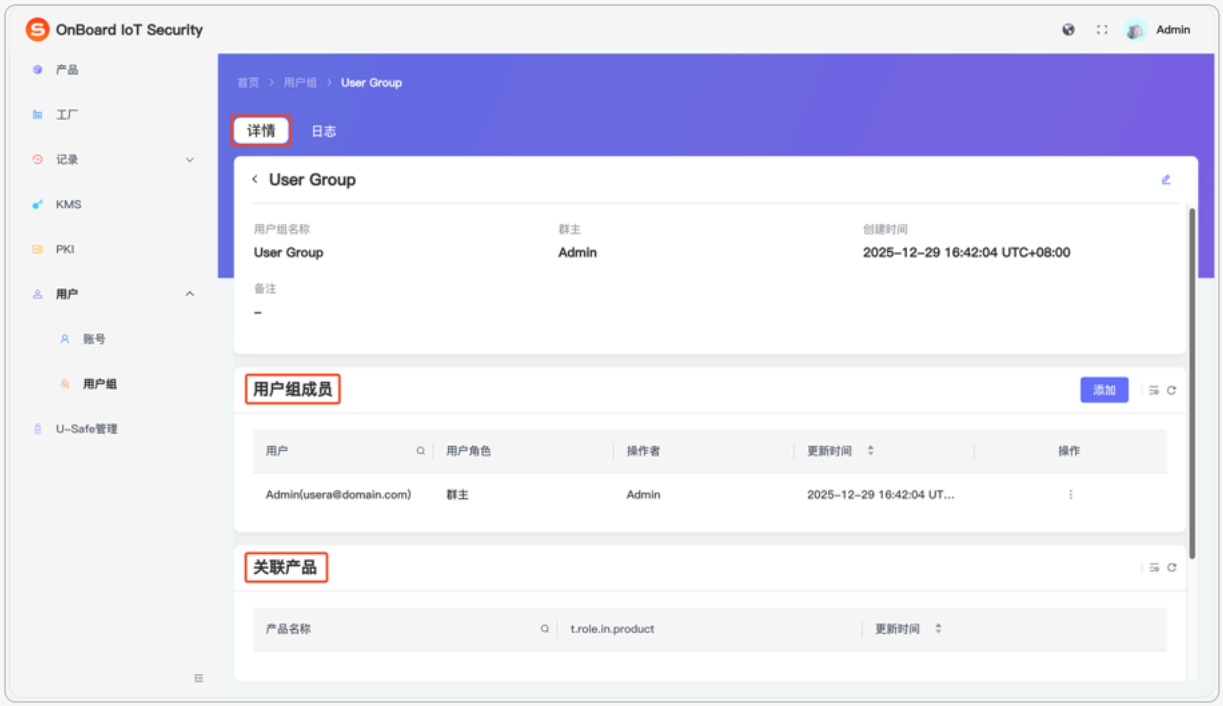
本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看用户组详情和日志。

查看详情

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，进入**用户组管理**页面。
3. 在**用户组管理**页面，点击对应的用户组名称。

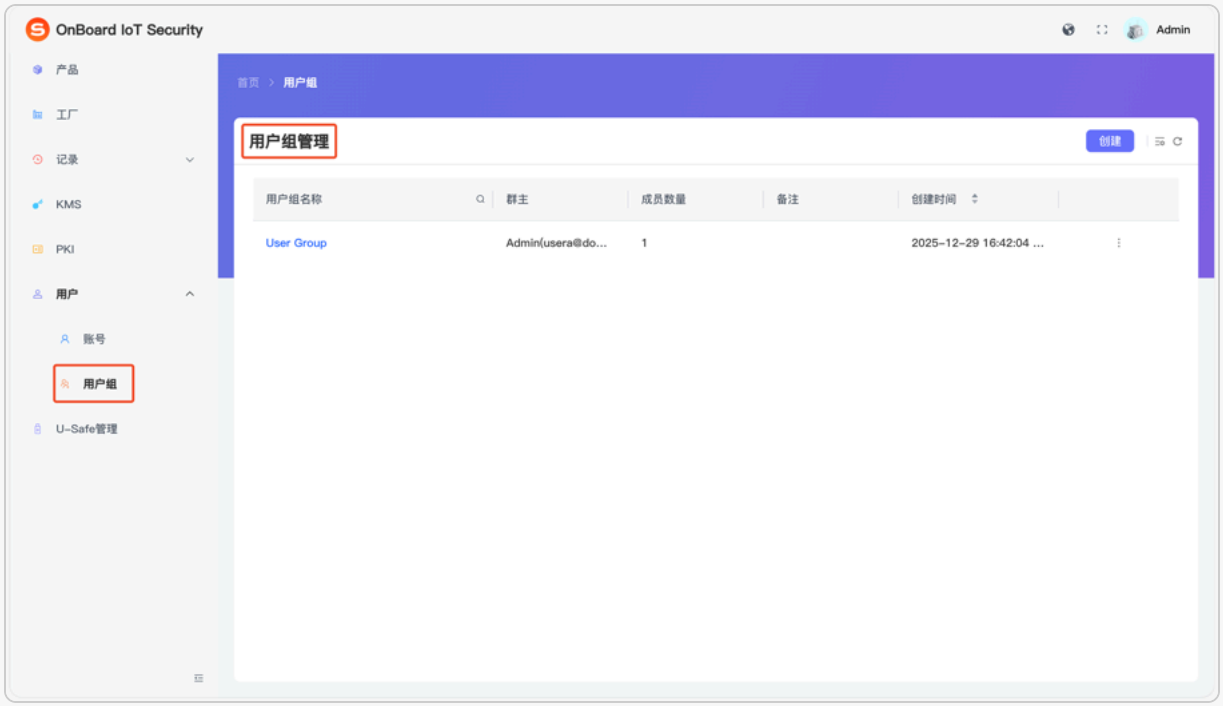


4. 在**详情**标签页，即可查看用户组信息（**用户组名称**，**群主**和**关联产品**等）。

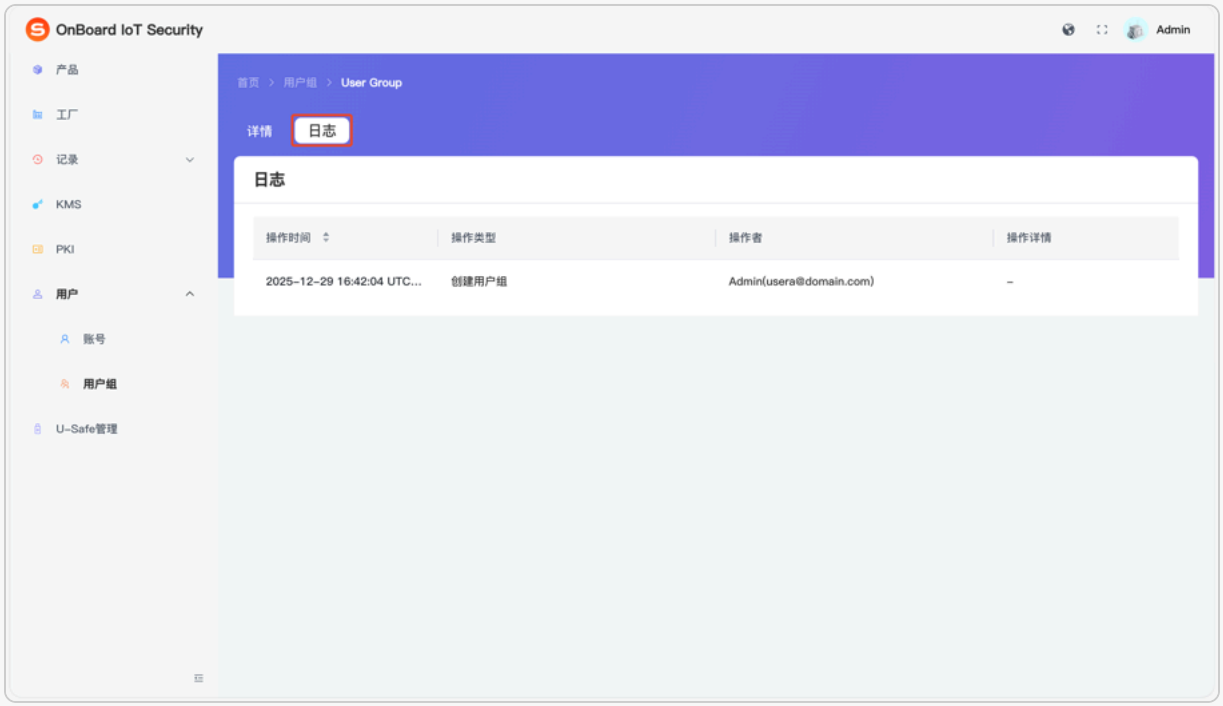


查看日志

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，进入**用户组管理**页面。
3. 在**用户组管理**页面，点击对应的用户组名称。



4. 在**日志**标签页，即可查看用户组的历史记录（**操作时间**和**操作类型**等）。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中管理用户组

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中完成以下操作从而调整用户组成员对 OBIS 的访问与操作权限：

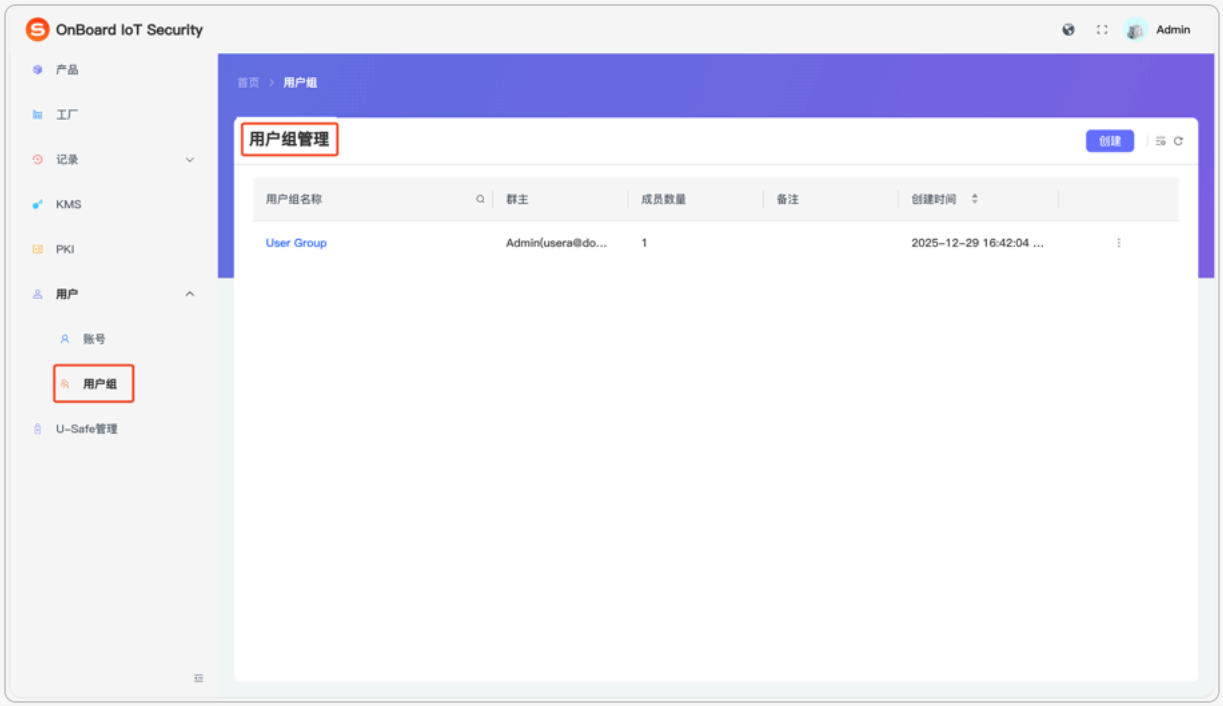
- 编辑用户组
- 添加用户组成员
- 更新用户组成员角色
- 移除用户组成员
- 删除用户组

前提条件

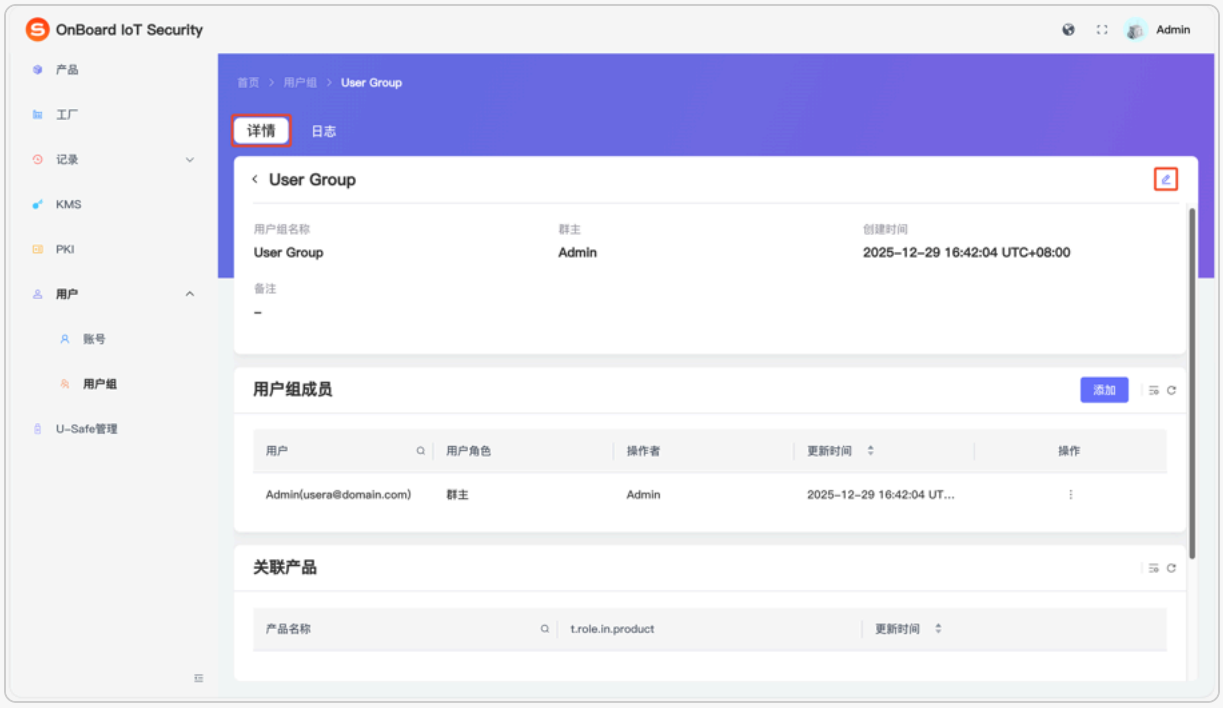
在开始之前，请确保你已获得 System Manager 的 OBIS 角色权限。

编辑用户组

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，进入**用户组管理**页面。
3. 在**用户组管理**页面，点击对应的用户组名称。

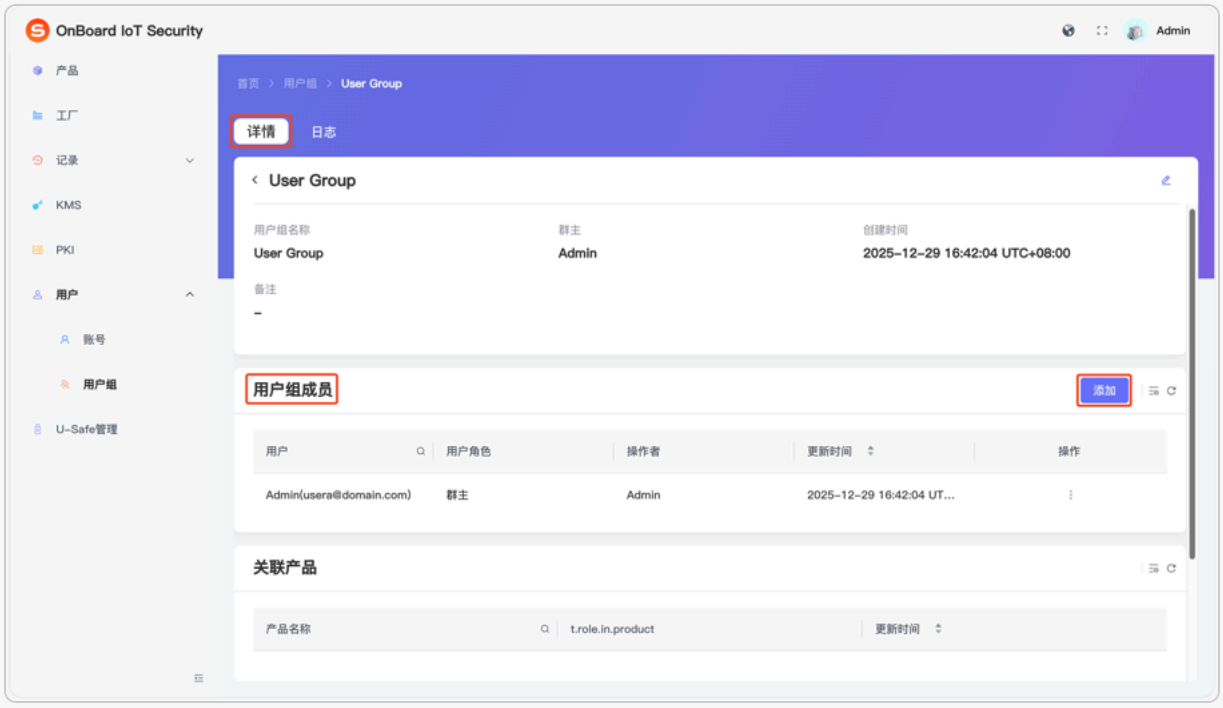


4. 在**详情**标签页，点击编辑 图标即可更改用户组信息（**用户组名称**，**群主**和**备注**）。

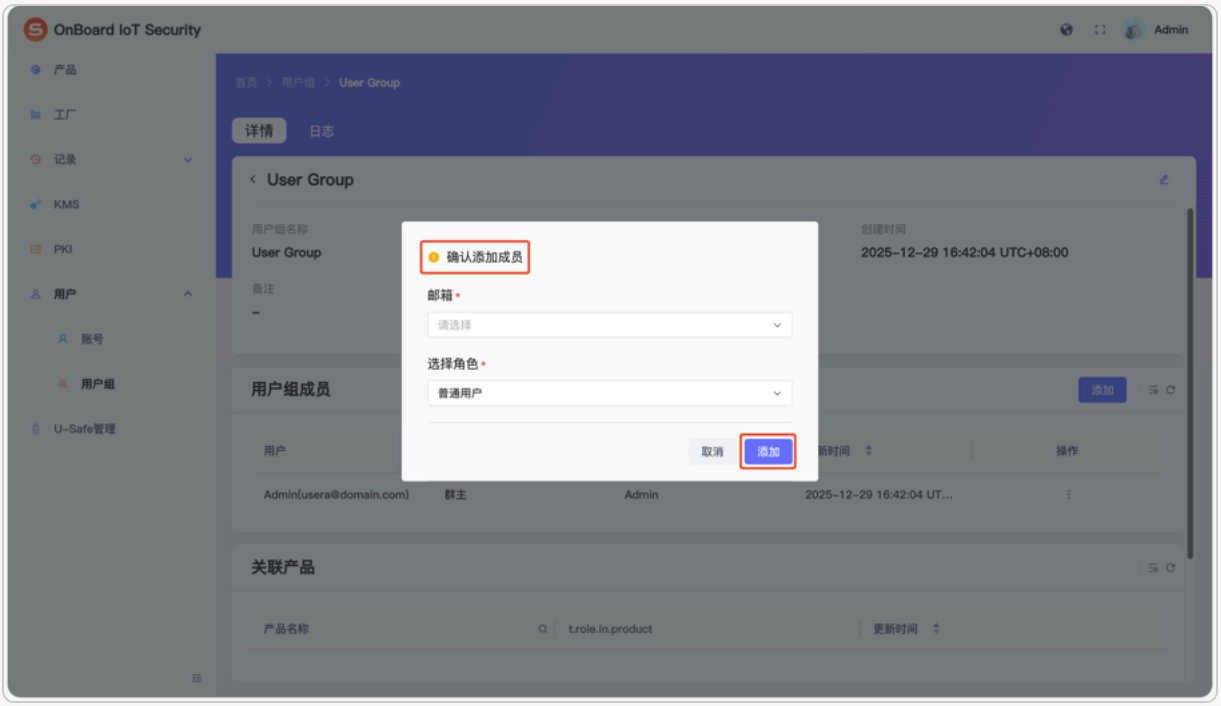


添加用户组成员

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，进入**用户组管理**页面。
3. 在**用户组管理**页面，点击对应的用户组名称。
4. 在**详情**标签页，找到**用户组成员**区域，点击**添加**。

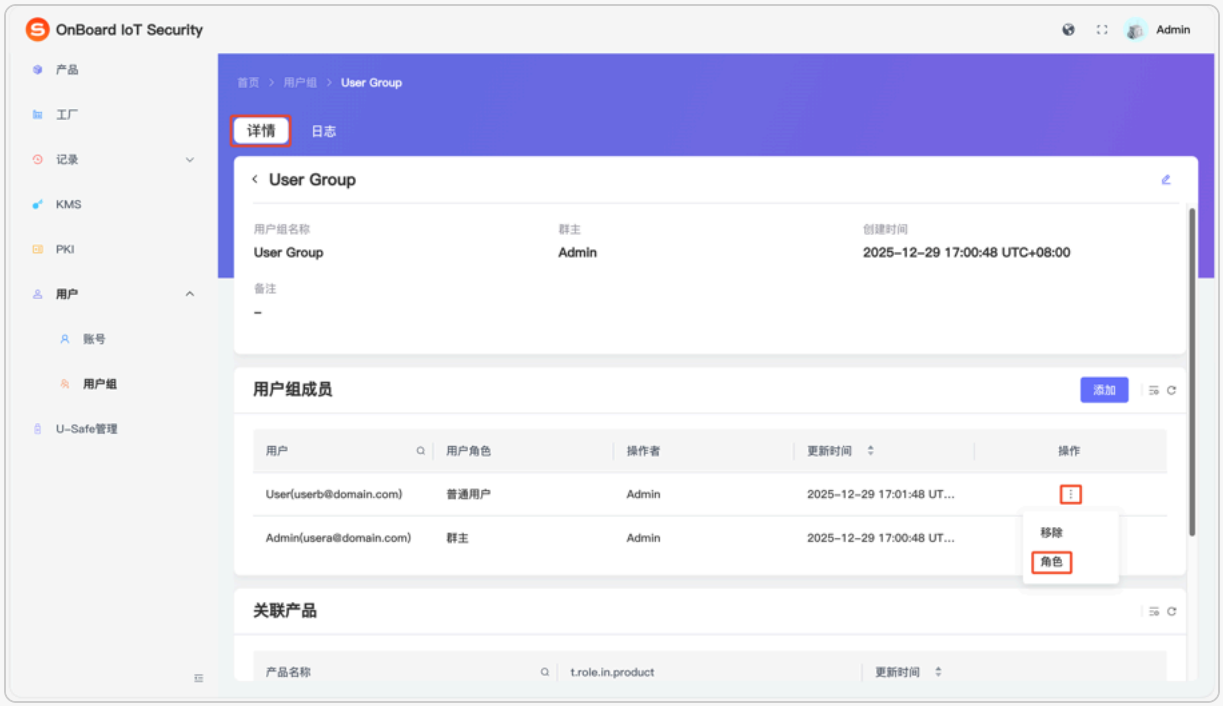


5. 在**确认添加成员**窗口，输入邮箱并分配对应的角色权限。然后，点击**添加**，即可完成用户组成员的添加。

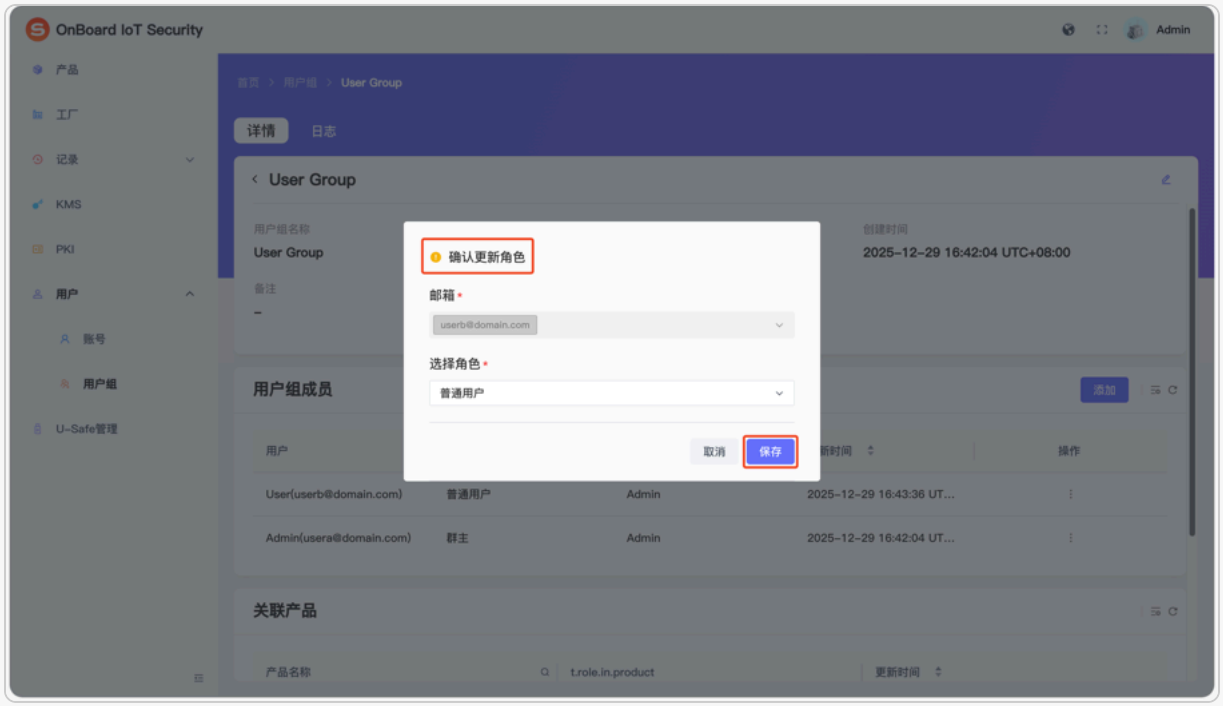


更新用户组成员角色

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，进入**用户组管理**页面。
3. 在**用户组管理**页面，点击对应的用户组名称。
4. 在**详情**标签页，找到**用户组成员**区域，点击对应成员的选项 ⋮ 图标并选择**角色**。

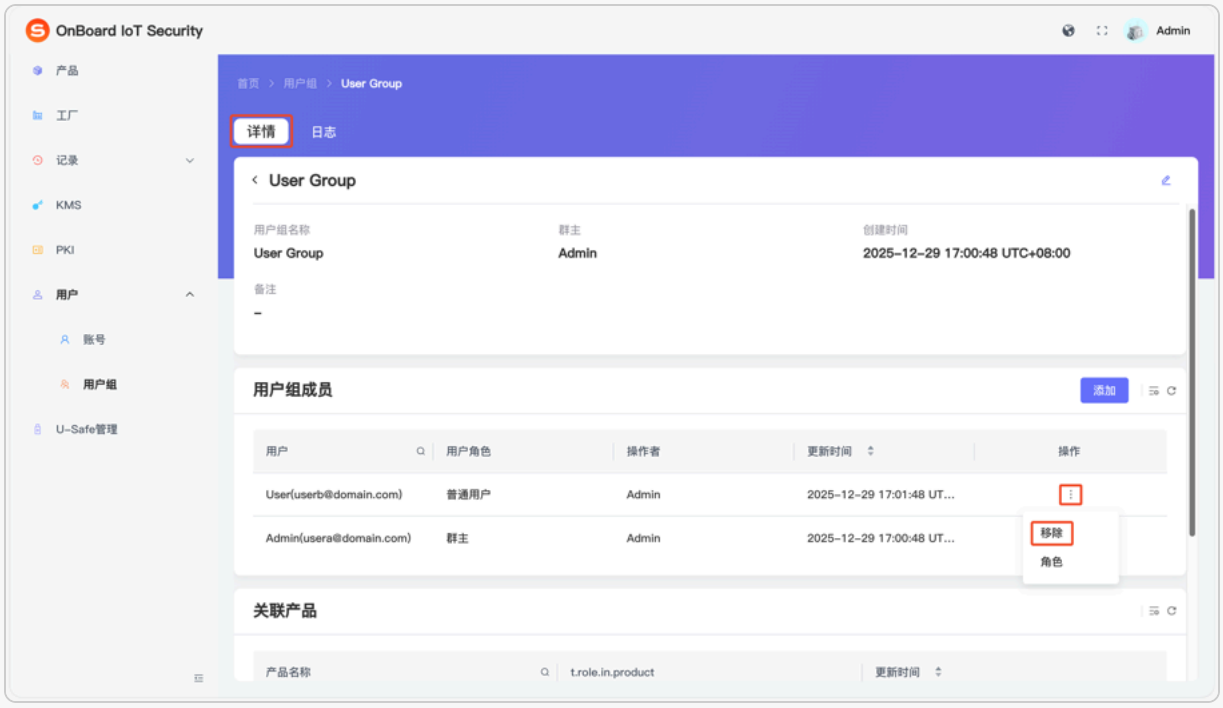


5. 在**确认更新角色**窗口，选择对应的角色权限。然后，点击**保存**，即可完成更新。

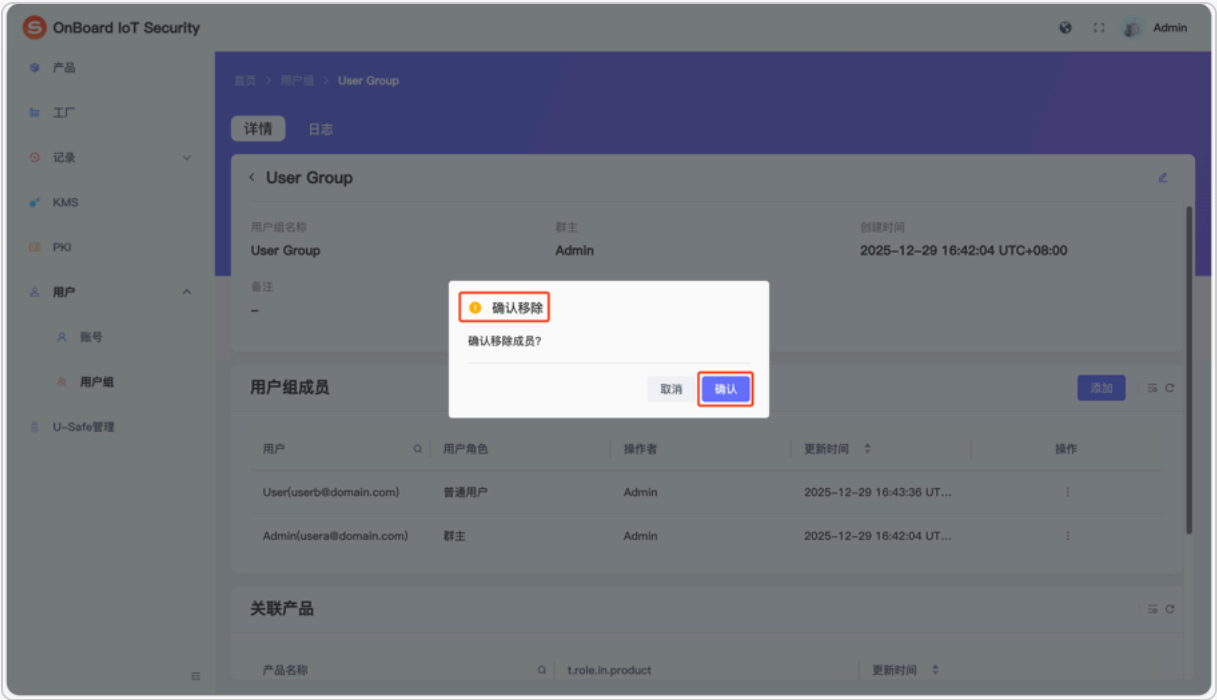


移除用户组成员

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，进入**用户组管理**页面。
3. 在**用户组管理**页面，点击对应的用户组名称。
4. 在**详情**标签页，找到**用户组成员**区域，点击对应成员的选项 ⋮ 图标并选择**移除**。



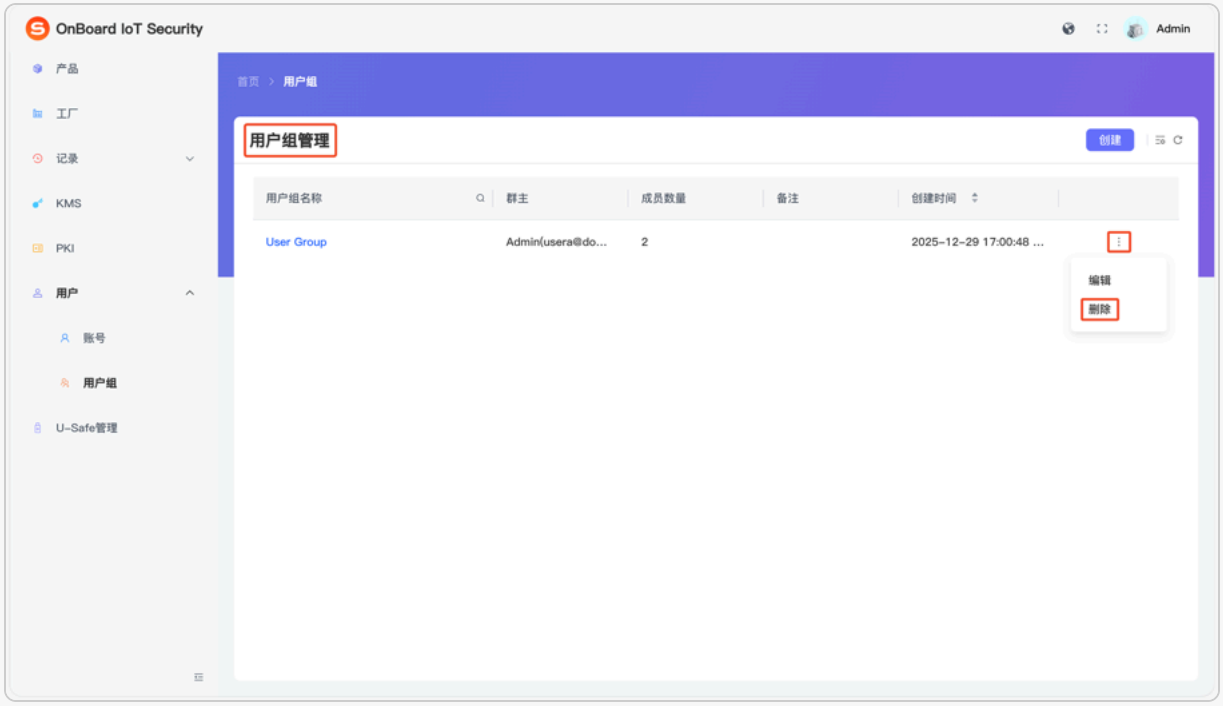
5. 在**确认移除**窗口，点击**确认**，即可移除该用户组成员。



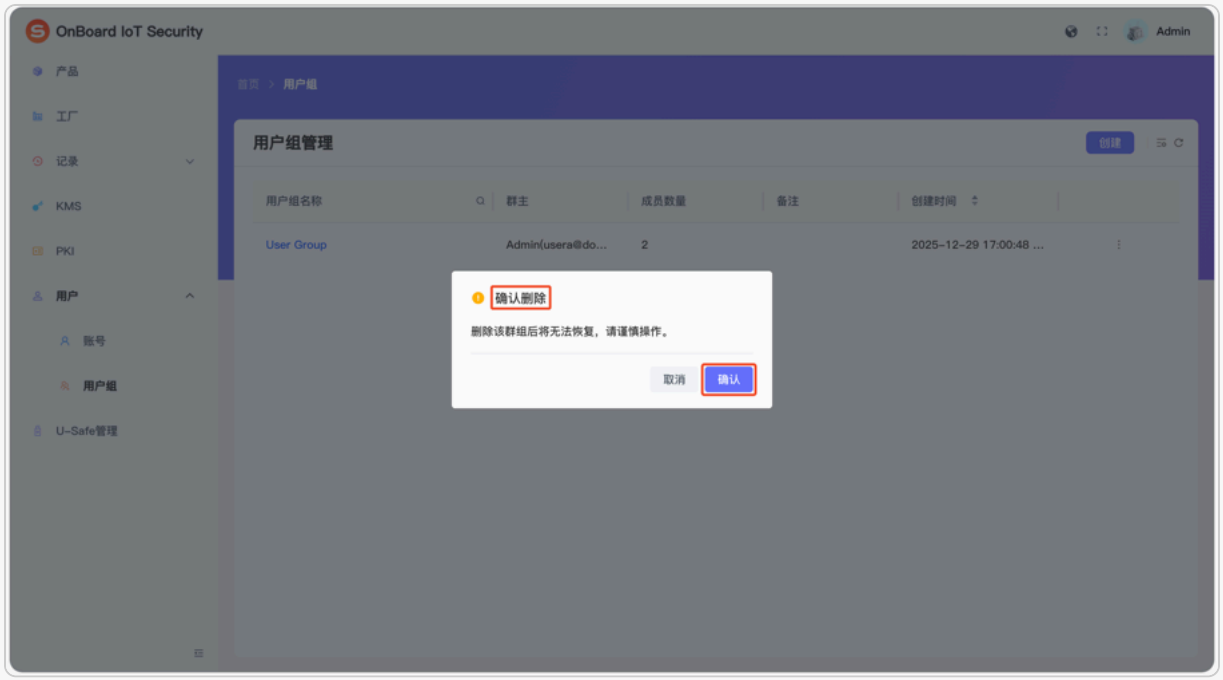
删除用户组

> **注意**
>
> 删除用户组后将无法恢复，请谨慎操作。

1. 登录 OBIS。
2. 点击左侧菜单中的**用户** > **用户组**，进入**用户组管理**页面。
3. 在**用户组管理**页面，点击对应用户组的选项 ⋮ 图标并选择**删除**。



4. 在**确认删除**窗口中点击**确认**，即可删除该用户组。



版本：v3.2.2

如何在 OnBoard™ IoT Security 中查看 U-Safe

本文将指导你如何在 OnBoard™ IoT Security (OBIS) 中查看工厂绑定的 U-Safe 的参数和状态。

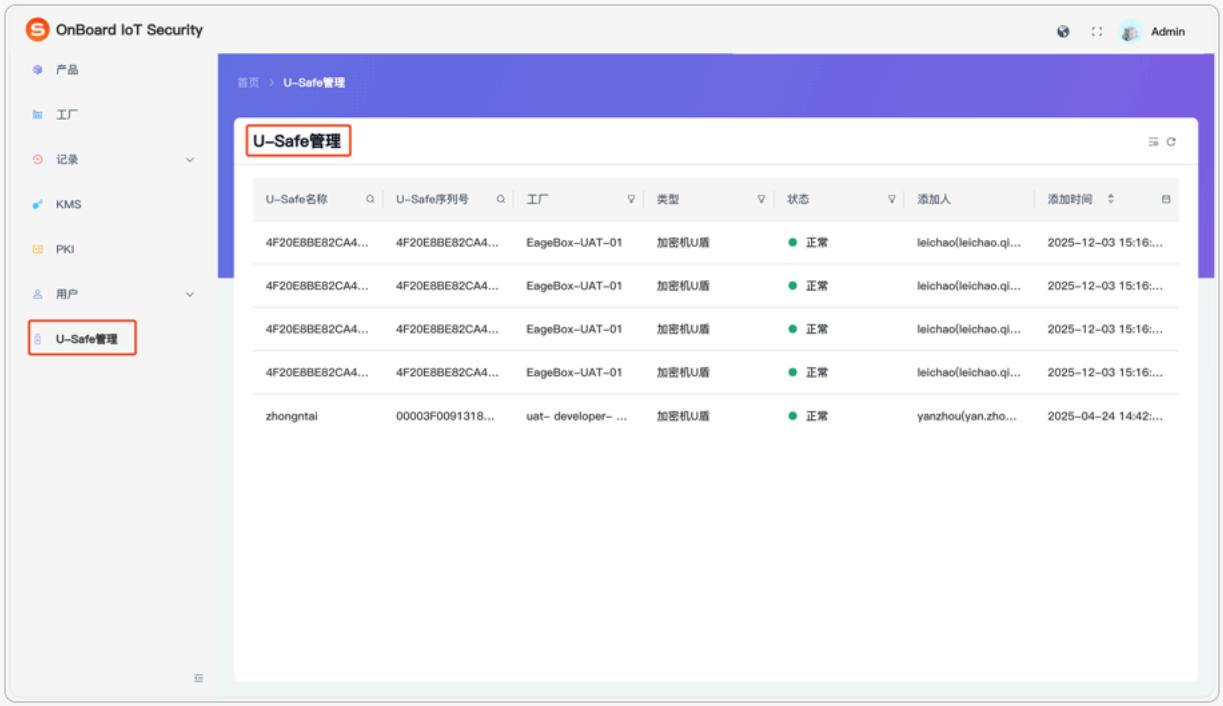
前提条件

在开始之前，请确保：

- 已完成工厂的创建并绑定 HSM U-Safe 和 Identity U-Safe。
- 你已获得 System Manager 的 OBIS 角色权限。

操作步骤

1. 登录 OBIS。
2. 点击左侧菜单中的 **U-Safe** 管理，在 **U-Safe 管理**页面，即可查看租户内工厂已绑定的所有 U-Safe 信息。



U-Safe 状态说明：

状态	说明
正常	在配置工厂时添加 HSM U-Safe 和 Identity U-Safe 后的状态，可正常使用。
已锁定	U-Safe 已被锁定，且无法正常使用。

版本：v3.2.2

OnBoard Edge Service 的概述

> **信息**
>
> OnBoard Edge Service 中的操作仅适用于本地工厂，云工厂的相关操作则可直接在 OnBoard™ IoT Security (OBIS) 中进行。请联系雪球协助完成部署。

OnBoard Edge Service 在工厂层面管理生产批次，本地执行授权约束，并使通过内网连接的烧录站能够执行生产操作。它提供一个内网可访问的网页门户，Factory Manager 可通过该门户监控生产状态并执行管理操作。OnBoard Edge Service 提供以下核心功能：

生产批次管理

支持来自一个或多个 OEM 的并发生产批次操作。Factory Manager 可通过网页门户监控批次状态、将批次分配给烧录站，并管理批次生命周期。

烧录站管理

Factory Manager 可通过网页门户管理烧录站，包括：

- 创建与配置烧录站，包括与 OnBoard Edge Service 连接的安全认证方式（Identity U-Safe、密码或无密码）
- 从 OnBoard Edge Service 将生产批次分配给烧录站（或者，操作员可直接在烧录站软件中指定批次 ID 或生产订单 ID）
- 向烧录站下发对应的烧录站软件包

生产设备记录管理

OnBoard Edge Service 从烧录站生成每台设备的生产记录，并提交到 OBIS 云平台。还可与现有工厂系统（如 MES 或工厂测试系统）互操作，交换生产记录。OnBoard Edge Service 在任何阶段都无法访问产品密钥材料，所有加密密钥材料的处理仅在 EdgeHSM 硬件边界内完成。

版本：v3.2.2

如何登录 OnBoard Edge Service

> **信息**
>
> OnBoard Edge Service 中的操作仅适用于本地工厂，云工厂的相关操作则可直接在 OnBoard™ IoT Security (OBIS) 中进行。

本文将指导你如何使用 "邮箱 + 密码" 或 "邮箱 + 验证码" 的方式登录 OnBoard Edge Service。

前提条件

在开始之前，请确保：

- 已完成工厂的创建和配置。具体操作请参考创建工厂，配置工厂和配置烧录站。
- 你已获得 Factory Manager 的工厂角色权限。
- 已联系雪球协助完成 OnBoard Edge Service 的部署。
- 已从雪球获取用于身份鉴权的 HSM U-Safe，并将对应的 HSM U-Safe（在配置工厂信息时添加）插入工厂本地服务器。
- 你的登录设备处于工厂服务内网环境。

> **提示**
>
> 工厂服务器至少需要插入 1 个 HSM U-Safe。插入多个 HSM U-Safe 可提升服务器性能，具体数量可根据产线需求采购。

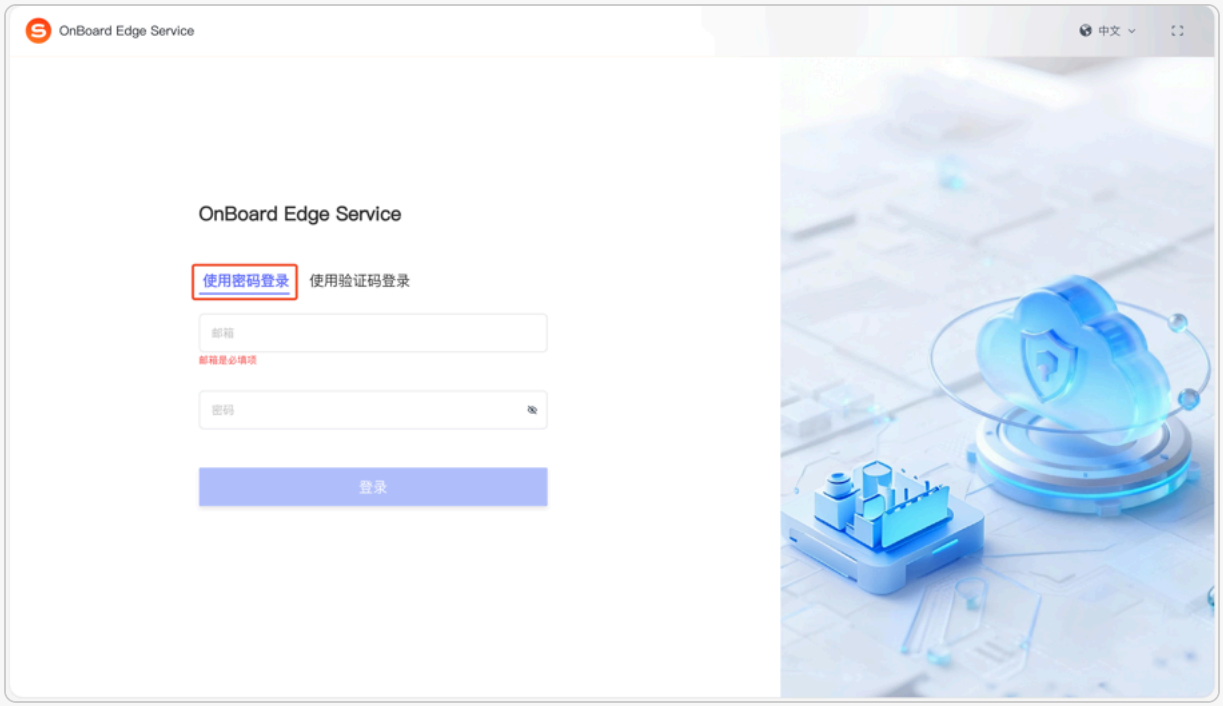
方式一：使用密码登录

> **信息**
>
> - 在断开外网环境下，OnBoard Edge Service 仅支持使用 "邮箱 + 密码" 的方式进行身份验证。
> - 在添加工厂管理员后，系统会自动生成初始密码并发送到对应的邮箱。目前，该随机密码可以重置但暂不支持自定义修改。

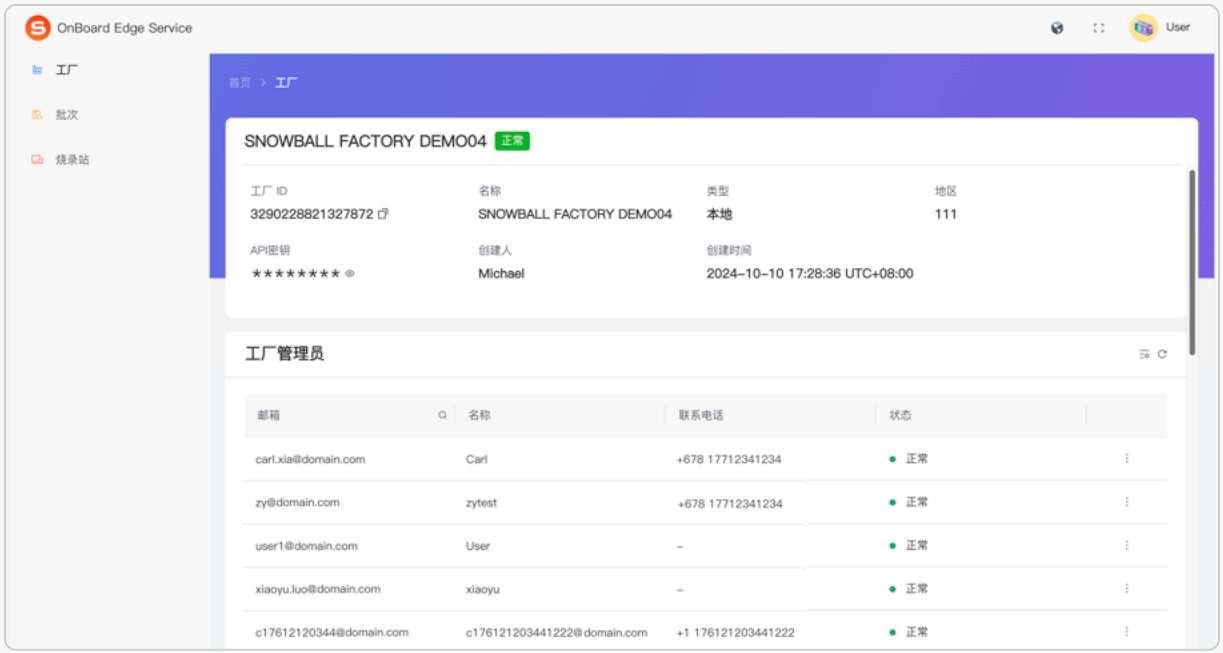
1. 在浏览器地址栏中输入以下地址：**http://<Service Address>:32334/login**。

   > **提示**
   >
   > 请将 **<Service Address>** 替换为实际服务 IP。具体服务地址获取方法，请参考查看服务地址。

2. 在 OnBoard Edge Service 登录页面，选择**使用密码登录**，输入你的邮箱和密码，并点击**登录**。



3. 登录成功后，页面会跳转至 OnBoard Edge Service 主界面。

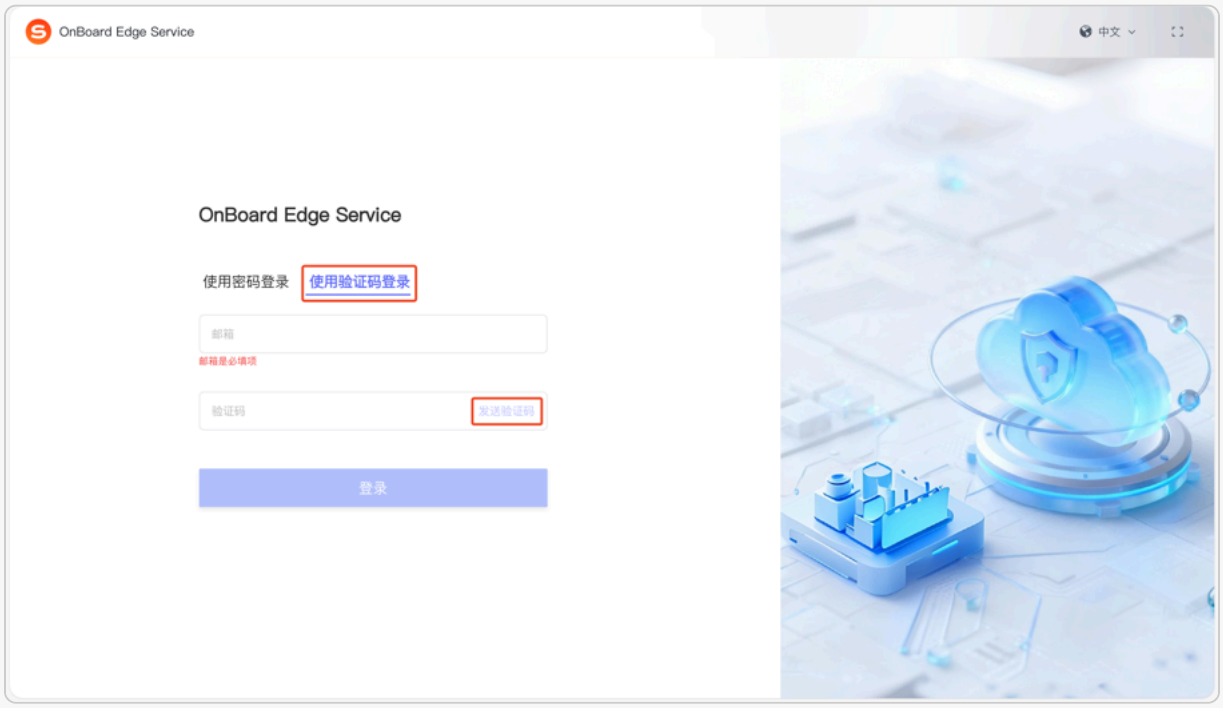


方式二：使用验证码登录

1. 在浏览器地址栏中输入以下地址：**http://<Service Address>:32334/login**。

   > **提示**
   >
   > 请将 **<Service Address>** 替换为实际服务地址。具体服务地址获取方法，请参考查看服务地址。

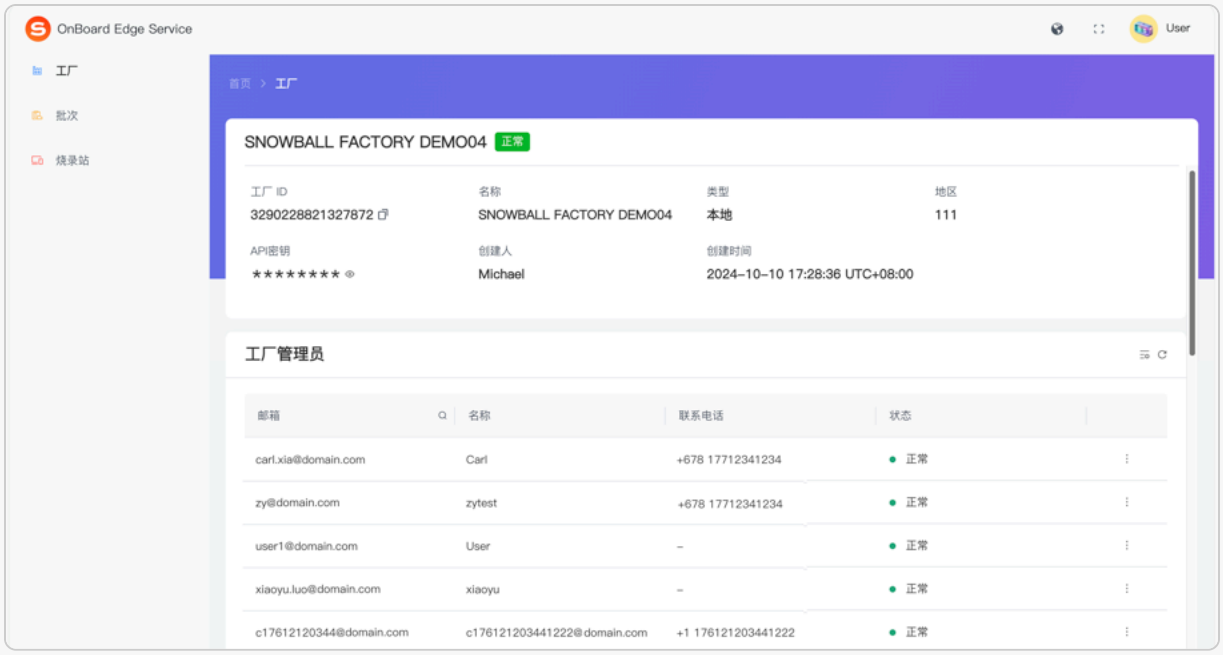
2. 在 OnBoard Edge Service 登录页面，选择**使用验证码登录**，输入你的邮箱，并点击**发送验证码**。



3. 登录你的邮箱，查收来自 OnBoard™ IoT Security（OBIS）的验证邮件并获取 6 位数字验证码。

4. 在 OnBoard Edge Service 登录页面，输入你邮件中收到的 6 位数字验证码，并点击**登录**。

5. 登录成功后，页面会跳转至 OnBoard Edge Service 主界面。



版本：v3.2.2

如何在 OnBoard Edge Service 中管理工厂

> **信息**
>
> OnBoard Edge Service 中的操作仅适用于本地工厂，云工厂的相关操作则可直接在 OnBoard™ IoT Security (OBIS) 中进行。

本文将指导你如何在 OnBoard Edge Service 中完成以下操作：

- 重置工厂管理员密码
- 删除工厂管理员
- 锁定或解锁 HSM U-Safe
- 锁定或解锁 Identity U-Safe

前提条件

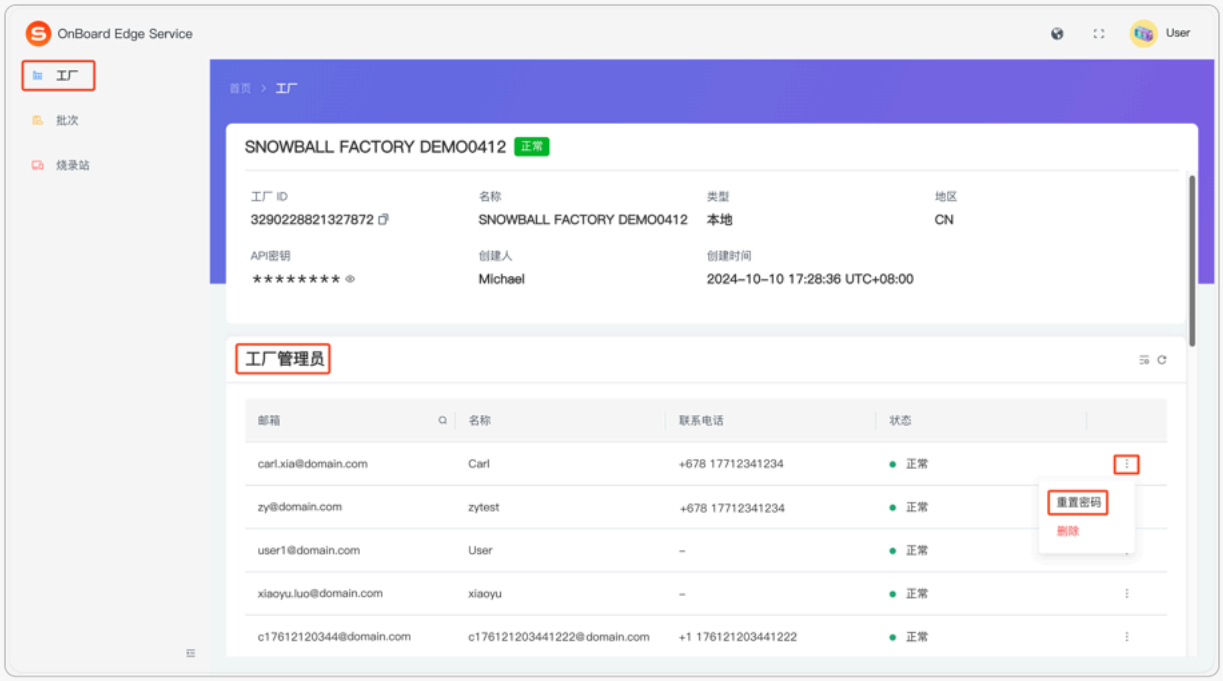
在开始之前，请确保你可以登录 OnBoard Edge Service。

重置工厂管理员密码

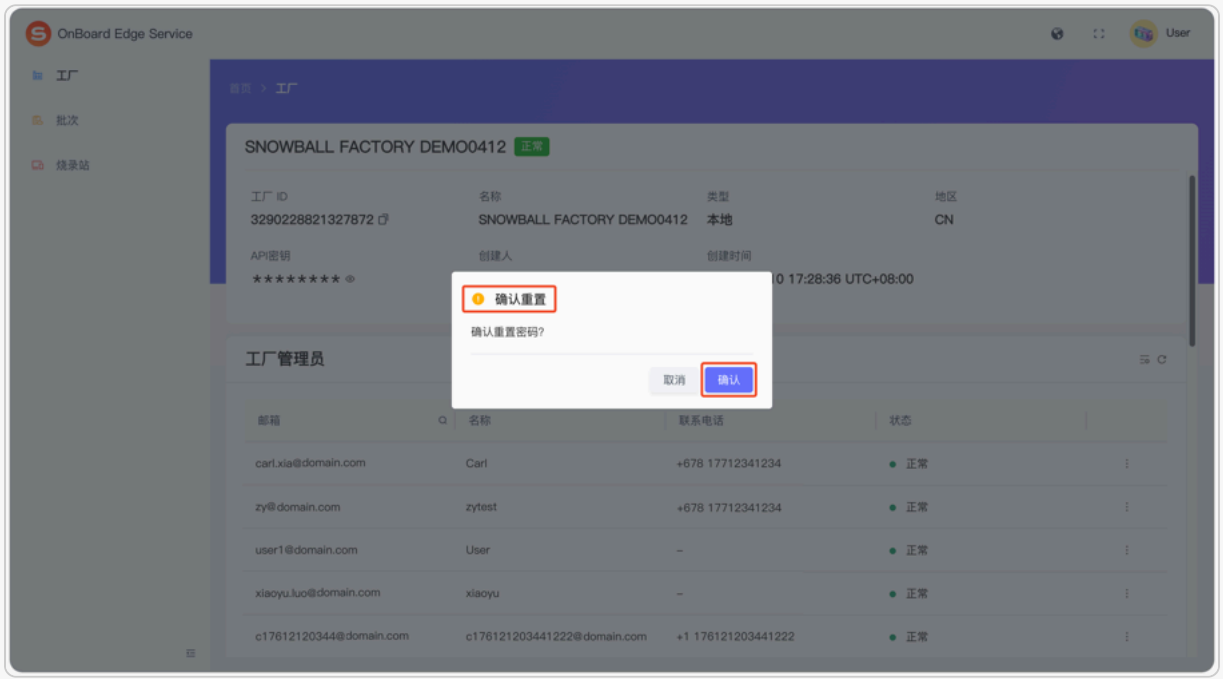
> **提示**
>
> 密码重置后，系统会自动生成随机密码并发送到对应的邮箱。

1. 登录 OnBoard Edge Service。

2. 点击左侧菜单中的**工厂**，找到**工厂管理员**区域，点击对应工厂管理员的选项 ⋮ 图标并选择**重置密码**。

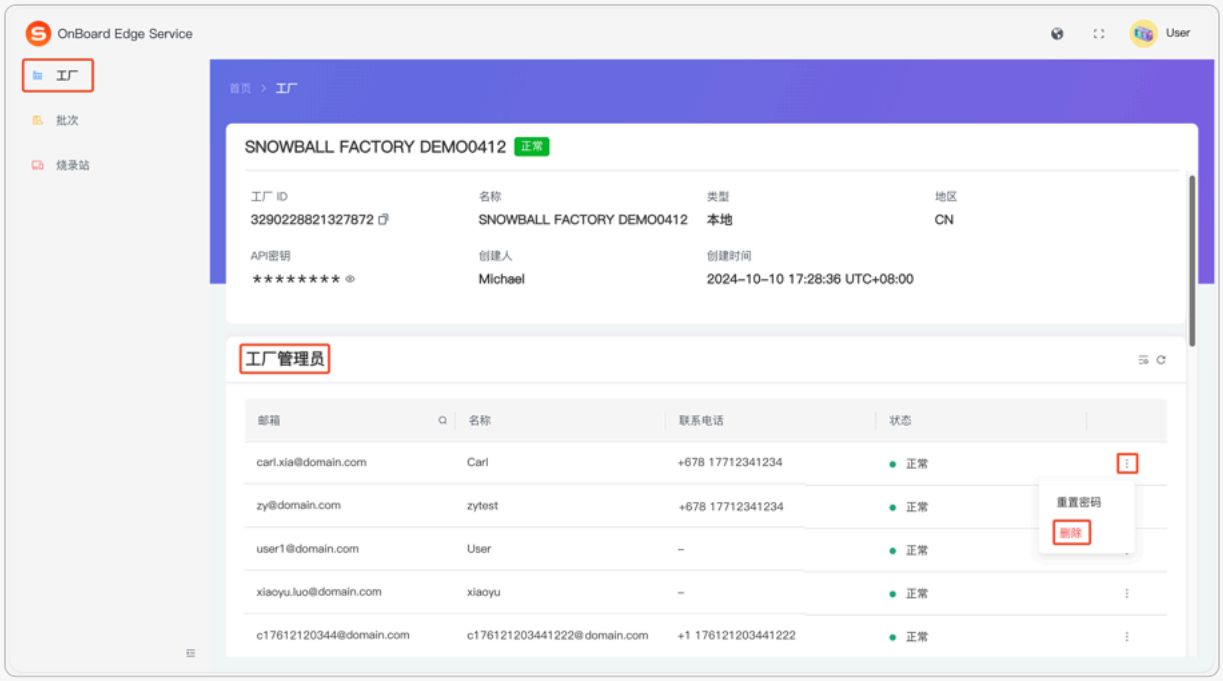


3. 在**确认重置**窗口中点击**确认**，即可重置工厂管理员密码。

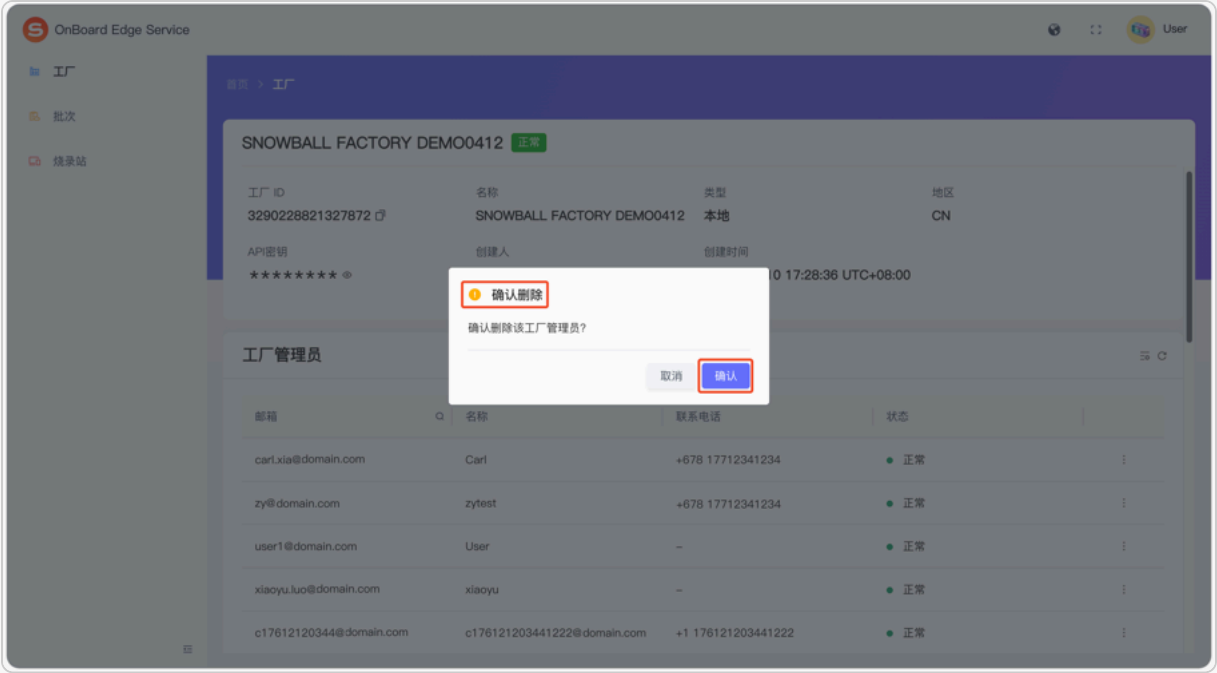


删除工厂管理员

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**工厂**，找到**工厂管理员**区域，点击对应工厂管理员的选项 ⋮ 图标并选择**删除**。



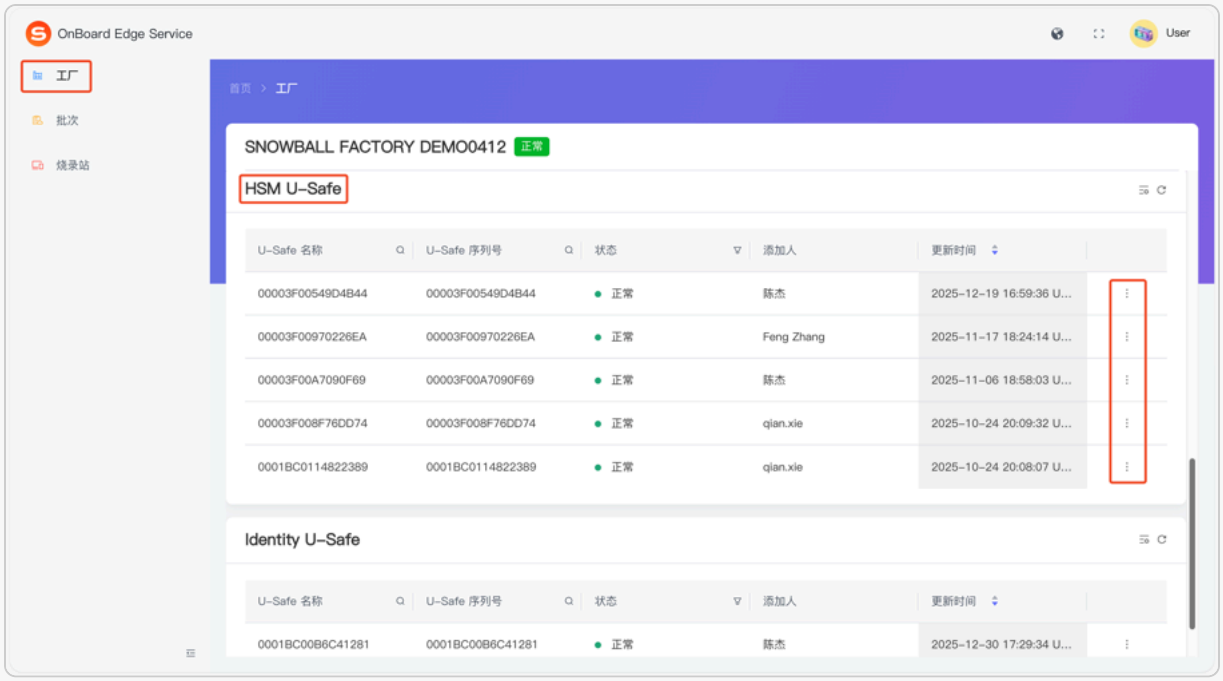
3. 在**确认删除**窗口中点击**确认**，即可删除该工厂管理员。



锁定或解锁 HSM U-Safe

> **提示**
>
> 锁定 HSM U-Safe 以禁止其使用和解绑。锁定后，状态会变更为**锁定**。

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**工厂**，找到 **HSM U-Safe** 区域，点击对应 HSM U-Safe 的选项图标并选择**锁定**或**解锁**。



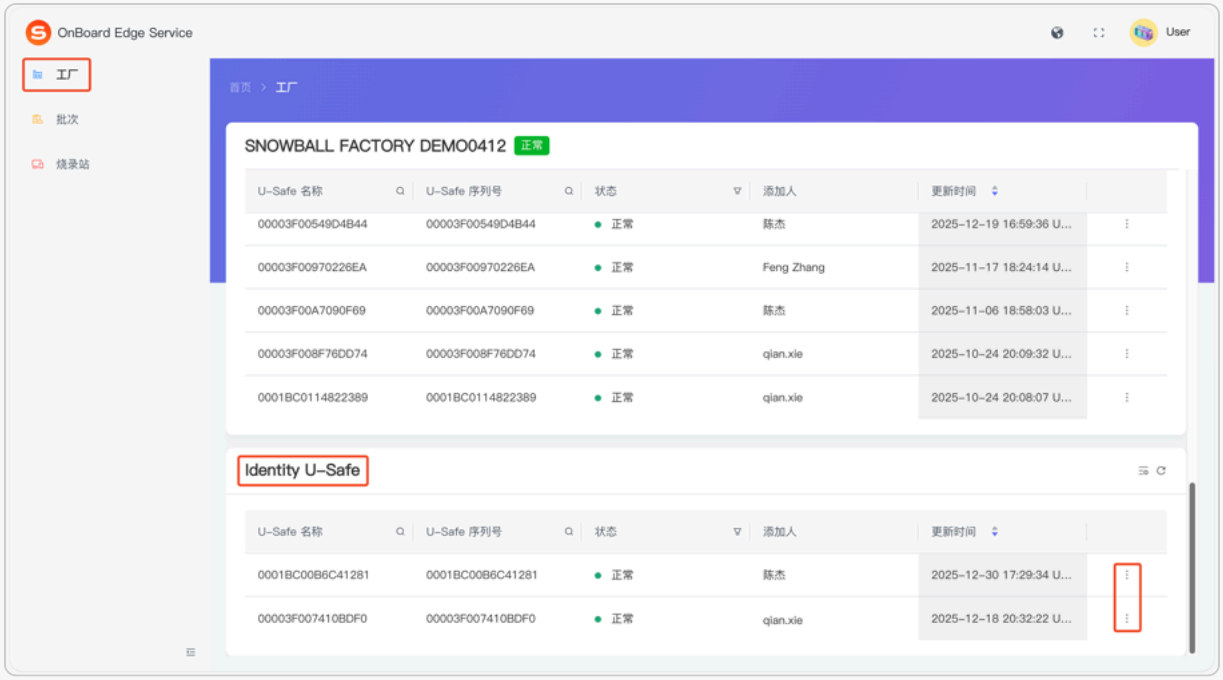
3. 在弹出的窗口中点击**确认**，即可锁定或解锁该 HSM U-Safe。

锁定或解锁 Identity U-Safe

> **提示**
>
> 锁定 Identity U-Safe 以禁止其使用和解绑。锁定后，状态会变更为**锁定**。

1. 登录 OnBoard Edge Service。

2. 点击左侧菜单中的**工厂**，找到 **Identity U-Safe** 区域，点击对应 Identity U-Safe 的选项 ⋮ 图标并选择**锁定**或**解锁**。



3. 在弹出的窗口中点击**确认**，即可锁定或解锁该 Identity U-Safe。

版本：v3.2.2

如何在 OnBoard Edge Service 中管理批次与分配烧录站

> **信息**
>
> OnBoard Edge Service 中的操作仅适用于本地工厂，云工厂的相关操作则可直接在 OnBoard™ IoT Security (OBIS) 中进行。

本文将指导你在 OnBoard Edge Service 中完成以下操作：

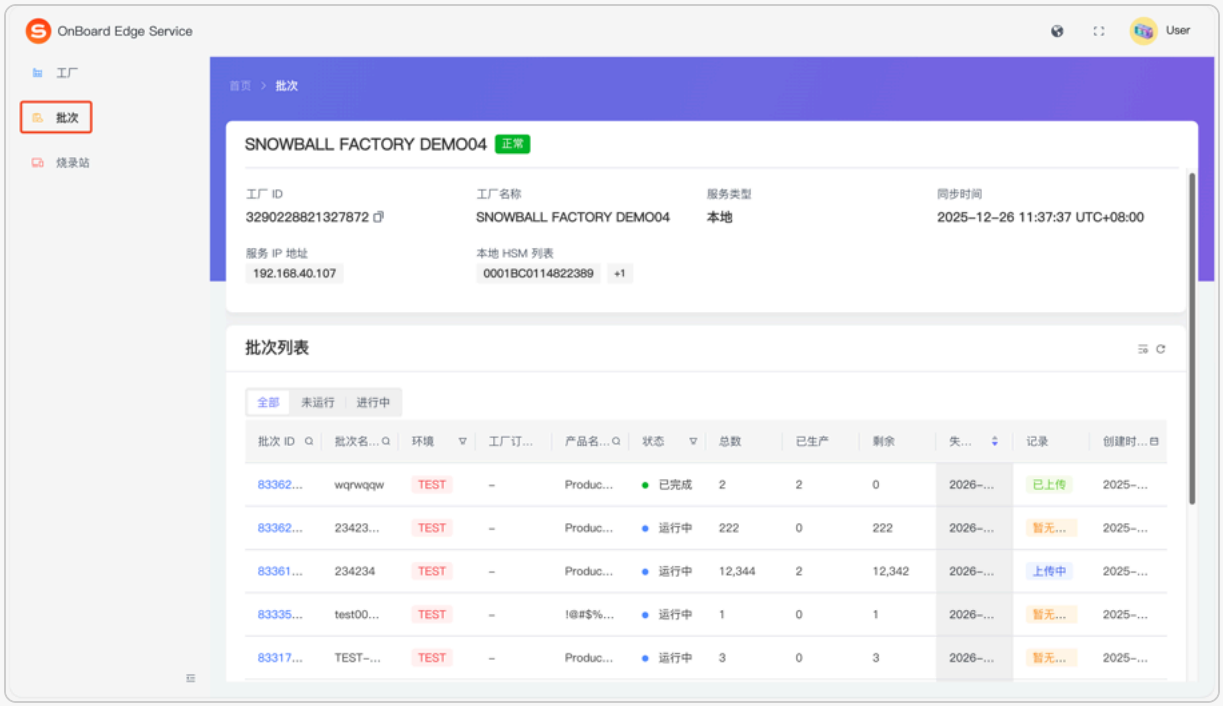
- 查看批次
- 为批次分配烧录站
- 为批次取消分配烧录站
- 关闭与删除批次

前提条件

在开始之前，请确保你可以登录 OnBoard Edge Service。

查看批次

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**批次**，进入**批次**页面。

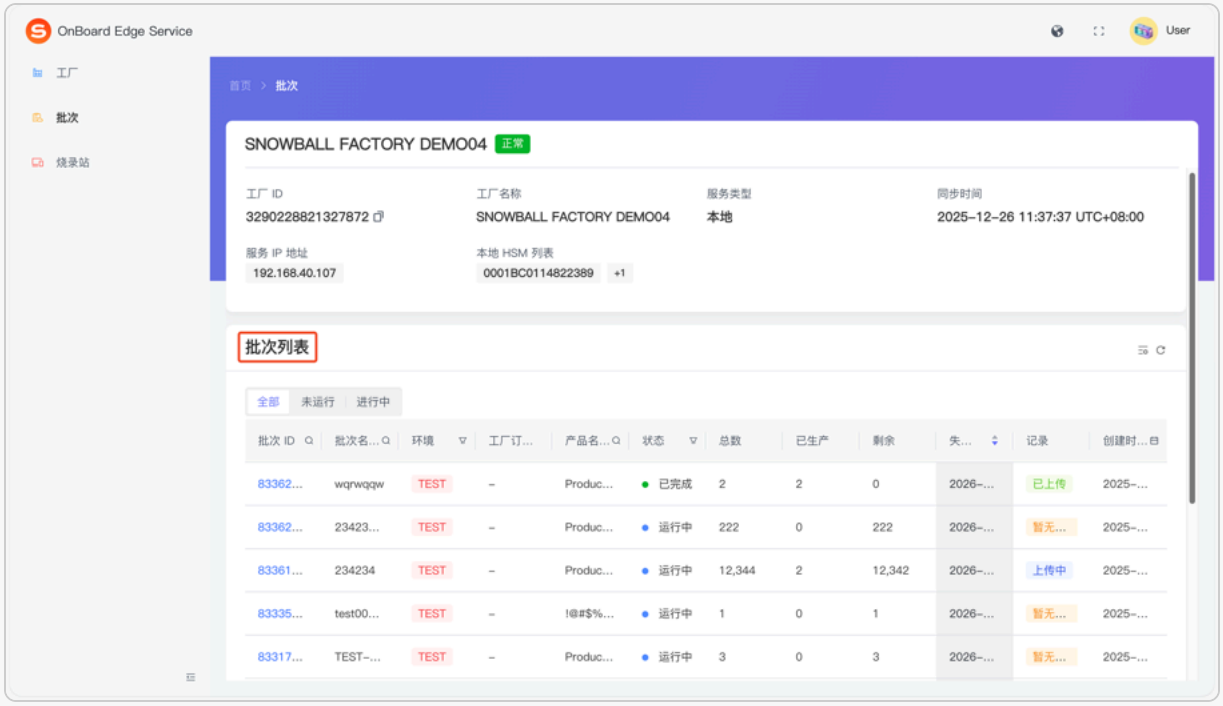


批次状态说明：

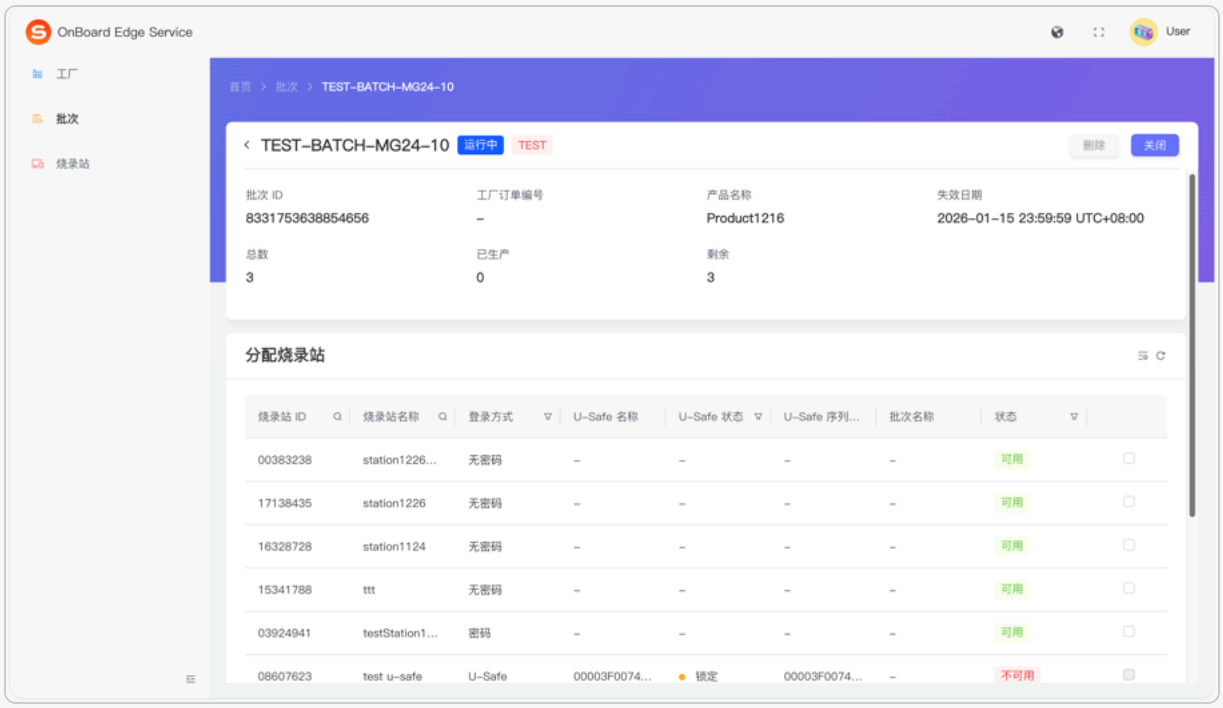
批次状态	可否开启	说明
运行中	-	创建批次后的默认状态，可绑定烧录站。
已停止	可以	批次已被关闭，可重新开启并绑定烧录站。

批次状态	可否开启	说明
已完成	不可以	批次烧录完成后的状态。
已过期	不可以	批次在完成烧录前已超过有效期。 **注释**：批次过期后，OnBoard Edge Service 针对该批次的大部分功能将无法正常操作，包括在Programming Station 客户端进行烧录。建议删除已过期批次。

3. 在**批次**页面，找到**批次列表**区域，点击对应的批次 ID，进入批次页面。



4. 在批次页面，即可查看批次信息。

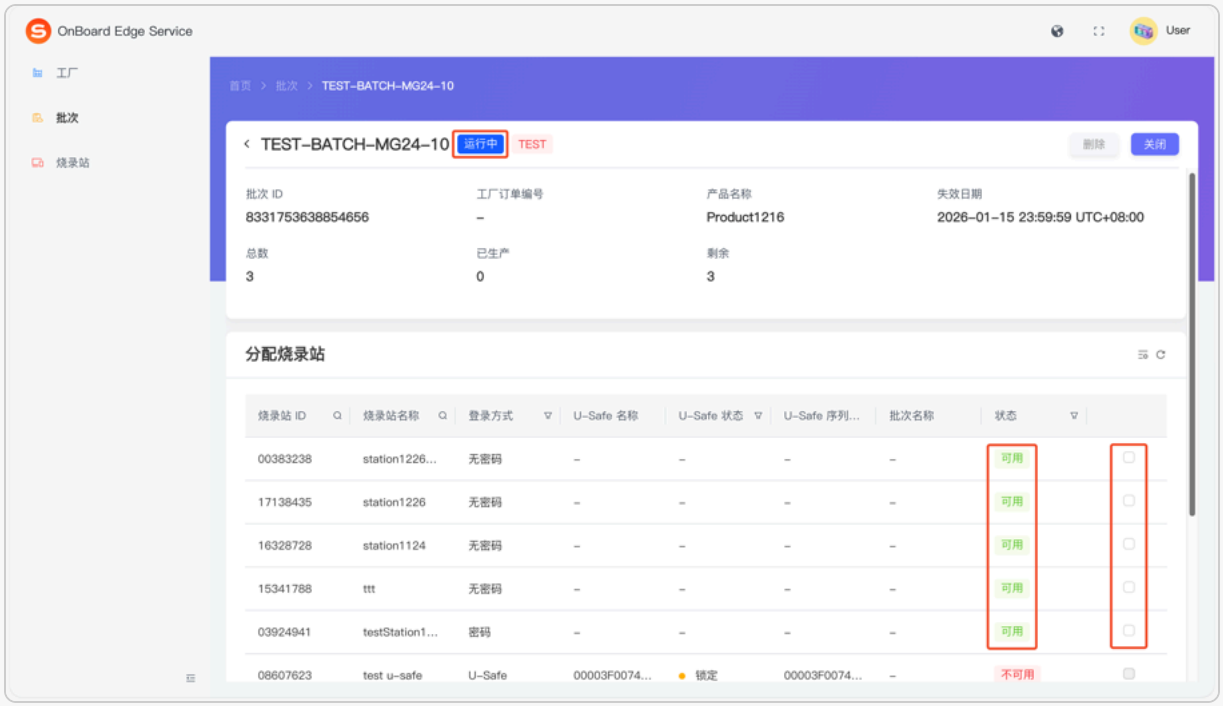


为批次分配烧录站

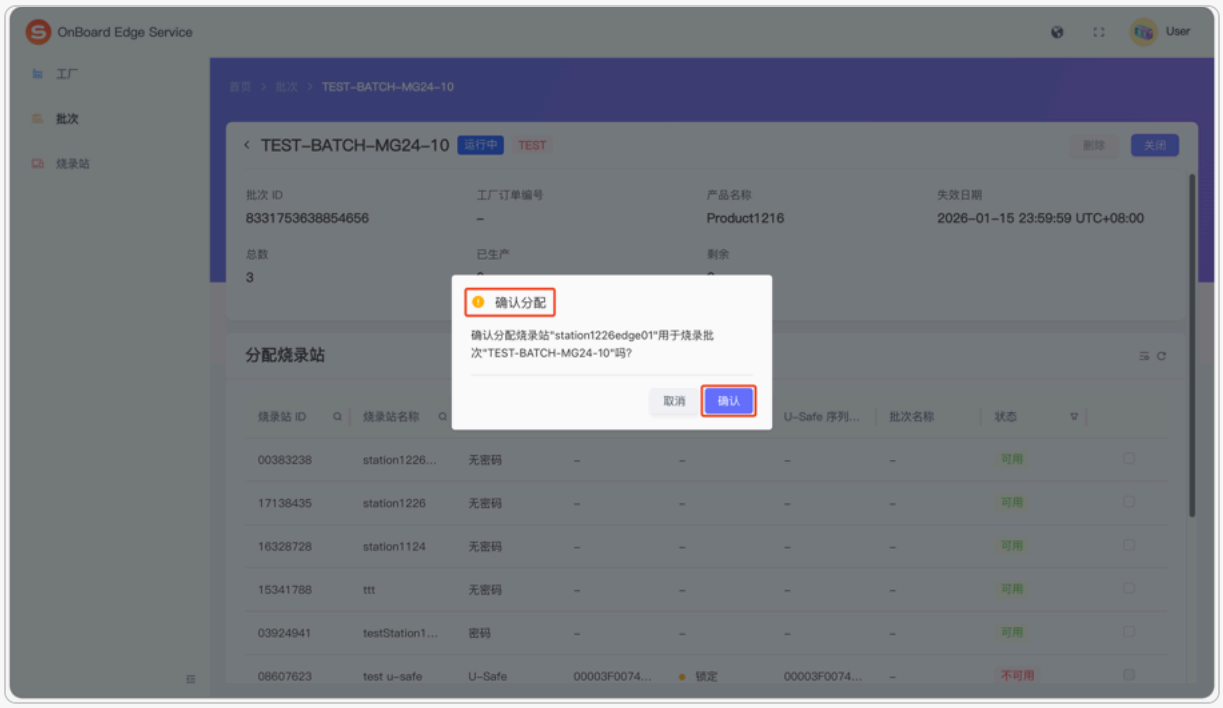
> **提示**
>
> 一个批次同时可以分配给多个烧录站，而一个烧录站同时只可烧录一个批次。

为批次分配状态为**可用**的烧录站用于烧录：

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**批次**，然后点击对应的批次 ID，进入批次页面。
3. 在批次页面，确认批次的状态为**运行中**，并勾选状态为**可用**的烧录站进行分配。



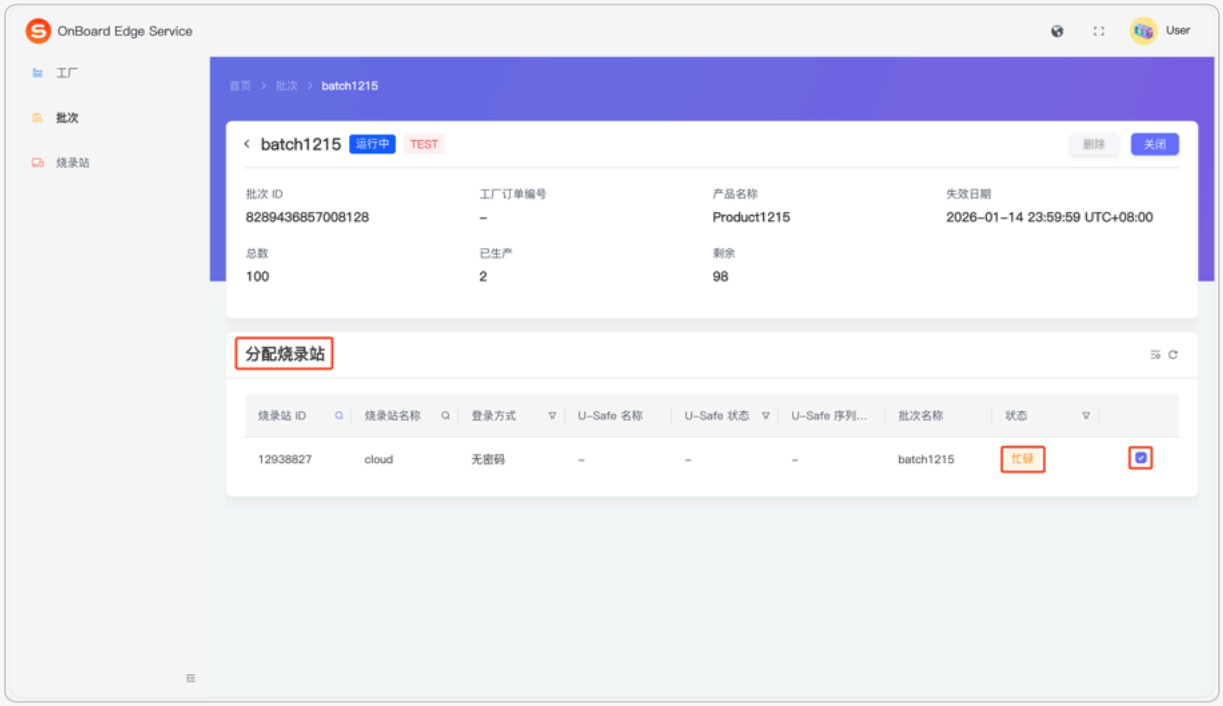
4. 在**确认分配**窗口，点击**确认**，即可完成烧录站的分配。分配后，烧录站状态会变更为**忙碌**。



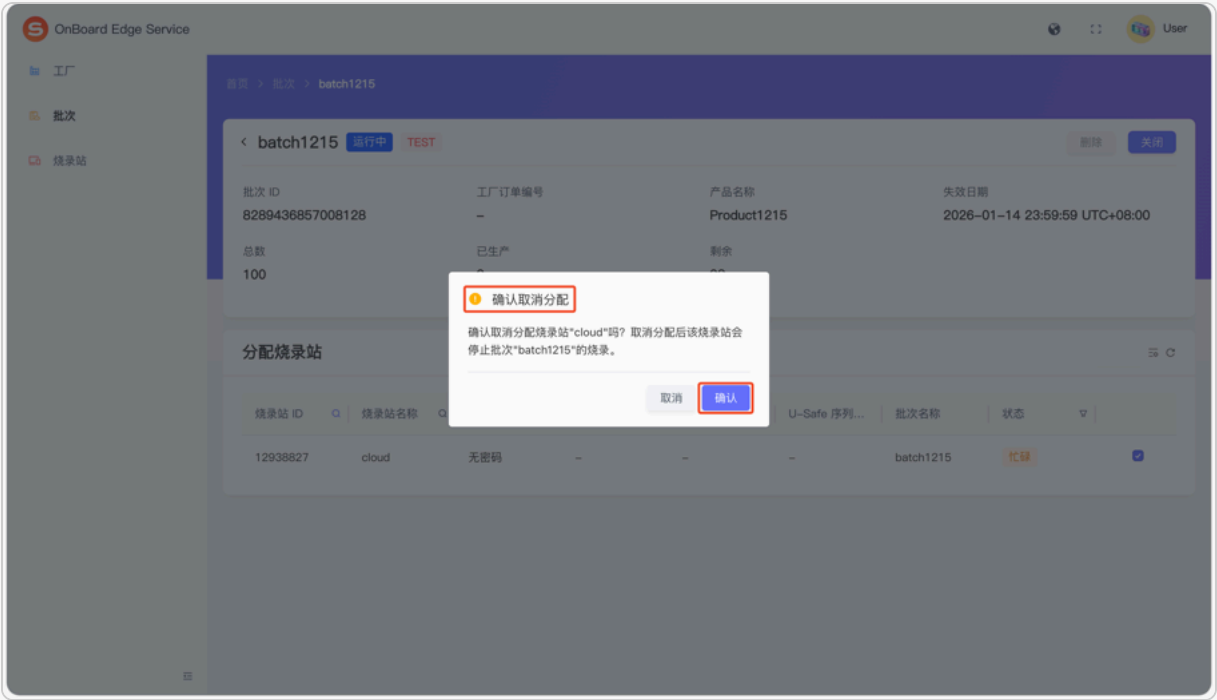
为批次取消分配烧录站

> **注意**
>
> 取消分配后，对应的烧录站无法继续用于该批次的烧录。

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**批次**，然后点击对应的批次 ID，进入批次页面。
3. 在批次页面的**分配烧录站**区域，找到对应批次，取消勾选状态为**忙碌**的烧录站。



4. 在**确认取消分配**窗口，点击**确认**，即可完成烧录站的取消分配。取消分配后，烧录站状态会变更为**可用**。

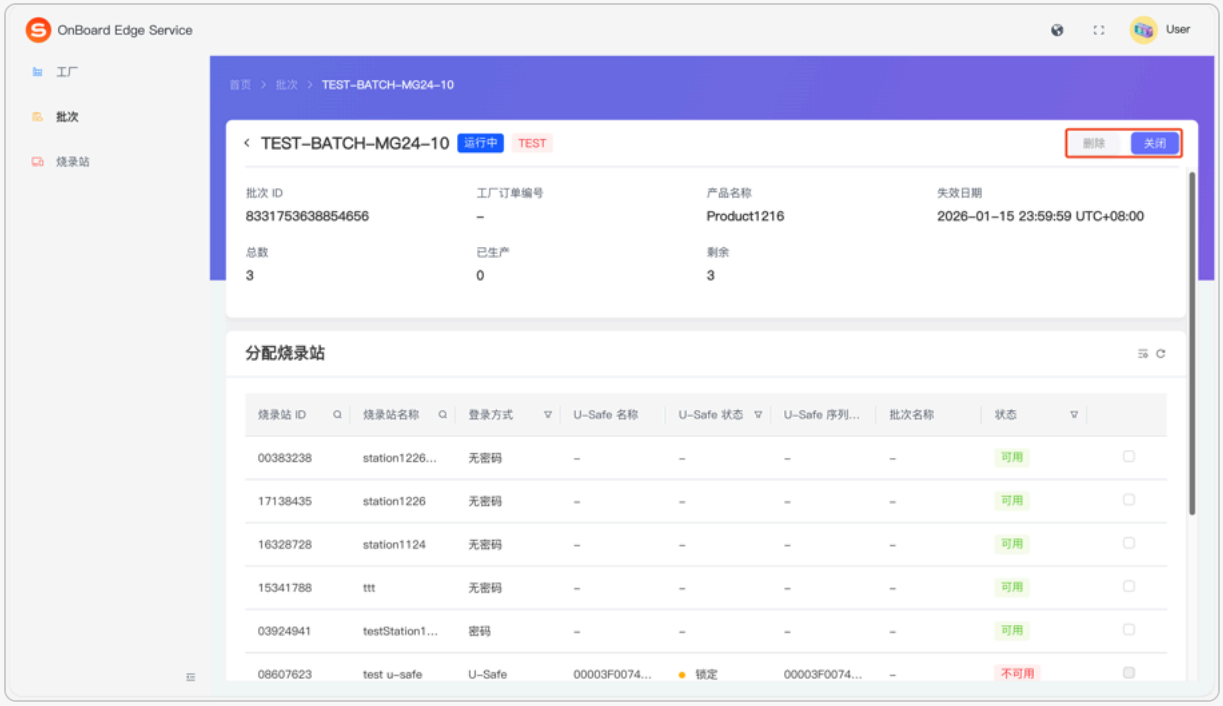


关闭或删除批次

> **信息**
>
> 批次删除后，该批次下已分配的烧录站将自动解除分配。

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**批次**，然后点击对应的批次 ID，进入批次页面。
3. 在批次页面，点击**关闭**或**删除**。

> **提示**
>
> 若**删除**按钮显示为灰色，表示当前不可用。请点击**关闭**将批次关闭。当批次处于非运行状态时，即可删除批次。


OnBoard Edge Service
User
工厂
批次
烧录站
首页 > 批次 > TEST-BATCH-MG24-10
TEST-BATCH-MG24-10
运行中
TEST
删除
关闭
批次 ID
8331753638854656
工厂订单编号
-
产品名称
Product1216
失效日期
2026-01-15 23:59:59 UTC+08:00
总数
3
已生产
0
剩余
3
分配烧录站
烧录站 ID
烧录站名称
登录方式
U-Safe 名称
U-Safe 状态
U-Safe 序列...
批次名称
状态
00383238 station1226... 无密码 - - - - 可用
17138435 station1226 无密码 - - - - 可用
16328728 station1124 无密码 - - - - 可用
15341788 ttt 无密码 - - - - 可用
03924941 testStation1... 密码 - - - - 可用
08607623 test u-safe U-Safe 00003F0074... 锁定 00003F0074... - 不可用

版本：v3.2.2

如何在 OnBoard Edge Service 中配置和管理烧录站

> **信息**
>
> OnBoard Edge Service 中的操作仅适用于本地工厂，云工厂的相关操作则可直接在 OnBoard™ IoT Security (OBIS) 中进行。

> **提示**
>
> 你可以根据实际需求，参考本文步骤在 OnBoard Edge Service 配置和管理烧录站，或者在 OBIS 配置和管理烧录站。

本文将指导你在 OnBoard Edge Service 中完成以下操作：

- 创建烧录站
- 为烧录站绑定 Identity U-Safe
- 为烧录站解绑 Identity U-Safe
- 删除烧录站
- 查看烧录站
- 查看烧录站密码

前提条件

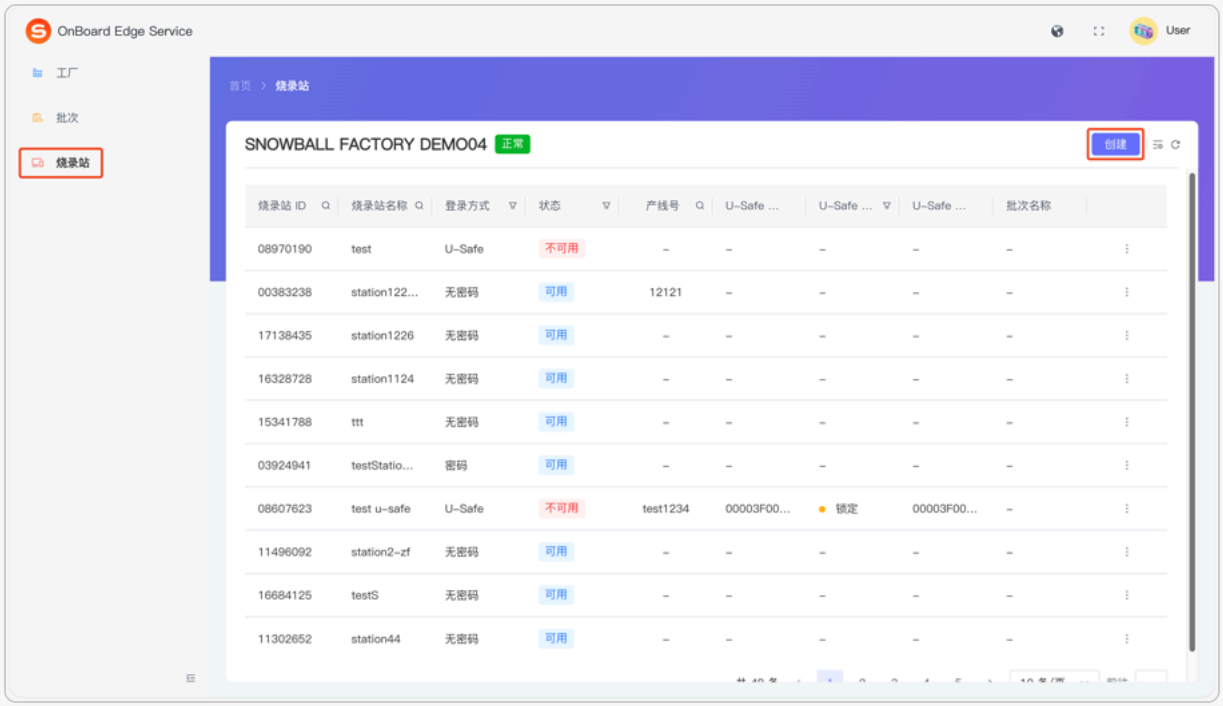
在开始之前，请确保：

- 你可以登录 OnBoard Edge Service。
- 已从雪球获取用于身份鉴权的 Identity U-Safe。

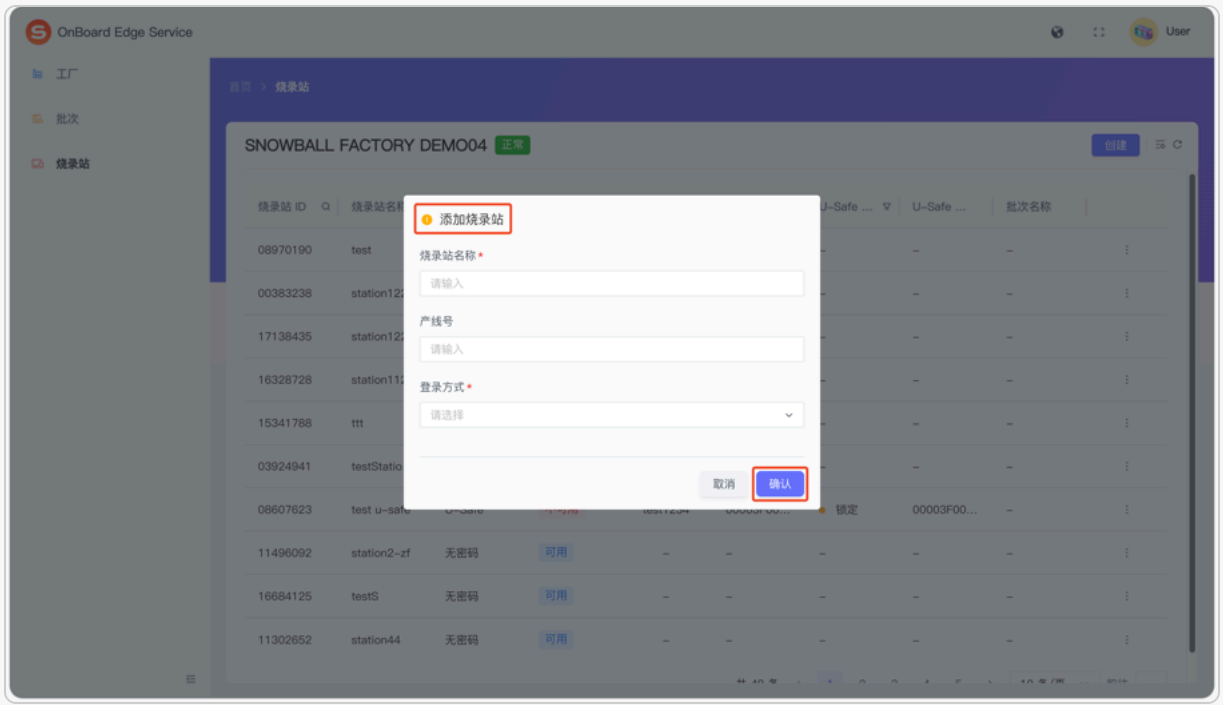
创建烧录站

> **提示**
>
> 为工厂创建烧录站后，你可以根据实际需求在 OnBoard Edge Service 为批次分配具体的烧录站。执行设备烧录前，需要根据烧录站 ID 查询并获取对应的批次信息和烧录数据。
>
> 支持两种烧录方式：
>
> - 使用烧录客户端（Programming Station）
> - 使用命令行工具（OBIS Programming CLI）

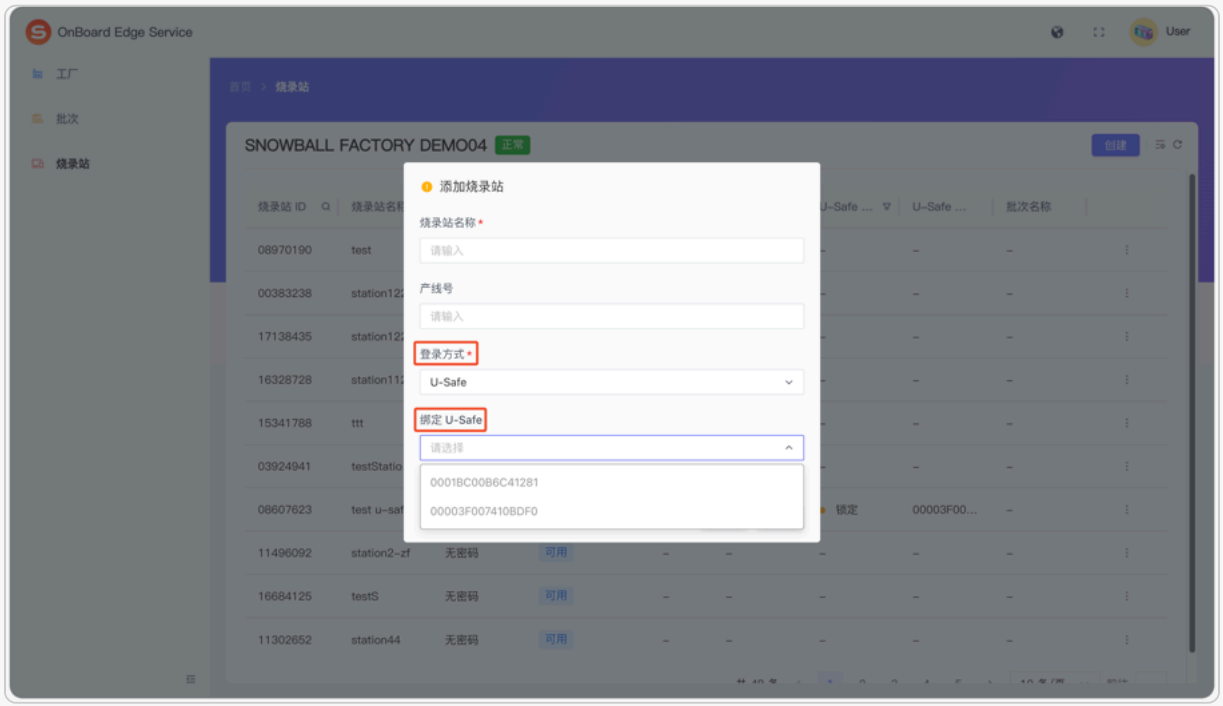
1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**烧录站**，在**烧录站**页面，点击**创建**。



3. 在**添加烧录站**窗口，填写**烧录站名称**和**产线号**，并选择**登录方式**。**登录方式**可选**无密码**，**密码**或 **U-Safe**。



> **提示**
>
> 如果登录方式为 **U-Safe**，你可以在**绑定 U-Safe** 下拉框中选择已添加的 Identity U-Safe 进行绑定，也可以在烧录站添加后再进行绑定。



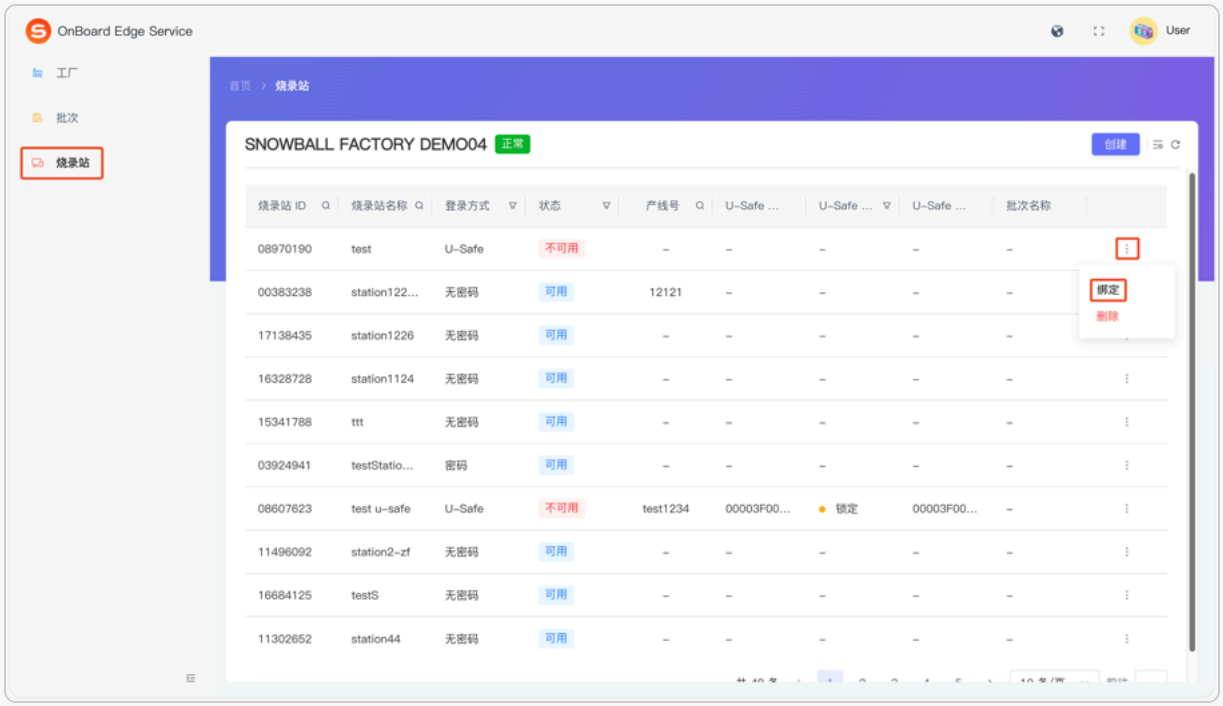
4. 点击**确认**，即可完成烧录站的创建。

为烧录站绑定 Identity U-Safe

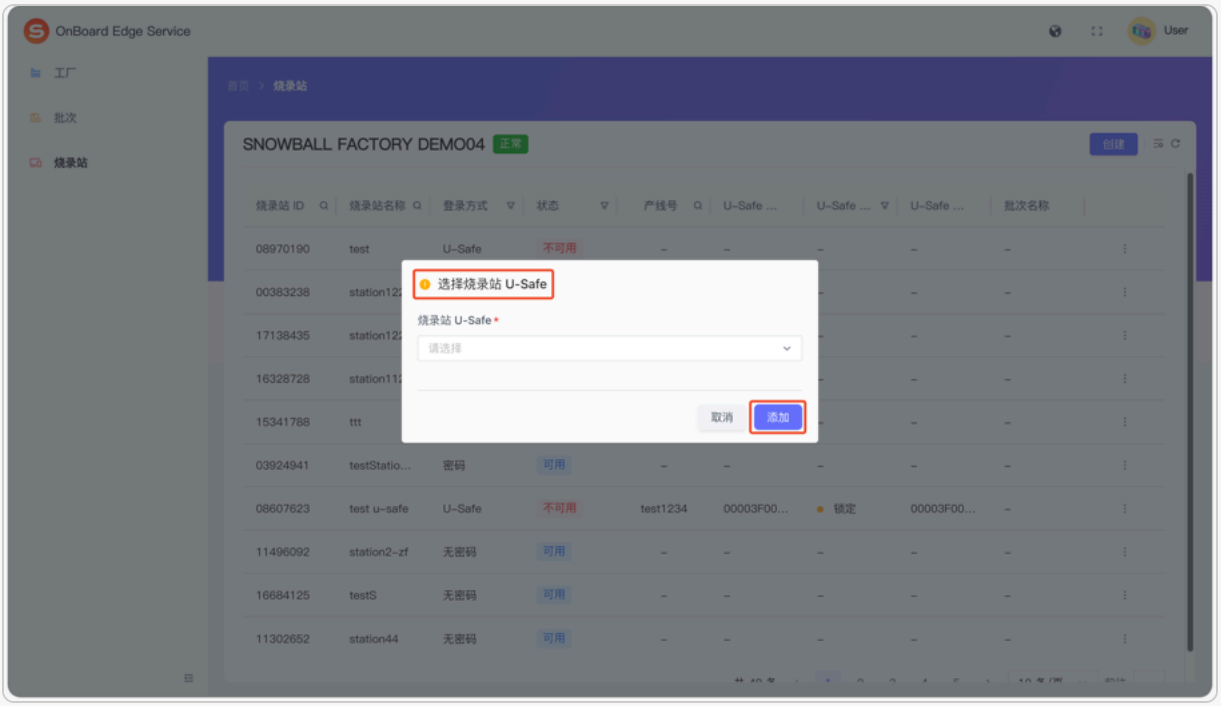
> **提示**
>
> 绑定后，可使用 Identity U-Safe 登录 Programming Station 客户端。

1. 登录 OnBoard Edge Service。

2. 点击左侧菜单中的**烧录站**，在**烧录站**页面，点击对应烧录站的选项 ⋮ 图标并选择**绑定**。



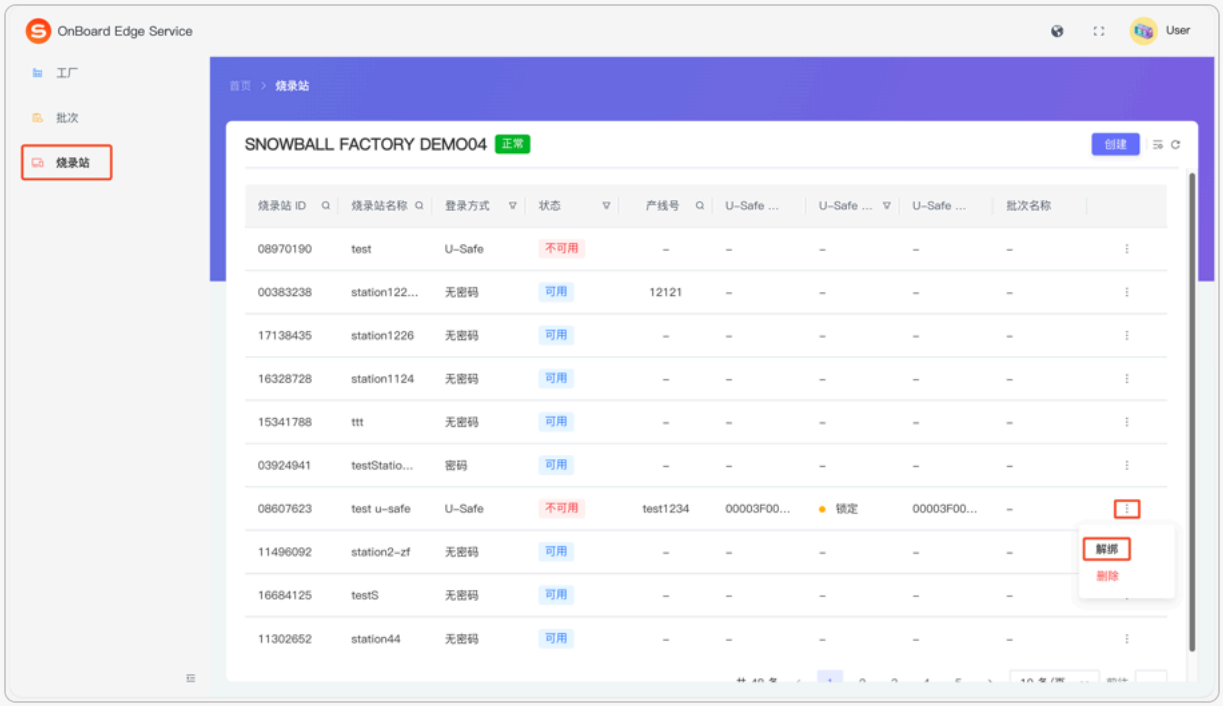
3. 在**选择烧录站 U-Safe** 窗口中，选择要绑定的 Identity U-Safe 并点击**添加**，即可为该烧录站绑定 Identity U-Safe。



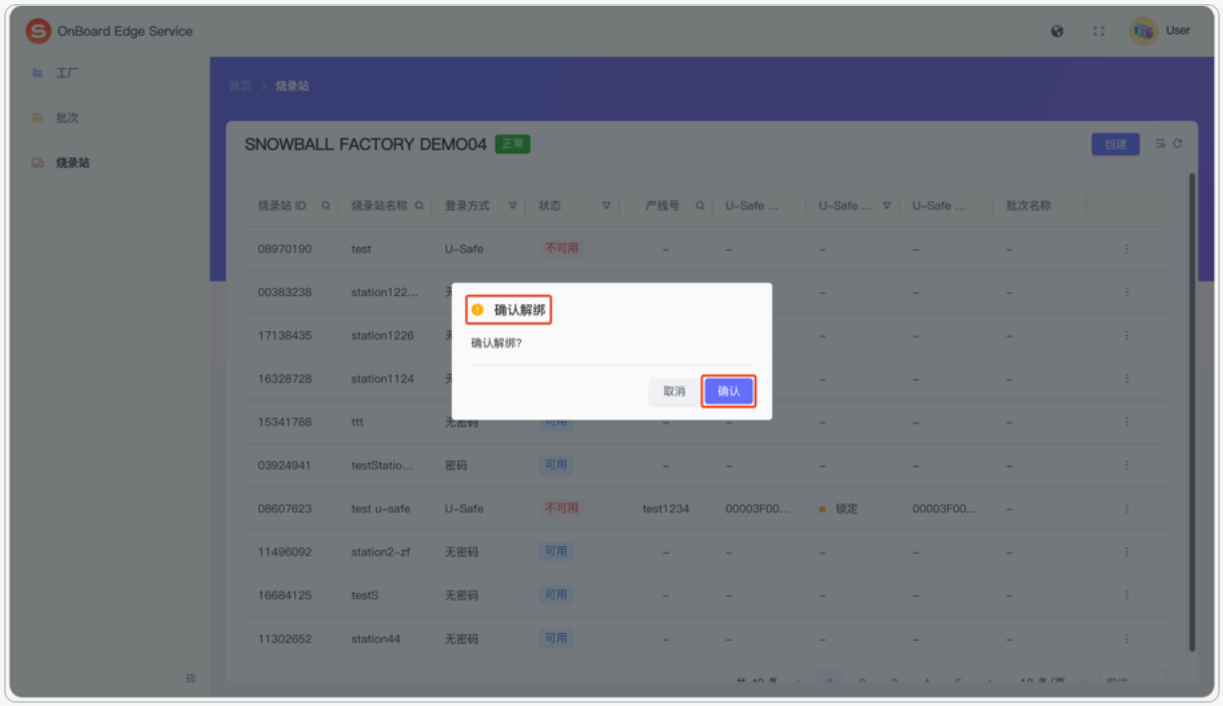
为烧录站解绑 Identity U-Safe

> **注意**
>
> 解绑后，不可使用 Identity U-Safe 登录 Programming Station。

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**烧录站**，在**烧录站**页面，点击对应烧录站的选项 ⋮ 图标并选择**解绑**。

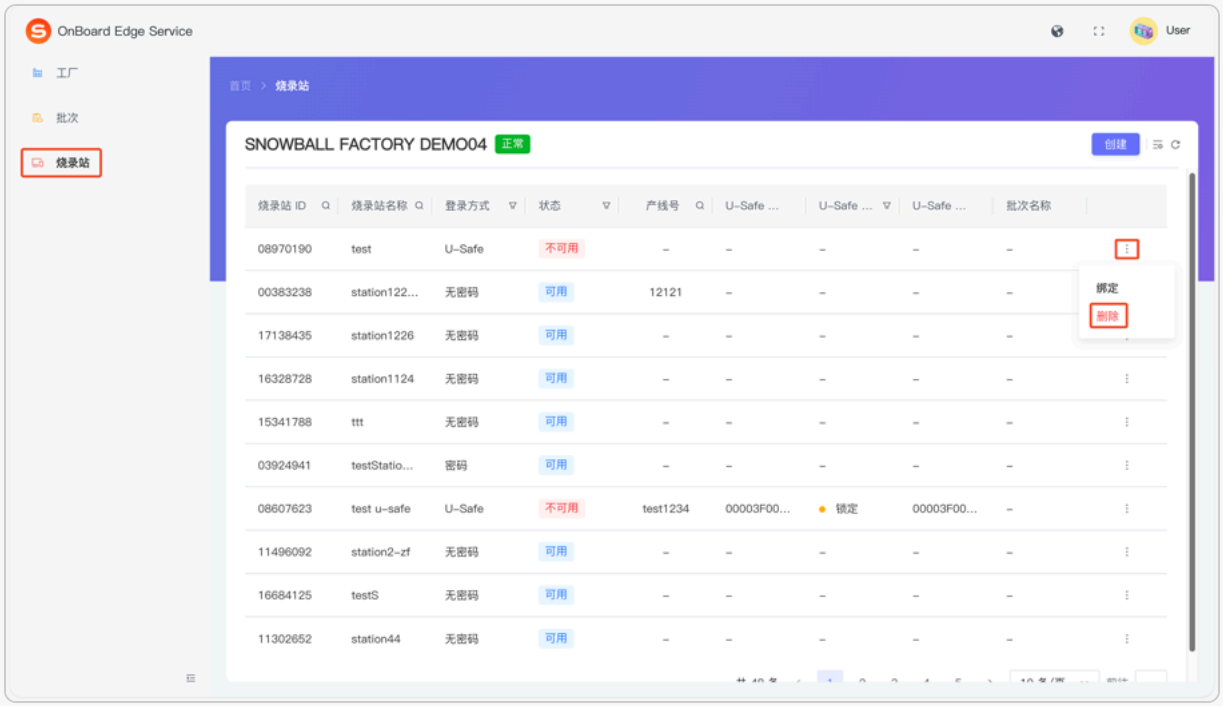


3. 在**确认解绑**窗口中点击**确认**，即可为该烧录站解绑 Identity U-Safe。

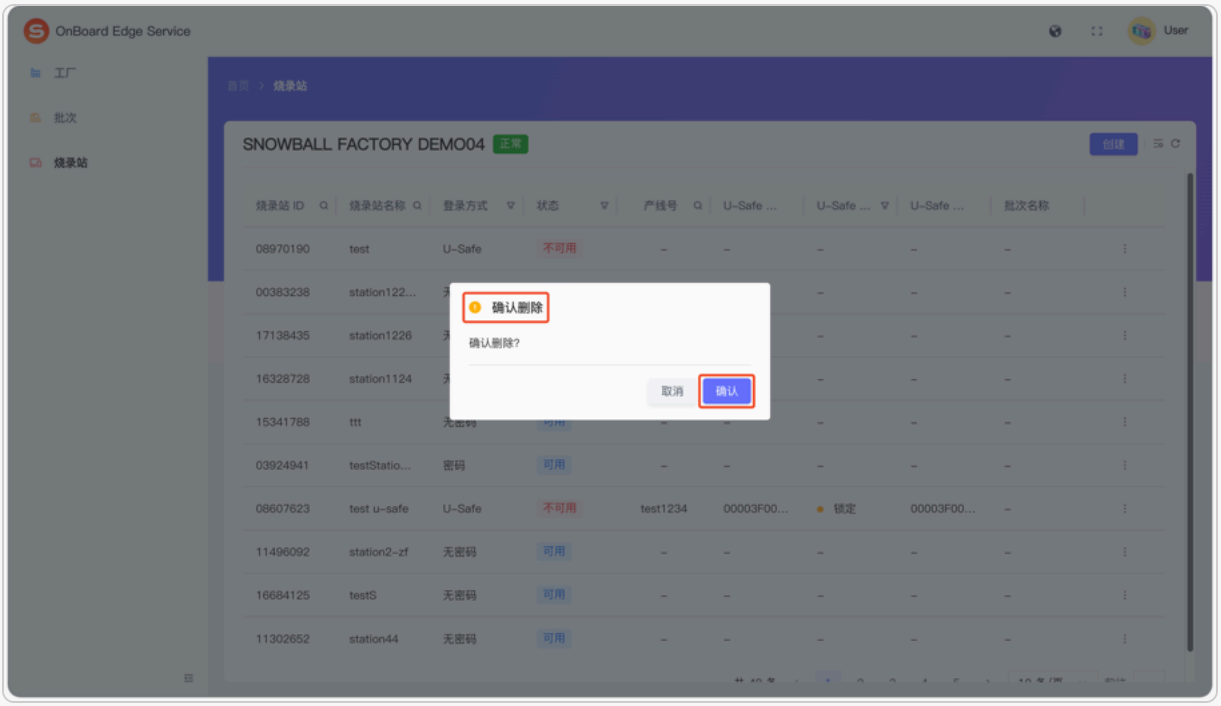


删除烧录站

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**烧录站**，在**烧录站**页面，点击对应烧录站的选项 ⋮ 图标并选择**删除**。



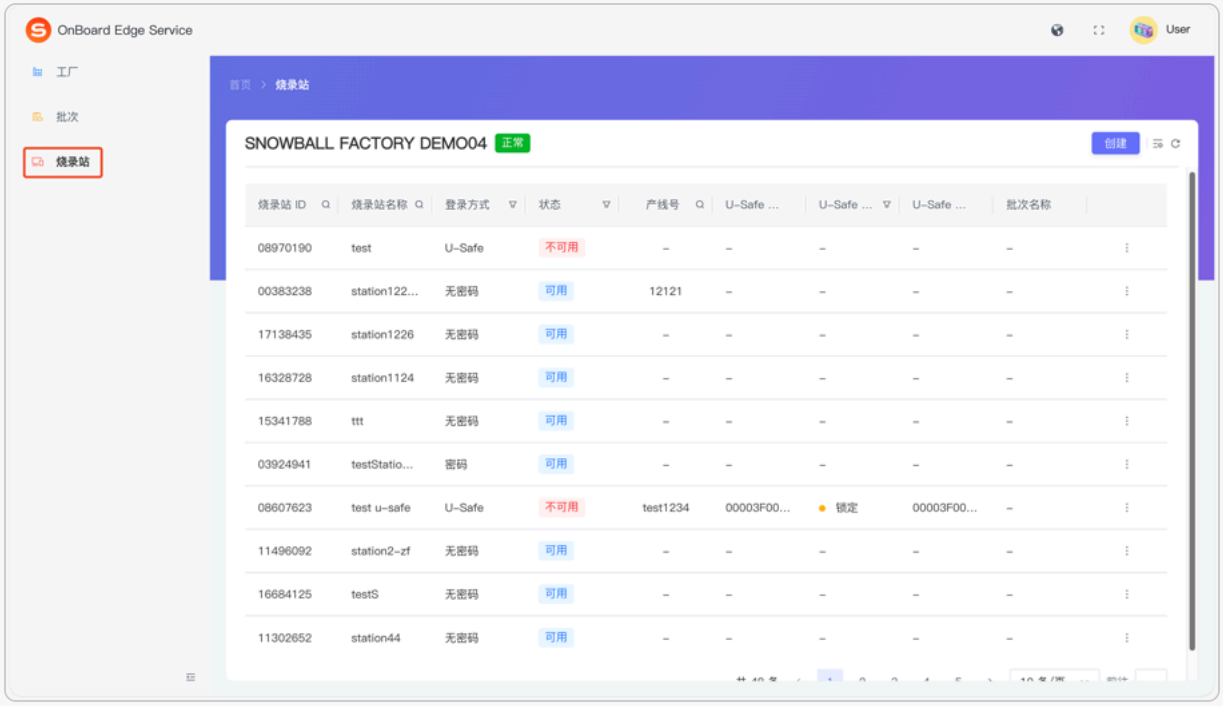
3. 在**确认删除**窗口中点击**确认**，即可删除该烧录站。



查看烧录站信息

1. 登录 OnBoard Edge Service。

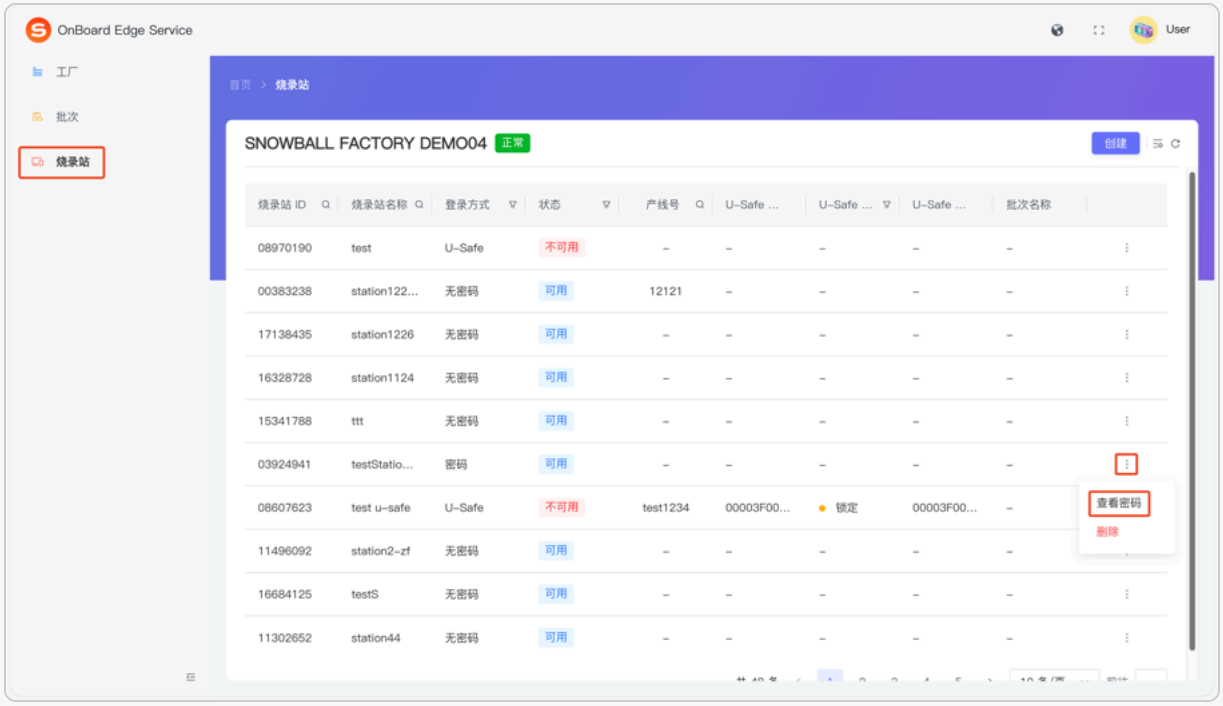
2. 点击左侧菜单中的**烧录站**，在**烧录站**页面，即可查看烧录站的相关信息（**烧录站 ID**，**烧录站名**称和**登录方式**等）。



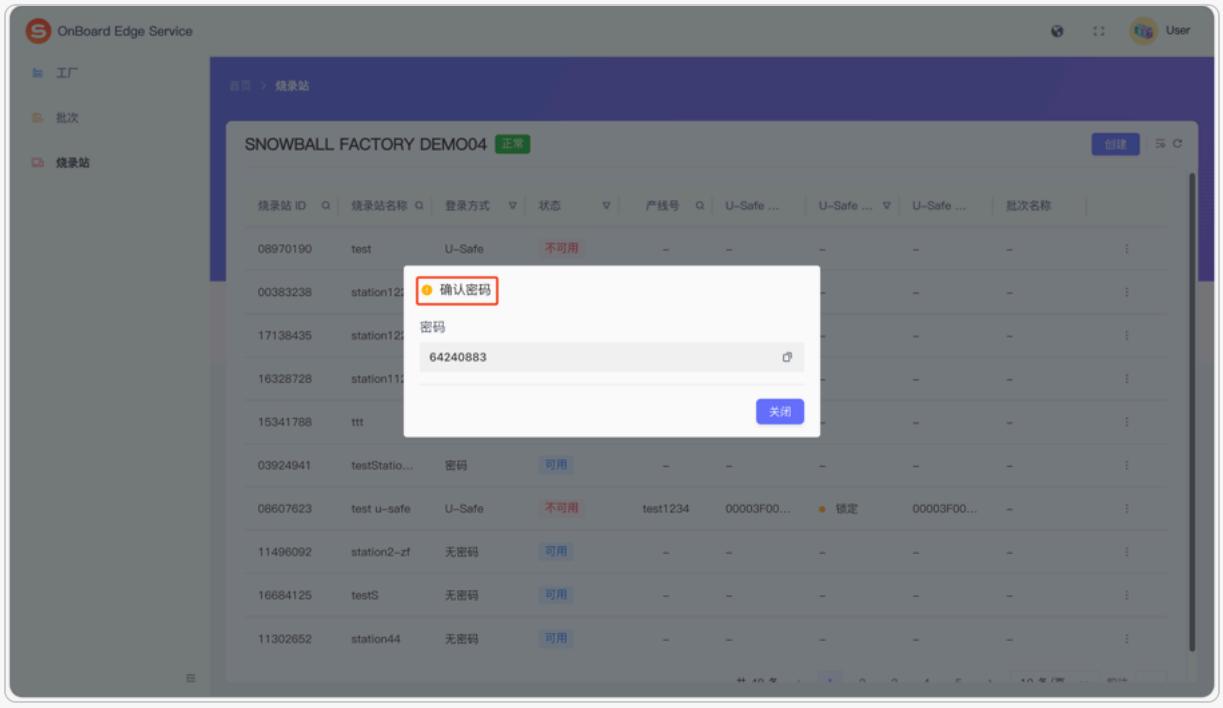
查看烧录站密码

> **提示**
>
> 当烧录站的登录方式为密码登录，系统会生成一个随机密码，该密码用于登录 Programming Station 客户端。该随机密码目前暂不支持修改。

1. 登录 OnBoard Edge Service。
2. 点击左侧菜单中的**烧录站**，在**烧录站**页面，点击对应烧录站的选项 ⋮ 图标并选择**查看密码**。



3. 在确认密码窗口，即可显示并复制烧录站的登录密码。



版本：v3.2.2

Programming Station 的概述

> **信息**
>
> Programming Station 客户端目前仅支持 Windows 操作系统。

Programming Station 是“最后一公里”执行端点，用于对设备进行物理烧录和配置。Programming Station 软件用于连接烧录设备并执行各类芯片烧录任务。作为整个芯片烧录流程中直接面向操作员的关键环节，它提供了简洁直观的操作环境，使操作员无需深入了解复杂的底层技术，即可轻松完成芯片烧录相关的所有操作。

Programming Station 软件通过分层组件架构封装并管理现有芯片厂商和烧录器厂商的工具，实现对不同芯片和产品需求的灵活支持，概况如下表所示：

组件	提供方	作用
Programming Station 客户端	OBIS	为烧录工程师提供用户界面 • 执行分配的生产批次，或手动输入生产批次标识或生产订单标识，以下载对应的烧录站软件包（该软件包整合了执行流程控制脚本 CLI 和芯片适配 CLI 工具） • 配置通道设置及 USB 端口与通道映射

组件	提供方	作用
Execution Process Control Scripts CLI	OEM	由 OEM 开发的脚本，用于编排烧录与设备配置流程
Chip Adaptation CLI	芯片厂商	芯片厂商提供的工具软件，支持特定芯片的烧录与配置接口
Programmer CLI	烧录器厂商	（如适用）烧录器厂商提供的驱动与工具（如 J-Link），直接与物理烧录硬件交互

这种分层架构使 OBIS 能够支持广泛的芯片平台和烧录器硬件，同时为所有配置操作保持一致的、集中化的管理模式。

版本：v3.2.2

如何安装 Programming Station

> **信息**
>
> Programming Station 客户端目前仅支持 Windows 操作系统。

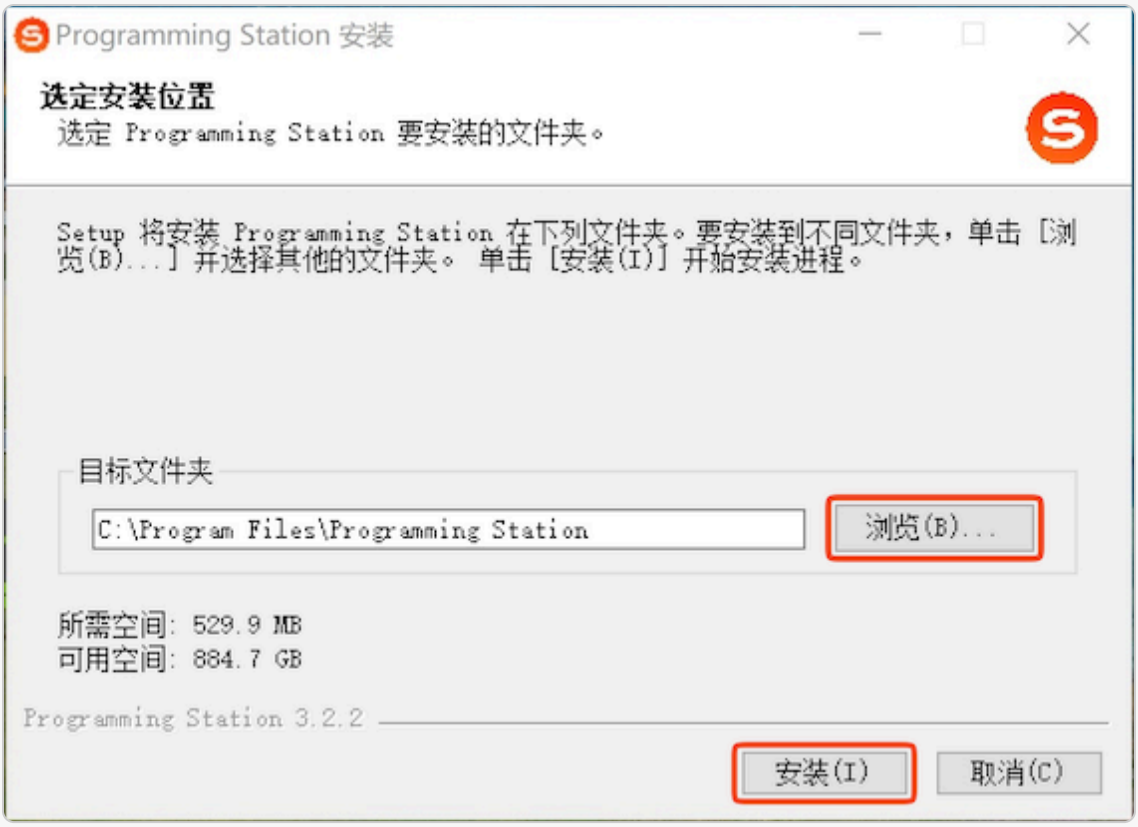
本文将指导你如何安装 Programming Station 客户端。

前提条件

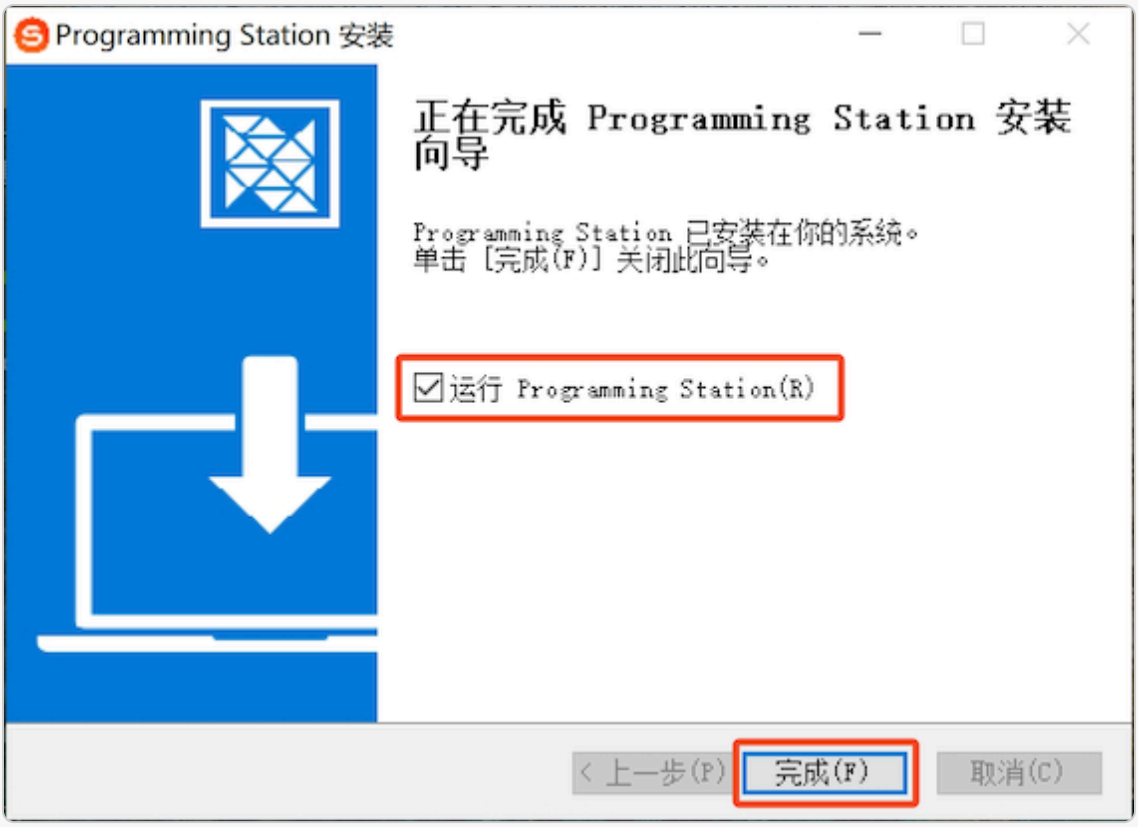
在开始之前，请确保你已从雪球获取 Programming Station 客户端的安装包。

操作步骤

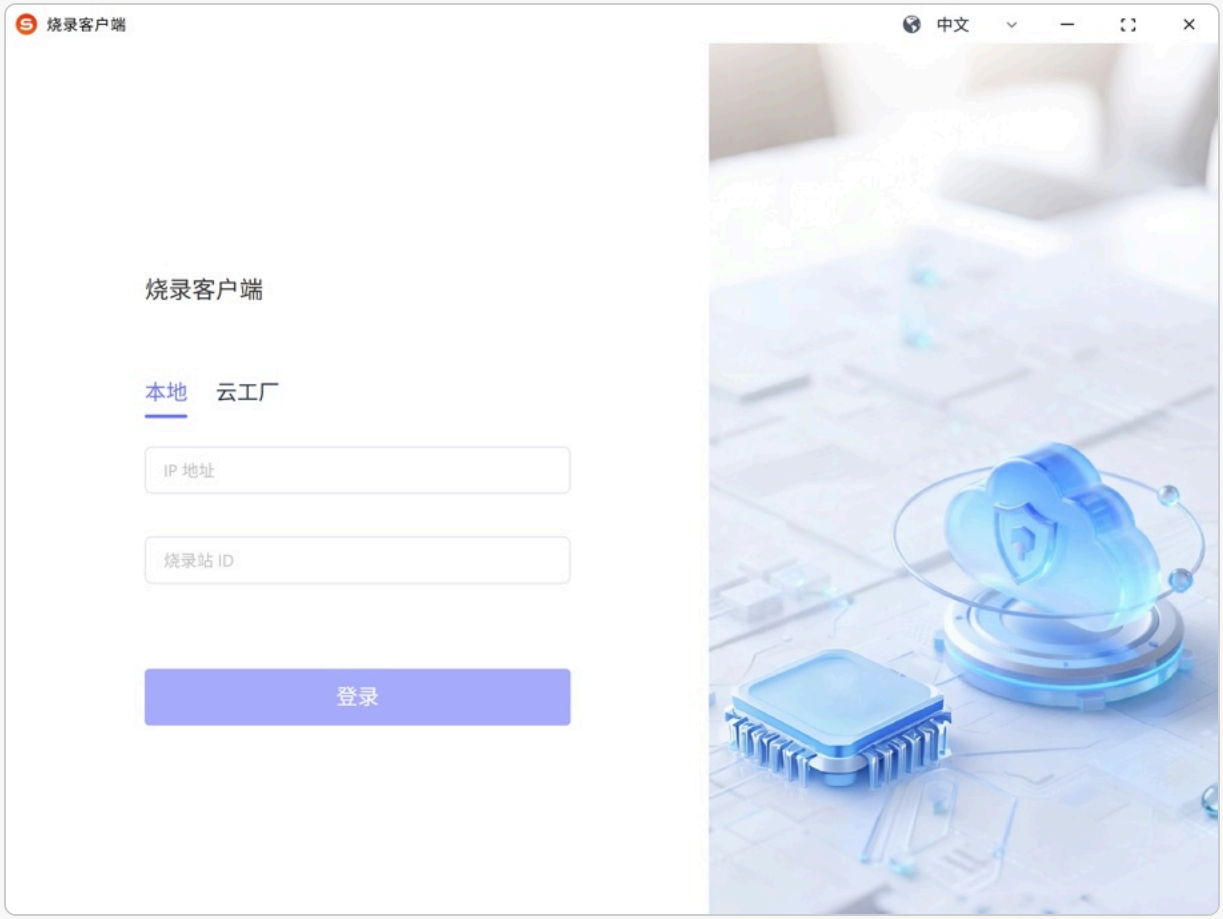
1. 运行 Programming Station 客户端的安装包以启动安装程序。
2. 在**选定安装位置**窗口，点击**浏览**为客户端指定一个安装位置（或使用默认路径），然后点击**安装**。



3. 在弹出的窗口中，你可以根据实际需求勾选**运行 Programming Station**，然后点击**完成**。



4. 勾选**运行 Programming Station** 后，Programming Station 客户端将自动运行。


烧录客户端
中文
烧录客户端
本地
云工厂
IP 地址
烧录站 ID
登录

版本：v3.2.2

如何登录 Programming Station

> **信息**
>
> Programming Station 客户端目前仅支持 Windows 操作系统。

本文将指导你如何使用无密码，密码或 Identity U-Safe 的方式登录 Programming Station 客户端。

前提条件

在开始之前，请确保：

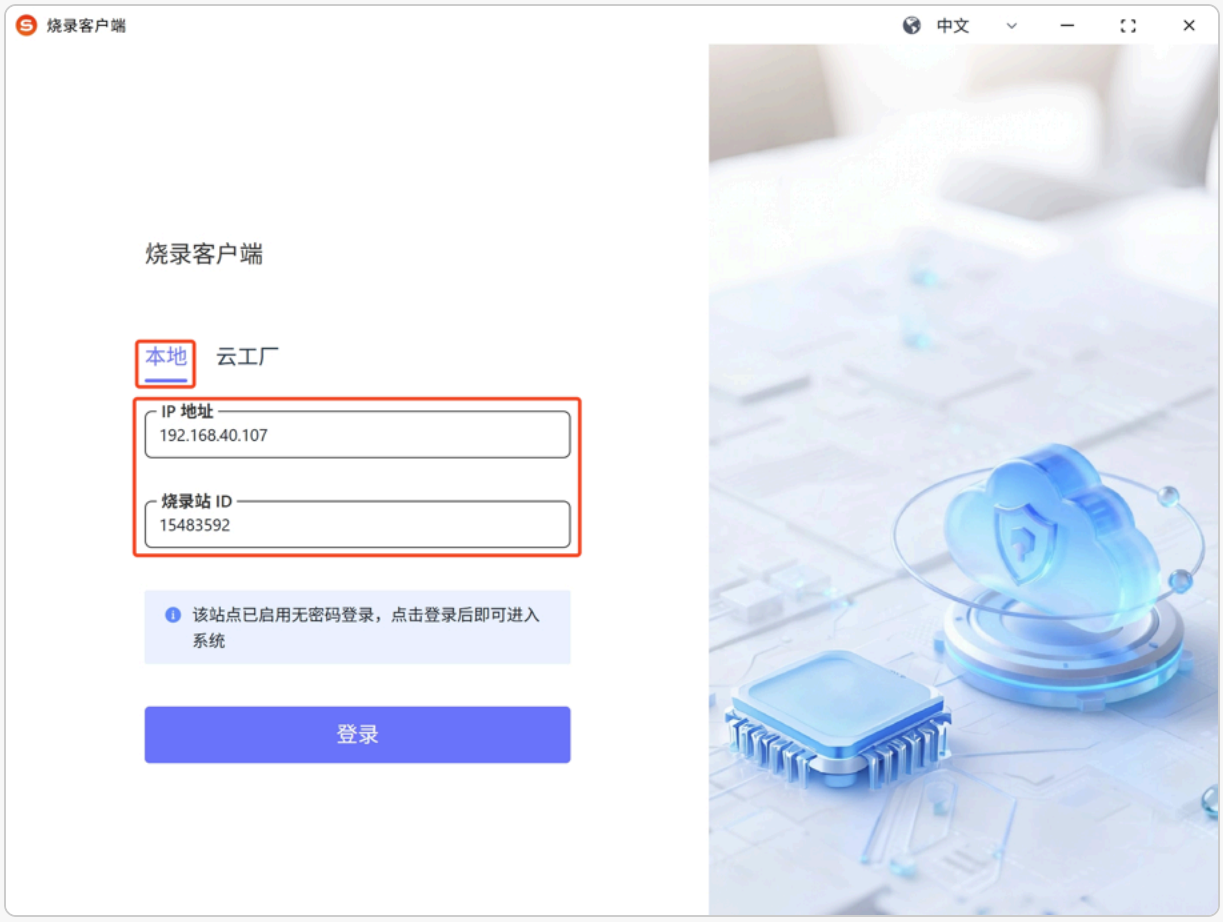
- 已在 OnBoard™ IoT Security (OBIS) (本地工厂 & 云工厂) 或 OnBoard Edge Service 完成烧录站的添加。
- 已安装 Programming Station 客户端。

本地工厂

> **提示**
>
> 请确保已联系雪球协助完成 OnBoard Edge Service 的部署，并且你的登录设备处于工厂服务内网环境。

方式一：无密码登录

1. 运行 Programming Station 客户端，然后选择**本地**。
2. 在服务器地址栏中输入 IP 地址（例如，192.168.0.1）。服务地址的获取方法请参考查看服务地址。



3. 输入对应的烧录站 ID，然后点击**登录**。登录后，页面会跳转至 Programming Station 客户端的主界面。



方式二：密码登录

> **提示**
>
> 初始登录密码是在工厂添加烧录站时由系统随机生成的，你可以在 OBIS 或 OnBoard Edge Service 查看烧录站登录密码。该随机密码目前暂不支持修改。

1. 运行 Programming Station 客户端，然后选择**本地**。
2. 在服务器地址栏中输入 IP 地址（例如，192.168.0.1）。服务地址的获取方法请参考查看服务地址。



3. 输入对应的烧录站 ID 和密码，然后点击**登录**。登录后，页面会跳转至 Programming Station 客户端的主界面。

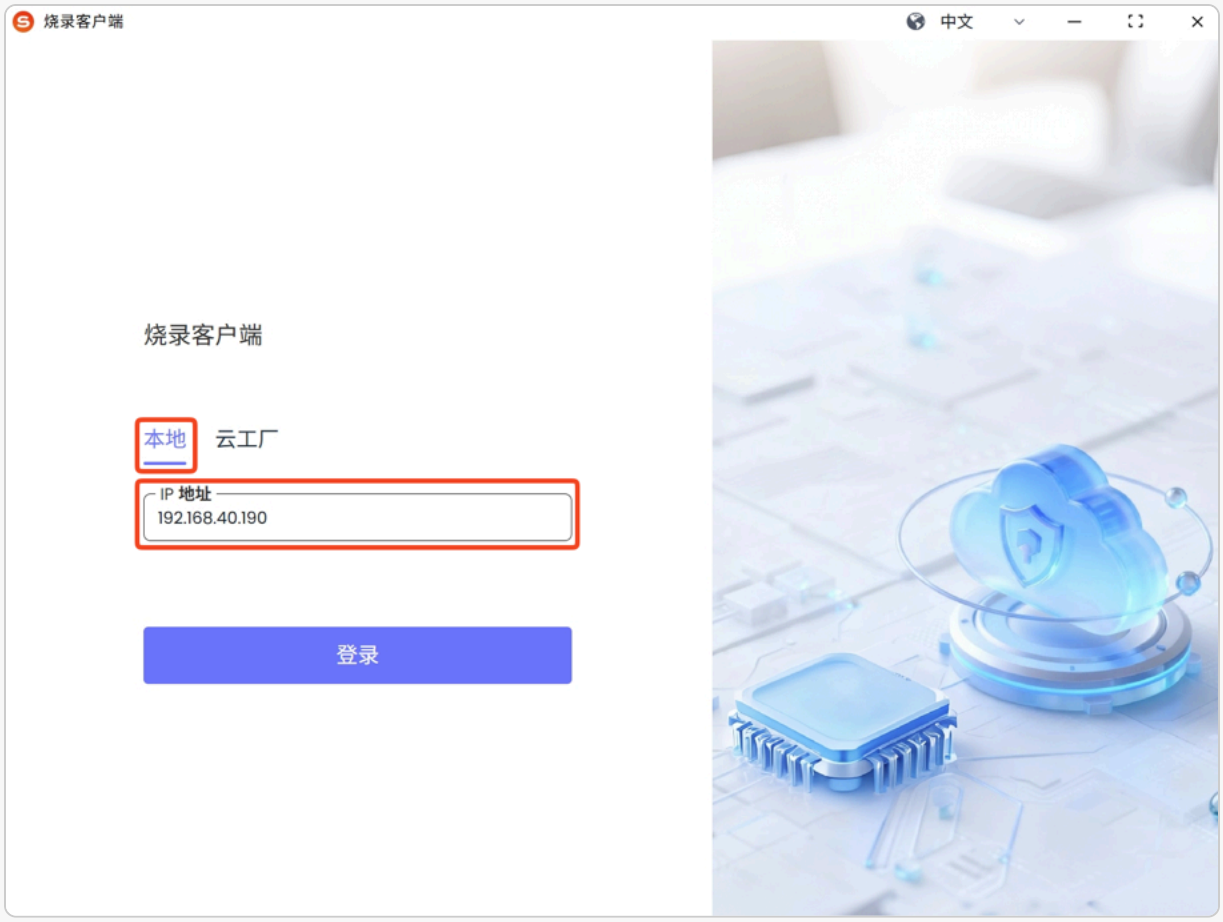


方式三：Identity U-Safe 登录

> **信息**
>
> 使用 Identity U-Safe 的方式登录时，需要从雪球获取用于身份鉴权的 Identify U-Safe，并将对应的 Identify U-Safe（在配置工厂时添加）插入已安装 Programming Station 客户端的设备。插入后，若 Identiy U-Safe 已绑定烧录站，客户端将识别烧录站 ID。
>
> 同一时间只能插入一个 Identify U-Safe，否则将无法登录。

1. 运行 Programming Station 客户端，然后选择**本地**并插入 Identify U-Safe。

2. 在服务器地址栏中输入 IP 地址（例如，192.168.0.1）。服务地址的获取方法请参考查看服务地址。



3. 点击**登录**。登录后，页面会跳转至 Programming Station 客户端的主界面。



云工厂

方式一：密码登录

> **提示**
>
> 初始登录密码是在工厂添加烧录站时由系统随机生成的，你可以在 OBIS 上查看烧录站登录密码。该随机密码目前暂不支持修改。

1. 运行 Programming Station 客户端，选择**云工厂**，然后输入对应的工厂 ID。



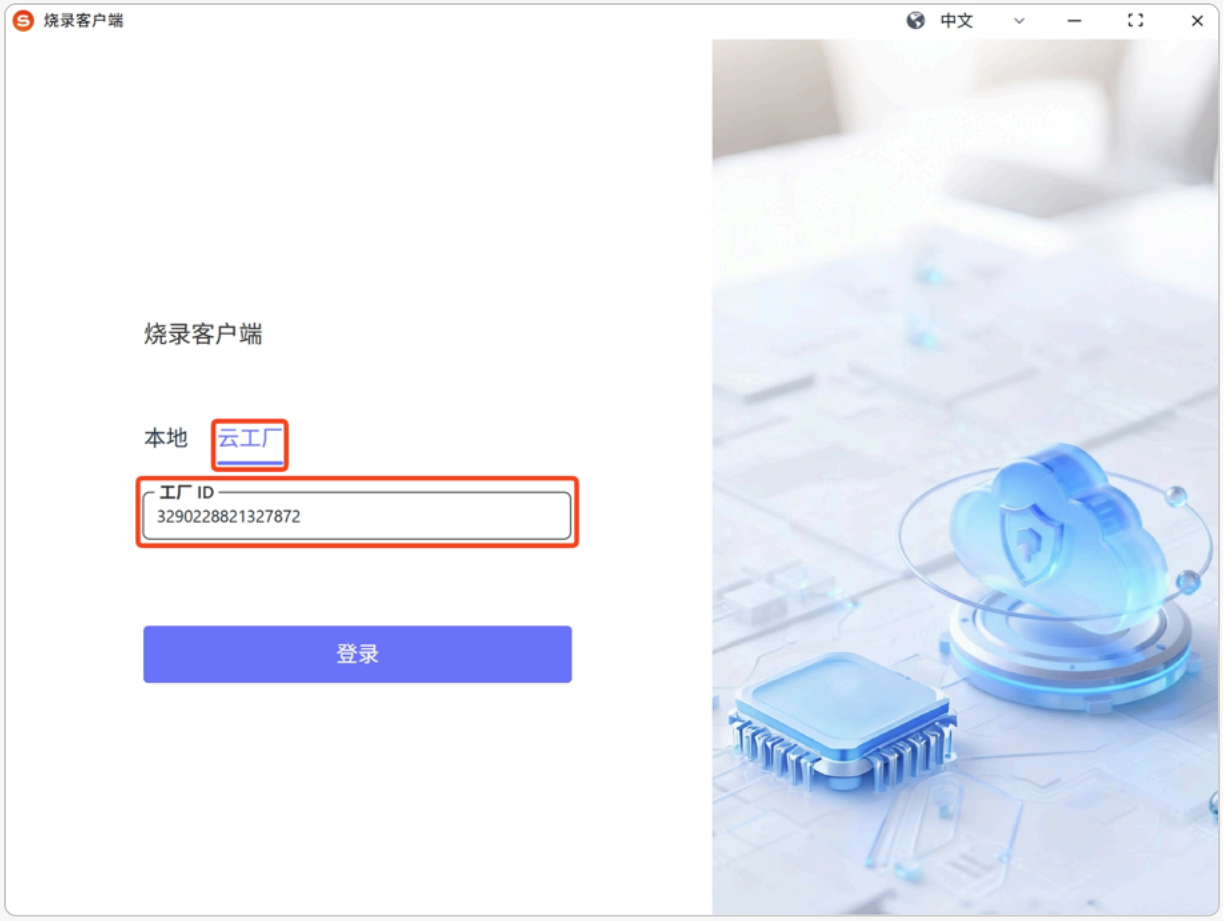
2. 输入对应的烧录站 ID 和密码，然后点击**登录**。登录后，页面会跳转至 Programming Station 客户端的主界面。



方式二：Identity U-Safe 登录

> **信息**
>
> 使用 Identity U-Safe 的方式登录时，需要从雪球获取用于身份鉴权的 Identify U-Safe，并将对应的 Identify U-Safe（在配置工厂时添加） Programming Station 客户端的设备。插入后，若 Identiy U-Safe 已绑定烧录站，客户端将识别烧录站 ID。同一时间只能插入一个 Identify U-Safe，否则将无法登录。

1. 运行 Programming Station 客户端，选择**云工厂**，然后输入对应的工厂 ID 并插入 Identify U-Safe。



2. 点击**登录**。登录后，页面会跳转至 Programming Station 客户端的主界面。


烧录客户端
烧录主控
批次详情
端口配置
TEST-QORVO-...
生产
批次 ID:
8806735210872832
订单编号:
LOS-ANGELES-001
脚本:
Qorvo-0115-v1.0.zip
开始
当前批次未完成，可以继续烧录。
关于
SNOWBALL
3.2.2
TEST-QORVO-QPG6200-LOCAL-0219
总量
10
已生产
5
剩余
5
1
就绪
01
02
03
04
并发负载: 0%
批次已更新
中文
2 X 2

版本：v3.2.2

如何在 Programming Station 中执行设备烧录

> **信息**
>
> Programming Station 客户端目前仅支持 Windows 操作系统。

本文将指导你如何使用 Programming Station 客户端执行设备烧录。

前提条件

在开始之前，请确保你可以登录 Programming Station 客户端。

操作步骤

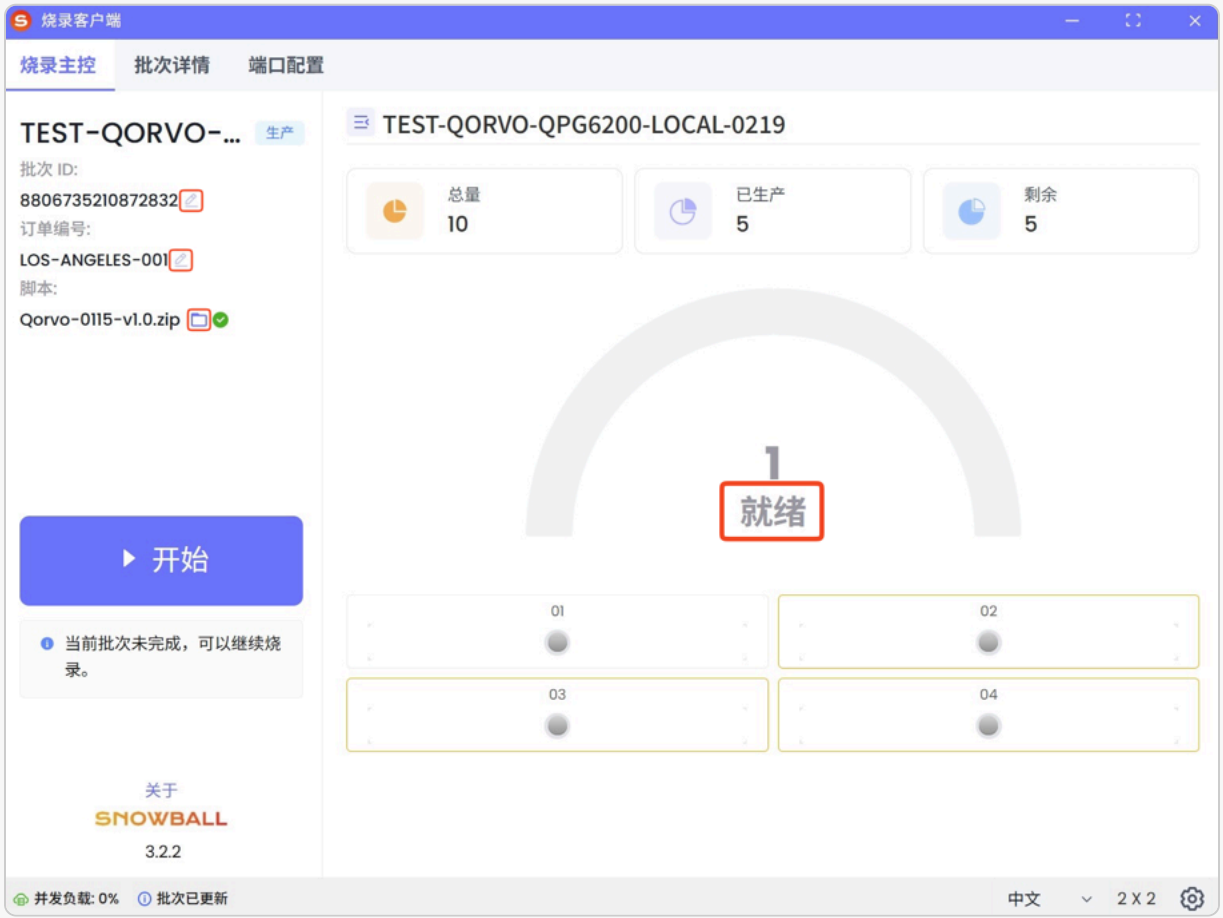
1. 登录 Programming Station 客户端，确认设备的状态为**就绪**。

提示

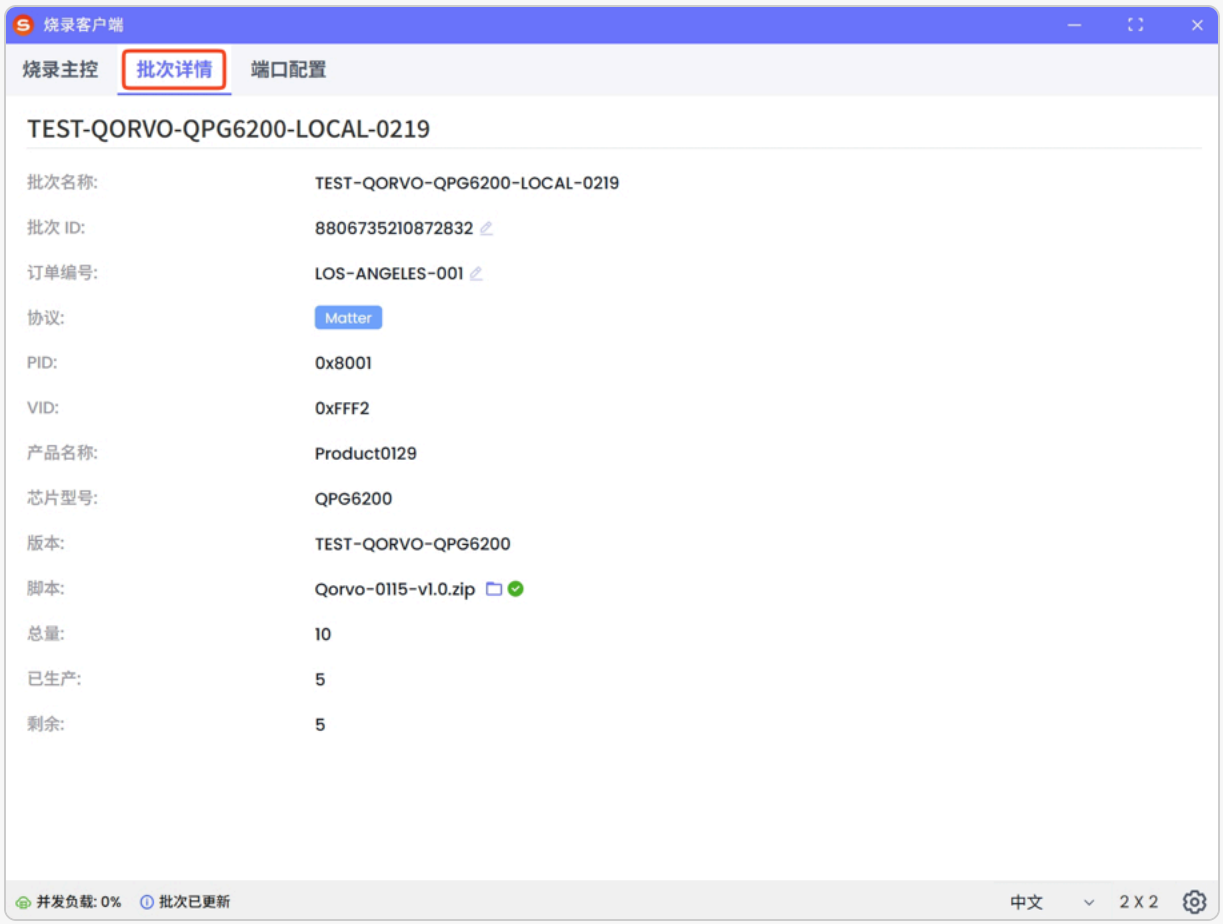
如果在创建批次时未选择烧录脚本，或者想更换为其他烧录脚本，可点击脚本区域右侧的文件夹图标，从本地选择所需的烧录脚本（任意格式）。另外，你可以点击右下角的设置图标，选择**脚本下载路径**，来自定义默认的脚本下载路径。

如果烧录站还未分配给任何批次：

- 点击**批次 ID** 区域右侧的编辑图标，然后输入批次 ID 从而获取对应的批次信息。
- 点击**订单编号**区域右侧的的编辑图标，输入订单编号并选择关联的批次名称，从而获取对应的批次信息。



2. 在**批次详情**标签页可以查看批次相关信息（**协议**，**VID**，和 **PID** 等）。



3. 在 Programming Station 客户端的主界面中，选择**端口配置**标签页，根据实际需求在布局区域设置烧录器连接数量。

提示

设定烧录器连接数量从而调整同时烧录的设备数量。当前系统支持设置的烧录器连接数量由**布局**区域的行数与列数的乘积决定，需同时满足以下条件：

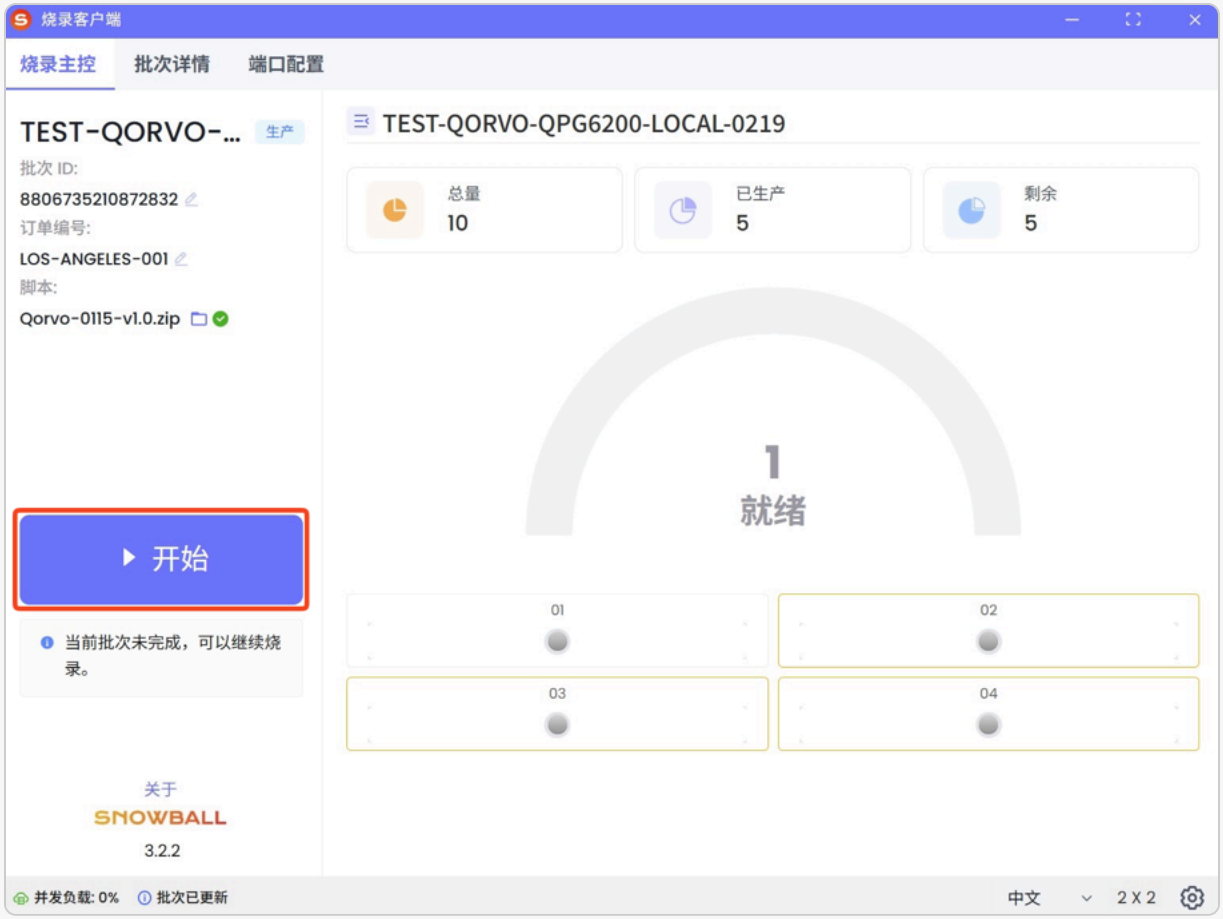
- 行数和列数必须为整数
- 行数 × 列数 ≤ 24
- 行数和列数均不得超过 16
- 当行的数大于 4 时，列数仅可为 1 或 2

端口可在**使用中端口**和**可用端口**之间拖拽并调整顺序，或通过下载并修改 Excel 模板（**Template.xlsx**）来更新顺序，调整后上传即可生效。



4. 确认信息无误后，在 Programming Station 客户端的主界面中点击**开始**，进行设备烧录。

> **提示**
>
> 客户端左下角的并发负载显示服务器在同时支持多个烧录站进行烧录时的负载情况：
>
> - 负载 ≥ 90%：
>   - 提示信息：“即将达到并发上限，请联系工厂管理员检查服务器”。
>   - 用户仍可继续烧录。
> - 负载 = 100%：
>   - 提示信息：“已达到并发上限，请联系工厂管理员检查服务器”。
>   - 用户仍可继续烧录。
>
> 烧录过程中无法切换到**端口配置**标签页。



5. 烧录结束后，点击对应的端口即可查看设备的烧录详情。

> **提示**
>
> 同时，也可在 OnBoard™ IoT Security (OBIS) 上查看设备档案及相关日志信息。


烧录客户端
烧录主控
批次详情
端口配置
TEST-QORVO-...
生产
批次 ID:
8806735210872832
订单编号:
LOS-ANGELES-001
脚本:
Qorvo-0115-v1.0.zip
1200000508
100%
成功
详情
hex"
"C:\Users\maryw\AppData\Roaming\SNB\ppcli\programmings\Qorvo\matter_factory_data_1200000508.hex"
[15:18:42] hexmerge:
[15:18:42] [Progress]:80% Executing command 9: program-hex
"C:\Users\maryw\AppData\Roaming\SNB\ppcli\programmings\Qorvo\qorvo_merged_QPG6200_1200000508.hex" --device QPG6200 --emulator-serialnumber 1200000508 --emulator-logs
"C:\Users\maryw\AppData\Roaming\SNB\Qorvo\emulator_logs"
[15:19:04] output:Programming
C:\Users\maryw\AppData\Roaming\SNB\ppcli\programmings\Qorvo\qorvo_merged_QPG6200_1200000508.hex Programming took 20.52 seconds. [Progress]:90%
[15:19:06] [Progress]:100% [Finish].
[15:19:08] out:b'{\r\n "success": true,\r\n "respCode": "8000",\r\n "respMsg": "NoError",\r\n "respData": null,\r\n "respTime": "1770189547"\r\n}\r\n'
[15:19:08] [EXIT] 0
[QRCode]: MT:634J0-C714EPGS7WE10
QRcode link
开始
当前批次未完成，可以继续烧录。
关于
SNOWBALL
3.2.2
并发负载: 0%
批次已更新
中文
2 X 2

版本：v3.2.2

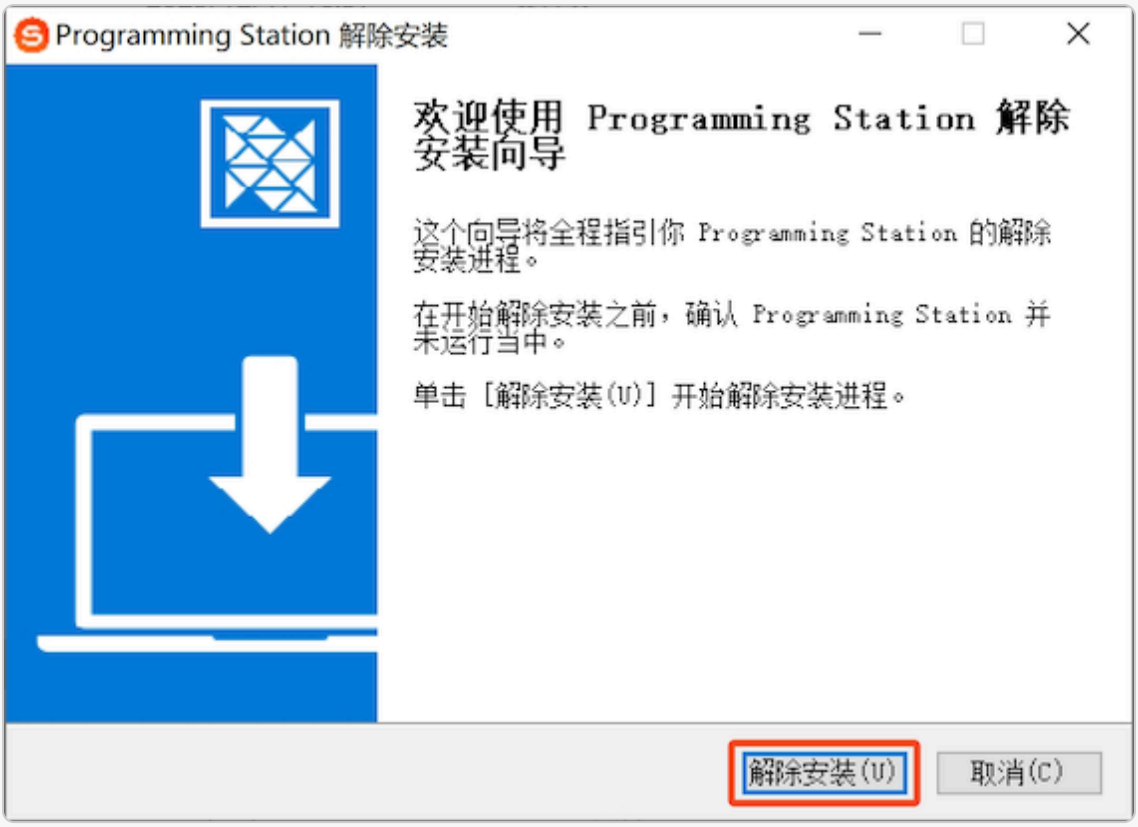
如何卸载 Programming Station

> **信息**
>
> Programming Station 客户端目前仅支持 Windows 操作系统。

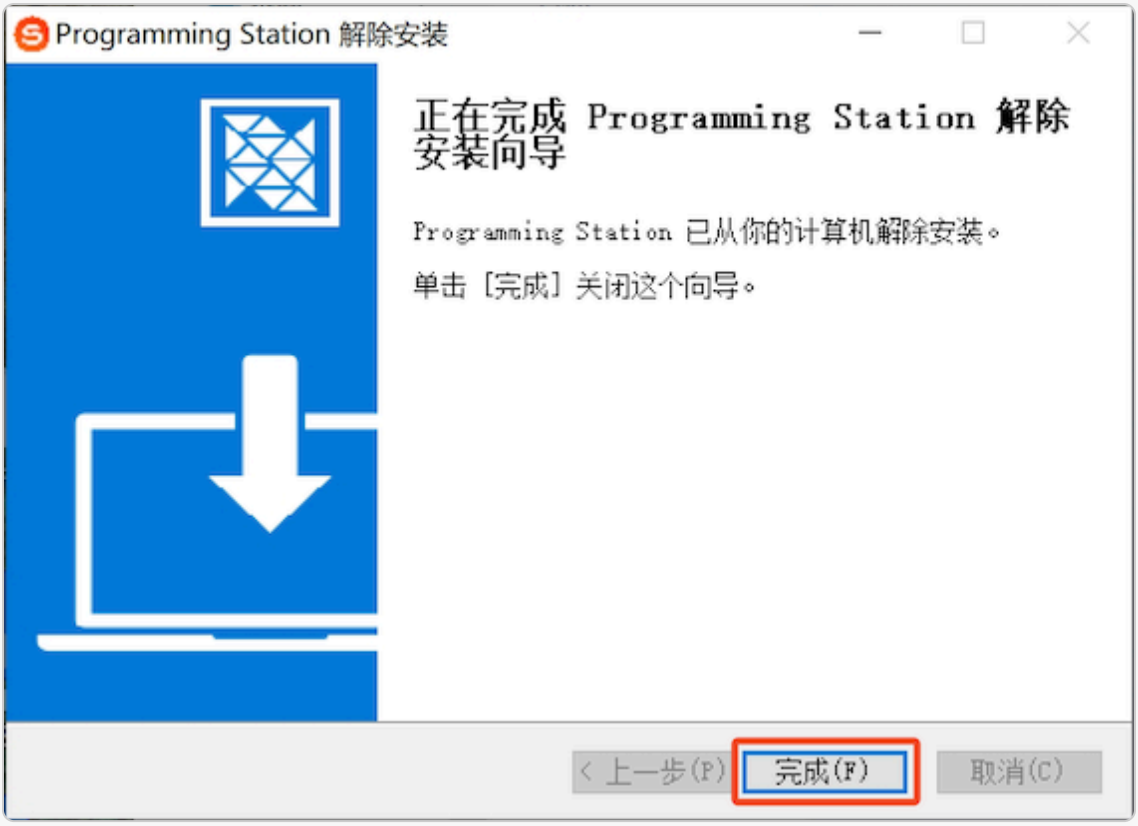
本文将指导你如何卸载 Programming Station 客户端。

操作步骤

1. 右击桌面或开始菜单中的客户端图标，然后在菜单中点击**打开文件所在的位置**。
2. 在安装目录中找到名为 **Uninstall Programming Station** 的文件并双击运行。
3. 在弹出的窗口中，点击**解除安装**以启动卸载。



4. 客户端卸载完成后，点击**完成**即可完成卸载。


Programming Station 解除安装
正在完成 Programming Station 解除安装向导
Programming Station 已从你的计算机解除安装。
单击［完成］关闭这个向导。
< 上一步(P)
完成(F)
取消(C)

版本：v3.2.2

Factory Edge Service API - v1.5

1.修订记录

版本	更改时间	备注
v1.0	2025/03/26	初始版本
v1.1	2025/12/17	增加"查询设备详情"接口「获取烧录状态可以使用该接口」
v1.2	2025/12/22	• 修改"查询设备详情"接口参数必填项校验和参数说明「支持使用 matter 的 qrcode 作为查询条件获取设备信息」 • 增加接口异常状态码信息 • 增加错误码列表
v1.3	2026/01/26	• 废弃"查询设备信息"和"更新设备信息"接口 • 增加"更新设备档案信息"接口
v1.4	2026/09/10	增加“上传设备生产数据”接口
v1.5	2026/10/16	• 更新“查询设备详情”、“更新设备档案信息”和“上传设备生产数据”接口说明 • 新增“查询设备综合信息”、“找回 Matter 二维码”、“上传设备测试文件”、“连接心跳”、“查询登录选项”、“查询工站登录方式”、“U-Safe 预登录”、“工站登录”、“工站退出登录”、“按订单查询批次”、“分配工站批次”、“查询工站批次信息”、“获取动态烧录数据”、“获取批次元数据”、“获取完整烧录数据（兼容接口）”、“提交中间烧录请求数据”、“上报烧录结果”、“SE APDU 步进烧录”、“获取烧录脚本地址”、“下载文件”、“上传设备事件日志”、“生成 Matter Verifier”、“生成 Matter 配网载荷”和“生成 Matter Unique ID”接口 • 补充已废弃接口“查询设备信息”和“更新设备信息”的兼容说明 • 补充公共调用约定及错误码说明

2. 概述

Factory Edge Service API 用于登录工站、管理生产批次、执行设备烧录、上传生产数据、查询设备信息及更新设备档案。该类接口采用 HTTP 协议传输数据，所有请求和响应均使用 UTF-8 编码。

本文将介绍接口规范、公共请求头，并提供各接口的请求参数、响应参数和调用示例。

3. 公共请求头

该类接口包含以下 Key （统一小写）：

Key	是否必填	说明
content-type	是	内容类型，除非特殊说明，否则固定使用 `application/json;charset=utf-8`
x-snbps-factory-id	否	工厂 ID
x-snbps-timestamp	否	请求时间。例如 2021-09-17T16:47:05.132+0800
x-snbps-lang	否	语言编号，默认使用英文：en-US，支持中文简体：zh-CN
x-snbps-request-id	否	请求 traceID

3.1 接口调用约定与鉴权

3.1.1 服务地址与请求格式

服务前缀为 `{BASE_URL}/api/iot-factory-edge-service`，其中 `{BASE_URL}` 为实际 Edge 地址（含协议、IP/域名和端口），不重复包含服务前缀。下文接口路径均相对于该前缀。

JSON 请求使用 UTF-8 编码；GET 参数通过 URL Query 传递，`multipart` 文件上传由客户端自动生成 boundary。

3.1.2 工站会话与接口校验

PS 工站调用 `/login` 后，从返回的 `respData.token` 获取会话令牌，并在后续请求的 `x-snbps-session` 请求头中传入。工站会话令牌与烧录 `traceId` / `uuid` 是不同标识；`x-snbps-request-id` 仅用于请求追踪，不能代替业务 `traceId`。

会话有效期由部署配置决定，调用受保护接口时会刷新有效期。

以下接口由 `UkeyPermissionFilter` 强制校验：

```
/assignBatch
/queryPPCBatchInfo
/v1/queryBatchByOrderNumber
/getPPData
/getProgrammingData
/uploadRequestData
/programmingResult
/executeApdu
/v1/eventLog/upload
/v1/matter/onboarding-payloads/recover
/v1/device/information/query
/v1/testFile/upload
/v1/deviceData/upload
```

校验信息缺失或失效时，返回 `HTTP 200`，业务码为 `8050`。

3.1.3 平台鉴权与工厂上下文

平台鉴权由部署配置独立控制。当前仓库的 `application.yml` 未将以下接口路径排除在平台鉴权范围之外：

- `/v1/device/detail`
- `/v1/updateDevicesProfile`
- `/v1/deviceData/upload`
- `/executeApdu`

- `/v1/eventLog/upload`
- `/v1/queryBatchByOrderNumber`
- 三个 `/generate*` 接口路径

当目标环境启用平台鉴权时，调用这些接口还需提供 `x-snbps-auth-token`。正式联调前应核对目标环境的鉴权配置，不能仅凭有效的工站会话令牌判断接口是否满足全部鉴权要求。

`x-snbps-factory-id` 用于传递公共工厂信息，当前工厂上下文主要通过平台登录信息或有效工站会话解析，不能仅通过该请求头切换工厂。

3.2 统一响应与错误处理

Factory Edge Service API 通常采用统一的 JSON 响应格式，各字段说明如下：

字段	类型	是否必填 / 条件	说明
success	boolean	响应	`true` 表示业务成功
respCode	string	响应	`8000` 表示成功；其他业务错误码见附录
respMsg	string	响应	状态说明，受语言设置影响
respData	object / array / null	响应	各接口的业务数据；无返回数据或业务失败时通常为 `null`
respTime	integer (int64)	响应	秒级 Unix 时间戳，非字符串

3.2.1 响应示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": null,
  "respTime": 1792108800
}
```

3.2.2 错误处理

业务异常（`ClientApiException`）和工站会话失效均通过 HTTP 200 返回，调用方不能仅根据 HTTP 状态码判断业务是否成功，还需检查 `success` 和 `respCode`。

参数绑定错误、Bean Validation 校验失败、平台鉴权失败及网关错误不保证返回 HTTP 200，应根据实际 HTTP 状态码和响应内容进行处理。

3.2.3 特殊响应

`/download` 接口成功时直接返回文件流，不使用统一 JSON 响应格式。

4. 接口索引（29 项）

本节列出 Factory Edge Service API 当前提供的 29 个接口，包括请求方法、相对路径、接口功能及调用身份要求。接口鉴权规则详见“接口调用约定与鉴权”章节。

编号	请求 Method	相对路径	接口名称	调用身份
5.1	POST	/v1/device/detail	查询设备详情	平台鉴权按部署配置
5.2	POST	/v1/updateDevicesProfile	更新设备档案信息	平台鉴权按部署配置
5.3	POST	/v1/deviceData/upload	上传设备生产数据	工站会话；平台鉴权见公共约定
5.4	POST	/v1/device/information/query	查询设备综合信息	工站会话；平台鉴权见公共约定
5.5	POST	/v1/matter/onboarding-payloads/recover	找回 Matter 二维码	工站会话；平台鉴权见公共约定
5.6	POST	/v1/testFile/upload	上传设备测试文件	工站会话；平台鉴权见公共约定
5.7	GET	/heartBeat	连接心跳	登录前 / 初始化
5.8	GET	/queryLoginOptions	查询登录选项	登录前 / 初始化
5.9	POST	/queryLoginType	查询工站登录方式	登录前 / 初始化
5.10	GET	/preLogin	U-Safe 预登录	登录前 / 初始化
5.11	POST	/login	工站登录	登录前 / 初始化
5.12	POST	/logout	工站退出登录	按需提供工站上下文
5.13	GET	/v1/queryBatchByOrderNumber	按订单查询批次	工站会话；平台鉴权见公共约定
5.14	POST	/assignBatch	分配工站批次	工站会话；平台鉴权见公共约定
5.15	GET	/queryPPCBatchInfo	查询工站批次信息	工站会话；平台鉴权见公共约定
5.16	GET	/getProgrammingData	获取动态烧录数据	工站会话；平台鉴权见公共约定
5.17	GET	/getMetadata	获取批次元数据	按需提供工站上下文
5.18	GET	/getPPData	获取完整烧录数据（兼容接口）	工站会话；平台鉴权见公共约定
5.19	POST	/uploadRequestData	提交中间烧录请求数据	工站会话；平台鉴权见公共约定
5.20	POST	/programmingResult	上报烧录结果	工站会话；平台鉴权见公共约定
5.21	POST	/executeApdu	执行 SE APDU 步进烧录	工站会话；平台鉴权见公共约定
5.22	POST	/getScriptFile	获取烧录脚本地址	按需提供工站上下文
5.23	GET	/download	下载文件	按需提供工站上下文
5.24	POST	/v1/eventLog/upload	上传设备事件日志	工站会话；平台鉴权见公共约定
5.25	POST	/generateVerifier	生成 Matter Verifier	平台鉴权按部署配置

编号	请求 Method	相对路径	接口名称	调用身份
5.26	POST	/generateOnboardingData	生成 Matter 配网载荷	平台鉴权按部署配置
5.27	POST	/generateUniqueId	生成 Matter Unique ID	平台鉴权按部署配置
5.28	POST	/queryDeviceData	查询设备信息（已废弃）	按需提供工站上下文
5.29	POST	/updateDeviceInfo	更新设备信息（已废弃）	按需提供工站上下文

4.1 接口范围说明

本接口索引覆盖 `ProgrammingStationController` 和 `ExternalSystemController` 当前注册的 29 条对外接口路由，不包含后台管理 Controller、调试用 `TestController` 及已整类注释的 `UsbKeyController`。

`/v1/testData/upload` 虽仍存在于过滤器或鉴权排除配置中，但当前控制器未注册该接口。设备生产数据统一通过 `/v1/deviceData/upload` 上传。

5. 接口列表

5.1 查询设备详情

根据 AND 逻辑组合的定位条件和过滤条件查询设备档案，并根据 `queryKeys` 返回指定分组或字段。本地未查询到匹配记录时，依次查询历史数据和云端数据。若查询结果包含不同产品或不同 `deviceId` 的多条记录，返回错误码 `8102`；若存在同一产品、同一设备的多条记录，则按生产时间选择最新记录返回。

5.1.1 请求 Method

`POST`

5.1.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/device/detail

5.1.3 请求 Body

该接口的 `Body` 包含以下参数：

> **提示**
> `deviceLocator` 中的 `chipUid`、`bleMac`、`serialNumber`、`matterQrcode`、`matterManualCode` 至少需要提供一项非空值，其余参数作为辅助查询条件，所有查询条件均采用 AND 逻辑组合。调用时应提供足够精确的参数，以确保定位目标设备。`queryKeys` 仅用于指定返回的数据分组或字段，不参与设备查询条件的匹配。

名称	类型	是否必填	说明
deviceLocator	object	是	用来检索设备的基础信息，选择 `deviceLocator` 作为查询条件时， `chipUid`， `bleMac`， `serialNumber`， `matterQrcode`， `matterManualCode` 至少必填一项

名称	类型	是否必填	说明
├──chipUid	string	否	烧录设备芯片中的 `mcuId`
├──bleMac	string	否	芯片蓝牙 mac 地址
├──serialNumber	string	否	烧录设备流水号
├──matterQrcode	string	否	matter 配网二维码
├──matterManualCode	string	否	matter 手工配网码
filters	object	否	提高定位设备的辅助过滤条件，用于提高设备查询精确度
├──productionTimeStart	string	否	设备烧录生产开始时间，日期格式：2025-12-16T12:02:58+08:00
├──productionTimeEnd	string	否	设备烧录生产完成时间，日期格式：2025-12-17T12:02:58+08:00
├──batchId	string	否	设备生产时的批次号
matter	object	否	matter 相关设备参数，携带该参数用于提高设备查询精确度
├──pid	string	否	matter 设备所属 PID，增加此参数条件有助于提高查询设备的精确度
├──vid	string	否	matter 设备所属 VID，增加此参数条件有助于提高查询设备的精确度
customParams	object	否	自定义条件过滤参数，详情请参考 customParams 参数说明
queryKeys	string[]	建议提供	以字符串数组形式传入，支持指定分组名（如 `matterData`）或字段名（如 `chipData.chipUid`）。本地设备档案匹配成功后，若未提供 `queryKeys` 或传入空数组，则返回空对象 `{}`，不会默认返回所有字段。支持的 key 请参考可查询键列表 (queryKeys V2)

customParams 参数说明

本接口支持通过该参数传入自定义查询条件，以实现更灵活的数据过滤。当需要根据业务中自定义上传的字段查询数据时，可将对应字段名与期望值构造成键值对，放入此对象中。

customParams 参数结构

- 类型：JSON 对象（键值对集合）。
- 键（key）：要匹配的扩展字段名（例如 `"customData"`）。
- 值（value）：该字段期望匹配的值（例如 `"customDataValue"`）。

示例

```
{"customData1": "customDataValue1", "customData2": "customDataValue2"}
```

表示查询设备上报数据 customData1 为 “customDataValue1” 且 customData2 为 “customDataValue2” 的记录。

名称	类型	是否必须	说明
key	string	是	用于查询设备档案信息中的基础 key 值。可使用的 key 由系统生成，或在烧录过程中自定义上传的 key 集合
value	string	是	每个 key 对应查询参数的值。

5.1.4 返回参数

> **提示**
> 下表展示的是查询所有 `queryKeys` 时返回的数据。在实际使用中，按需查询可适当提升接口性能。

名称	类型	说明
respCode	string	`8000` 表示成功，其他均表示异常，具体异常信息可见于 `respMsg`
respMsg	string	响应信息，例如 `NoError` 或 `DataNotFound`
respData	object	查询结果数据
├──matterData	object	Matter 数据相关
├──iterations	string	迭代次数
├──commissioningFlows	string	配网流程
├──discoverableMode	string	设备可发现模式
├──vid	string	厂商 ID
├──pid	string	产品 ID
├──discriminator	string	区分码
├──enableRotatingDeviceId	string	启用旋转设备标识符
├──manualCode	string	手工配对码
├──passcode	string	配对码
├──qrcode	string	matter 配网二维码
├──salt	string	盐值
├──saltLength	string	盐值长度
├──uniqueid	string	设备范围内唯一标识符
├──verifier	string	密码验证器
├──serialNumber	string	matter 烧录设备流水号

名称	类型	说明
├──dacCertMd5	string	DAC 证书 MD5
├──cdFileMd5	string	CD 文件 MD5
├──cdFileName	string	CD 文件名称
├──paiCertMd5	string	PAI 证书 MD5
├──factoryData	object	Factory Data 配置相关
├──vendorName	string	厂商名称
├──hardwareVersion	string	硬件版本
├──hardwareVersionString	string	硬件版本字符串
├──partNumber	string	部件号
├──productUrl	string	产品网址
├──productLabel	string	产品标签
├──productName	string	产品名称
├──startAddress	string	起始地址
├──programmingData	object	设备烧录结果相关
├──programmingStatus	string	设备烧录状态： • 成功: `success` • 失败: `failure` • 进行中: `pending`
├──programmingLog	string	烧录日志
├──programmingTime	string	烧录时间
├──chipData	object	芯片数据相关
├──chipUid	string	烧录设备芯片中的 mcuId
├──bleMac	string	芯片蓝牙 mac 地址
├──firmwareData	object[]	固件相关
├──fwName	string	固件名称
├──fwMd5	string	固件 MD5
├──customData	object	客户方在生产时上报的自定义数据

5.1.5 请求示例

```
{
  "deviceLocator": {
    "chipUid": "f9e203b4cf3ce6323637364250 55f321",
    "bleMac": "f9:e2:03:b4:cf:3c:e6:32",
    "serialNumber": "8360224622727172372"
  },
  "filters": {
    "productionTimeStart": "2025-12-16T12:02:58+08:00",
    "batchId": "7603594830282752"
  },
  "matter": {
    "pid": "8001",
    "vid": "FFF2"
  },
  "customParams": {
    "customData1": "nPK3DqGQQwdE3fYxaofURw==",
    "customData2": "1"
  },
  "queryKeys": [
    "matterData.vid",
    "matterData.pid",
    "matterData.manualCode",
    "matterData.passcode",
    "matterData.qrcode",
    "matterData.salt",
    "matterData.uniqueid",
    "matterData.verifier",
    "matterData.serialNumber",
    "matterData.dacCertMd5",
    "matterData.paiCertMd5",
    "factoryData.vendorName",
    "factoryData.hardwareVersion",
    "factoryData.hardwareVersionString",
    "factoryData.partNumber",
    "factoryData.productUrl",
    "factoryData.productLabel",
    "factoryData.productName",
    "programmingData.programmingStatus",
    "chipData.chipUid",
    "chipData.bleMac",
    "firmwareData.fwName",
    "firmwareData.fwMd5"
  ]
}
```

5.1.6 响应示例

注意：本响应示例展示全部结果信息。在实际使用中，请根据 `queryKeys` 查询参数获取指定的结果数据。

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "matterData": {
      "vid": "FFF2",
      "pid": "8001",
```

5.1.7 当前实现说明

- 支持的数据分组：`matterData`、`factoryData`、`programmingData`、`chipData`、`firmwareData`、`customData`、`seData`。
- 字段返回规则：`seData` 仅支持整组返回，指定 `seData.xxx` 时也不会按子字段过滤。`queryKeys` 中包含未知分组时，不会返回额外数据；查询字段为空或对应数据未存储时，可能返回空值。
- 烧录数据：`programmingData.programmingStatus` 支持 `success`、`failure` 和 `pending` 三种状态。当前本地数据构造逻辑未向 `programmingLog` 赋予实际日志内容，不应依赖该字段获取烧录日志。
- 接口鉴权：本接口不属于 `UkeyPermissionFilter` 强制校验工站会话的接口，平台鉴权要求见“接口调用约定与鉴权”章节。

5.2 更新设备档案信息

工厂在修改设备烧录数据后，可以通过此接口实现设备信息的同步，将修改信息同步至 OBIS 统一管理。

5.2.1 请求 Method

`POST`

5.2.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/updateDevicesProfile

5.2.3 请求 Body

该接口的 `Body` 包含以下参数：

名称	类型	是否必填	说明
deviceLocator	object	是	用来检索设备的基础信息，选择 `deviceLocator` 作为查询条件时，`chipUid`，`bleMac`，`serialNumber`，`matterQrcode`，`matterManualCode` 至少必填一项。
├──chipUid	string	否	烧录设备芯片中的 `mcuId`
├──bleMac	string	否	芯片蓝牙 mac 地址
├──serialNumber	string	否	烧录设备流水号
├──matterQrcode	string	否	matter 配网二维码
├──matterManualCode	string	否	matter 手工配网码
filters	object	否	提高定位设备的辅助过滤条件，用于提高设备查询精确度
├──productionTimeStart	string	否	设备烧录生产开始时间，日期格式：2025-12-16T12:02:58+08:00
├──productionTimeEnd	string	否	设备烧录生产完成时间，日期格式：2025-12-17T12:02:58+08:00
├──batchId	string	否	设备生产时的批次号
matter	object	否	matter 相关设备参数，携带该参数用于提高设备查询精确度

名称	类型	是否必填	说明
├──pid	string	否	matter 设备所属 PID，增加此参数条件有助于提高查询设备的精确度
├──vid	string	否	matter 设备所属 VID，增加此参数条件有助于提高查询设备的精确度
customParams	object	否	自定义条件过滤参数，详情请参考 customParams 参数说明
updateParams	object[]	否	更新项以数组形式传入，每项包含 `type`、`key` 和 `value` 字段，类型均为 `string`。传入空数组时不执行字段更新。支持的 `key` 及操作范围请参考 updateParams 参数说明

updateParams 参数说明

本接口支持通过该参数传入系统支持的 key，以实现对指定设备字段记录的修改。当需要更新设备字段记录时，可将字段名，目标值及操作类型（type）组合成数组后一并传入。

updateParams 参数结构

- 类型：JSON 数组。
- 当 type 为 `add` 时，新增一条 key 对应的字段记录；如果该 key 已存在，则更新其对应的字段值。
- 部分设备数据支持删除，当 type 为 `delete` 时，删除 key 对应的字段记录。
- 该参数支持的 key 详情请参考可更新键列表 (updateDevicesProfile)。

示例

```
[
    {
        "type": "add",
        "key": "qrcode",
        "value": "MT:XXX-S0XXXX"
    },
    {
        "type": "delete",
        "key": "factoryData.productURL"
    },
    {
        "type": "add",
        "key": "factoryData.qqbb",
        "value": "abcd"
    },
    {
        "type": "add",
        "key": "firmwareData.bootloader-xxx-1.1.s37",
        "value": "00000001"
    }
]
```

以上示例表示：

1. 新增 `qrcode` 字段，并将其值设置为 MT:XXX-S0XXXX；如果对应设备中已存在该字段，则覆盖原有值。
2. 删除对应设备 `factoryData` 中的 `productURL` 字段。
3. 在对应设备的 `factoryData` 中新增 `qqbb` 字段，并将其值设置为 `abcd`。

4. 对应设备的固件列表中新增固件 bootloader-xxx-1.1.s37，并将其 `startAddress` 设置为 `00000001`。如果设备中已存在同名固件，则将其起始地址更新为该值。

5.2.4 返回参数

名称	类型	说明
respCode	string	`8000` 表示成功，其他均表示异常，具体异常信息可见于 `respMsg`
success	bool	`true` 表示成功
respMsg	string	响应信息，例如 `NoError` 或 `DataNotFound`

5.2.5 请求示例

```
{
  "deviceLocator": {
    "chipUid": "f9e203b4cf3ce6323637364250 55f321"
  },
  "filters": {
    "productionTimeStart": "2025-12-16T12:02:58+08:00",
    "batchId": "7603594830282752"
  },
  "matter": {
    "pid": "8001",
    "vid": "FFF2"
  },
  "customParams": {
    "customData1": "nPK3DqGQQwdE3fYxaofURw=="
  },
  "updateParams": [
    {
      "type": "add",
      "key": "qrcode",
      "value": "MT:XXX-S0XXXX"
    },
    {
      "type": "delete",
      "key": "factoryData.productURL",
      "value": ""
    },
    {
      "type": "add",
      "key": "factoryData.qqbb",
      "value": "abcd"
    },
    {
      "type": "add",
      "key": "firmwareData.bootloader-xxx.s37",
      "value": "00000000"
    }
  ]
}
```

5.2.6 响应示例

```
{
```

5.2.7 操作规则与边界

更新参数 `updateParams` 中各字段的要求如下：

字段	类型	是否必填 / 条件	说明
updateParams[].type	string	按操作传入	`add` 表示新增或覆盖；`delete` 仅适用于支持删除操作的集合字段
updateParams[].key	string	是	支持附录三中列出的系统键或集合前缀
updateParams[].value	string	add 时传入	仅接收字符串，日期和整数文本需符合目标字段的格式要求

设备定位与更新范围

`deviceLocator`、`filters`、`matter` 和 `customParams` 的设备定位规则与“查询设备详情”接口一致。更新操作仅针对本地设备档案，不查询历史数据或云端数据。

字段取值规则

- 环境值仅支持 `test` 和 `prod`。
- `programmingStatus` 支持 `Complete`、`Failed` 和 `Unreported`，其他值当前按未完成状态处理。

字段更新与删除规则

- **支持删除的字段：** `factoryData.*`、`firmwareData.*`、`otherFile.*`、`attachment.*` 和 `programmingScript`。
- **不支持删除的字段：** 标量键及 `extend.*` 当前直接按 `value` 赋值，不支持通过 `delete` 操作删除。
- **未知字段：** 当前会跳过未知 `key`，不保证返回参数错误。

其他处理规则

- `updateParams` 为空数组时，不执行字段更新，但仍会尝试读取目标设备档案及元数据。
- 更新完成后，记录将标记为等待同步状态。收到 HTTP 响应不代表数据已完成云端同步。

5.3 上传设备生产数据

工厂可通过此接口上传设备生产过程中的烧录、测试及组装数据，并根据 `traceId` 维护对应的生产记录。烧录结束时复用 `programmingResult` 接口的处理逻辑，测试及组装数据通过后续异步流程同步。

5.3.1 请求 Method

`POST`

5.3.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/deviceData/upload

5.3.3 请求头

本接口的 `Content-Type` 为 `application/json`。除公共请求头外，还需提供以下请求头：

字段	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

平台鉴权要求见“接口调用约定与鉴权”章节。

5.3.4 请求 Body

该接口的 `Body` 包含以下参数：

名称	类型	是否必填 / 条件	说明
traceId	string	按场景传入	生产记录跟踪 ID，兼容别名 `uuid`。烧录场景使用 `/getProgrammingData` 返回的 `uuid`；未提供时可创建新记录。传入的值须对应已有生产记录或文件任务，否则返回 `8006`
scenarioType	string	测试 / 自定义场景必填	生产场景名称，如 `RF-Test`。组装批次固定为 `assembly`，已有烧录会话固定为 `programming`；其他场景缺失时返回 `8069`
deviceId	string	测试 / 组装对接必填	设备 MCU ID，未提供时使用 `deviceRecord.deviceId`
batchId	string	组装对接必填	生产批次 ID，仅纯数字字符串按 ID 处理。组装场景根据批次类型识别，不能仅通过 `scenarioType=assembly` 判断
deviceProfileId	string	否	已有设备档案 ID，应与 `deviceId` 对应；未提供时根据设备 ID 匹配已有档案
stationId	string	否	工站编码，未提供时使用当前会话对应的工站，不限于数字
factoryId	string	否	工厂 ID，新建记录时仅纯数字字符串生效，否则使用当前会话对应的工厂
deviceSn	string	否	设备序列号
result	boolean	烧录场景建议提供	烧录结果，`true` 表示成功，`false` 表示失败；烧录结束时未提供则按失败处理
failureReason	string	否	失败原因，写入生产记录
timestamp	string	否	当前接口未使用该字段设置生产记录时间，记录创建时间由服务端生成
status	string	否	自定义状态。烧录和组装场景可能由系统执行结果生成并覆盖；其他非空值写入 `customStatus`
deviceRecord	object	按场景对接约定传入	设备生产数据，兼容别名 `deviceProfile`，具体字段见 deviceRecord 参数说明

5.3.5 deviceRecord 参数说明

`deviceRecord` 用于上传设备在当前生产环节产生的数据。

名称	类型	是否必填 / 条件	说明
deviceId	string	条件必填	设备 MCU ID，当请求 Body 顶层的 `deviceId` 为空时使用该字段

名称	类型	是否必填 / 条件	说明
firmwareVersion	string	烧录场景按需提供	固件版本
seid	string	SE 烧录时按需提供	SEID，字段名使用小写
cplc	string	SE 烧录时按需提供	CPLC 数据
matterData	map	Matter 烧录时按需提供	实际烧录的 Matter 数据
factoryData	map	否	实际烧录的 Factory Data，字段名不能为空，值必须为字符串
customData	map	否	自定义数据，最多 50 个字段；字段名不能为空且不超过 64 个字符，值必须为字符串，否则返回 `8002`。本次上传将整体替换原有 `customData`
testData	map	测试 / 组装对接必填	测试数据，兼容旧字段名 `diagnosticData`。值可以是文本或文件引用列表；同名字段均为列表时追加，否则覆盖

5.3.6 场景处理规则

不同生产场景下的参数要求及处理规则如下：

场景	traceId 与批次	数据与状态
烧录	使用已创建的烧录会话 `traceId`，通常来自 `/getProgrammingData` 接口	根据实际烧录内容提供 `deviceRecord`，并明确传入 `result`。烧录结束时复用 `/programmingResult` 的处理逻辑，仅设置 `scenarioType=programming` 不会创建完整的烧录流程
测试 / 自定义	首次上传可不提供 `traceId`，后续上传使用响应返回的 `traceId`；`scenarioType` 必填	按对接要求提供 `deviceId` 和 `deviceRecord.testData`。若存在对应设备档案，则关联该档案；否则可仅创建生产记录，`deviceProfileId` 返回 `null`
组装	`batchId` 指定组装批次；`traceId` 可来自 `/getProgrammingData` 接口或文件任务，首次未提供时也可创建新记录	提供 `deviceId` 和 `deviceRecord.testData`，场景固定为 `assembly`。`testData.result` 参与结果状态和产量计算，同一设备在同一批次中的成功组装记录按当前实现去重

参数长度与格式限制

数据库字段长度限制如下：

字段	最大长度
traceId	64 字符

字段	最大长度
deviceId	128 字符
deviceSn	64 字符
failureReason	1024 字符
status / scenarioType	64 字符

上述长度限制并不代表接口会自动校验所有请求字段。ID 值应在 Java `Long` 类型的取值范围内。字段超长、数字格式不合法或不符合数据库约束时，可能导致数据保存失败。纯数字字符串也必须符合相应的长度和取值范围要求。

其他处理规则

- 烧录数据： 按对接约定，`deviceRecord.firmwareVersion`、`deviceRecord.seid`、`deviceRecord.cplc` 和 `deviceRecord.matterData` 至少提供一项，但当前接口未对此要求执行强制校验。
- 测试与组装数据： 使用 `deviceRecord.testData` 上传测试数据。请求 Body 顶层的 `customData` 不会被识别为 `deviceRecord.customData`。

5.3.7 请求示例

烧录场景

使用 `/getProgrammingData` 返回的 `uuid` 作为 `traceId`，并提交烧录结果及设备数据。

```
{
  "traceId": "<uuid returned by getProgrammingData>",
  "deviceId": "DEVICE-001",
  "result": true,
  "deviceRecord": {
    "firmwareVersion": "1.2.3",
    "matterData": {
      "qrcode": "<actual commissioning payload>"
    },
    "customData": {
      "mac": "AA:BB:CC:DD:EE:FF"
    }
  }
}
```

测试场景

首次测试上报可不提供 `traceId`，通过 `scenarioType` 指定测试场景，并在 `testData` 中提交测试数据。

```
{
  "scenarioType": "RF-Test",
  "deviceId": "DEVICE-001",
  "status": "success",
  "deviceRecord": {
    "testData": {
      "voltage": "3.3",
      "result": "success"
    },
    "customData": {
      "fixture": "A01"
```

组装场景

通过 `batchId` 指定组装批次，首次上报可不提供 `traceId`。

```
{
  "batchId": "<actual assembly batch ID>",
  "deviceId": "DEVICE-001",
  "deviceRecord": {
    "testData": {
      "eol": "pass",
      "result": "success"
    }
  }
}
```

5.3.8 文件与生产数据关联

需要关联测试文件时，先调用 `POST /v1/testFile/upload` 上传文件，并保存返回的 `traceId`。随后使用相同的 `traceId` 调用本接口，通过 `testData` 中的 `fileName` 引用对应文件。

如果已有生产记录的 `traceId`，也可以在文件上传时直接传入。`fileTraceId` 为内部关联字段，无需在本接口请求顶层额外提供。

请求示例：

```
{
  "traceId": "<traceId returned by testFile/upload>",
  "scenarioType": "RF-Test",
  "deviceId": "DEVICE-001",
  "deviceRecord": {
    "testData": {
      "report": [
        {
          "fileName": "report.txt"
        }
      ]
    }
  }
}
```

5.3.9 返回参数

本接口采用统一 JSON 响应格式，公共响应字段见“统一响应与错误处理”章节。`respData` 包含以下字段：

字段	类型	说明
traceId	string	服务端最终使用的跟踪 ID，后续上传时原样回传
deviceProfileId	integer (int64) / null	已关联的设备档案 ID，未关联时为 `null`
programmingRecordId	integer (int64)	设备生产记录 ID

5.3.10 响应示例

```
{
```

5.3.11 主要错误码

错误码	说明
8002	自定义字段不符合要求
8006	`traceId` 未关联已有生产记录或文件任务
8007	数据保存失败
8032	批次不存在
8037	设备建档失败
8050	工站会话失效
8069	测试或自定义场景缺少 `scenarioType`

5.3.12 HTTP 调用示例与续传

以下示例使用 `curl` 首次上传测试数据。`BASE_URL` 仅包含协议、主机和端口，例如 `http://edge-host:8005`；`SESSION` 为工站调用 `/login` 获取的会话令牌。如目标环境启用平台鉴权，还需提供 `x-snbps-auth-token` 请求头。

```
curl -X POST "$BASE_URL/api/iot-factory-edge-service/v1/deviceData/upload" \
  -H "Content-Type: application/json;charset=utf-8" \
  -H "x-snbps-session: $SESSION" \
  -d '{
    "scenarioType": "RF-Test",
    "deviceId": "DEVICE-001",
    "status": "success",
    "deviceRecord": {
      "testData": {"voltage": "3.3", "result": "success"},
      "customData": {"fixture": "A01"}
    }
  }'
```

续传规则与注意事项

- **跟踪 ID**：首次上传成功后，保存响应中的 `respData.traceId`。后续上传同一生产记录时，在请求顶层传入该值，并继续提供 `scenarioType`。不同的独立测试不应复用已有 `traceId`。
- **标识使用**：`x-snbps-request-id` 仅用于请求追踪，不能代替业务 `traceId`。传入不存在且未关联文件任务的 `traceId` 时，返回 `8006`。
- **测试数据**：同一 `traceId` 下，`testData` 的同名列表字段会追加数据，重复请求可能产生重复的附件引用。
- **自定义数据**：`customData` 每次上传均整体替换原有数据，续传时应提交需要保留的全部自定义字段。
- **重复请求**：网络超时后，不能直接假定上一次请求未执行，也不能将重复上传视为完全幂等操作。

5.4 查询设备综合信息

本接口用于查询本地设备档案中的通用信息、Matter 数据及指定的自定义数据。与 `/v1/device/detail` 不同，本接口不支持通过 `queryKeys` 指定返回字段。

5.4.1 请求 Method

`POST`

5.4.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/device/information/query

5.4.3 请求头

除公共请求头外，本接口还需提供以下请求头：

字段	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.4.4 请求 Body

该接口的 `Body` 包含以下参数，至少提供一个非空设备标识。

字段	类型	是否必填 / 条件	说明
deviceId	string	至少提供一项	按设备档案中的 `deviceId` 精确匹配，最长 128 字符
chipId	string	至少提供一项	按设备档案中的 `chipUid` 精确匹配，最长 128 字符
uniqueId	string	至少提供一项	按 Matter 数据中的 `uniqueId`、`uniqueid` 及对应的 `snake_case` 字段匹配，最长 128 字符

5.4.5 接口说明

- **查询条件**：多个非空设备标识按 AND 关系组合，所有条件均满足时才匹配。
- **查询范围**：仅查询本地设备档案，不查询历史数据库或云端。未找到匹配记录时返回 `8101`。
- **查询结果**：存在多条匹配记录时，返回查询结果中的第一条，不保证设备标识的唯一性，也不保证仅返回烧录成功的设备。
- **烧录状态**：当前 `provisionStatus` 固定返回 `Success`，不能单独作为判断设备实际烧录状态的依据。

5.4.6 请求示例

```
{
  "deviceId": "DEVICE-001"
}
```

5.4.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下数据对象：

字段	类型	说明
generalData	object	设备身份、烧录相关信息及产品版本信息
matterData	object / null	Matter 证书及配网数据，非 Matter 设备可能返回 `null`
customData	object	设备序列号、业务标识、MAC 地址及测试、标定数据

未记录的字段可能为空。

5.4.8 返回示例


```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "generalData": {
      "deviceId": "DEVICE-001",
      "provisionStatus": "Success",
      "provisionTime": "2026-10-16T02:00:00Z",
      "sourceProgramingBatchId": "10",
      "provisionFactoryId": "20",
      "provisionStationId": "30",
      "productId": "40",
      "productName": "sample product",
      "productVersionId": "50",
      "productVersionName": "1.0"
    },
    "matterData": null,
    "customData": {
      "serialNumber": "SN-001",
      "customId": "CUSTOM-001",
      "macAddress": "AA:BB:CC:DD:EE:FF",
      "factoryTestData": null,
      "calibrationParams": null
    }
  },
  "respTime": 1792108800
}
```


5.4.9 响应字段说明

generalData：通用信息

> **提示**
>
> `sourceProgramingBatchId` 为实际响应字段名，保留原有的 `Programing` 拼写。

字段	类型	说明
deviceId	string	设备 ID
provisionStatus	string	烧录状态，当前固定返回 `Success`
provisionTime	string	烧录时间，采用 UTC ISO 8601 格式
sourceProgramingBatchId	string	来源烧录批次 ID
provisionFactoryId	string	烧录工厂 ID
provisionStationId	string	烧录工站 ID，对应设备档案中的工站主键
productId	string	产品 ID

字段	类型	说明
productName	string	产品名称
productVersionId	string	产品版本 ID
productVersionName	string	产品版本名称

matterData：Matter 数据

字段	类型	说明
dacCert	string	DAC 证书，沿用存储的 Base64 DER 或 PEM 格式
paiCert	string	PAI 证书，沿用存储的 Base64 DER 或 PEM 格式
certDeclaration	string	Certification Declaration（CD），Base64 编码
discriminator	integer	Matter Discriminator，十进制数值
spake2pIterationCount	integer	SPAKE2+ 迭代次数
passcode	string	Matter 配网密码
manualPairingCode	string	Matter 手动配对码
setupPayload	string	Matter 配网载荷
spake2pVerifier	string	SPAKE2+ Verifier，沿用存储值
spake2pSalt	string	SPAKE2+ Salt，沿用存储值

customData：自定义数据

> **提示**
>
> `factoryTestData` 和 `calibrationParams` 返回前会递归清理已定义的凭据字段，不返回设备私钥。

字段	类型	说明
serialNumber	string	设备序列号
customId	string	业务标识
macAddress	string	MAC 地址
factoryTestData	JSON	工厂测试数据
calibrationParams	JSON	设备标定参数

5.4.10 主要错误码

错误码	说明
8002	查询参数全部为空
8007	存储数据解析失败
8050	鉴权失败
8101	未找到匹配设备

其他错误码见附录。

5.5 找回 Matter 二维码

本接口用于在组装阶段，根据组装批次关联的投入烧录批次，以及 Matter VID、PID、Discriminator 和 BLE 广播中的 Rotating Device Identifier（RDI），查找已成功烧录的本地设备档案，恢复对应的 Matter 配网数据。

5.5.1 请求 Method

`POST`

5.5.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/matter/onboarding-payloads/recover

5.5.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.5.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填	说明
batchId	string	是	组装批次 ID，须已关联投入烧录批次，且处于 `RUNNING` 状态、未过期
matterProductId	string	是	Matter PID，十六进制字符串，不带 `0x` 前缀；字段名不能使用 `productId` 或 `pid`
matterVendorId	string	是	Matter VID，十六进制字符串，不带 `0x` 前缀；字段名不能使用 `vendorId` 或 `vid`
discriminator	string	是	十六进制字符串，解析后的数值范围为 `0x000-0xFFF`，例如 `0F00`，不能使用十进制字符串
rotatingDeviceIdentifier	string	是	BLE 广播中的 RDI，18 字节（36 位十六进制字符），不带 `0x` 前缀，须使用实际广播数据

5.5.5 接口说明

- **查询范围**：仅查询组装批次关联的投入烧录批次中已成功烧录的本地设备档案，不通过云端反查 RDI。
- **设备匹配**：根据 VID、PID、Discriminator 和 RDI 匹配设备。RDI 不匹配或未找到对应设备时，均返回 `8101`。
- **数据格式**：VID 和 PID 按数据库中的文本值匹配，调用时应保持与批次数据一致的格式。

5.5.6 请求示例

```
{
  "batchId": "<assembly batch ID>",
  "matterProductId": "8001",
  "matterVendorId": "FFF2",
  "discriminator": "0F00",
  "rotatingDeviceIdentifier": "<actual 36-character hexadecimal BLE advertising RDI>"
}
```

5.5.7 返回参数

接口采用统一 JSON 响应格式。当前实现的 `respData` 返回以下四个字段，并非仅返回 `setupPayload`：

字段	类型	说明
setupPayload	string	Matter 二维码配网载荷
deviceId	string	设备 ID
deviceProfileId	integer	设备档案 ID
manualCode	string	Matter 手动配网码，字段名为 `manualCode`，不能使用 `manualPairingCode`

5.5.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "setupPayload": "<actual QR code payload>",
    "deviceId": "DEVICE-001",
    "deviceProfileId": 200,
    "manualCode": "<actual manual pairing code>"
  },
  "respTime": 1792108800
}
```

5.5.9 主要错误码

错误码	说明
8002	Discriminator 参数无效
8009	批次已过期

错误码	说明
8010	批次未处于运行状态
8032	组装批次或关联关系不存在
8050	鉴权失败
8101	未找到匹配设备

其他错误码见附录。

5.6 上传设备测试文件

本接口用于上传设备测试文件。接口接收文件后，先将文件保存至本地并创建异步上传任务。接口收到 HTTP 响应仅表示文件已被接收并完成本地处理，不代表文件已成功上传至云存储。

5.6.1 请求 Method

`POST`

5.6.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/testFile/upload

5.6.3 请求头

本接口的 `Content-Type` 为 `multipart/form-data`。除公共请求头外，还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.6.4 请求 Body

该接口使用 `multipart/form-data` 上传文件，包含以下参数：

字段	类型	是否必填	说明
traceId	string（form field）	否	已有脚本跟踪 ID 可传入；未提供时自动生成。接口返回的 `traceId` 应原样传递给生产数据上传接口
files	file[]（multipart parts）	是	上传的文件列表，至少包含一个非空文件。同一请求上传多个文件时，重复使用 `files` 字段。文件名不得包含路径，不得为 `.` 或 `..`，同一请求中的文件名不得重复

5.6.5 接口说明

- **文件处理**： 接口接收文件后，先保存至本地，再创建异步上传任务。
- **任务跟踪**： 可传入已有的 `traceId`，未提供时由接口自动生成。返回的 `traceId` 应原样用于后续生产数据上传接口。
- **上传状态**： HTTP 请求成功不代表文件已上传至云存储。后台异步上传过程中发生的错误，不一定会体现在首次 HTTP 响应中。
- **文件大小限制**： 文件大小受部署环境的 `multipart` 配置限制。`application.yml` 默认单文件最大 2 GB、单次请求最大 4 GB； `dev` 配置分

别为 200 MB 和 250 MB。实际限制可能被环境变量或外部配置覆盖。

5.6.6 请求示例

```
curl -X POST "$BASE_URL/api/iot-factory-edge-service/v1/testFile/upload" \
  -H "x-snbps-session: $SESSION" \
  -F "files=@./report.txt" \
  -F "files=@./photo.png"
```

5.6.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
traceId	string	文件任务跟踪 ID
fileCount	integer	本次接收的文件数量
status	integer	文件异步上传任务状态：`1` 待同步；`2` 上传中；`3` 成功；`4` 等待重试；`5` 失败；`6` 本地已清理

5.6.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "traceId": "example-file-trace-id",
    "fileCount": 2,
    "status": 1
  },
  "respTime": 1792108800
}
```

5.6.9 主要错误码

错误码	说明
8057	上传参数错误
8058	文件清单或文件名不符合要求
8059	文件本地存储失败
8060	文件上传任务保存失败
8061	后台云存储上传失败
8051	存储空间不足或并发处理压力

后台异步上传过程中产生的错误不一定出现在首次 HTTP 响应中。其他错误码见附录。

5.7 连接心跳

本接口用于烧录客户端检测 Edge 服务的连通性。

5.7.1 请求 Method

`GET`

5.7.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/heartBeat

5.7.3 请求参数

本接口为初始化接口，不要求提前获取工站会话令牌，无需提供请求参数。

5.7.4 接口说明

客户端可根据需要定期调用本接口。心跳请求成功仅表示 Edge HTTP 服务可访问，不代表其他业务服务均正常运行。

5.7.5 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/heartBeat
```

5.7.6 返回参数

接口采用统一 JSON 响应格式，成功时 `respData` 为 `null`。

5.7.7 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": null,
  "respTime": 1792108800
}
```

5.8 查询登录选项

本接口用于兼容旧版客户端，查询工站绑定批次是否要求使用 U-Safe 和生产密码。

5.8.1 请求 Method

`GET`

5.8.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/queryLoginOptions

5.8.3 请求参数

本接口为初始化接口，不要求提前获取工站会话令牌。该接口通过 Query 参数传递以下字段：

字段	类型	是否必填 / 条件	说明
usafeSN	string	与 stationId 至少提供一项	旧版 U-Safe 编号
stationId	string	与 usafeSN 至少提供一项	工站编码，提供时优先使用

5.8.4 接口说明

- **查询条件**： 工站及其绑定批次必须存在。
- **兼容性**： 本接口用于兼容旧版客户端。新登录流程可先调用 `/queryLoginType` 查询工站登录方式。

5.8.5 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/queryLoginOptions?stationId=STATION-001
```

5.8.6 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
requireUsafe	boolean	是否要求 U-Safe，当前固定返回 `true`
requirePassword	boolean	是否要求生产密码，由绑定批次的生产密码开关决定

5.8.7 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "requireUsafe": true,
    "requirePassword": false
  },
  "respTime": 1792108800
}
```

5.8.8 主要错误码

错误码	说明
8007	未找到工站或绑定批次

其他错误码见附录。

5.9 查询工站登录方式

本接口用于查询指定工站的登录方式，并兼容客户端信息上报。

5.9.1 请求 Method

`POST`

5.9.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/queryLoginType

5.9.3 请求 Body

本接口为初始化接口，不要求提前获取工站会话令牌。该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
factoryId	integer（int64）	云工厂必填	云模式下校验指定工厂存在且为云工厂
stationId	string	是	工站编码
clientPcMessage	string	否	客户端信息的 JSON 对象字符串，自 3.3.2 起标记为废弃，新客户端应通过 `/login` 上报

5.9.4 接口说明

云工厂登录仅支持密码认证或 Identity U-Safe 认证。

5.9.5 请求示例

```
{
  "factoryId": 20,
  "stationId": "STATION-001"
}
```

5.9.6 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
loginType	integer	登录方式：`1` 无密码；`2` 密码；`3` Identity U-Safe

5.9.7 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "loginType": 2
  },
  "respTime": 1792108800
}
```

5.9.8 主要错误码

错误码	说明
8016	工站不存在
8036	云工厂 ID 错误

其他错误码见附录。

5.10 U-Safe 预登录

本接口用于执行 U-Safe 双向认证的第一步，获取服务端证书、挑战数据及签名。

5.10.1 请求 Method

`GET`

5.10.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/preLogin

5.10.3 请求参数

本接口为初始化接口，不要求提前获取工站会话令牌。该接口通过 Query 参数传递以下字段：

字段	类型	是否必填 / 条件	说明
usafeSN	string	是	Identity U-Safe 编号
nonce	string	是	客户端生成的随机挑战
type	integer	否	兼容参数，当前方法未使用其值
factoryId	integer（int64）	云模式按工厂传入	用于云模式下的工厂范围匹配

5.10.4 接口说明

- **适用范围**： 本接口用于 U-Safe 双向认证，不适用于直接密码登录。
- **签名处理**： 响应中的 `sign` 为服务端签名，不能直接作为客户端登录签名。客户端应验证服务端身份，并按照现有 U-Safe 认证协议对 `challenge` 签名，再调用 `/login`。

5.10.5 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/
preLogin?usafeSN=USAFE-001&nonce=%3C%E5%AE%A2%E6%88%B7%E7%AB%AF%E9%9A%8F%E6%9C%BA%E6%95%B0%3E&factoryId=20
```

5.10.6 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
cert	string	服务端认证证书
challenge	string	服务端挑战数据，供客户端签名
sign	string	服务端签名，供客户端验证服务端身份

5.10.7 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "cert": "<authentication certificate>",
    "challenge": "<server challenge>",
    "sign": "<server signature>"
  },
  "respTime": 1792108800
}
```

5.10.8 主要错误码

错误码	说明
8020	U-Safe 不存在
8031	U-Safe 不可用
8024 / 8026	认证或签名验证失败

其他错误码见附录。

5.11 工站登录

本接口用于完成工站身份认证并获取工站会话令牌。登录成功后，烧录客户端在后续需要工站会话的接口请求中，通过 `x-snbps-session` 请求头传递返回的 `token`。

5.11.1 请求 Method

`POST`

5.11.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/login

5.11.3 请求 Body

本接口为初始化接口，不要求提前获取工站会话令牌。该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
stationId	string	新登录流程必填	非空时执行工站认证；未提供时使用旧版 U-Safe 认证流程

字段	类型	是否必填 / 条件	说明
factoryId	integer（int64）	云工厂必填	工厂 ID
password	string	loginType=2 时必填	使用现有 RSA 公钥加密后的密码，不能传入明文密码
usafeSN	string	loginType=3 或旧版流程	Identity U-Safe 编号，须与工站匹配
sign	string	U-Safe 认证时必填	客户端对 `/preLogin` 返回的挑战数据生成的签名
usafeCert	string	否	兼容字段，当前控制器分支未直接使用
clientPcMessage	string	否	客户端信息的 JSON 对象字符串

5.11.4 接口说明

- **登录方式：** 根据 `/queryLoginType` 返回的登录方式执行对应认证流程。
- **无密码登录：** `NO_PASSWORD`（`loginType=1`）无需传入密码。
- **密码登录：** `PASSWORD`（`loginType=2`）须传入 RSA 加密后的密码。
- **U-Safe 登录：** `IDENTITY_USAFE`（`loginType=3`）须先调用 `/preLogin`，再使用客户端签名完成登录。
- **会话令牌：** 登录返回的 `token` 用于工站会话认证，不应与业务操作中的 `uuid` 或 `traceId` 混用。

5.11.5 请求示例

```
{
  "factoryId": 20,
  "stationId": "STATION-001",
  "password": "<RSA-encrypted password>",
  "clientPcMessage": "{\"hostname\":\"PS-01\"}"
}
```

5.11.6 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
stationId	string	工站编码
token	string	工站会话令牌
factoryId	integer（int64）	工厂 ID
partnerCode	string	租户编码

5.11.7 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
```

5.11.8 主要错误码

错误码	说明
8012	工站重复登录
8014	密码错误
8016	工站不存在
8017	不支持的登录方式
8020 / 8021	U-Safe 不存在或不匹配
8036	云工厂 ID 错误

其他错误码见附录。

5.12 工站退出登录

本接口用于删除当前工站的会话缓存，完成工站退出登录。

5.12.1 请求 Method

`POST`

5.12.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/logout

5.12.3 请求头

参数	是否必填 / 条件	说明
x-snbps-session	有效退出登录时需要	工站会话令牌，使用 `/login` 接口返回的 `token`

5.12.4 请求参数

本接口无需提供请求参数。

5.12.5 接口说明

- **会话识别**：需要提供有效的 `x-snbps-session`，才能识别并清理对应工站的会话。
- **校验规则**：本接口未纳入 `UkeyPermissionFilter` 的强制校验路径，但不能因此省略有效退出登录所需的会话令牌。
- **退出结果**：无法识别工站上下文时，不应认为已成功清理指定工站的会话。

5.12.6 请求示例

```
curl -X POST "$BASE_URL/api/iot-factory-edge-service/logout" \
  -H "x-snbps-session: $SESSION"
```

5.12.7 返回参数

接口采用统一 JSON 响应格式，`respData` 为 `null`。

5.12.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": null,
  "respTime": 1792108800
}
```

5.13 按订单查询批次

本接口用于查询当前工厂下指定订单号对应的运行中批次。

5.13.1 请求 Method

`GET`

5.13.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/queryBatchByOrderNumber

5.13.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.13.4 请求参数

该接口通过 Query 参数传递以下字段：

字段	类型	是否必填	说明
orderNumber	string	是	工厂订单号，不能为空或仅包含空白字符

5.13.5 接口说明

- **查询范围**： 仅查询当前工厂下与指定订单号对应、状态为 `RUNNING` 的批次。
- **批次绑定**： 本接口仅执行查询，不会自动将查询到的批次绑定至当前工站。

5.13.6 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/v1/queryBatchByOrderNumber?orderNumber=ORDER-001
```

5.13.7 返回参数

接口采用统一 JSON 响应格式，`respData` 为批次信息数组，未找到匹配批次时返回空数组 `[]`。数组中的每个对象包含以下字段：

字段	类型	说明
batchId	integer（int64）	批次 ID
batchName	string	批次名称

5.13.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": [
    {
      "batchId": 10,
      "batchName": "BATCH-001"
    }
  ],
  "respTime": 1792108800
}
```

5.13.9 主要错误码

错误码	说明
8039	订单号为空
8050	鉴权失败

其他错误码见附录。

5.14 分配工站批次

本接口用于将当前登录工站绑定至指定批次，并同步工站与批次的绑定信息。

5.14.1 请求 Method

`POST`

5.14.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/assignBatch

5.14.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.14.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填	说明
batchId	integer（int64）	是	待绑定的批次 ID，必须属于当前工厂，且批次未过期、处于运行状态

5.14.5 接口说明

- **批次要求**：只能绑定当前工厂下未过期且处于运行状态的批次。
- **绑定限制**：已通过 Edge 管理端绑定、且不允许覆盖绑定关系的工站，不能通过本接口重新分配批次。
- **重复绑定**：工站已绑定指定批次时，再次绑定相同批次将返回 `8011`。

5.14.6 请求示例

```
{
  "batchId": 10
}
```

5.14.7 返回参数

接口采用统一 JSON 响应格式，`respData.batchInfo.batches` 为批次详情数组，其结构与 `/queryPPCBatchInfo` 返回的批次详情相同。

5.14.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "batchInfo": {
      "batches": [
        {
          "batchId": "10",
          "batchName": "BATCH-001",
          "programmingType": "SE",
          "bindType": 2
        }
      ]
    }
  },
  "respTime": 1792108800
}
```

5.14.9 主要错误码

错误码	说明
8009	批次已过期
8010	批次已关闭
8011	工站已绑定该批次

错误码	说明
8013	数据解析失败
8016	工站不存在
8032	批次不存在
8035	与 Edge 管理端绑定关系冲突
8050	鉴权失败

其他错误码见附录。

5.15 查询工站批次信息

本接口用于查询工站关联的批次信息、烧录脚本及服务器压力信息。

5.15.1 请求 Method

`GET`

5.15.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/queryPPCBatchInfo

5.15.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.15.4 请求参数

该接口通过 Query 参数传递以下字段：

字段	类型	是否必填	说明
usafeSN	string	否	兼容旧版 U-Safe 工站查询
batchId	integer（int64）	否	传入正数时查询指定的运行中批次；两个参数均未提供时，按当前会话工站查询

5.15.5 接口说明

- **查询与绑定**： 本接口仅查询批次信息，不会自动执行 `/assignBatch` 绑定操作。
- **烧录类型**： `programmingType` 用于区分 `MCU`、`SE`、`SE_MCU` 等烧录类型，具体以实际批次值为准。
- **枚举使用**： 烧录类型应以 `ProgrammingType` 枚举及实际接口响应为准。旧版注释中的 `MATTER`、`SE_MATTER` 不能直接作为请求枚举值。

5.15.6 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/queryPPCBatchInfo
```

5.15.7 返回参数

接口采用统一 JSON 响应格式，主要返回工站关联批次和服务器压力信息。

字段	类型	说明
usafeSN	string	U-Safe 编号
batchInfo.batches	array	工站关联的批次详情数组，无可用批次时可能返回空数组
serverLoadInfo.pressureLevel	integer	服务器压力等级
serverLoadInfo.pressurePercentage	number	服务器压力百分比

5.15.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "batchInfo": {
      "batches": [
        {
          "batchId": "10",
          "batchName": "BATCH-001",
          "total": 100,
          "produced": 10,
          "toBeProduced": 90,
          "programmingType": "SE",
          "scriptName": "burn-1.0",
          "scriptMd5": "<md5>",
          "scriptUrl": "<script download URL>"
        }
      ]
    },
    "serverLoadInfo": {
      "pressureLevel": 1,
      "pressurePercentage": 10.0
    }
  },
  "respTime": 1792108800
}
```

5.15.9 批次详情字段说明

`batchInfo.batches` 数组中的每个对象包含批次、芯片、烧录脚本及生产进度等信息，具体字段如下：

字段	类型	说明
batchId	string	批次 ID

字段	类型	说明
batchName	string	批次名称
chips	object[]	芯片信息数组，其中 `chipPart`、`chipVendor`、`objectid` 均为 string
env	string	环境信息
extData	string	扩展数据
mark	string	批次标记
notes	string	批次备注
orderNo	string	订单号
produced	integer	已生产数量
toBeProduced	integer	待生产数量
total	integer	批次总数量
productName	string	产品名称
version	string	产品版本
region	string	地区
secureDebugReopen	integer	安全调试开关
supportProtocols	string[]	支持的协议列表
scriptName	string	烧录脚本名称
scriptMd5	string	烧录脚本 MD5 校验摘要
scriptUrl	string	烧录脚本下载地址
vid	string	Matter VID
pid	string	Matter PID
bindType	integer	绑定方式：1 Edge 管理端绑定；2 客户端手动绑定
programmingType	string	烧录类型，如`MCU`、`SE`、`SE_MCU`，应根据实际响应值选择烧录流程

5.15.10 主要错误码

错误码	说明
8013	数据解析失败

错误码	说明
8015	工站未绑定批次
8016	工站不存在
8050	鉴权失败

其他错误码见附录。

5.16 获取动态烧录数据

本接口用于为当前工站绑定的批次生成设备烧录数据，并建立本次烧录的生产记录。

5.16.1 请求 Method

`GET`

5.16.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/getProgrammingData

5.16.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 token

5.16.4 请求参数

该接口通过 Query 参数传递以下字段：

字段	类型	是否必填 / 条件	说明
uuid	string	否	单次烧录跟踪 ID；未提供时由 `getProgrammingData` 生成，后续调用需保留并复用
ppcliVersion	string	否	客户端脚本版本
usafeSN	string	兼容旧流程	自 3.1.1 起不再作为新流程的主要参数；存在会话工站时优先使用工站信息
chipIndex	integer	否	自 3.1.1 起不再使用

5.16.5 接口说明

- **烧录跟踪**： 首次调用建议不传 `uuid`。获取返回的 `uuid` 后，应在后续 `/uploadRequestData`、`/executeApdu` 或烧录结果上报中复用。
- **SE 烧录**： SE-only 场景下，即使返回的 `bundles` 为空数组，也会创建烧录记录。
- **文件下载**： 大文件的 `downloadUrl` 应以实际响应为准。
- **敏感材料**： 敏感生产材料不会通过 `bundle` 原文下发。

5.16.6 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/getProgrammingData?ppcliVersion=1.0.0
```

5.16.7 返回参数

接口采用统一 JSON 响应格式，`respData` 为批次 Bundle 对象，包含以下字段：

字段	类型	说明
batchId	string	批次 ID
batchName	string	批次名称
uuid	string	本次烧录跟踪 ID
bundles	object[]	本次烧录的数据条目；SE-only 场景下可能返回空数组

5.16.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "batchId": "10",
    "batchName": "BATCH-001",
    "uuid": "example-programming-trace",
    "bundles": []
  },
  "respTime": 1792108800
}
```

5.16.9 主要错误码

错误码	说明
8013	Bundle 数据解析失败
8030	无可用工站批次
8050	鉴权失败

其他错误码见附录。

5.17 获取批次元数据

本接口用于获取完整的批次元数据及固件等文件的下载地址，供客户端准备设备烧录。

5.17.1 请求 Method

`GET`

5.17.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/getMetadata

5.17.3 请求头

建议提供以下请求头，以便解析当前工站和工厂上下文：

参数	是否必填 / 条件	说明
x-snbps-session	建议提供	工站会话令牌，使用 `/login` 接口返回的 `token`

本接口未列入 `UkeyPermissionFilter` 的强制校验路径。

5.17.4 请求参数

该接口通过 Query 参数传递以下字段：

字段	类型	是否必填 / 条件	说明
uuid	string	否	单次烧录跟踪 ID；未提供时可由后续 `getProgrammingData` 生成，后续调用需保留
ppcliVersion	string	否	客户端脚本版本
usafeSN	string	兼容旧流程	自 3.1.1 起不再作为新流程的主要参数；存在会话工站时优先使用工站信息
chipIndex	integer	否	自 3.1.1 起不再使用

5.17.5 接口说明

- **元数据获取**： 返回完整的批次 Bundle，执行必要的元数据处理，并移除指定的密钥材料。
- **生产记录**： 本接口不会代替 `/getProgrammingData` 创建本次烧录记录，响应中的 `uuid` 不能作为新生产会话已创建的依据。
- **工站上下文**： 提供有效的 `x-snbps-session` 可用于解析当前工站信息。

5.17.6 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/getMetadata?usafeSN=USAFE-001
```

5.17.7 返回参数

接口采用统一 JSON 响应格式，`respData` 为批次 Bundle 对象，包含以下字段：

字段	类型	说明
batchId	string	批次 ID
batchName	string	批次名称
uuid	string	响应中的烧录跟踪 ID，不代表已创建生产记录
bundles	object[]	完整批次数据条目

5.17.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "batchId": "10",
    "batchName": "BATCH-001",
    "uuid": "example-programming-trace",
    "bundles": []
  },
  "respTime": 1792108800
}
```

上述示例中的空 `bundles` 仅用于展示响应结构，不代表所有批次均返回空数组。

5.17.9 主要错误码

错误码	说明
8013	数据解析失败
8030	无可用工站批次

其他错误码见附录。

5.18 获取完整烧录数据（兼容接口）

本接口为旧版完整 Bundle 获取接口，其服务方法已标记为 `@Deprecated`。新客户端建议使用 `/getMetadata` 和 `/getProgrammingData`。

5.18.1 请求 Method

`GET`

5.18.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/getPPData

5.18.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.18.4 请求参数

该接口通过 Query 参数传递以下字段：

字段	类型	是否必填 / 条件	说明
uuid	string	否	单次烧录跟踪 ID；新流程中未提供时由 `getProgrammingData` 生成，后续调用需保留

字段	类型	是否必填 / 条件	说明
ppcliVersion	string	否	客户端脚本版本
usafeSN	string	兼容旧流程	自 3.1.1 起不再作为新流程的主要参数；存在会话工站时优先使用工站信息
chipIndex	integer	否	自 3.1.1 起不再使用

5.18.5 接口说明

- **批次查询**： 当前接口根据 `usafeSN` 定位批次，生成新的 `uuid`，并返回处理后的完整烧录数据。
- **无可用批次**： 未找到可用批次时，可能返回 `null`。
- **兼容性**： 本接口用于兼容旧客户端，新客户端应使用 `/getMetadata` 和 `/getProgrammingData`。

5.18.6 请求示例

```
GET {BASE_URL}/api/iot-factory-edge-service/getPPData?usafeSN=USAFE-001
```

5.18.7 返回参数

接口采用统一 JSON 响应格式，`respData` 为批次 Bundle 对象，包含以下字段：

字段	类型	说明
batchId	string	批次 ID
batchName	string	批次名称
uuid	string	本次返回的烧录跟踪 ID
bundles	object[]	处理后的完整烧录数据条目

5.18.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "batchId": "10",
    "batchName": "BATCH-001",
    "uuid": "example-programming-trace",
    "bundles": []
  },
  "respTime": 1792108800
}
```

示例中的空 `bundles` 仅用于说明响应结构，不能作为所有批次的实际返回内容。SE-only 场景下，`getProgrammingData` 可返回空 `bundles`。

5.18.9 Bundle 通用结构

`bundles` 数组中的每个 Bundle 对象包含以下字段：

字段	类型	说明
type	string	Bundle 类型
section	string	Bundle 所属 Section
ownerId	string	所属对象 ID
authToken	string	认证令牌
authCert	string	认证证书
authSign	string	认证签名
items	object[]	Bundle 数据条目列表

`items` 数组中的每个条目包含以下字段：

字段	类型	说明
itemId	string	条目 ID
key	string	条目标识
objectId	string	对象 ID
parentId	string	父对象 ID
fieldType	string	字段类型
attribute	JSON object	条目属性，具体结构由字段类型定义

不同批次的 Bundle 字段及内容可能不同。客户端应以服务端实际返回的数据结构和下载地址为准，不应将空 `bundles` 示例视为所有批次的实际数据。

5.18.10 主要错误码

错误码	说明
8050	鉴权失败

本接口的其他异常行为受旧版流程影响，具体错误码见附录。

5.19 提交中间烧录请求数据

本接口用于根据批次 Bundle 条目上报芯片执行结果，并由对应处理器生成下一阶段的烧录数据。

5.19.1 请求 Method

`POST`

5.19.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/uploadRequestData

5.19.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.19.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
batchId	string	是	当前工厂的批次 ID
uuid	string	业务上必填	本次 `/getProgrammingData` 返回的烧录跟踪 ID；IDRule 取号要求使用有效且稳定的 `traceId`
bundles	object[]	按需	上报的 Bundle 数据；为空时当前实现返回 `respData=null`
bundles[].type	string	按模板	Bundle 类型
bundles[].section	string	按模板	Bundle 所属 Section
bundles[].deviceId	string	IDRule 场景建议提供	稳定的设备标识；未提供时尝试从 `uuid` 对应记录中解析，首次取号后不得更换
bundles[].items	object[]	非空 Bundle 中必填	Bundle 条目列表，不允许空 `items` 或 `null` 条目
bundles[].items[].key	string	是	非空的条目标识
bundles[].items[].response	object[]	按条目协议	上一步执行结果列表，具体字段由 `fieldType` 及对应处理器协议定义

5.19.5 接口说明

- **接口用途**： 本接口用于提交中间烧录数据，不是最终烧录结果上报接口。
- **数据结构**： `items[].key` 和 `response` 必须依据实际批次模板及处理器协议填写，不能直接使用示例中的占位符。
- **设备标识**： IDRule 场景下，同一设备应复用稳定的设备标识。首次取号后，不得在后续烧录步骤中更换 `deviceId`。
- **空数据处理**： `bundles` 为空时，当前实现返回 `respData=null`。

5.19.6 请求示例

```
{
  "batchId": "10",
  "uuid": "<uuid returned by getProgrammingData>",
  "bundles": [
    {
      "section": "<template section>",
      "deviceId": "DEVICE-001",
```

5.19.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
batchId	string	批次 ID
batchName	string	批次名称
uuid	string	本次烧录跟踪 ID
bundles	object[]	对应处理器生成的下一阶段烧录数据条目

5.19.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "batchId": "10",
    "batchName": "BATCH-001",
    "uuid": "example-programming-trace",
    "bundles": []
  },
  "respTime": 1792108800
}
```

5.19.9 主要错误码

错误码	说明
8002	请求参数无效
8003	数据处理失败
8032	批次不存在
8040–8046	证书或密钥协商相关错误
8065	IDRule 规则无效
8066	号段已耗尽
8067	状态冲突
8068	缺少稳定设备标识
8050	鉴权失败

其他错误码见附录。

5.20 上报烧录结果

本接口用于结束已有的设备烧录记录，并生成或更新设备档案，同时兼容旧版客户端。

5.20.1 请求 Method

`POST`

5.20.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/programmingResult

5.20.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.20.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
uuid	string	是（对接约定）	已有烧录记录的 traceId
batchId	integer（int64）	建议提供	烧录记录中没有批次信息时用于补充定位
result	boolean	建议明确提供	`true` 表示成功，`false` 或 `null` 按失败处理
failureReason	string	否	烧录失败原因
deviceProfile	object	否	CPS 实际烧录数据，字段参见生产数据上传接口中的 `deviceRecord`，包括 `deviceId`、`firmwareVersion`、`seid`、`cplc`、`matterData`、`factoryData` 和 `customData`
changeKeys	object	否	兼容字段，接受 `null`、空字符串或 JSON 对象，其他类型返回 `8002`；当前烧录结果处理流程未进一步使用该字段

5.20.5 接口说明

- **记录处理**： 当烧录记录不存在，或无法定位当前工厂批次时，当前方法会跳过设备建档并正常返回。
- **重复上报**： 相同状态重复上报，或烧录记录已处于成功状态时，不会重新处理。
- **结果确认**： 接口返回 `respCode="8000"` 不代表一定生成了新的设备档案，应通过设备查询接口确认。

- **自定义数据：** `customData` 仅允许字符串类型的值，最多包含 50 项。
- **兼容性：** 新版统一生产数据上传流程可使用 `/v1/deviceData/upload` 完成烧录结果上报。

5.20.6 请求示例

```
{
  "uuid": "<current programming uuid>",
  "batchId": 10,
  "result": true,
  "deviceProfile": {
    "deviceId": "DEVICE-001",
    "firmwareVersion": "1.2.3",
    "customData": {
      "fixture": "A01"
    }
  }
}
```

5.20.7 返回参数

接口采用统一 JSON 响应格式，成功时 `respData` 为 `null`。

5.20.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": null,
  "respTime": 1792108800
}
```

5.20.9 主要错误码

错误码	说明
8002	请求参数无效
8007	数据处理或存储错误
8037	设备档案生成失败
8050	鉴权失败

其他错误码见附录。

5.21 SE APDU 步进烧录

本接口用于执行 SE 设备的 APDU 步进烧录。客户端按照服务端返回的 APDU 指令列表执行操作，并提交执行结果，以获取下一阶段指令。当 `nextStep` 返回 `EOF` 时，表示烧录流程结束。

5.21.1 请求 Method

`POST`

5.21.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/executeApdu

5.21.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.21.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
batchId	string	是	当前工厂的 SE 配置批次 ID，不能省略并仅依赖工站绑定关系
cplc	string	SE 对接时提供	SE 设备的 CPLC 协议值，通常为 84 个字符，用于 SE 设备记录
seId	string	SE 对接时提供	SE 标识，字段名区分大小写，使用 `seId`
nfcUid	string	动力钥匙场景	设备物理编号的协议 Hex 值，兼容输入别名 `nfcId`；首步与续步必须保持一致
programmingTraceId	string	BOF 可省略，其他步骤必填	烧录跟踪 ID，兼容输入别名 `uuid`；`SE_MCU` 场景复用 Matter 会话的 `traceId`，响应仅返回 `programmingTraceId`
currentStep	string	是	首次调用使用 BOF；后续调用原样使用上次响应中的 `nextStep`，不得自行构造步骤游标
commandResults	object	续步必填	上一步 APDU 执行结果；BOF 不传，续步要求 `succeed=true` 且 `results` 与上次指令对应
commandResults.succeed	boolean	续步必填	上一步是否执行成功；`false` 将中止流程并返回 `8053`
commandResults.results	object[]	续步按上次响应提供	APDU 执行结果列表，每项包含 `index`（integer）及 `command`、`checker`、`result`（string）；`index` 应与上一步指令的字符串序号按数值对应

5.21.5 接口说明

- **首次调用**： 使用 `currentStep="BOF"` 开始烧录，并保存响应中的 `programmingTraceId`，用于后续步骤。
- **指令执行**： 客户端必须按照 `commands` 的顺序执行 APDU，并在续步请求中提交上一批指令的真实执行结果。
- **步骤控制**： 后续请求应原样使用上次返回的 `nextStep`，不得解析或修改服务端步骤游标。收到 `EOF` 后停止循环，不得再次提交 `EOF`。
- **会话恢复**： 返回 `8054` 表示当前会话无法继续，应从 `BOF` 重新开始，不得自动重放之前的 APDU 指令。
- **会话有效期**： 当前 SE 会话有效期以服务端配置常量为准。

- **动力钥匙**： 首步和续步中的 `nfcUid`（或 `nfcId`）必须保持一致，且不能将 `seId` 作为 NFC UID 使用。

5.21.6 请求示例

首次调用使用 `BOF`：

```
{
  "batchId": "10",
  "cplc": "<device CPLC>",
  "seId": "<device SEID>",
  "currentStep": "BOF"
}
```

5.21.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段 类型 说明 nextStep

字段	类型	说明
nextStep	string	服务端返回的下一步骤游标；EOF 表示流程结束
programmingTraceId	string	本次 SE 烧录跟踪 ID
commands	object[]	本步骤需要执行的 APDU 指令列表
commands[].index	string	APDU 指令序号
commands[].checker	string	APDU 响应校验器
commands[].command	string	需要执行的 APDU Hex 指令

5.21.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "nextStep": "<server step cursor>",
    "programmingTraceId": "example-programming-trace",
    "commands": [
      {
        "index": "1",
        "checker": "<server-provided checker>",
        "command": "<server-provided APDU hex>"
      }
    ]
  },
  "respTime": 1792108800
}
```

5.21.9 APDU 续步请求示例

客户端完成上一批 APDU 指令后，使用服务端返回的 `programmingTraceId` 和 `nextStep` 提交执行结果。

```
{
  "batchId": "10",
  "programmingTraceId": "<programmingTraceId from previous response>",
  "currentStep": "<nextStep from previous response>",
  "cplc": "<same device CPLC>",
  "seId": "<same device SEID>",
  "commandResults": {
    "succeed": true,
    "results": [
      {
        "index": 1,
        "command": "<original command>",
        "checker": "<original checker>",
        "result": "<actual card response hex>"
      }
    ]
  }
}
```

动力钥匙场景下，首次请求和每次续步请求均需携带相同的 `nfcUid`（或输入别名 `nfcId`）。

原始生产密钥文件不应额外下载至客户端。相关生产材料由 Edge 准备，并通过当前步骤的 APDU 指令返回。

完整烧录成功并到达 `EOF` 后，沿用现有产量统计和设备档案处理流程。若卡端明确执行失败，不得将 `commandResults.succeed` 上报为 `true`。

5.21.10 动力钥匙设备绑定规则

- 设备标识： 当前动力钥匙流程必须同时提供 `nfcUid`（兼容输入别名 `nfcId`）和 `seId`。
- 数据格式： 两个标识均按设备协议传入偶数位十六进制字符串。当前分配入口要求字段非空、长度不超过 64 字符，并统一转换为大写。
- 别名使用： 不得同时提交值不一致的 `nfcUid` 和 `nfcId`。首次调用及所有续步请求均应保持相同的 `nfcUid` 和设备身份。
- 材料绑定： 生产材料按工厂、批次、NFC UID 和 SEID 的设备绑定关系保存。同一设备即使更换 `traceId` 或重新发送 `BOF`，也会优先复用已绑定的材料。
- 绑定冲突： 当 NFC UID 或 SEID 已存在绑定记录，但另一设备标识不一致时，返回 `8073`。
- 材料耗尽： 无可用生产材料时返回 `8072`。
- 历史数据： 批次中存在尚未迁移的旧 Redis 或旧生产记录分配状态时，也可能返回 `8073`，不能通过重复发送 `BOF` 重新领取材料。

5.21.11 主要错误码

错误码	说明
8006	批次或烧录记录缺失
8047	SE 配置缺失
8048	数据解析失败
8049	CPLC 相关错误
8052	步骤执行结果不匹配

错误码	说明
8053	上一步执行失败
8054	会话无法继续，需要重新开始
8055	APDU 处理失败
8056	配置解密失败
8070–8074	生产数据、材料分配或配置相关错误
8050	鉴权失败

其他错误码见附录。

5.22 获取烧录脚本地址

本接口用于获取指定批次对应的烧录脚本下载地址。

5.22.1 请求 Method

`POST`

5.22.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/getScriptFile

5.22.3 请求头

建议提供以下请求头，以便解析当前工站和工厂上下文：

参数	是否必填	说明
x-snbps-session	建议提供	工站会话令牌，使用 `/login` 接口返回的 `token`

本接口未列入 `UkeyPermissionFilter` 的强制校验路径。

5.22.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
batchId	integer（int64）	按批次查询时必填	批次 ID
productId	integer（int64）	否	兼容产品维度查询参数
chipPart	string	按需	芯片型号
chipVendor	string	按需	芯片厂商

5.22.5 接口说明

- **下载地址**： 客户端应使用接口实际返回的 `fileUrl` 下载烧录脚本。
- **脚本校验**： 接口返回 HTTP 成功不代表客户端已完成本地脚本下载及校验。

5.22.6 请求示例

```
{
  "batchId": 10
}
```

5.22.7 返回参数

接口采用统一 JSON 响应格式，respData 包含以下字段：

字段	类型	说明
fileUrl	string	服务端返回的烧录脚本下载地址

5.22.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "fileUrl": "<server-generated script download URL>"
  },
  "respTime": 1792108800
}
```

5.22.9 主要错误码

错误码	说明
8022	获取烧录脚本失败

其他错误码见附录。

5.23 下载文件

本接口用于根据本地文件记录 ID 下载文件。

5.23.1 请求 Method

`GET`

5.23.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/download

5.23.3 请求头

需要工厂上下文的调用应提供以下请求头：

参数	是否必填 / 条件	说明
x-snbps-session	需要工厂上下文时	有效的工站会话令牌，使用 `/login` 接口返回的 `token`

5.23.4 请求参数

该接口通过 Query 参数传递以下字段：

字段	类型	是否必填	说明
fileId	integer（int64）	是	本地文件记录 ID
fileName	string	否	客户端期望的下载文件名

5.23.5 接口说明

- **响应格式**： 下载成功时返回二进制文件流，并包含 `Content-Disposition` 等下载响应头，不使用统一的 `ClientR` JSON 响应结构。
- **下载地址**： 客户端可直接使用 `/getScriptFile` 或 `/getMetadata` 返回的完整下载 URL，避免手动拼接。
- **会话要求**： 需要工厂上下文时，应携带有效的 `x-snbps-session`。

5.23.6 请求示例

```
curl -L "$BASE_URL/api/iot-factory-edge-service/download?fileId=100&fileName=burn.zip" \
  -H "x-snbps-session: $SESSION" \
  -o burn.zip
```

5.23.7 返回参数

成功时返回文件二进制数据流，不返回 JSON 格式的 `respData`。

5.23.8 主要错误码

错误码	说明
8022	文件记录不存在
8023	文件下载失败

其他错误码见附录。

5.24 上传设备事件日志

本接口用于接收设备烧录过程中的事件日志，按照配置的优先级过滤后通过 MQTT 上报。发送失败时，保存相应的补偿记录。

5.24.1 请求 Method

`POST`

5.24.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/v1/eventLog/upload

5.24.3 请求头

除公共请求头外，本接口还需提供以下请求头：

参数	是否必填	说明
x-snbps-session	是	工站会话令牌，使用 `/login` 接口返回的 `token`

5.24.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
programmingTraceId	string	是（非 null）	本次烧录跟踪 ID，兼容输入别名 `uuid`
events	object[]	是（非 null）	事件日志列表，空数组可能被直接忽略
events[].eventName	string	建议提供	事件名称
events[].timestamp	string	建议提供	客户端事件时间，格式沿用事件协议
events[].source	string	建议提供	事件来源
events[].priority	string	建议提供	事件优先级，使用 1–5 的数字字符串，数值越大优先级越高；空值或非法值会被过滤
events[].status	string	建议提供	事件状态：`success` 或 `failure`
events[].content	string	否	日志内容，当前实现超过 256 字符时会截断

5.24.5 接口说明

- **事件过滤**： 服务端根据配置的优先级阈值过滤事件。低于阈值或优先级无效的事件不会继续上报。
- **日志同步**： HTTP 请求成功仅表示接口已接收并处理请求，不代表所有事件均已成功同步至云端。
- **失败补偿**： MQTT 发送失败时，保存相应的补偿记录。

5.24.6 请求示例

```
{
  "programmingTraceId": "<current programming traceId>",
  "events": [
    {
      "eventName": "ProgrammingFinished",
      "timestamp": "2026-10-16T10:00:00+08:00",
      "source": "PS",
      "priority": "5",
      "status": "success",
      "content": "Burn completed"
    }
  ]
}
```

5.24.7 返回参数

接口采用统一 JSON 响应格式，成功时 `respData` 为 `null`。

5.24.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": null,
  "respTime": 1792108800
}
```

5.24.9 主要错误码

错误码	说明
8050	鉴权失败

此外，请求参数绑定或校验失败也可能产生相应错误，其他错误码见附录。

5.25 生成 Matter Verifier

本接口用于调用 Matter SDK，生成 Matter 配对密码、Salt 和 Verifier。

5.25.1 请求 Method

`POST`

5.25.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/generateVerifier

5.25.3 请求头

当前 `application.yml` 未将本接口列入平台鉴权排除项。如果部署环境启用了平台鉴权，还需提供以下请求头：

参数	是否必填 / 条件	说明
x-snbps-auth-token	启用平台鉴权时	平台鉴权令牌，与工站会话令牌 `x-snbps-session` 不同

5.25.4 请求参数

本接口无需提供请求参数。

5.25.5 接口说明

- **生成参数**： 当前固定使用 `saltLen=32` 字节、`iterations=1000`，不支持通过请求 Body 覆盖。
- **运行依赖**： 服务端需要部署对应的原生 Matter SDK。
- **状态判断**： 外层 `respCode` 与 `respData.status` 含义不同。前者表示接口业务响应状态，后者为 Matter SDK 返回的状态码。
- **示例数据**： 返回示例中的数值仅用于说明字段类型。

5.25.6 请求示例

```
POST {BASE_URL}/api/iot-factory-edge-service/generateVerifier
```

5.25.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
status	integer	Matter SDK 返回的状态码
passcode	integer	Matter 配对密码
salt	string	Base64 编码的 Salt
verifier	string	Base64 编码的 Verifier

5.25.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "status": 0,
    "passcode": 20202021,
    "salt": "<Base64 salt>",
    "verifier": "<Base64 verifier>"
  },
  "respTime": 1792108800
}
```

5.26 生成 Matter 配网载荷

本接口用于根据指定的 Matter 参数调用 Matter SDK，生成二维码配网载荷和手工配网码。

5.26.1 请求 Method

`POST`

5.26.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/generateOnboardingData

5.26.3 请求头

当前 `application.yml` 未将本接口列入平台鉴权排除项。如果部署环境启用了平台鉴权，还需提供以下请求头：

参数	是否必填 / 条件	说明
x-snbps-auth-token	启用平台鉴权时	平台鉴权令牌，与工站会话令牌 `x-snbps-session` 不同

5.26.4 请求参数

本接口通过 Query 参数或表单参数传递以下字段，不使用 JSON 请求 Body。

字段	类型	是否必填	说明
vid	integer	是	Matter Vendor ID（VID），使用十进制整数，与 Matter 二维码找回接口使用的十六进制文本不同
pid	integer	是	Matter Product ID（PID），使用十进制整数
passcode	integer	是	Matter 配对密码
commissioningFlow	integer	是	Matter SDK 配网流程参数
discoveryMode	integer	是	Matter SDK 设备发现模式参数

5.26.5 接口说明

- **参数传递**： 虽然本接口使用 `POST` 方法，但五个参数均通过 `@RequestParam` 绑定，客户端应通过 URL Query 或表单参数传递，不应使用 JSON 请求 Body。
- **参数校验**： 各参数的有效取值范围由底层 Matter SDK 处理。
- **VID/PID 格式**： 本接口使用十进制整数，与 `/v1/matter/onboarding-payloads/recover` 接口使用的十六进制字符串不同。

5.26.6 请求示例

```
POST {BASE_URL}/api/iot-factory-edge-service/
generateOnboardingData?vid=65522&pid=32769&passcode=20202021&commissioningFlow=0&discoveryMode=2
```

5.26.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
status	integer	Matter SDK 返回的状态码
discriminator	integer	Matter 配网 `Discriminator`
qrcode	string	Matter 二维码配网载荷
manualcode	string	Matter 手工配网码，注意字段名为全小写的 `manualcode`

5.26.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
```

5.27 生成 Matter Unique ID

本接口用于调用 Matter SDK 生成 Matter Unique ID。

5.27.1 请求 Method

`POST`

5.27.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/generateUniqueId

5.27.3 请求头

当前 `application.yml` 未将本接口列入平台鉴权排除项。如果部署环境启用了平台鉴权，还需提供以下请求头：

参数	是否必填 / 条件	说明
x-snbps-auth-token	启用平台鉴权时	平台鉴权令牌，与工站会话令牌 `x-snbps-session` 不同

5.27.4 请求参数

本接口无需提供请求参数或请求 Body。

5.27.5 接口说明

- **生成方式**： 由 Matter SDK 生成 Unique ID，无需客户端提供生成参数。
- **状态判断**： Matter SDK 成功状态码通常为 `0`，外层接口业务成功码为 `8000`，两者含义不同。

5.27.6 请求示例

```
POST {BASE_URL}/api/iot-factory-edge-service/generateUniqueId
```

5.27.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
status	integer	Matter SDK 返回的状态码
result	string	Base64 编码的 Unique ID

5.27.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "status": 0,
    "result": "<Base64 uniqueId>"
  },
```

5.28 查询设备信息（旧接口，已废弃）

本接口用于兼容旧版客户端的设备信息查询。新客户端应使用 `/v1/device/detail`，或根据工站查询场景使用 `/v1/device/information/query`。

5.28.1 请求 Method

`POST`

5.28.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/queryDeviceData

5.28.3 请求头

建议提供以下请求头，以便解析当前工站和工厂上下文：

参数	是否必填 / 条件	说明
x-snbps-session	建议提供	工站会话令牌，使用 `/login` 接口返回的 `token`

本接口未列入 `UkeyPermissionFilter` 的强制校验路径。

5.28.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
batchId	string	是	批次 ID
paramKey	string	是	用于定位参数所属的 Key
paramField	string	是	用于定位具体字段
paramValue	string	是	用于定位设备的字段值
queryKeys	string[]	是	支持的 key 请参考可查询键列表 (queryKeys V1)

5.28.5 接口说明

- **兼容性：** 本接口已在 v1.3 文档中声明废弃，仅保留用于兼容旧版客户端。
- **查询来源：** 当批次已删除、文件无法读取或本地设备档案未命中时，接口可能回源查询历史记录或云端数据。
- **查询参数：** `queryKeys` 应遵循旧版 Bundle 查询协议，不能直接使用新版 QueryKeys V2 的字段定义。

5.28.6 请求示例

```
{
  "batchId": "10",
  "paramKey": "<legacy protocol key>",
  "paramField": "<legacy protocol field>",
  "paramValue": "<actual value>",
  "queryKeys": [
```

5.28.7 返回参数

接口采用统一 JSON 响应格式，`respData` 包含以下字段：

字段	类型	说明
batchId	string	批次 ID
paramKey	string	查询使用的参数 Key
paramField	string	查询使用的字段
paramValue	string	查询使用的字段值
data	map<string,map<string,object>>	按旧版数据结构返回的设备信息

5.28.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
  "respData": {
    "batchId": "10",
    "paramKey": "<legacy protocol key>",
    "paramField": "<legacy protocol field>",
    "paramValue": "<actual value>",
    "data": {}
  },
  "respTime": 1792108800
}
```

5.28.9 主要错误码

错误码	说明
8006	批次不存在
8003	旧版设备信息查询处理失败

其他错误码见附录。

5.29 更新设备信息（旧接口，已废弃）

本接口用于兼容旧版设备数据模型的更新操作。新客户端应使用 `/v1/updateDevicesProfile`。

5.29.1 请求 Method

`POST`

5.29.2 请求 URL

{BASE_URL}/api/iot-factory-edge-service/updateDeviceInfo

5.29.3 请求头

建议提供以下请求头，以便解析当前工站和工厂上下文：

参数	是否必填 / 条件	说明
x-snbps-session	建议提供	工站会话令牌，使用 `/login` 接口返回的 `token`

本接口未列入 `UkeyPermissionFilter` 的强制校验路径。

5.29.4 请求 Body

该接口的 `Body` 包含以下参数：

字段	类型	是否必填 / 条件	说明
batchId	integer（int64）	是	批次 ID，注意与旧版查询接口使用的 string 类型不同
paramKey	string	是	用于定位参数所属的 Key
paramField	string	是	用于定位具体字段
paramValue	string	是	用于定位设备的字段值
data	map<string,map<string,string>>	是	需要更新的设备数据

5.29.5 接口说明

- **兼容性**： 本接口沿用旧版设备数据模型，新客户端应使用 `/v1/updateDevicesProfile`。
- **批次处理**： 当前服务在指定批次不存在时会直接返回，控制器仍可能返回请求中的 `data`。
- **更新确认**： 返回 `respCode="8000"` 及 `data` 回显，不代表数据一定已成功持久化。不能仅凭接口返回成功判断设备信息已更新。

5.29.6 请求示例

```
{
  "batchId": 10,
  "paramKey": "<legacy protocol key>",
  "paramField": "<legacy protocol field>",
  "paramValue": "<actual value>",
  "data": {}
}
```

5.29.7 返回参数

接口采用统一 JSON 响应格式，`respData` 为请求中的 `data` 对象，类型为 `map<string, map<string, string>>`。该字段仅回显请求数据，不代表重新查询得到的实际持久化结果。

5.29.8 返回示例

```
{
  "success": true,
  "respCode": "8000",
  "respMsg": "NoError",
```

6. 附录

6.1 附录一：可查询键列表 (queryKeys V2)

数据项	键
Matter 配网相关- PBKDF（密码派生）算法的迭代次数（1000 次）	matterData.iterations
Matter 配网相关-盐值长度（32 字节）	matterData.saltLength
Matter 配网相关-配网流程类型（0 表示标准流程）	matterData.commissioningFlows
设备可发现模式	matterData.discoverableMode
Matter 设备所属 VID	matterData.vid
Matter 设备所属 PID	matterData.pid
区分码	matterData.discriminator
Matter 配网参数-是否启用轮换设备 ID（true 表示启用）	matterData.enableRotatingDeviceId
Matter 手工配网码	matterData.manualCode
Matter 配网密码	matterData.passcode
Matter 配网参数二维码	matterData.qrcode
Matter 配网相关-用于密码派生算法的盐值	matterData.salt
Matter 设备标识-设备唯一标识符（Base64 编码）	matterData.uniqueid
Matter 配网参数-校验值	matterData.verifier
Matter 烧录设备流水号	matterData.serialNumber
Matter DAC 证书 MD5 值	matterData.dacCertMd5
Matter PAI 证书 MD5 值	matterData.paiCertMd5
Matter CD 文件名称	matterData.cdFileName
Matter CD 文件 MD5 值	matterData.cdFileMd5
工厂数据相关，参考 OBIS 产品界面配置	factoryData.vendorName
工厂数据相关，参考 OBIS 产品界面配置	factoryData.hardwareVersion
工厂数据相关，参考 OBIS 产品界面配置	factoryData.hardwareVersionString

数据项	键
工厂数据相关，参考 OBIS 产品界面配置	factoryData.partNumber
工厂数据相关，参考 OBIS 产品界面配置	factoryData.productUrl
工厂数据相关，参考 OBIS 产品界面配置	factoryData.productLabel
工厂数据相关，参考 OBIS 产品界面配置	factoryData.productName
工厂数据相关，参考 OBIS 产品界面配置	factoryData.startAddress
烧录状态	programmingData.programmingStatus
烧录日志	programmingData.programmingLog
烧录时间	programmingData.programmingTime
烧录设备芯片中的 mcuId	chipData.chipUid
芯片蓝牙 mac 地址	chipData.bleMac
固件名称 (返回固件列表)	firmwareData.fwName
固件 md5 (返回固件列表)	firmwareData.fwMd5
自定义字段	customData.自定义字段

6.1.1 补充查询键

> **提示**
> 本接口的响应字段遵循 `queryKeys V2` 命名规则，Matter 数据使用 `matterData.manualCode`、`matterData.qrcode` 和 `matterData.uniqueid`，与“查询设备综合信息”接口中的 `manualPairingCode`、`setupPayload` 不同，请勿混用。

数据项	键	说明
SE 数据	seData	返回设备档案中完整的 SE 数据对象
整组查询	matterData / factoryData / programmingData / chipData / firmwareData / customData	不带点号的分组名用于查询该分组支持的字段，具体返回内容以当前接口实现为准

6.2 附录二：错误码列表

下表列出了接口使用的客户端业务错误码及对应说明。

- 不同接口仅涉及其中部分错误码，具体原因及处理方式见对应接口说明。
- 参数校验、平台鉴权及框架处理过程中产生的错误可能使用其他错误码，不限于本表。
- SDK 内部的 `status=0` 与接口统一响应中的 `respCode="8000"` 含义不同，不应混用。

respCode	枚举名称	说明
8000	NO_ERROR	操作成功（NoError）
8001	REQUEST_FAILED	请求失败（RequestFailed）
8002	PARAM_ERROR	参数错误（ParamError）
8003	METHOD_ERROR	方法错误（MethodError）
8004	AUTH_ERROR	鉴权失败（AuthError）
8005	USAFE_TYPE_ERROR	U-safe 类型错误（USafeTypeError）
8006	DATA_NOT_FOUND	数据不存在（DataNotFound）
8007	DATA_ERROR	数据错误（DataError）
8008	NOT_ALLOWED	不允许执行该操作（NotAllowed）
8009	BATCH_EXPIRED_ERROR	批次已过期
8010	BATCH_STATUS_CLOSED	批次已关闭
8011	STATION_BATCH_HAS_BIND	该批次已绑定当前工站
8012	STATION_HAS_LOGIN	工站已登录（StationHasLogin）
8013	BATCH_BUNDLE_PARSE_ERROR	解析批次生成的 Bundle 文件失败
8014	STATION_PASSWORD_ERROR	工站密码错误（StationPasswordError）
8015	STATION_NOT_BIND_BATCH	工站未绑定任何批次
8016	STATION_NOT_EXIST	工站不存在
8017	UNKNOWN_LOGIN_TYPE	未知登录类型
8018	FACTORY_NOT_EXIST	工厂不存在
8019	FACTORY_STATUS_LOCKED	工厂已锁定
8020	USAFE_NOT_FOUNT	未找到 U-safe
8021	STATION_USAFE_NOT_MATCH	U-safe 与工站 ID 不匹配
8022	FETCH_BURN_SCRIPT_ERROR	获取烧录脚本失败

respCode	枚举名称	说明
8023	DOWNLOAD_BURN_SCRIPT_ERROR	下载烧录脚本失败
8024	PRELOGIN_AUTH_ERROR	登录前鉴权失败
8025	HSM_CERT_SYNC_REQUIRED	未找到 HSM 认证证书。请在云端编辑工厂 HSM 信息，以触发证书同步
8026	SIGN_VERIFY_ERROR	签名验证失败
8027	BATCH_HAS_DONE	批次已完成
8028	BATCH_FILE_NOT_EXIST	操作失败，产品批次文件不存在
8029	BATCH_IMPORT_FAILED	导入批次失败
8030	PS_BURN_STATION_RECORDING_FINISHED	U-safe SN 已完成烧录，或未绑定任何批次
8031	USAFE_UNAVAILABLE	U-safe 不可用
8032	BATCH_NOT_FOUND	未找到批次
8033	BATCH_DOWNLOAD_STATUS_NOT_WAIT	批次下载状态不是等待中
8034	DEVICES_PROGRAMMER_DATA_UPLOAD_ERROR	上传设备烧录器数据失败
8035	STATION_HAS_EDGE_BIND	工站已绑定来自 Edge Service 的批次
8036	CLOUD_FACTORY_ID_ERROR	云端工厂 ID 错误
8037	DEVICE_PROFILE_GENERATE_ERROR	生成设备档案失败
8038	UNSUPPORTED_CERTIFICATE_FORMAT	不支持的证书格式
8039	ORDER_NUMBER_CANNOT_BE_EMPTY	订单号不能为空
8040	CERT_CHAIN_VERIFY_FAILED	Silicon Labs 证书链验证失败
8041	PUBKEY_SIGNATURE_VERIFY_FAILED	芯片临时公钥签名验证失败
8042	ECDH_KEY_AGREEMENT_FAILED	ECDH 密钥协商失败
8043	SESSION_KEY_NOT_FOUND	会话密钥不存在或已过期
8044	KEY_WRAP_FAILED	密钥封装失败
8045	DAC_PUBKEY_SIGNATURE_VERIFY_FAILED	DAC 公钥签名验证失败
8046	BUNDLE_PUBLIC_KEY_NOT_FOUND	未找到指定 keyCode 对应的 Bundle 公钥材料
8047	SE_PROFILE_NOT_FOUND	未找到批次对应的 SE Profile

respCode	枚举名称	说明
8048	SE_PROFILE_PARSE_ERROR	解析 SE Profile 失败
8049	CPLC_INVALID	CPLC 无效，长度必须为 84
8050	PERMISSION_CHECK_DENY	授权验证失败，请重新获取验证权限
8051	SERVER_PERFORMANCE_LIMIT	已达到并发限制，请联系工厂管理员检查服务器
8052	APDU_SESSION_INVALID	executeApdu 会话无效，当前步骤的命令执行结果不符合预期
8053	APDU_STEP_FAILED	上一步 APDU 执行失败
8054	APDU_RESTART_REQUIRED	APDU 会话无效或已过期，请重新开始该设备的烧录流程
8055	APDU_PROCESS_FAILED	处理 APDU Profile 步骤失败
8056	SE_PROFILE_DECRYPT_ERROR	解密 SE Profile 失败
8057	TEST_FILE_PARAM_ERROR	测试文件上传参数无效
8058	TEST_FILE_MANIFEST_ERROR	测试文件清单无效
8059	TEST_FILE_STORAGE_ERROR	测试文件本地保存失败
8060	TEST_FILE_TASK_ERROR	保存测试文件上传任务失败
8061	TEST_FILE_UPLOAD_ERROR	上传测试文件至云端存储失败
8064	TEST_DATA_TASK_ERROR	保存测试数据上报任务失败
8065	ID_RULE_INVALID	ID Rule 描述信息无效
8066	ID_RULE_RANGE_EXHAUSTED	ID Rule 可分配范围已耗尽
8067	ID_RULE_CONFLICT	ID Rule 状态与批次描述信息冲突
8068	ID_RULE_DEVICE_KEY_REQUIRED	解析 ID Rule 时需要稳定的设备标识
8069	SCENARIO_TYPE_NOT_BE_NULL	scenarioType 不能为空
8070	DATA_INVALID	生产数据无效
8071	KEY_DECRYPT_FAILED	解密批次保护密钥失败
8072	DATA_EXHAUSTED	生产数据已耗尽
8073	ASSIGNMENT_CONFLICT	生产数据分配与批次或设备存在冲突
8074	PROFILE_INVALID	烧录要求配置 PERSONALIZE 操作及对应的 Applet 选择信息

respCode	枚举名称	说明
8101	DEVICE_NOT_FOUND	未找到设备，请检查查询参数
8102	MULTI_DEVICES_FOUND	查询到多个设备，请增加筛选条件
8103	QUERY_CONDITION_REQUIRED	请在 deviceLocator 中至少指定一个有效查询条件

6.3 附录三：可更新键列表 (updateDevicesProfile)

更新设备档案信息接口支持的 key:

数据项	键	值说明
设备扩展字段	extend.扩展字段的key	支持的扩展字段：vid, pid, chipId, qrcode, pairingCode
env	environment	test / prod
生产时间	manufactureTime	• string • 时间格式：2020-11-11T01:11:11+08:00
版本名	productVersion	string
批次名	productBatch	string
工厂名	factory	string
烧录站名	programmingStation	string
烧录状态	programmingStatus	Complete / Failed / Unreported
DAC 证书序列号	dacSerialNo	string
DAC 证书 commonName	dacCommonName	string
DAC 证书的 VID	dacVid	string
DAC 证书的 PID	dacPid	string
主体密钥标识	subjectKeyIdentifier	string
颁发机构密钥标识	authorityKeyIdentifier	string
DAC 生效日期	notBefore	• string • 时间格式：2020-11-11T01:11:11+08:00
DAC 失效日期	notAfter	• string • 时间格式：2020-11-11T01:11:11+08:00
配对码	pairingCode	string
二维码	qrcode	string

数据项	键	值说明
DAC 有效期的到期时间（日期，时间，时区）	expiryDate	• string • 时间格式：2020-11-11T01:11:11+08:00
DAC 剩余有效天数	leftDays	string
实际签发出的 DAC 证书	certificate	string
factoryData 中的字段	factoryData.key 名称	• string • 支持删除，可删除工厂数据中对应 key 的条目
脚本文件名称	programmingScript	• string • 支持删除，可删除对应名称的脚本文件；只能存在一个脚本文件，当上传一个新的脚本文件名称时，将会覆盖原有的脚本文件记录。
固件	firmwareData.固件名称	• 可更新对应名称固件的 startAddress； • 支持删除，可删除对应固件
otherFiles	otherFile.文件名称	• 可更新 otherFile 中对应文件名称的 key； • 支持删除，可删除对应名称的文件
attachmentFiles	attachment.文件名称	• 可更新 attachment 中对应文件名称的 key； • 支持删除，可删除对应名称的文件

6.4 附录四：可查询键列表 (queryKeys V1)

> **提示**
> - 如果某个数据项在产品中未配置，则接口返回该数据项的值为空。
> - 键 `fw1.fileName` 中的数字 `1` 表示第一个固件。

数据项	键
CD 文件 MD5	cdFile.md5
CD 文件名	cdFile.fileName
Commissioning flow	matterCommissioningParameter.commissioningFlows
DAC MD5	dacTemplate.md5
Discovery mode	matterCommissioningParameter.discoverableMode
Discriminator	matterCommissioningParameter.discriminator
EnableRotatingDeviceId	matterCommissioningParameter.enableRotatingDeviceId
hardwareVersion	hardwareVersion
hardwareVersionString	hardwareVersionString

数据项	键
mcuId	mcuId
PAI MD5	paiCert.md5
partNumber	partNumber
PID	pid
productLabel	productLabel
productUrl	productUrl
salt	matterCommissioningParameter.salt
softwareVersion	softwareVersion
softwareVersionString	softwareVersionString
SPAKE2+Iterations	matterCommissioningParameter.iterations
UniqueID	matterCommissioningParameter.uniqueId
vendorName	vendorName
verifier	matterCommissioningParameter.verifier
VID	vid
二维码	matterCommissioningParameter.qrcode
固件文件名	fw1.fileName
设备烧录结果	programmingStatus
手工配网码	matterCommissioningParameter.manualCode
原始配网密码	matterCommissioningParameter.passcode

6.5 典型业务调用流程

本节介绍工站烧录、测试与组装场景下的典型接口调用顺序。具体请求参数、响应结构及处理规则见对应接口说明。

6.5.1 工站烧录

工站完成登录、批次查询及分配后，获取烧录数据，执行设备烧录并上报结果。典型调用流程如下：

```
queryLoginType
    ↓
preLogin（按需）
    ↓
login
    ↓
```

对于 SE 烧录，还需注意以下处理规则：

- SE-only 场景： 可在 BOF 阶段创建生产记录。
- SE_MCU 混合烧录场景： 应复用 MCU 或 Matter 烧录阶段的同一 `traceId`，再通过 `executeApdu` 执行 SE 烧录操作，直至 EOF。

6.5.2 测试与组装

测试和组装场景通过 `/v1/deviceData/upload` 上报生产数据。如需关联测试文件，应先上传文件，再提交对应的生产数据。典型调用流程如下：

```
login
    ↓
testFile/upload（按需）
    ↓
deviceData/upload
```

调用时需注意：

- 测试场景： 首次上传且无附件时，可不提供 `traceId`，但必须提供 `scenarioType`。后续上传使用接口返回的 `traceId`。
- 文件关联： 如需关联测试文件，先调用 `/v1/testFile/upload` 上传文件，再将返回的 `traceId` 用于 `/v1/deviceData/upload`。
- 组装场景： 使用实际组装批次的 `batchId`，并通过 `deviceRecord.testData.result` 上报执行结果。

根据业务需要，扫码时可先调用 `/v1/matter/onboarding-payloads/recover` 恢复 Matter 配网数据；需要查询设备详细信息时，可调用 `/v1/device/information/query`。

版本：v3.2.2

OnBoard™ IoT Security FW Signing and Pushing CLI - v1.0.0

1. 修订记录

版本	更改时间	备注
v1.0.0	2025/02/17	初始版本

2. 概述

OnBoard™ IoT Security (OBIS) FW Signing and Pushing CLI 是专为 OBIS 打造的命令行工具，专门用于固件的签名与推送。它集成了多项核心功能，包括基于 PKI（公钥基础设施）的证书签名，KMS（密钥管理系统）的密钥配置，固件签名，公钥导出，以及向 OBIS 的固件推送。

本文将介绍 OBIS FW Signing and Pushing CLI 的常用命令，并提供各命令的参数说明与使用示例，帮助你高效完成固件签名与推送操作。

3. 语法

`snb [命令] [子命令] [参数]`

4. 命令列表

4.1 密钥配置命令

设置访问密钥（Access Key）和私密密钥（Secret Key）。

4.1.1 命令

`config`

4.1.2 参数

参数	类型	是否必填	说明
`--accessKey`	string	是	设置访问密钥，可在 OBIS 的用户页面中查看。
`--secretKey`	string	是	设置私密密钥，可在 OBIS 的用户页面中查看。

4.1.3 示例

```
snb config --accessKey <AK> --secretKey <SK>
```

4.2 固件签名命令

使用指定密钥对输入数据执行签名操作。

4.2.1 命令

`kms`

4.2.2 子命令

`sign`

4.2.3 参数

参数	类型	是否必填	说明
`--algorithm`	string	是	配置签名算法，当前支持 `ecdsa_sha_256`。
`--keyId`	string	是	指定执行签名操作时使用的密钥 ID。
`--inputFile`	string	否	指定需要进行签名的目标文件。
`--inputData`	string	否	输入需要签名的数据，格式为十六进制的字符串。
`--inputHash`	string	否	输入需要签名的预哈希数据，格式为十六进制的字符串。
`--outputFile`	string	否	指定签名结果的输出文件，格式为 **.der**。默认为标准输出（stdout）。

参数	类型	是否必填	说明
`--resourcePackageId`	string	否	指定产品的资源包 ID。当访问产品密钥时，此参数为必填项。

> **提示**
>
> - 必须在 `inputFile`，`inputData` 或 `inputHash` 中提供至少一个参数。
> - 当输出至标准输出（stdout）时，**.der** 格式的签名结果以十六进制字符串形式表示。

4.2.4 示例

```
snb kms sign --algorithm ecdsa_sha_256 --keyId <keyID> --
inputFile ./firmware.bin --outputFile ./signedResult.der
```

4.3 公钥导出命令

导出指定密钥对的公钥。

4.3.1 命令

`kms`

4.3.2 子命令

`exportPubKey`

4.3.3 参数

参数	类型	是否必填	说明
`--keyId`	string	是	指定要导出的密钥 ID。
`--outputFormat`	string	是	指定导出密钥的输出格式，可选 `PEM` 或 `DER`，默认格式为 `DER`。
`--outputFile`	string	是	指定导出密钥的输出文件，默认为标准输出（stdout）。
`--resourcePackageId`	string	否	指定产品的资源包 ID。当所使用的访问密钥属于个人账户时，此参数为必填项。

提示

当输出至标准输出（stdout）时，**.der** 格式的签名结果以十六进制字符串形式表示。

4.3.4 示例

```
snb kms exportPubKey --keyId <keyID> --outputFormat PEM --
outputFile pubKey.pem
```

4.4 证书签名命令（例如 Matter DAC）

使用指定的证书模板对证书签名请求（CSR）文件进行签名。

4.4.1 命令

`pki`

4.4.2 子命令

`sign`

4.4.3 参数

参数	类型	是否必填	说明
`--inputFormat`	string	是	指定输入文件的格式（`PEM` 或 `DER`），默认格式为 `PEM`。
`--certTemplateId`	string	是	指定证书（例如 Matter DAC）的模板 ID。
`--outputFormat`	string	是	指定输出文件的格式（`PEM` 或 `DER`），默认格式为 `DER`。
`--csrFile`	string	否	指定 CSR 文件的路径。
`--csrContent`	string	否	提供 CSR 文件的内容。

参数	类型	是否必填	说明
`--outputFile`	string	否	指定输出文件的文件名，默认为标准输出（stdout）。

> **提示**
>
> - 必须在 `csrFile` 或 `csrContent` 中提供至少一个参数。
> - 当输出至标准输出（stdout）时，**.der** 格式的签名结果以十六进制字符串形式表示。

4.4.4 示例

```
snb pki sign --csrFile ./csr.der --inputFormat DER --
certTemplateId 1F118749-EEB1-11EF-A430-B8CEF6C25058
```

4.5 固件推送命令

推送签名后的固件文件至 OBIS。

4.5.1 命令

`obis`

4.5.2 子命令

`push`

4.5.3 参数

参数	类型	是否必填	说明
`--file`	string	是	指定要推送至 OBIS 的文件。
`--fileType`	string	是	指定推送文件的类型：`Application`（默认），`Bootloader` 或 `Others`。
`--resourcePackageId`	string	是	指定 OBIS 中目标产品下的资源包 ID。
`--env`	string	是	指定推送的目标环境：`TEST` 或 `PROD`。
`--startAddress`	string	否	指定固件的烧录起始地址，格式为以 `0x` 开头的十六进制数。
`--version`	string	否	设置固件文件版本，最大长度为 16 个字符。允许使用的字符包括：文本，数字 (0-9)，字母 (a-z, A-Z)，点 (.)，下划线 (_) 和连字符 (-)。

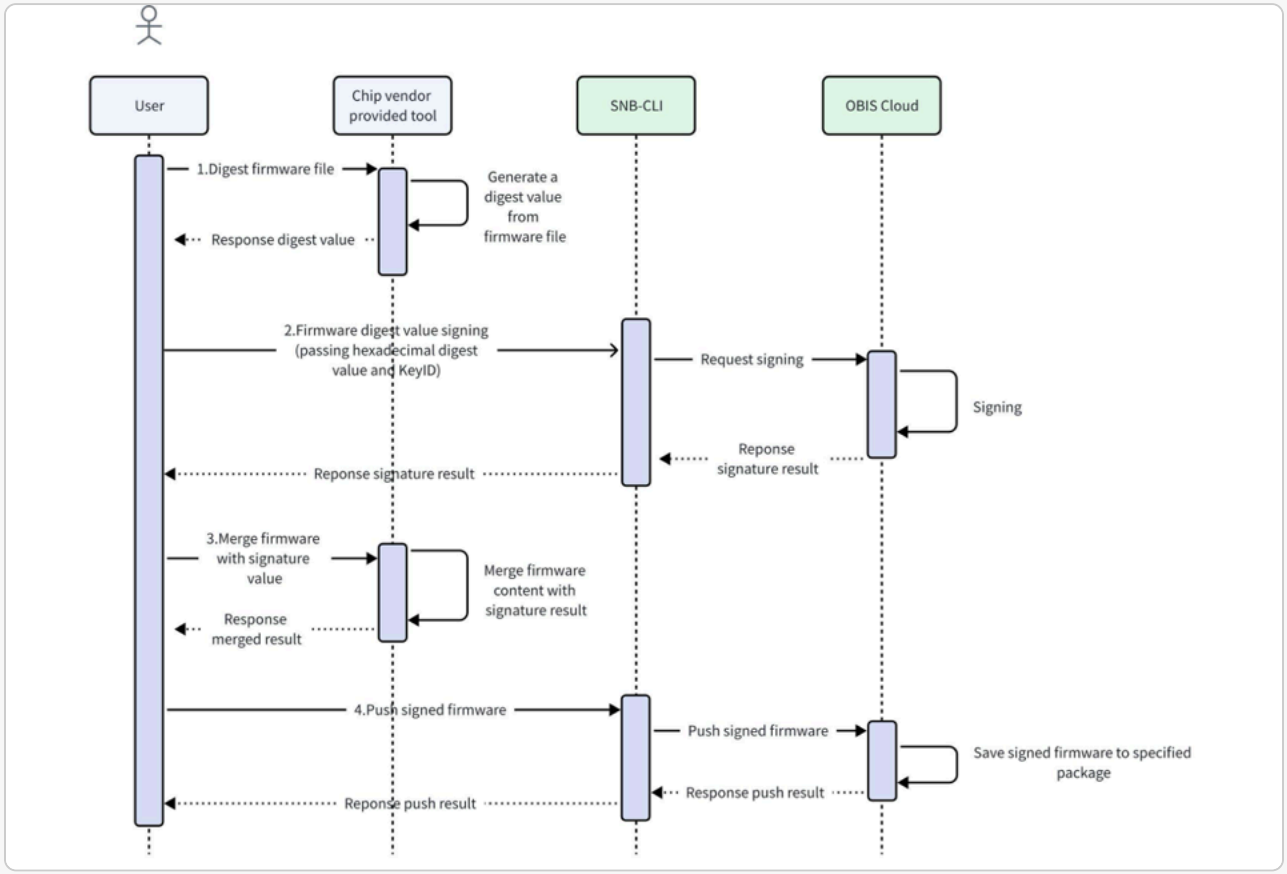
提示

- 当推送文件的格式为 **.bin** 时，参数 `startAddress` 为必填项。
- 参数 `version`，`env` 和 `file` 的组合必须是唯一的，不允许重复。

4.5.4 示例

```
snb obis push --file ./lighting.s37 --fileType Application --
resourcePackageId <id> --env TEST --version 1.0.0
```

5. 固件签名和推送示例



版本：v3.2.2

OnBoard™ IoT Security Programming CLI - v3.0.5

1. 修订记录

版本	更新时间	备注
v1.0.0	2025/05/12	初始版本
v3.0.0	2026/01/22	增加了以下九个命令： • 登出 Edge Service • 查询登录类型 • 查询批次列表 • 绑定批次 • 推送烧录事件 • 解压烧录脚本 • 查看日志配置 • 开启日志收集 • 关闭日志收集 删除了以下三个命令： • 查询 Identity U-Safe 信息 • 查询工具版本 • 下载工具更新包 修改了以下两个命令： • 登录 Edge Service 命令新增了 `--stationId` 参数 • 查询批次信息命令的输出（output）中新增了 `scriptUrl` 字段
v3.0.1	2026/03/13	登录 Edge Service 命令中的 `--stationId` 参数由必填改为可选
v3.0.2	2026/03/17	在查询烧录脚本命令章节新增烧录脚本下载说明，并更新相应的命令示例
v3.0.3	2026/06/02	• 增加了下载大型固件命令 • 登录 Edge Service 命令新增了 `--stationVersion` 参数
v3.0.4	2026/09/19	• 新增 PPCLI（Programming Process Control CLI）及 Telink 专用烧录接口，并补充 UI（Programming Station 客户端）调用说明； • 补充从登录、批次准备、烧录脚本准备到设备烧录的完整调用流程； • 补充烧录过程中的进程启动、输出读取、结果判定及异常处理说明； • 更新相关命令参数及烧录流程说明。
v3.0.5	2026/09/22	新增“执行批次 APDU 操作”命令，包括参数、调用示例、HTTP 映射、返回结果及错误码说明。

2. 概述

OnBoard™ IoT Security (OBIS) Programming CLI 是专为 OBIS 打造的命令行工具，专门用于设备烧录。它功能全面，你可以通过它登录 Edge Service，查询 Identity U-Safe 列表，查询和推送批次数据，推送烧录结果，以及查询和下载烧录脚本。

本文将介绍 OBIS Programming CLI 的常用命令，并提供各命令的参数说明与使用示例，帮助你高效完成设备烧录及相关操作。本文同时介绍 UI（Programming Station 客户端）通过 PPCLI（Programming Process Control CLI）或专用烧录接口启动设备烧录的接入方式，以及从登录、批次准备、烧录脚本准备到设备烧录及结果处理的完整调用流程。

> **备注**
>
> 更多信息请参考：
>
> - [烧录脚本自定义修改指南]
> - 烧录接口调用时序图

3. OBIS Programming CLI 语法

`obiscli [命令] [子命令] [参数]`

> **信息**
>
> 命令行参数中包含空格的字符串必须使用双引号 `"` 括起来，否则命令解析可能失败。
>
> - 正确示例：`--chipVendor="Test Labs"`
> - 错误示例：`--chipVendor=Test Labs`

4. OBIS Programming CLI 命令

4.1 查询工具帮助

用于查询命令行工具的使用说明及可用参数信息。

4.1.1 参数

`--help`

4.1.2 示例

```
obiscli.exe --help
positional arguments:
                OBIS CLI Tool Command
    login       Login to OBIS Local Service.
    logout      logout to OBIS Local Service.
    usafe       OBIS usafe commands.
    batch       OBIS batch commands.
    script      OBIS batch commands.
    obisexe     OBIS commands.
    logCollect  OBIS batch commands.
    addPath     add sys Path.

options:
  -h, --help    show this help message and exit
```

4.2 登录 Edge Service

用于登录 Edge Service。

4.2.1 命令

`login`

4.2.2 参数

参数	类型	是否必填	说明
`--ip`	string	是	Programming Station 客户端 IP，默认值：127.0.0.1
`--port`	string	是	Programming Station 客户端端口，默认值：32333
`--stationId`	string	否	烧录站 ID
`--stationVersion`	string	否	Programming Station 客户端版本
`--factorytype`	string	否	工厂类型，默认值：本地
`--password`	string	否	登录密码，仅在启用密码登录时需要填写
`--usafesn`	string	否	U-Safe Serial Number（序列号）

4.2.3 示例

```
# 无密码登录:
obiscli login --ip=192.168.40.100 --port=32333 --stationId 81045379
# u-safe 登录:
obiscli login --ip=192.168.40.100 --port=32333 --usafesn=00003F00AB013668 --stationId 81045379
# 密码登录:
obiscli login --ip=192.168.40.100 --port=32333 --password=12345678 --stationId 81045379
或
obiscli login --ip=192.168.40.100 --port=32333 --password=12345678
输出:
{
    "stationid": "94221070",
    "token": "b69a9fe7-9df4-4bdc-819e-9b84a7bc0b23",
    "tokenexpire": "unlimited"
}
```

4.3 登出 Edge Service

用于登出 Edge Service。

4.3.1 命令

`logout`

4.3.2 示例

```
# 登出
obiscli logout
```

4.4 查询登录类型

用于查询登录 Edge Service 的类型。

4.4.1 命令

`login`

4.4.2 子命令

`queryType`

4.4.3 参数

参数	类型	是否必填	说明
`--ip`	string	是	Programming Station 客户端 IP，默认值：127.0.0.1
`--port`	string	是	Programming Station 客户端端口，默认值：32333
`--stationId`	string	是	烧录站 ID

4.4.4 示例

```
obiscli login queryType --ip=192.168.40.205 --port=32333 --stationId=97123586

输出：（1 无密码、 2 密码登录、3 U-safe登录）
{
    "loginType": 3
}
```

4.5 查询登录选项

用于查询登录 Edge Service 的具体选项。

4.5.1 命令

`login`

4.5.2 子命令

`queryoptions`

4.5.3 参数

参数	类型	是否必填	说明
`--ip`	string	是	Programming Station 客户端 IP，默认值：127.0.0.1
`--port`	string	是	Programming Station 客户端端口，默认值：32333
`--usafesn`	string	是	U-Safe Serial Number（序列号）
`--factoryType`	string	否	工厂类型，默认值：本地
`--stationId`	string	否	烧录站 ID

4.5.4 示例

```
obiscli login queryoptions --factoryType=local --ip=192.168.40.100 --port=32333 --usafesn=00003F00AB013668
```

4.6 查询 Identity U-Safe 列表

用于列出已插入本地设备的 Identity U-Safe。

4.6.1 命令

`usafe`

4.6.2 子命令

`list`

4.6.3 示例

```
obiscli usafe list
输出:
{
    "usafelist": [
        {
            "usafesn": "0001BC00B86DA2BB"
        }
    ]
}
```

4.7 查询批次信息

用于查询待烧录设备对应批次的信息。

4.7.1 命令

`batch`

4.7.2 子命令

`query`

4.7.3 示例

```
obiscli batch query

输出:
{
    "batches": [
        {
            "batchId": "7346273795440640",
            "batchName": "UI-jason-0925",
            "chips": [
                {
                    "chipPart": "QPG6200",
                    "chipVendor": "Qorvo",
                    "objectid": "4bb81efb504248fe813c83a4ed4602ac"
                }
            ],
            "env": "test",
            "extData": "VID : 0xFFF2  PID : 0x8001",
```

4.8 查询批次列表

用于查询工厂订单号对应的批次列表。

4.8.1 命令

`batch`

4.8.2 子命令

`queryList`

4.8.3 参数

参数	类型	是否必填	说明
`--orderNumber`	string	是	工厂订单号

4.8.4 示例

```
obiscli batch query

输出:
obiscli batch queryList --orderNumber "LDS-19990101-1030BTC"

输出:
[
    {
        "batchId": "8725582734950400",
        "batchName": "Test-Order-5"
    }
]
```

4.9 绑定批次

用于绑定批次到烧录站。

4.9.1 命令

`batch`

4.9.2 子命令

`assign`

4.9.3 参数

参数	类型	是否必填	说明
`--batchId`	string	是	批次 ID

4.9.4 示例

```
obiscli batch assign --batchId 7176076344164352

输出:
```

4.10 查询批次完整数据

用于查询待烧录到设备的完整数据。该命令通常由 PPCLI 或厂商烧录脚本调用，UI 无需直接调用。

4.10.1 命令

`batch`

4.10.2 子命令

`getfulldata`

4.10.3 示例

```
obiscli batch getfulldata
```

4.11 查询批次固定数据

用于查询待烧录到设备的固定数据。该命令通常由 PPCLI 或厂商烧录脚本调用，UI 无需直接调用。

4.11.1 命令

`batch`

4.11.2 子命令

`getFixData`

4.11.3 示例

```
obiscli batch getFixData
```

4.12 查询批次变化的数据

用于查询待烧录到设备中的变化数据。该命令通常由 PPCLI 或厂商烧录脚本调用，UI 无需直接调用。

4.12.1 命令

`batch`

4.12.2 子命令

`getChangingData`

4.12.3 示例

```
obiscli batch getChangingData

输出:
{
    "batchId": "7115488079380480",
    "batchName": "V1.8.5 Dual Button 0906 MP",
    "bundles": [
        {
            "type": "dataBundle",
            "section": "Matter configuration",
            "items": [
```

4.13 推送批次数据

用于推送需要上传到 Edge Service 的数据。该命令通常由 PPCLI 或厂商烧录脚本调用，UI 无需直接调用。

4.13.1 命令

`batch`

4.13.2 子命令

`putdata`

4.13.3 参数

参数	类型	是否必填	说明
`--batchId`	string	是	批次 ID
`--chipVendor`	string	是	芯片供应商名称，例如：Silicon Labs
`--chipPart`	string	是	芯片型号
`--data`	json	是	推送数据
`--uuid`	string	是	通用唯一识别码

4.13.4 示例

```
obiscli batch putdata --uuid=24a1c3da-459a-429b-a717-29989f926d43 --batchId=4808741360762880 --chipVendor="Silicon Labs" --chipPart=EFR
[{\"items\": [{\"key\": \"dacTemplate\", \"response\": [{\"csr\":
\"MIHmMIGNAgEAMCsxKTAnBgNVBAMMIERBQyA1SFNmblpxVSBNdmlkOjE0NDEgTXBpZDowMTIzMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEXeJ5k1UtgF7YJDRm37XvHu47m
FNF1JjcucbEgaIR+KAAMAoGCCqGSM49BAMCA0gAMEUCIQDeR2rM9FOdCg1GTJyAOpQR2FtARVwIw3nEjXv9t7mg6wIgBcvUbH1G8NAlB/I7rOY48b/Fgnf5UkEOMx/7eekcU0k=
```

4.14 执行批次 APDU 操作

用于在需要 APDU（Application Protocol Data Unit，应用协议数据单元）交互的烧录流程中，通过平台服务执行 APDU 操作。该命令通常由 PPCLI 或厂商烧录脚本调用，UI 无需直接调用。

4.14.1 命令

`batch`

4.14.2 子命令

`executeApdu`

4.14.3 参数

参数	类型	是否必填	说明
`--batchId`	string	是	当前烧录批次 ID，应与本次设备数据所属批次一致。

参数	类型	是否必填	说明
`--uuid`	string	是	本次设备数据或烧录记录的 `uuid`，应使用当前烧录流程中获取的标识，不能使用烧录器 `serialno` 替代。
`--data`	string	否	JSON 对象格式的额外请求字段，解析后合并到请求体顶层。例如 `programmingTraceId` 以及服务端协议要求的命令响应数据。具体字段、类型及各阶段是否需要传入，以实际服务端协议为准。

提示

- `--batchId` 和 `--uuid` 在当前参数解析器中未设置为必填参数，但命令不会从当前绑定批次或数据缓存中自动补全。调用时应传入有效的 `batchId` 和 `uuid`；省略时，请求中的对应值为 `null`，最终业务校验由服务端完成。
- 调用前应已成功登录，并获取当前批次及本次设备数据的相关标识。服务地址及会话 token 由 OBIS Programming CLI 从本地配置读取，无需在本命令中重复传入登录参数。
- `--data` 应为有效的 JSON 对象，不能传入文件路径、普通字符串或数组。当前实现将其中的字段直接合并到请求体顶层，不会额外增加 `data` 层；同名字段会覆盖原有值，因此不应在 `--data` 中重复传入 `batchId` 或 `uuid`。
- 本命令不支持 `--chipVendor`、`--chipPart` 或 `--serialno` 参数。
- 是否需要多次调用 `executeApdu`、如何提交上一轮命令响应以及如何判断 APDU 流程结束，由具体服务端协议确定。

4.14.4 示例

不包含额外请求字段时：

```
obiscli batch executeApdu --batchId="<BATCH_ID>" --uuid="<UUID>"
```

需要传入 `programmingTraceId` 时，其他业务字段按服务端协议补充：

```
obiscli batch executeApdu --batchId="<BATCH_ID>" --uuid="<UUID>" --
data="{\"programmingTraceId\":\"<PROGRAMMING_TRACE_ID>\"}"
```

以上两个示例展示不同的参数形式，不表示必须连续执行两次。`--data` 的具体使用方式及其中包含的业务字段，以实际服务端协议为准。

上述 `--data` 中的 `\"` 转义适用于 Windows `cmd.exe` 示例。应用程序通过进程 API 调用时，应直接在参数数组中传入 JSON 字符串，无需额外添加供 Shell 使用的引号或转义。

4.14.5 HTTP 映射

OBIS Programming CLI 将 `executeApdu` 请求发送至以下接口：

项目	说明
Method	`POST`
Path	`/api/iot-factory-edge-service/executeApdu`
Content-Type	`application/json`
Session	`x-snbps-session` 使用当前登录 token
Request ID	`x-snbps-request-id` 由 OBIS Programming CLI 自动生成

项目	说明
Timestamp	`x-snbps-timestamp` 由 OBIS Programming CLI 自动生成

本地模式使用已保存的 IP 和 port；云端模式使用已保存的 host，并在配置存在时自动加入 `factoryId`。

例如，包含 `programmingTraceId` 的命令对应的请求体如下：

```
{
  "uuid": "<UUID>",
  "batchId": "<BATCH_ID>",
  "programmingTraceId": "<PROGRAMMING_TRACE_ID>"
}
```

4.14.6 返回结果与退出码

OBIS Programming CLI 解析 HTTP 响应后，当 `respCode` 为字符串 `"8000"` 且 `respData` 不为 `null` 时，将 `respData` 本身格式化为 JSON 输出，并以退出码 `0` 结束。

CLI 不会同时输出外层的 `respCode` 和 `respMsg`。`respData` 的具体业务结构由服务端定义，当前 CLI 未规定固定字段。

退出码	说明
`0`	本次接口调用成功。调用脚本仍需根据业务协议处理返回数据，不能仅据此判断整个设备烧录流程已经完成
`1`	`--data` 不是有效的 JSON 文本
`45`	HTTP 客户端调用失败，CLI 输出 `executeApdu failed: Network error`
`46`	成功响应中的 `respData` 缺失或为 `null`。当前实现不会仅因空对象 `{}` 或空数组 `[]` 返回该错误
服务端 `respCode` 转换后的整数值	当 `respCode` 不为 `"8000"` 时，CLI 输出 `respMsg`，并以对应的业务错误码退出
`200`	出现未捕获异常，例如响应无法解析，或业务错误码无法转换为整数

APDU 步骤完成后，PPCLI 或厂商烧录脚本继续按照原烧录流程执行后续操作，并通过 `batch putresult` 上报最终烧录结果。`executeApdu` 不替代 `batch putresult`。

4.15 推送烧录结果

用于推送烧录结果到 Edge Service。该命令通常由 PPCLI 或厂商烧录脚本调用，UI 无需直接调用。

4.15.1 命令

`batch`

4.15.2 子命令

`putresult`

4.15.3 参数

参数	类型	是否必填	说明
`--batchId`	string	是	批次 ID

参数	类型	是否必填	说明
`--changeKeys`	string	是	如果不使用服务数据，请输入 key。支持的 key 请参考可查询键列表 (queryKeys V2)
`--result`	string	是	烧录结果
`--uuid`	string	是	通用唯一识别码

4.15.4 示例

```
obiscli batch putresult --batchId=4808741360762880 --uuid=24a1c3da-459a-429b-a717-29989f926d43 --result=success --changeKeys=''
```

4.16 推送烧录事件

用于推送烧录事件到 Edge Service。

4.16.1 命令

`batch`

4.16.2 子命令

`uploadEvent`

4.16.3 参数

参数	类型	是否必填	说明
`--batchId`	string	是	批次 ID
`--uuid`	string	是	设备唯一标识符
`--event`	string	否	烧录事件

4.16.4 示例

```
obiscli batch uploadEvent --batchId=4808741360762880 --uuid=24a1c3da-459a-429b-a717-29989f926d43 --event "{\"uuid\":\"24a1c3da-459a-429b-a717-29989f926d43\", \"events\": [{\"eventName\": \"Generate Trace ID\", \"timestamp\": \"2026-01-09T09:46:02.672Z\", \"source\": \"script\", \"priority\": \"1\", \"status\": \"success\", \"content\": \"\"}, {\"eventName\": \"Fetch Batch Dynamic Data\", \"timestamp\": \"2026-01-09T09:46:04.538Z\", \"source\": \"script\", \"priority\": \"1\", \"status\": \"success\", \"content\": \"\"}, {\"eventName\": \"Fetch Device ID\", \"timestamp\": \"2026-01-09T09:46:07.340Z\", \"source\": \"script\", \"priority\": \"1\", \"status\": \"success\", \"content\": \"\"}, {\"eventName\": \"Fetch CSR\", \"timestamp\": \"2026-01-09T09:46:27.433Z\", \"source\": \"script\", \"priority\": \"1\", \"status\": \"success\", \"content\": \"\"}, {\"eventName\": \"Program Factory Data\", \"timestamp\": \"2026-01-09T09:46:33.631Z\", \"source\": \"script\", \"priority\": \"1\", \"status\": \"success\", \"content\": \"\"}, {\"eventName\": \"Report Programming Result\", \"timestamp\": \"2026-01-09T09:46:35.726Z\", \"source\": \"script\", \"priority\": \"1\", \"status\": \"failure\", \"content\": \"['[ERROR][ppcli]1200000486:Command failed with exception: 2\\n result.exit_code:2 retry_times:1.']\"}]}"
```

4.17 查询烧录脚本

> **提示**
>
> 建议直接使用查询批次信息命令（`batch query`）获取待烧录设备对应的批次信息，从中获取脚本下载链接，然后执行下载烧录脚本命令（`script download`）进行下载。

用于检查待烧录批次是否存在对应的烧录脚本。

4.17.1 命令

`script`

4.17.2 子命令

`query`

4.17.3 参数

参数	类型	是否必填	说明
`--batchId`	string	是	批次 ID
`--chipPart`	string	否	芯片型号
`--chipVendor`	string	否	芯片供应商名称，例如：Silicon Labs
`--productId`	string	否	产品 ID

4.17.4 示例

> **提示**
>
> 当 `scriptExist` 为 `false` 时，需要执行下载烧录脚本命令（`script download`）下载脚本；若为 `true`，则无需再次下载。

```
obiscli script query --batchId=4808741360762880 --chipVendor="Silicon Labs" --chipPart=EFR32MG24B210F1536IM48
输出：
{
    "scriptExist": bool,
    "path": ...
}
```

4.18 下载烧录脚本

用于下载烧录脚本。

4.18.1 命令

`script`

4.18.2 子命令

`download`

4.18.3 参数

参数	类型	是否必填	说明
`--url`	string	是	烧录脚本的下载链接
`--extractFolder`	string	否	烧录脚本下载后的解压路径
`--savePath`	string	否	烧录脚本下载后的保存路径

4.18.4 示例

```
obiscli script download --url="http://192.168.40.171:8005/api/iot-factory-edge-service/
download?fileId=1866310684527935298&fileName=0d232bf966e516f4bf2a75e1880e59ee.zip" --
savePath="C:\\Users\\Administrator\\AppData\\Roaming\\SNB\\chipVendorCmd" --
extractFolder="C:\\Users\\Administrator\\AppData\\Roaming\\SNB\\chipVendorCmd"
```

4.19 解压烧录脚本

用于解压烧录脚本。

4.19.1 命令

`script`

4.19.2 子命令

`extract`

4.19.3 参数

参数	类型	是否必填	说明
`--extractFolder`	string	否	文件的解压路径
`--zipFile`	string	否	压缩文件的路径，文件格式必须是 **.zip**

4.19.4 示例

```
obiscli script extract --zipFile="C:\\Users\\Administrator\\AppData\\Roaming\\SNB\\chipVendorCmd\\Silicon Labs.zip"
--extractFolder="C:\\Users\\Administrator\\AppData\\Roaming\\SNB\\chipVendorCmd"
```

4.20 日志配置

用于配置日志收集的相关参数。

4.20.1 命令

`logCollect`

4.20.2 子命令

`config`

4.20.3 参数

参数	类型	是否必填	说明
`--logLevel`	string	否	日志级别控制
`--collectLogPath`	string	否	日志文件的存放目录，默认值：**C:\Users\<User Name>\AppData\SNB**
`--collectLogSuffix`	string	否	日志文件的后缀，默认值：`.log`
`--mqttIp`	string	否	MQTT Broker（服务器）的 IP
`--mqttPassword`	string	否	连接 MQTT Broker 用户对应的密码
`--mqttPort`	string	否	MQTT Broker 的端口号
`--mqttUser`	string	否	连接 MQTT Broker 的用户名

4.20.4 示例

```
obiscli logCollect config --logLevel=INFO
```

4.21 查看日志配置

用于查看日志收集的相关参数。

4.21.1 命令

`logCollect`

4.21.2 子命令

`showConfig`

4.21.3 示例

```
obiscli logCollect showConfig
输出：
{
  "DEFAULT": {
    "log_level": "INFO",
    "log_file_path": "C:\\Users\\Administrator\\AppData\\Roaming\\iot-log-collecter\\collecter.log",
    "log_rotation": "2 hour",
    "log_retention": "1 days",
    "send_times_save_status": "1",
    "collect_log_path": "C:\\Users\\Administrator\\AppData\\Roaming\\SNB",
    "collect_log_suffix": "[\".log\"]",
    "send_log_mode": "1",
    "source_type": "\"PS\" # ps:programming station  es:edge service"
  },
  "MQTT": {
    "broker": "192.168.40.144",
    "port": "32222",
    "user": "snowball",
    "password": "vX4hR9tV7iE8pO9d",
    "connect_crt": "",
    "connect_key": "",
```

4.22 开启日志收集

用于开始日志收集。

4.22.1 命令

`logCollect`

4.22.2 子命令

`start`

4.22.3 示例

```
obiscli logCollect start
输出:
iot-log-collect.exe run successfully.
```

4.23 关闭日志收集

用于停止日志收集。

4.23.1 命令

`logCollect`

4.23.2 子命令

`stop`

4.23.3 示例

```
obiscli logCollect stop
输出:
iot-log-collecter.exe stop successfully.
```

4.24 下载大型固件

用于下载大型固件文件。

4.24.1 命令

`downloadFw`

4.24.2 参数

参数	类型	是否必填	说明
`--url`	string	是	固件文件的下载链接
`--savePath`	string	否	固件文件下载后的保存路径

4.24.3 示例

```
obiscli downloadFw --url <URL> --savePath <Path>
```

5. PPCLI 烧录接口

5.1 概述

PPCLI（Programming Process Control CLI）用于控制设备烧录流程。UI（Programming Station 客户端）接入设备烧录时，先通过 OBIS Programming CLI 完成登录、批次绑定及烧录脚本准备，再通过烧录脚本包提供的 PPCLI（Programming Process Control CLI）入口查询烧录器并启动设备烧录。

烧录过程中所需的批次数据获取、证书请求、按需执行的 APDU 交互及烧录结果上报，由 PPCLI 或厂商烧录脚本通过 OBIS Programming CLI 完成，UI 通常无需直接调用相关接口。

5.2 Python PPCLI

适用于烧录脚本包中包含 `ppcli.py` 及对应 Python 运行环境的场景。`ppcli.py` 为 PPCLI 入口，`process.py` 为其调用的厂商烧录实现。UI 应通过 `ppcli.py` 启动烧录，不应直接执行 `process.py`；具体入口以烧录脚本包说明为准。

5.2.1 使用前提

调用 Python PPCLI 前，应满足以下条件：

- 已登录 Edge Service，并完成当前批次绑定；
- 已下载并准备与当前批次、芯片型号匹配的烧录脚本包，且相关依赖文件完整；
- 将子进程的工作目录（`cwd`）设置为烧录脚本包的实际入口目录，即包含 `ppcli.py` 的目录，而非 ZIP 下载目录或 UI 安装目录；后续相对路径均以该目录为基准；
- 当前进程及其子进程能够通过 `PATH` 找到正确版本的 `obiscli.exe`，并与前序 OBIS Programming CLI 使用相同的 Windows 用户及配置环境，以共享登录会话和当前批次绑定信息；仅 UI 使用 OBIS CLI 的绝对路径，不能保证 PPCLI 内部调用的 `obiscli.exe` 能够被正确找到；
- 烧录器驱动已正确安装并连接，目标设备已上电；
- `--type`、`--speed`、芯片供应商及芯片型号等参数应与实际烧录脚本包保持一致。

5.2.2 查询烧录器

用于查询当前可用的烧录器。

5.2.2.1 命令

`adapter`

5.2.2.2 子命令

`list`

5.2.2.3 参数

参数	类型	是否必填	说明
`--chipVendor`	string	是	交付包识别的芯片供应商名称，通常来自批次 `chips[].chipVendor`；必要时按烧录脚本包约定进行映射。大小写及空格不能自行假设可互换。
`--type`	string	是	烧录器类型，具体取值以烧录脚本包配置为准

5.2.2.4 示例

> **提示**
> `--type=1` 为当前示例使用的值，具体取值以实际烧录脚本包为准。

```
".\python-3.12.9-embed-amd64\python.exe" ".\ppcli.py" adapter list --chipVendor="<CHIP_VENDOR>" --type=1
输出：
{
  "adapterList": [
    {
      "serialno": "<SERIALNO>"
    }
  ]
}
```

查询成功后，从 `adapterList` 中选择目标烧录器，并将其 `serialno` 原样传入 `program` 命令。

`serialno` 按字符串处理。如果返回值包含前导零，例如 `0008`，应保留完整字符串。如果 `adapterList` 为空，则不应启动设备烧录；返回结果缺少必要字段或无法按约定格式解析时，应按烧录器查询失败处理。不能仅根据进程退出码为 `0` 判断存在可用烧录器。

5.2.3 启动烧录

用于通过指定烧录器启动设备烧录。

5.2.3.1 命令

`program`

5.2.3.2 参数

参数	是否必填	类型	说明
`--serialno`	string	是	来自本烧录脚本包 `adapter list` 返回的烧录器/通道选择标识，应保留原始字符串；不能使用 `batchId`、`stationId`、芯片 UID 或 `uuid` 替代
`--type`	string	是	按烧录脚本包约定填写，并与 `adapter list` 使用的类型保持一致；示例中的 `1` 具体含义由烧录脚本包定义
`--chipVendor`	string	否	当前批次所选芯片的供应商；UI 调用时应传入；如烧录脚本包使用厂商别名，应根据配置完成映射
`--chipPart`	string	否	当前批次所选芯片型号，通常来自 `chips[].chipPart`；UI 调用时应传入
`--speed`	string	否	烧录通信速率；示例为 `115200`，具体单位、有效范围及缺省行为以烧录脚本包说明为准，不统一解释为所有厂商的 UART 波特率或调试时钟
`--extInfo`	string	否	厂商扩展参数；仅在烧录脚本包有明确约定时传入，无明确约定时省略，不应直接传入任意批次 JSON

5.2.3.3 示例

当前核对的 Python PPCLI 中，`--serialno` 和 `--type` 为必填参数；其他上述参数在语法上可省略。但厂商实现可能依赖芯片供应商和芯片型号，因此 UI 调用时应传入 `--chipVendor` 和 `--chipPart`。program 无需传入 `batchId`，当前批次通过前序 `obiscli batch assign` 命令完成绑定。

```
".\python-3.12.9-embed-amd64\python.exe" ".\ppcli.py" program --chipVendor="<CHIP_VENDOR>" --chipPart="<CHIP_PART>"
--serialno="<SERIALNO>" --speed=115200 --type=1
```

> **提示**
>
> `--chipVendor`、`--chipPart` 和 `--type` 的具体取值应以实际 Python PPCLI 烧录脚本包约定为准，不应直接使用 Telink 专用烧录接口示例中的 `TELINK` 或 `telinkevk`。

如烧录脚本包明确要求传入厂商扩展参数，可增加 `--extInfo="<EXT_INFO>"`。

5.2.4 参数检查

可使用以下命令检查实际 Python PPCLI 包支持的参数：

```
".\python-3.12.9-embed-amd64\python.exe" ".\ppcli.py" adapter list --help
".\python-3.12.9-embed-amd64\python.exe" ".\ppcli.py" program --help
```

> **提示**
>
> `--help` 仅用于查看命令及参数定义，不执行设备烧录。

5.2.5 PPCLI 内部调用

用于说明启动设备烧录后，PPCLI 与 OBIS Programming CLI 之间的调用关系。UI 主要负责准备运行环境、选择烧录器、启动 `program` 并处理输出。启动烧录后，PPCLI 根据当前批次及芯片配置执行具体烧录流程，主要包括：

1. 调用 `obiscli batch getChangingData` 获取当前设备的变化数据及烧录标识；
2. 固定数据未缓存时，调用 `obiscli batch getFixData` 获取并缓存固定数据，再与变化数据合并；
3. 调用 `process.py` 中的厂商烧录逻辑，根据芯片及批次配置完成设备访问、证书及 Factory Data 处理、固件写入和验证等操作；
4. 根据烧录流程需要，通过 `obiscli batch putdata` 与 Edge Service 进行数据交互，该调用可能发生在烧录过程中，例如提交 CSR 并获取 DAC；需要 APDU 交互时，可通过 `obiscli batch executeApdu` 执行相应的 APDU 烧录步骤；
5. 烧录完成后，根据执行结果通过 `obiscli batch putresult` 上报 `success` 或 `failure`。前置数据获取失败、缺少有效 `uuid` 或 `batchId`，或流程异常中断时，可能无法完成结果上报。

UI 无需重复获取设备变化数据、申请 DAC 或上报同一烧录结果，上述操作由 PPCLI 或烧录脚本统一完成。

5.3 Telink 专用烧录接口

部分 Telink 烧录脚本包使用专用可执行程序直接查询烧录器并启动设备烧录。UI 应根据实际交付的烧录脚本包选择对应的烧录入口。

Python PPCLI 与 Telink 专用可执行程序属于两种不同的接入方式，二者选择其一，无需依次执行。如果 Python PPCLI 内部已调用 Telink 烧录工具，UI 仅需调用外层 PPCLI，避免对同一设备重复启动烧录。

5.3.1 查询烧录器

用于通过 Telink 专用烧录程序查询当前可用的烧录器。

5.3.1.1 命令

`adapter`

5.3.1.2 子命令

`list`

5.3.1.3 参数

参数	类型	是否必填	说明
`--chipVendor`	string	是	芯片供应商标识，本示例使用 `TELINK`，不应直接替换为 Python PPCLI 使用的厂商名称
`--type`	string	是	烧录器类型，本示例使用 `telinkevk`，不应与 Python PPCLI 示例中的 `--type=1` 混用

5.3.1.4 示例

```
".\Cmd\_download\_tool.exe" adapter list --chipVendor="TELINK" --type=telinkevk
```

上述路径表示 `Cmd` 目录下 `_download` 子目录中的 `_tool.exe`，不表示名为 `Cmd_download_tool.exe` 的单个文件。 可执行程序路径应以实际交付的烧录脚本包为准。例如，如果实际交付包中的可执行程序位于根目录，可使用：

```
".\Cmd_download_tool.exe" adapter list --chipVendor="TELINK" --type=telinkevk
```

不同烧录脚本包可能使用不同的目录结构，UI 应根据烧录脚本包配置确定可执行程序路径及工作目录，不应硬编码固定路径，也不应在目标文件不存在时自动猜测替代程序。

5.3.2 启动烧录

用于通过指定的 Telink 烧录器启动设备烧录。

5.3.2.1 命令

`program`

5.3.2.2 参数

参数	类型	是否必填	说明
`--serialno`	string	是	目标烧录器的设备 ID，由同一 Telink 专用程序的 `adapter list` 命令查询获得，应保留返回的原始字符串

5.3.2.3 示例

```
".\Cmd\_download\_tool.exe" program --serialno="<SERIALNO>"
```

如果实际交付包中的可执行程序位于根目录，可使用：

```
".\Cmd_download_tool.exe" program --serialno="<SERIALNO>"
```

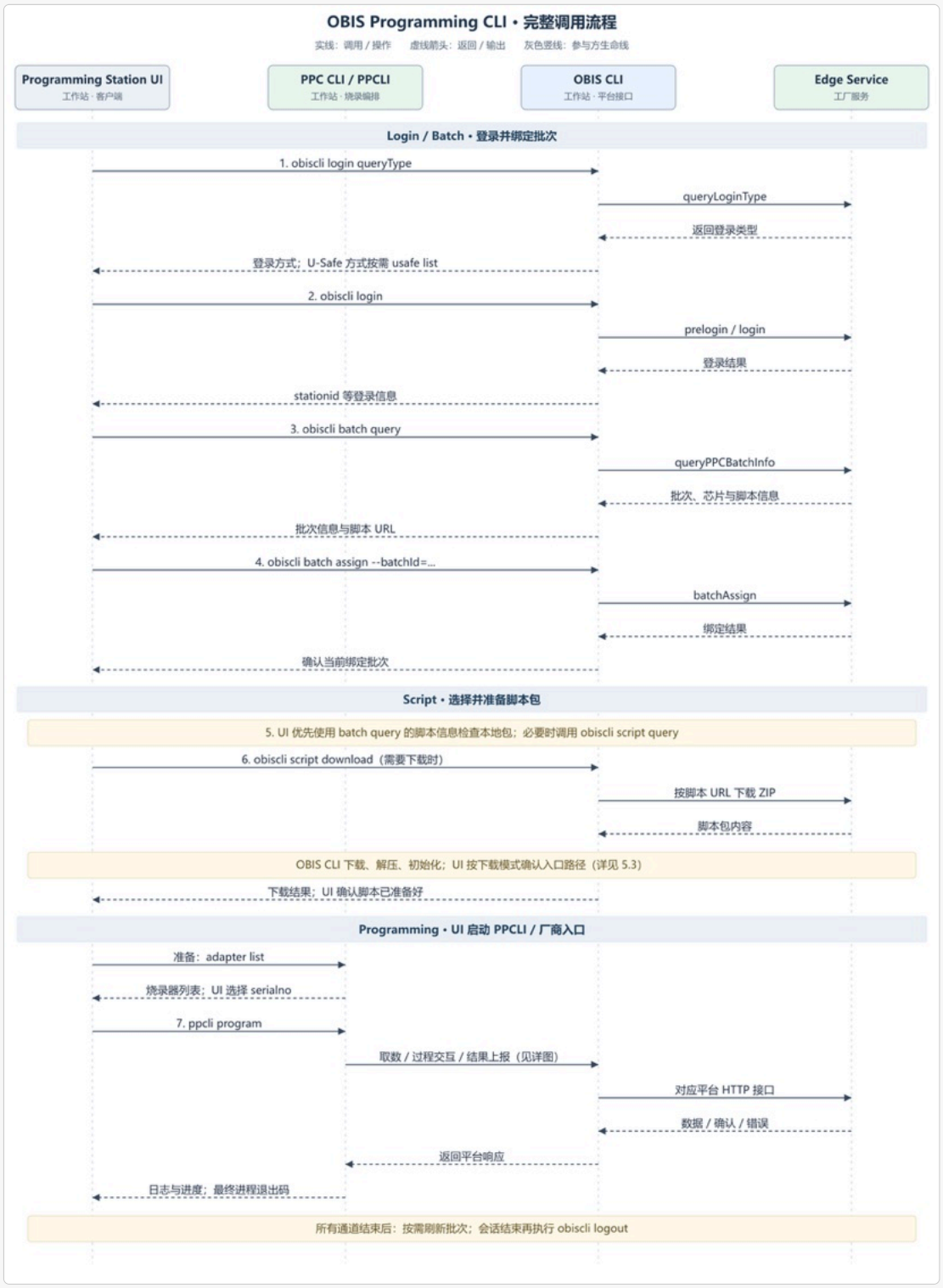
Telink 专用烧录接口的 `program` 命令不使用 Python PPCLI 示例中的 `--chipVendor`、`--chipPart`、`--speed` 和 `--type` 参数，不应将这些参数直接追加到 Telink 专用烧录命令中。其他烧录配置以实际 Telink 烧录脚本包说明为准。

Telink 专用烧录接口的返回结果、进度格式、超时及退出码以对应版本的烧录脚本包为准。

6. 设备烧录执行流程

6.1 总体调用流程

用于说明 UI 从登录 Edge Service、准备批次和烧录脚本，到调用 PPCLI 完成设备烧录的总体流程。



阶段	调用方 → 接收方	命令/处理	后续动作的前提
1	UI → OBIS CLI	`login queryType`；U-Safe 登录方式下，按需执行 `usafe list`	获取登录类型及所需身份信息

阶段	调用方 → 接收方	命令/处理	后续动作的前提
2	UI → OBIS CLI	`login`	登录成功，并保存客户端所需的 `stationid`
3	UI → OBIS CLI	`batch query`；或按订单执行 `batch queryList`	获取完整批次信息，并确定 `batchId`、芯片及烧录脚本；仅包含 ID 的批次列表不能替代完整批次信息
4	UI → OBIS CLI	`batch assign`	批次绑定成功，后续启动的设备烧录使用当前绑定批次
5	UI	优先读取 `batch query` 返回的脚本 URL、名称及版本/摘要；必要时执行 `script query`	确定复用匹配的本地烧录脚本包或重新下载
6	UI → OBIS CLI	`script download`	烧录脚本下载、解压及初始化完成，并定位实际入口目录；本地缓存命中且脚本包已准备完成时可跳过下载
7（准备）	UI → PPCLI / 包入口	`adapter list`	成功查询可用烧录器，并选择目标 `serialno`
7	UI → PPCLI / 包入口	`program`	实时接收烧录输出并等待进程结束，根据执行结果处理烧录成功、失败或结果上报异常
结束	UI → OBIS CLI	按需执行 `batch query` 刷新批次统计；会话结束时执行 `logout`	所有通道的烧录及结果上报完成后再结束当前登录会话

第 5 步优先使用 `batch query` 返回的烧录脚本信息，不要求 UI 固定调用 `script query`；仅在实际流程需要补充查询时调用。批次返回结构随版本可能存在外层封装，应按实际使用的 OBIS Programming CLI 版本解析，不应根据 HTTP 原始响应直接假定 CLI 的 JSON 结构。

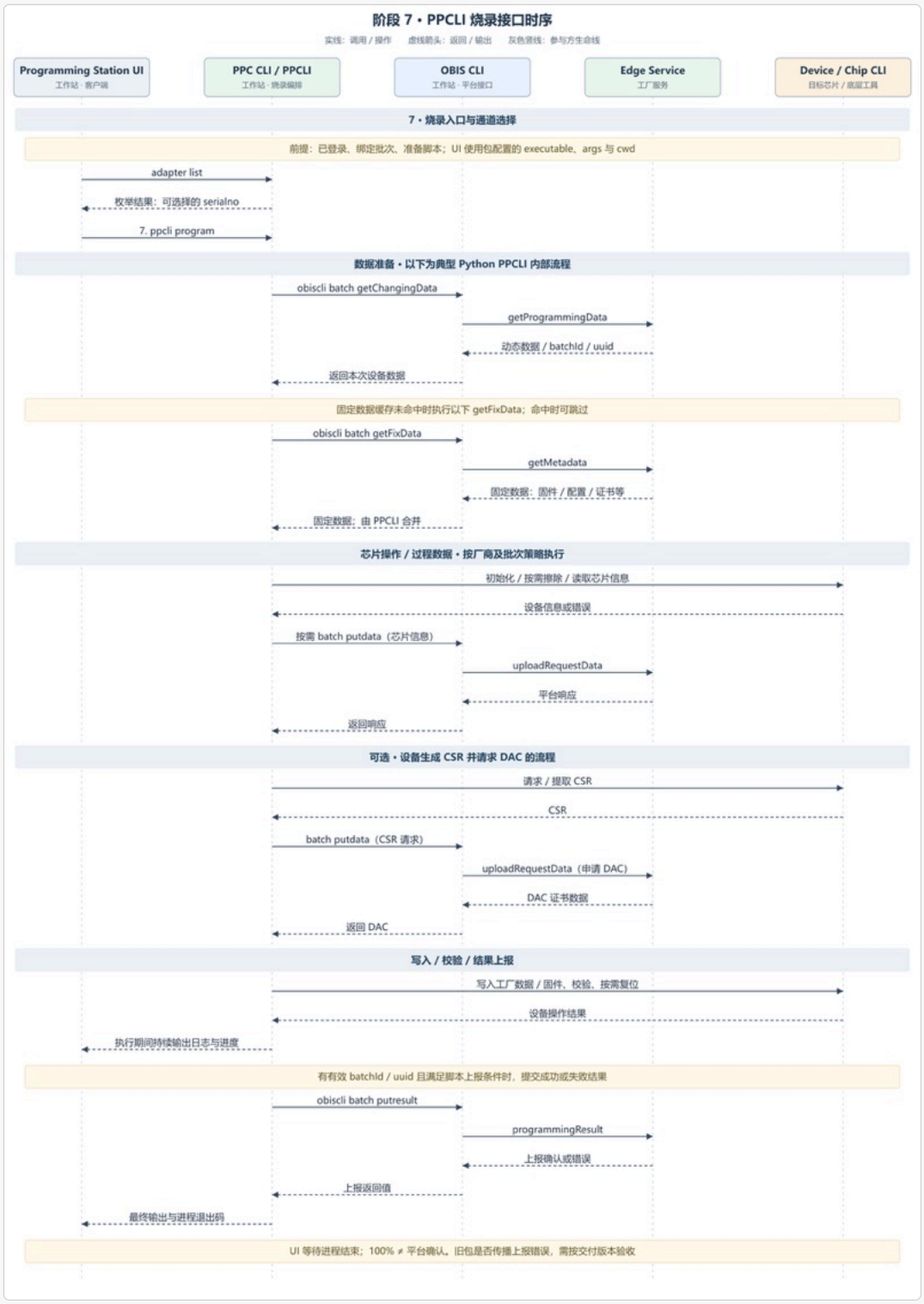
6.2 烧录阶段调用流程

用于说明总体调用流程中阶段 7 的设备烧录过程，包括 UI 启动烧录、烧录数据获取、设备烧录及结果上报。

提示

- `getChangingData` 和 `getFixData` 为 OBIS Programming CLI 命令，分别用于获取设备变化数据和固定数据；对应的 HTTP 接口分别为 `getProgrammingData` 和 `getMetadata`，CLI 命令名与 HTTP 接口名不应混用；
- 固定数据已缓存时，可跳过 `getFixData`；
- 设备擦除、Device ID 获取、CSR/DAC 处理、Factory Data 写入、固件烧录及复位等具体步骤，由芯片及批次配置决定；
- 非 Matter 设备或不采用设备端生成 CSR 的流程，不一定包含 CSR/DAC 处理；
- `putdata` 用于烧录过程中的数据及请求/响应交互，例如 CSR/DAC 请求；对应 HTTP 接口 `uploadRequestData`；
- `putresult` 用于上报最终烧录结果，对应 HTTP 接口 `programmingResult`；`putdata` 与 `putresult` 用途不同，不能互相替代。

图中展示 Python PPCLI 的典型调用流程；Telink 专用烧录接口的内部处理流程以实际交付版本为准。



6.3 完整命令示例

用于说明从登录 Edge Service 到完成设备烧录的基本命令调用顺序。以下示例使用 Python PPCLI，Windows 命令采用 cmd.exe 语法。实际参数应根据现场环境、批次及烧录脚本包配置填写。执行过程中，应确认当前步骤执行成功后再执行下一步。Python PPCLI 与 Telink 专用烧录接口为两种不同的烧录入口，应根据实际烧录脚本包选择其中一种。

```
REM 1. 查询登录方式；U-Safe 登录时按需执行 obiscli usafe list
obiscli login queryType --ip="<EDGE_IP>" --port=32333

REM 2. 根据 queryType 返回的登录方式登录；以下为无密码登录示例
REM 密码登录增加 --password="<PASSWORD>"；U-Safe 登录增加 --usafesn="<USAFE_SN>"
obiscli login --ip="<EDGE_IP>" --port=32333 --stationVersion="<UI_VERSION>"

REM 3. 查询批次，并获取 batchId、chips、scriptUrl 等信息
obiscli batch query

REM 4. 绑定当前批次
obiscli batch assign --batchId="<BATCH_ID>"

REM 5. 从批次查询结果中获取 scriptUrl，并检查本地是否已有匹配且准备完成的烧录脚本包

REM 6. 本地不存在可用烧录脚本包时下载
obiscli script download --url="<SCRIPT_URL>"

REM 根据下载结果定位实际入口目录，并切换工作目录
cd /d "<SCRIPT_ENTRY_DIRECTORY>"

REM 7（准备）. 查询烧录器并选择 SERIALNO
".\python-3.12.9-embed-amd64\python.exe" ".\ppcli.py" adapter list --chipVendor="<CHIP_VENDOR>" --type=1

REM 7. 启动设备烧录
".\python-3.12.9-embed-amd64\python.exe" ".\ppcli.py" program --chipVendor="<CHIP_VENDOR>" --chipPart="<CHIP_PART>"
--serialno="<SERIALNO>" --speed=115200 --type=1

REM program 执行完成后查看返回码；UI 应直接获取子进程返回码，不依赖此 echo
echo %ERRORLEVEL%

REM 按需刷新批次统计
obiscli batch query

REM 所有通道烧录及结果上报完成后登出
obiscli logout
```

使用 Telink 专用烧录接口时，仅替换阶段 7 的烧录器查询和设备烧录命令：

```
REM 7（准备）. 查询 Telink 烧录器并选择 SERIALNO
".\Cmd\_download\_tool.exe" adapter list --chipVendor="TELINK" --type=telinkevk

REM 7. 启动设备烧录
".\Cmd\_download\_tool.exe" program --serialno="<SERIALNO>"
```

登录、批次绑定及烧录脚本准备流程保持不变。Telink 专用烧录程序的可执行文件路径和工作目录应以实际交付的烧录脚本包为准。

6.4 烧录脚本下载与入口目录

用于说明 UI 下载烧录脚本后定位实际烧录入口的处理方式。

仅传入 `--url` 时，`script download` 使用默认缓存目录，并可通过返回结果中的 `scriptExist` 和 `path` 判断脚本是否已存在及其所在位置。

如果同时指定 `--savePath` 和 `--extractFolder`，UI 不应假定命令输出一定包含烧录程序的实际入口路径。应根据指定的解压目录及烧录脚本包结构定位实际入口，并确认入口程序及相关依赖文件存在。

`--savePath` 表示下载文件的保存目录。烧录脚本 ZIP 包内部可能包含一级或多级子目录，因此下载目录不一定等于实际烧录入口目录。

烧录脚本启动前，还应完成对应烧录脚本包要求的初始化及依赖检查。

6.5 UI 进程与结果处理

用于说明 UI 调用 PPCLI 或 Telink 专用烧录程序时的进程启动、输出读取及结果处理要求。

项目	接入要求
进程启动	分别设置 `executable`（可执行文件路径）、`args[]`（命令参数）和 `cwd`（工作目录）。优先使用绝对可执行文件路径，并设置 `shell=false`，直接启动目标进程。`args[]` 中的参数值无需额外添加供 Shell 使用的引号。
参数来源	芯片厂商和型号来自当前选择的批次；`serialno` 来自当前烧录入口的 `adapter list` 返回结果；`type`、`speed` 及入口路径根据烧录脚本包配置确定。
输出读取	异步持续读取 `stdout` 和 `stderr`，直到进程结束且输出读取完成，避免因仅处理单一输出流导致缓冲区阻塞。Python PPCLI 通常使用 UTF-8 编码，旧版 EXE 的输出编码以实际版本为准
枚举输出	按烧录脚本包约定解析 `adapter list` 完整返回结果，并处理空列表、字段缺失、异常输出及非零退出码。对于多行 JSON 输出，应按完整结果解析，不应逐行作为独立 JSON 处理。
烧录输出	烧录输出可能包含日志和事件信息，不应将整个 `stdout` 固定按单一 JSON 解析。当前核对的 Python PPCLI 可输出 `[Progress]:<数值>%`、`[Finish]`. 等进度或完成信息，失败时可能输出 `{"errMsg":"..."}`；其他烧录脚本包的输出格式可能不同。
成功判定	等待烧录进程正常结束，并根据退出码及烧录脚本包约定的最终结果判断执行状态。进度达到 100% 或输出 `[Finish]`. 仅用于表示烧录进度或流程完成，不应据此提前释放烧录通道，也不能单独作为平台结果上报成功的依据。
结果上报	应确认烧录脚本是否将 `putresult` 上报失败反映到最终执行状态；如果未反映，应单独核对平台记录。批次统计变化不能单独作为某一 `uuid` 已成功上报的依据
失败、超时或取消	保留 serialno、烧录脚本版本、退出码、`stdout`、`stderr` 及已获取的 `uuid`。超时或进程终止不得标记为烧录成功。重新烧录前，应确认设备状态及上一次烧录结果，避免重复获取烧录数据或申请证书。
并发处理	一个活动烧录任务对应一个 `serialno`。实际并发能力取决于烧录脚本包、驱动及服务能力。共享同一 OBIS 登录会话时，烧录过程中不应切换当前批次或执行 `logout`，也不应替换正在执行的烧录脚本包。

6.5.1 UI 进程调用示例

以下示例用于说明 Python PPCLI 模式下 UI 启动烧录进程时的参数组织方式，不是新增的 CLI 参数或固定接口格式：

- `executable`：Python 可执行文件路径；
- `args`：传递给 Python PPCLI 的命令及参数；
- `cwd`：烧录脚本的实际入口目录；
- `shell`：示例设置为 `false`，表示直接启动目标进程，不通过 Shell 执行整条命令。

```
{
  "executable": "<入口目录>\\python-3.12.9-embed-amd64\\python.exe",
  "args": [
    "<入口目录>\\ppcli.py",
    "program",
    "--chipVendor=<CHIP_VENDOR>",
    "--chipPart=<CHIP_PART>",
    "--serialno=<SERIALNO>",
    "--speed=115200",
    "--type=1"
```

6.6 接入验证

用于说明 UI 完成烧录接口接入后建议验证的主要场景。

验证项	验证内容
正常流程	验证登录、批次查询与绑定、脚本准备、烧录器查询、单设备烧录、进程退出状态及平台记录确认和登出流程
参数与路径	验证包含空格的目录或芯片供应商名称能够正常处理；`serialno` 的前导零保持不变；实际可执行文件、工作目录及 `type` 映射正确。
设备异常	验证无设备、设备拔出、`serialno` 错误及设备占用等场景下，UI 能够正确结束任务并保留错误信息。
平台异常	验证设备数据获取失败、证书请求失败及烧录结果上报失败等场景能够正确区分，且不会仅根据烧录进度达到 100% 判断烧录成功。
连续/并发烧录	验证脚本缓存命中及重新下载均能正常启动；多通道烧录时 `serialno`、`uuid` 互不混淆；烧录任务执行期间不改变共享批次或退出当前会话。

版本：v3.2.2

OnBoard™ IoT Security 术语库

B

BYOK (Bring Your Own Key)

一种密钥管理模型，其中客户保留对加密密钥生成过程的所有权与控制权，同时将密钥导入或委托给外部平台进行受管使用。

C

CLI (Command-Line Interface)

一种基于文本的接口，用于通过输入命令与软件系统进行交互，通常用于自动化操作、脚本编写，以及与开发和生产工作流程的集成。

CSR (Certificate Signing Request)

一种由设备或安全环境生成的加密请求，其中包含公钥及身份信息，用于向证书颁发机构（CA）申请签发数字证书。

D

Device Identity

在设备初始化或配置过程中分配给设备的加密身份，通常由一组唯一的密钥对及其对应的数字证书表示。

Digital Assets

在设备生产与配置过程中使用的与产品相关的资产，包括固件、加密密钥、证书以及配置文件。

E

EdgeHSM

一种部署在工厂现场的硬件保护型加密执行设备，旨在在外部下发的授权下，本地执行安全敏感操作。

Edge Service

OnBoard™ IoT Security 的 Edge Service，提供工厂本地端的生产管理能力。通常部署在产线的本地服务器上，缓存处理存储批次数据和设备数据，基于 Edge HSM 提供密码学服务。

H

HSM (Hardware Security Module)

一种防篡改硬件设备，用于安全地生成、存储和使用加密密钥。

HSM U-Safe

登录 OnBoard Edge Service 时，用于进行鉴权的 U 盾设备。

I

Identity U-Safe

登录 Programming Station 客户端时，用于进行鉴权的 U 盾设备。

K

KMS (Key Management System)

用于管理加密密钥全生命周期的系统，包括密钥的生成、存储、分发、轮换、吊销和销毁。

M

MES (Manufacturing Execution System)

用于管理、监控和控制制造运营的工厂侧系统，包括生产计划、过程跟踪和质量控制。

O

OBIS

OnBoard™ IoT Security，雪球为物联网（IoT）领域构建的安全服务平台。它是一个安全控制平台（security control plane），用于管理产品数字资产在设备制造生命周期中的使用方式。

OBIS CLI

OnBoard™ IoT Security CLI，通过使用 U-Safe for PPC 的认证与 Local Service 建立通道，并向 Local Service 获取对应的设备数据，供上层应用使用。

OBIS Edge Web Portal

OBIS Edge 端的门户网站，工厂管理员可通过该网站实现工厂本地的可视化管理。

OBIS Programming Station software

一种受控的“最后一公里”软件环境，安装在工厂的烧录站上，用于设备的物理连接、烧录与初始化。

OBIS Web Portal

OnBoard™ IoT Security 的门户网站，用户可以通过该网站进行产品相关的操作和管理，例如产品管理，工厂管理，用户管理，密钥管理和证书管理等。

ODM (Original Design Manufacturer)

为 OEM 提供端到端产品设计与制造服务的制造商。ODM 通常基于自身参考设计或平台开发产品，并根据 OEM 规格进行定制，最终产品以 OEM 品牌出货。

OEM (Original Equipment Manufacturer)

设计、拥有并以自有品牌销售产品的实体。即使部分或全部制造活动委托给外部合作伙伴，OEM 仍对产品规格、质量和法规合规性承担最终责任。

OIDC (OpenID Connect)

基于 OAuth 2.0 协议构建的身份层，用于身份验证和单点登录（SSO）。OBIS 支持基于 OIDC 的企业身份与访问管理系统集成。

P

PKI (Public Key Infrastructure)

用于管理数字证书及公私钥对的一整套技术、策略和流程框架，支持安全的身份验证、加密通信和数字签名。

PPC (Programming Process Control) CLI

Programming Process Control Command Line Interface，用于执行烧录流程控制的命令行工具。

Production Order

生产订单（在不同系统中也称为工单或制造订单）是由 OEM 在其 SAP、ERP 或 MES 平台中创建的指令，用于授权并跟踪特定数量产品的生产过程。

Programming CLI

Programming Command Line Interface， 用于执行设备烧录的命令行工具。

Programming Station client

工厂烧录客户端，专门用在工厂产线，用来完成芯片烧录操作。

S

Secure Provisioning

将设备安全地初始化的端到端流程，包括其身份、加密材料、固件、配置以及实现认证操作和生命周期管理所需的安全设置。

SnowBall

雪球数智科技有限公司，简称雪球。一家在安全设备配置和加密基础设施方面具有深厚专业技术的科技公司。

SSO (Single Sign-On)

一种身份验证机制，允许用户使用同一组凭据访问多个系统。

版本：v3.2.2

OnBoard™ IoT Security 常见问题

本文列举了你在使用 OnBoard™ IoT Security（OBIS）时可能遇到的问题及其解决方案。

注册与登录

如何自注册 OBIS 账号？

当你的组织已启用了自注册功能时，请在 OBIS 登录页面点击**注册**，完成注册后可通过验证码方式登录。

为什么无法自注册账号？

如果提示"你的组织未启用自注册，请联系管理员"，请联系 System Manager 确认你的组织是否已启用自注册功能。

为什么无法通过验证码方式登录 OBIS？

- 如果提示 "邮箱不存在，请检查后重试"，请联系 System Manager 确认是否已根据你的电子邮件为你创建了 OBIS 账号。
- 如果提示 "此账号已被禁用，如有疑问，请联系管理员"，请联系 System Manager 检查你的账号状态，只有状态为"**启用**"的账号才能登录 OBIS。更多信息请参考管理账号。

为什么无法通过 SSO 方式登录 OBIS?

- 如果提示 "你的组织未启用SSO登录"，请联系 System Manager 确认你的组织是否已启用 SSO 登录方式。
- 如果提示 "SSO服务异常，请稍后再试"，请联系 System Manager 检查你组织的 SSO 服务是否异常。
- 如果提示 "你的组织不允许通过SSO自动创建账号，请联系管理员"，请联系 System Manager 确认你的组织是否已启用通过 SSO 自动创建账号的功能。

用户与角色

是否可以给用户分配多个角色，分配后可以更新吗?

- System Manager 可以为用户分配多个 OBIS 角色，并且可以在分配后进行更改。更多信息请参考管理账号。
- Product Owner 和 Product Manager 可以为用户分配多个产品角色，并且可以在分配后进行更改。更多信息请参考管理成员。

可以添加其他租户的用户和用户组到产品团队里吗?

可跨租户添加用户，但无法跨租户添加用户组。被添加的用户必须是状态为**启用**的用户。例如：Product Manager 来自租户 A，可以添加来自租户 B 的用户加入其产品团队。更多信息请参考管理成员。

为什么给产品团队添加用户时提示 "邮箱地址有误，请检查后再试"?

请检查该用户是否已有 OBIS 账号，只有 OBIS 用户才能被邀请加入产品团队。请先联系 System Manager 为该用户创建账号，然后再添加。

产品

为什么产品列表里的某些产品无法点击查看?

产品列表包含你参与的及租户内所有已创建的产品。请确认你是否为该产品团队的成员。如果不是，请联系该产品的 Product Owner 或 Product Manager 邀请你加入该产品团队并为你分配角色。被添加后，你将收到一封包含链接的电子邮件，可通过该链接查看产品。

为什么产品状态显示为草稿?

产品创建完成后，产品的初始状态为**草稿**状态。创建版本后，产品的状态将变更为**已发布**状态。更多信息请参考产品状态说明。

当前 OBIS 支持哪些芯片?

具体列表请参考当前支持的芯片设置。后续将根据需求新增支持的芯片厂商，系列和型号。

在产品配置 Factory Data 时的 "Mandatory" 是什么意思?

如果勾选某个工厂数据 **Mandatory** 列的复选框，成员在创建版本时必须输入该工厂数据的值。

如何查看设备烧录后的二维码？

完成设备烧录后，你可以在产品的设备标签页选择对应的设备，然后在设备页面查看二维码。

为什么有些批次烧录的设备不显示在产品的设备标签页中？

设备标签页仅显示当前租户内工厂烧录的设备信息，其他租户烧录的批次不会显示。

版本

为什么版本标签页显示为灰色？

如果**版本**标签页显示为灰色，表示当前不可用。请检查该产品必要的资源与配置项是否均已完成配置。当所有内容配置完成后，**版本**标签页将自动启用。

为什么无法编辑或删除版本？

编辑或删除版本前请确保：

- 此版本不是**待审核**状态（如果版本是**待审核**状态，此版本将无法编辑或删除）。
- 此版本未关联任何批次（如果使用此版本创建批次后，此版本将无法编辑或删除）。

批次

为什么批次标签页显示为灰色？

如果**批次**标签页显示为灰色，表示当前不可用。请检查版本是否已创建完成。创建版本后，**批次**标签页将自动启用。

为什么在创建批次时，列出的部分版本为灰色且无法选择？

创建批次时，只可选择状态为**已通过**的版本。如果在创建产品时开启了版本审核，PROD 环境下的版本创建后将需要 Product Owner 或 Product Manager 进行审核（批准或拒绝）。

- 批准后，版本状态变为**已通过**并可用于创建批次。
- 拒绝后，版本状态变为**未通过**，需要修改并重新审核。

工厂

为什么工厂列表里的某些工厂无法点击查看？

工厂列表包含你参与的及租户内所有已创建的工厂。请确认你是否为该工厂的工厂管理员。如果不是，请联系该工厂的工厂管理员邀请你加入该工厂。被添加后，你将收到一封电子邮件，内含用于登录工厂服务管理页面的凭证。

为什么在创建批次时，列出的部分工厂显示为待定且无法选择?

创建批次时，只可选择状态为**正常**的工厂。本地工厂创建后，工厂的初始状态为**待定**状态，只有当给本地工厂添加 HSM U-Safe 后，工厂才会由**待定**状态变更为**正常**状态。更多信息请参考工厂状态说明。

为什么无法解绑 U-Safe?

如果 U-Safe 已锁定，则无法解绑。更多信息请参考解绑，锁定或解锁 HSM U-Safe 和 Identity U-Safe。

证书

为什么无法删除证书?

- 如果提示 "证书无法删除。它包含以下子证书" 并列出子证书列表，你需要先删除该证书下的所有子证书。
- 如果提示 "无法删除证书，证书已被以下列表引用" 并列出引用的资源列表，你需要先解除所有资源对该证书的引用。

OnBoard Edge Service & Programming Station

如何查看本地工厂用于登录 OnBoard Edge Service 和 Programming Station 客户端的服务地址?

在联系雪球协助完成 OnBoard Edge Service 的部署后，在**工厂**的**批次**标签页中，找到**服务信息**区域的**服务地址** 。更多信息请参考查看批次。

为什么无法登录 OnBoard Edge Service?

请检查是否已满足登录 OnBoard Edge Service 的前提条件。

Programming Station 客户端的密码?

初始登录密码是在为工厂添加烧录站后由系统随机生成的，你可以在 OBIS （本地工厂 & 云工厂）或 OnBoard Edge Service （本地工厂）中查看烧录站登录密码。该随机密码目前暂不支持修改。

版本：v3.2.2

发布说明

本文概述了 OnBoard™ IoT Security (OBIS) 各版本的发布日期、主要功能更新及相关文档内容，所涵盖的版本信息截至 **v3.2.2**。有关 CLI 和 API 各版本的主要更新与变更，请参考对应文档的修订记录。

2026 年发布版本

v3.2.2 (2026 年 2 月 9 日)

v3.2.2 版本对产品、工厂、证书、记录以及 Programming Station 等功能进行了更新。

功能分类	主要更新	更新说明	相关文档
产品	调整并优化批次页面	• **批次状态**，**批次同步状态**及**烧录数据同步状态**根据不同取值以不同颜色区分显示 • 批次支持以下搜索条件：**批次 ID**，**批次名称**，**版本名称**，**环境**，**创建时间**，**批次状态**，**批次同步状态**和**烧录数据同步状态**	产品 - 查看批次

功能分类	主要更新	更新说明	相关文档

工厂	调整并优化工厂管理员 (Factory Manager) 权限配置	仅工厂管理员可查看对应工厂的详情页并配置工厂	工厂：
			• 管理工厂与烧录站：
			  ◦ 本地工厂：
			  ▪ 配置与管理工厂
			  ▪ 配置与管理烧录站
			  ◦ 云工厂：
			  ▪ 配置与管理工厂
			  ▪ 配置与管理烧录站
			  ▪ 管理批次与分配烧录站
			• 查看工厂

功能分类	主要更新	更新说明	相关文档

证书	支持选择平台预置的 Google Cast 模板，同时在产品创建版本时可添加自定义证书或 Google 证书	• 创建证书时，可直接选择平台预置的 Google Cast 模板，并根据 Google Cast 的证书层级（Root / Manufacturer / Model / Device）自动获取对应配置
		• 在创建自定义证书时，支持配置 **Extended Key Usage**
		• 产品支持新增非 Matter 证书配置项，可在创建版本时添加非 Matter 证书，以便工厂生产支持其他协议的设备
		• 证书链详情页可展示新增证书模板，证书列表中可明确标识证书是否包含私钥
			• 产品：
			  ◦ 配置产品
			  ◦ 创建版本
			• PKI - 创建证书

功能分类	主要更新	更新说明	相关文档
记录	新增烧录档案与日志记录功能，在保留已烧录数据的同时，优化筛选与搜索能力，并调整记录查看权限	• 设备档案信息优化：记录设备在生产阶段的更多关键信息，包括芯片配置，Matter 配置，烧录的 DAC，工厂数据，固件版本，烧录脚本等 • 设备基础信息支持自定义扩展字段：除系统固定字段外，可定义自定义扩展字段，并可基于扩展字段进行模糊搜索。当前支持的扩展字段包括：**VID**，**PID**，**Chip UID**，**QR Code** 和 **Pairing Code** • 新增烧录日志：记录设备烧录过程中的所有关键事件及结果，以及烧录环境信息，便于追踪和	• 产品 - 查看烧录设备 • 记录 - 查看设备历史

功能分类	主要更新	更新说明	相关文档
		排查 • 权限控制优化：仅允许对应产品的任意产品角色权限或对应工厂的工厂管理员 (Factory Manager) 查看设备记录，确保数据安全	

功能分类	主要更新	更新说明	相关文档
Programming Station	• 优化 UI 布局，支持新版端口配置 • 新增批次详情查看功能，可通过订单号获取对应批次信息 • 支持自动识别 Identity U-Safe 已绑定的烧录站 ID，登录时无需手动输入	• 烧录客户端页面样式优化：展示本次烧录总进度，并可双击端口查看对应端口的烧录进度 • 多端口烧录支持：支持 1-24 个端口同时连接烧录，并可自由设置端口布局排列，提升操作灵活性 • 批次信息查询优化：支持通过订单编号获取对应批次，并可在烧录端查看批次的更多详细信息，便于追踪与管理 • 烧录站 ID 自动识别：插入 Identity U-Safe 后，客户端会自动识别已绑定的烧录站 ID，登录时无需手动输入	Programming Station: • 安装 • 登录 • 执行设备烧录

v3.2.1 (2026 年 1 月 15 日)

v3.2.1 版本对开通与登录、用户组、产品、版本、批次、工厂、KMS、OnBoard Edge Service 以及 Programming Station 等功能进行了更新。

功能分类	主要更新	更新说明	相关文档
开通与登录	新增 SSO 登录和企业邮箱自注册功能	• 支持 SSO 登录：通过企业 IdP 提供的账号直接登录 OBIS，无需单独记住 OBIS 账号 • 企业邮箱自注册：可使用公司邮箱完成注册，并自动分配默认角色，快速加入企业租户	开通与登录

功能分类	主要更新	更新说明	相关文档

用户组	新增**用户组**菜单	• 用户组管理：可将相关用户通过用户组进行关联，实现统一管理
		• 产品权限赋能：在产品中可直接添加用户组，将用户组内所有成员一次性添加至产品
			• 产品 - 管理成员
			• 用户：
			  ◦ 用户组：
			  ▪ 创建用户组
			  ▪ 查看用户组
			  ▪ 管理用户组

功能分类	主要更新	更新说明	相关文档

产品	• 拆分密钥和证书资源包
	• 支持下载产品资源文件
	• 新增产品归档功能
	• 创建产品时移除标记（Mark）字段
		拆分密钥与证书资源包：
		• 用户需提前将产品使用的所有证书与密钥添加至对应资源包
		• 产品的 Resource Manager 可在配置项（如 Matter Config、Chip Config）中从资源包选择所需证书或密钥
		
		下载产品资源文件：
		• 产品成员可下载上传的文件
		• 支持在资源配置页下载，包括：Firmware、Matter Config 的 CD File、Factory Data 文件、Programming
			• 快速入门
			• 产品：
			  ◦ 创建产品
			  ◦ 配置产品
			  ◦ 管理产品

功能分类	主要更新	更新说明	相关文档
		Script、Attachment File、Other File 产品归档功能： • 可对不活跃产品进行归档/取消归档，并在列表中标记归档状态，可快速筛选归档/未归档产品 • 归档产品不可编辑（信息修改、资源配置、版本或批次创建等），但已创建批次的烧录不受影响 移除产品 Mark 字段： • 创建产品时无需填写 Mark 字段，产品页面不再显示该字段 • 已存在产品的	

功能分类	主要更新	更新说明	相关文档
		Mark 字段会自动拼接到产品名称中	

版本	支持标签切换，长名称，量产标记及固件多选添加	• 标签页切换：创建版本时，可切换标签页查看产品其他信息
		• 版本名称：支持长度 128 字符
		• 量产标记：可标记版本为**量产**，标记后将在批次中显示量产标记
		• 固件添加优化：
		  ◦ 点击**添加**后展示固件列表，支持按文件名或版本搜索
		  ◦ 支持多选添加固件，提高操作效率
			产品 - 创建版本

功能分类	主要更新	更新说明	相关文档
批次	• 支持标签切换，长名称，版本量产信息显示，简化工厂选择，脚本自动匹配 • 完善批次及数据同步状态管理及重新下发功能	批次优化： • 标签页切换：创建批次时，可切换标签页查看产品其他信息 • 批次名称：支持长度 128 字符 • 版本信息展示：选择版本时，显示是否为量产及配置区域 • 工厂选择简化：只需选择工厂，无需选择工厂管理员 • 脚本自动匹配：根据所选版本配置的芯片，自动匹配烧录脚本 批次状态优化： • 批次状态：标识批次完整生命周期，包括：**生成中**，**已生成**，**生成**	产品： • 创建批次 • 查看批次

功能分类	主要更新	更新说明	相关文档
		失败，运行中，已完成，已停止，已过期 • 批次同步状态：标识批次数据同步到工厂的状态。包括：**等待中，下发中，下发失败，已同步，已删除** • 烧录数据同步状态：标识批次烧录的数据同步状态。包括：**无记录，上报中，已上报** • 重新下发功能：对于下发失败的批次（批次状态为**生成失败**或批次同步状态为**下发失败**），可在批次列表点击**重新下发**，将批次重新下发到工厂	

功能分类	主要更新	更新说明	相关文档
工厂	支持创建云工厂	• 云工厂创建：支持在 OBIS 中创建云工厂，并创建支持密码登录或 Identity U-Safe 登录的烧录站 • 云工厂管理：支持在 OBIS 中管理云工厂的批次及烧录站分配，包括批次的启停操作	工厂： • 创建工厂 • 管理工厂与烧录站： 云工厂： ▪ 配置与管理工厂 ▪ 配置与管理烧录站 ▪ 管理批次与分配烧录站 • 查看工厂
KMS	新增**完成**，**下载**和**归档**选项	• 密钥创建：选择**完成**后可继续创建密钥 • 密钥下载：支持下载非对称密钥的公钥 • 密钥归档：支持归档不活跃密钥，列表中优先显示活跃密钥，已归档密钥默认隐藏，便于日常管理	KMS - 管理密钥

功能分类	主要更新	更新说明	相关文档
OnBoard Edge Service	• 新增**工厂，批次**和**烧录站**菜单 • 支持多 HSM U-Safe 烧录	• 工厂管理：支持在 OnBoard Edge Service 中管理工厂，包括批次管理、烧录站分配，以及烧录站的配置与管理。 • 多 HSM U-Safe 烧录：Edge-Service 服务器可插入多个 HSM U-Safe，提高烧录性能与并发能力。	OnBoard Edge Service: • 登录 • 管理工厂 • 管理批次与分配烧录站 • 配置与管理烧录站

功能分类	主要更新	更新说明	相关文档

Programming Station	• 新增云工厂登录方式
	• 支持切换烧录脚本及自定义下载路径
	• 新增烧录并发负载显示与提示
		• 云工厂登录：可通过密码或 Identity U-Safe 登录
		• 本地烧录脚本管理：支持切换烧录脚本并自定义脚本下载路径
		• 烧录并发负载提示：
		  ◦ 客户端左下角实时显示烧录负载情况
		  ◦ 当负载 ≥ 90% 时显示预警提示，负载达到 100% 时显示高负载提示，便于操作监控
			Programming Station:
			• 登录
			• 执行设备烧录

版本：v3.2.2

如何在 OnBoard™ IoT Security 中为 Microsoft 工作帐户配置 SSO 登录

> **提示**
>
> OnBoard™ IoT Security (OBIS) 中的 Single Sign-On (SSO) 登录功能允许用户通过企业身份提供商 (IdP) 使用公司邮箱进行登录，OBIS 将自动为其创建账号并分配默认角色权限。

本文将指导你如何在 OBIS 中为 Microsoft 工作帐户配置和启用 **SSO 登录**功能。启用后，用户可通过工作帐户和密码登录 OBIS。

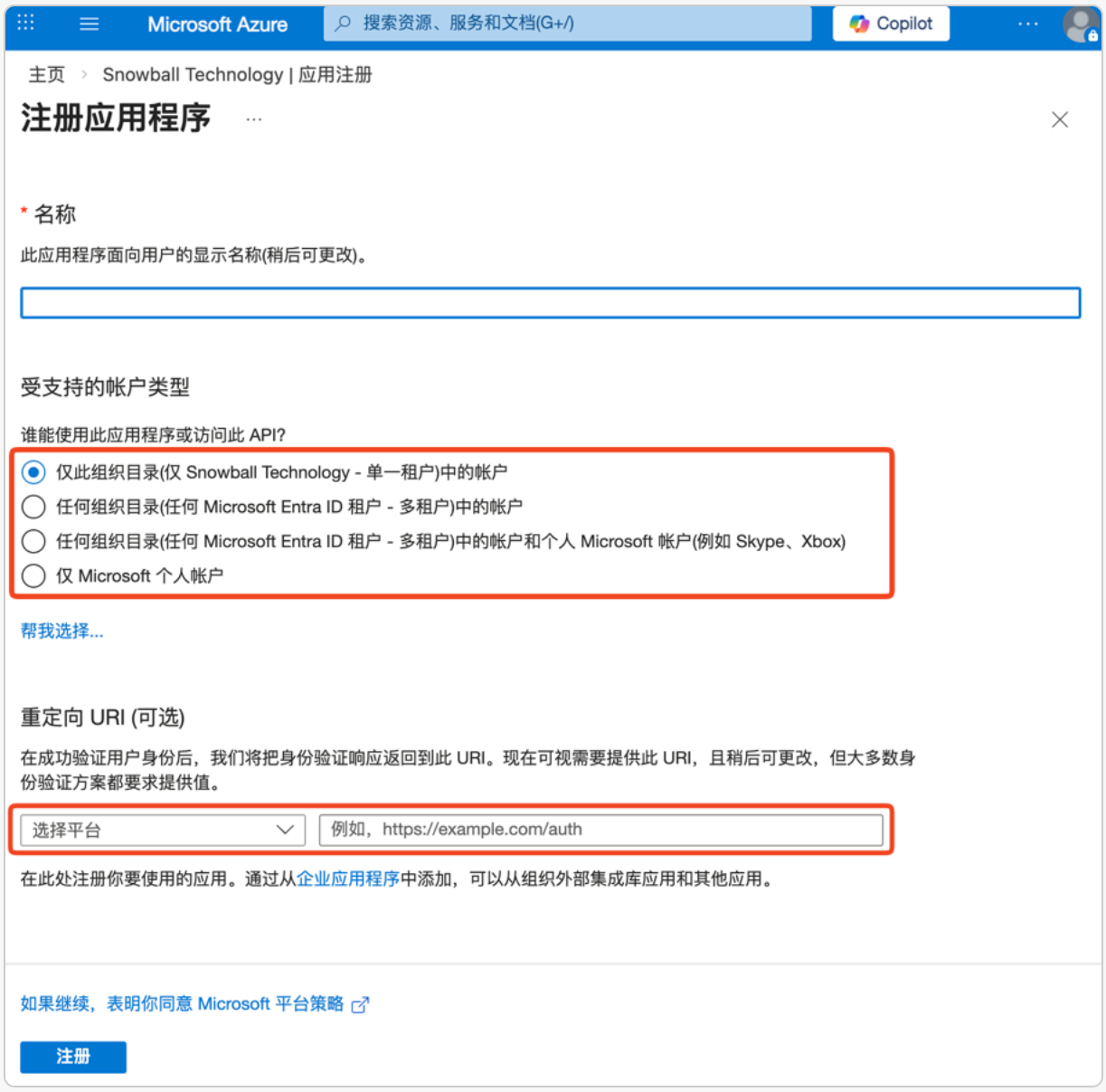
前提条件

在开始之前，请确保你已从雪球获取用于配置 SSO 所需的重定向 URL。

操作步骤

1. 登录 Microsoft Azure 门户。
2. 在顶部的搜索框中，搜索并选择 **Microsoft Entra ID**。
3. 点击左侧菜单中的**管理** > **应用注册** > **新注册**。在**注册应用程序**页面，输入应用程序名称并选择**受支持的账户类型**。

4. 在**重定向 URI** 区域，选择平台为 **Web** 并输入从雪球获取的重定向 URL 并点击**注册**。

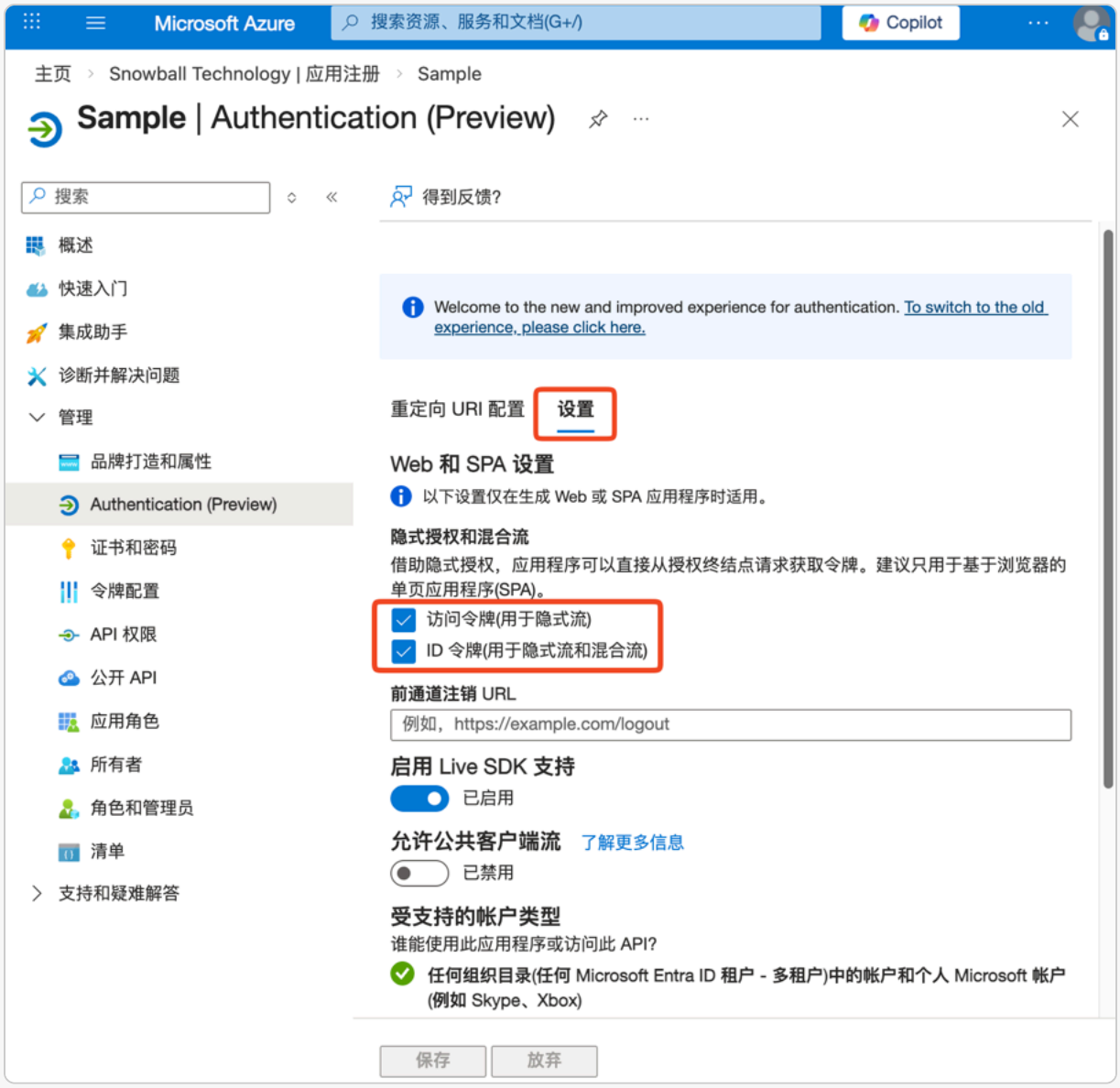


后续如果需要修改重定向 URI，可在应用程序的左侧菜单中点击**管理** > **Authentication (Preview)**，然后在**重定向 URI 配置**标签页中进行修改。



5. 在应用程序的左侧菜单中，点击**管理** > **Authentication (Preview)**，在**设置**标签页中勾选**访问令牌(用于隐式流)** 和 **ID 令牌(用于隐式流和混合流)** 复选框。

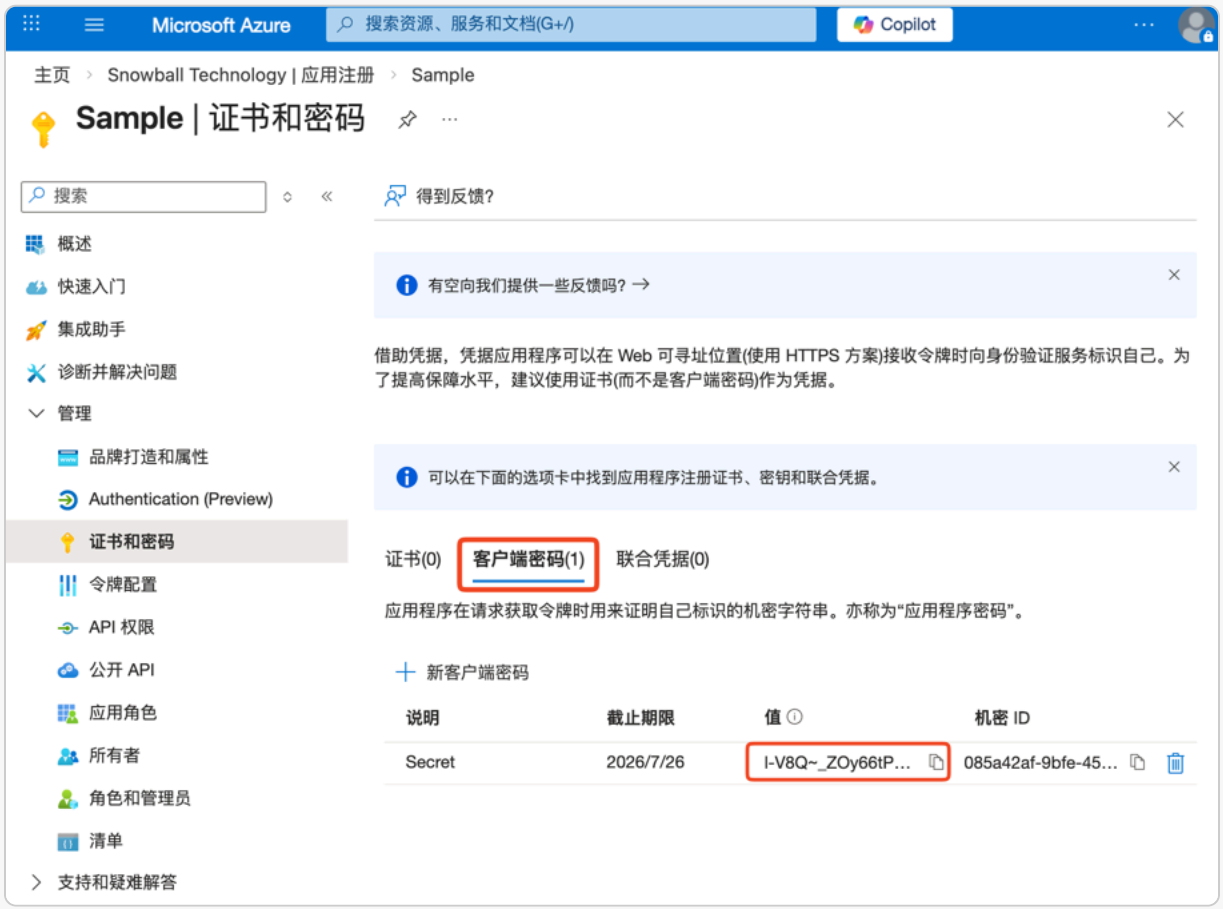
> **提示**
>
> 访问令牌用于调用获取用户信息接口，ID 令牌用于 SSO 登录过程中的参数校验。



6. 在应用程序的左侧菜单中，点击**管理** > **证书和密码**，选择**客户端密码**标签页并点击**新客户端密码**。在**添加客户端密码**面板输入客户端密码说明并选择**截止期限**，然后点击**添加**。

7. 记录客户端密码（Client Secret）的值。离开此页面后，该密码将不会再次显示。

> **提示**
>
> 客户端密码即当前授权应用的 Client Secret，用于回调验证应用信息，确保授权请求来源可信。



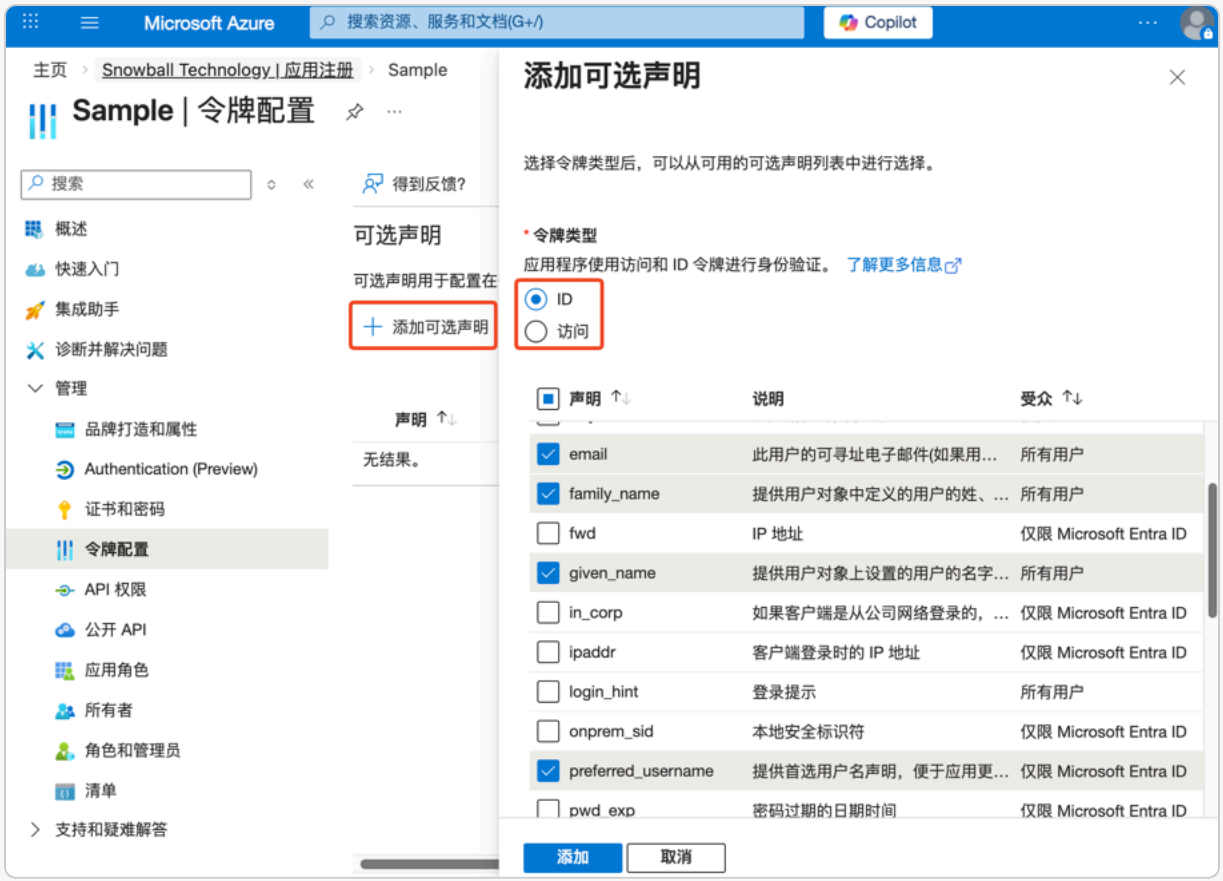
8. 在应用程序的左侧菜单中，点击**管理** > **令牌配置** > **添加可选声明**，然后分别配置和添加 ID 令牌和访问令牌。

> **提示**
>
> 此处的可选的声明对应微软在认证回调中返回给平台对应的信息。

   - 选择**令牌类型**为 **ID**，然后勾选 **email**，**family_name**，**given_name** 和

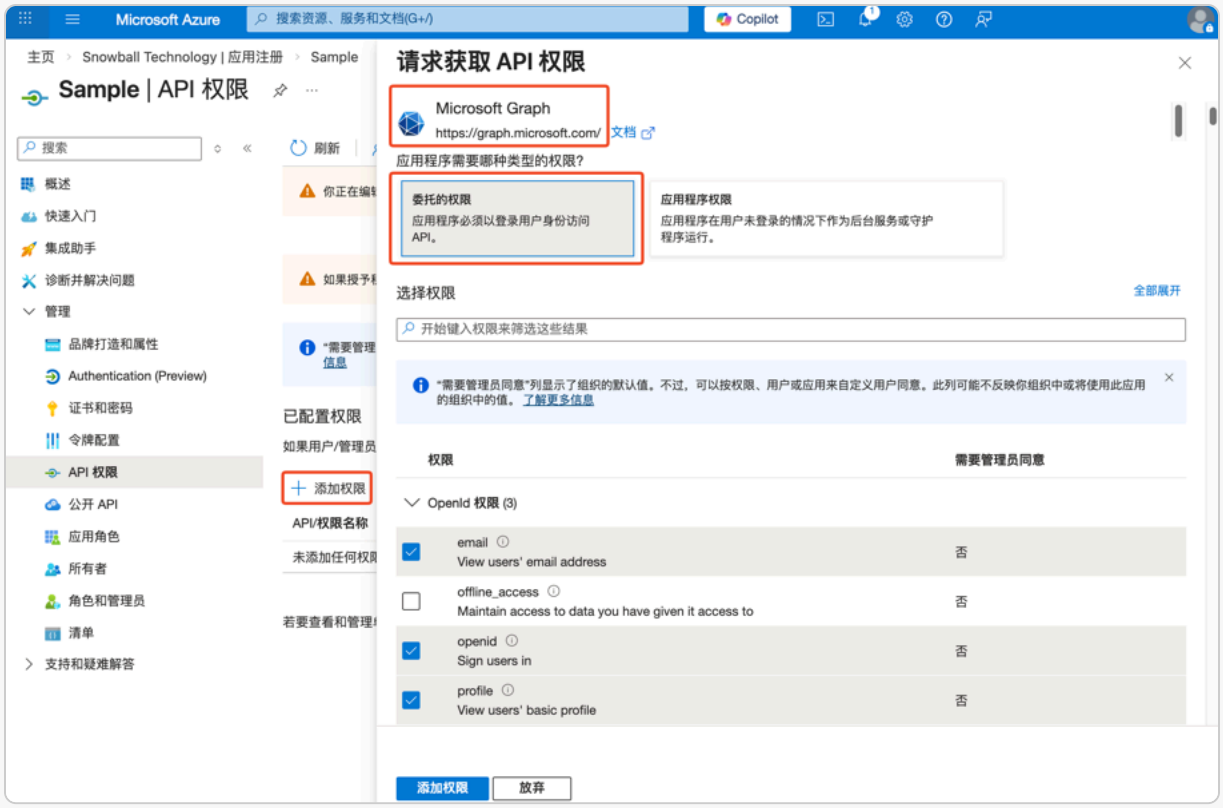
preferred_username

- 选择**令牌类型**为**访问**，然后勾选 **email**，**family_name**，**given_name** 和 **preferred_username**



9. 在应用程序的左侧菜单中，点击**管理** > **API 权限** > **添加权限**。在**请求获取 API 权限**面板选择 **Microsoft Graph**，并选择**委托的权限**，在 **OpenId 权限**区域中勾选对应的权限（建议勾选 **email**，**openid** 和 **profile**）并点击**添加权限**。

> **提示**
>
> 此处配置的 API 权限对应 OBIS 中 SSO 设置的范围配置。



10. 完成应用程序配置后，请向雪球提供以下信息，我们将据此在 OBIS 中为你的租户启用 **SSO 登录**功能。

- **邮箱域名**：Microsoft 工作帐户的邮箱域名
- **签发机构**：在应用程序的**概述**页面，点击**终结点**，然后复制对应的**机构 URL**
- **客户端 ID**：在应用程序的**概述**页面，复制**应用程序(客户端) ID**
- **客户端密钥**：在第七步记录的客户端密码


Microsoft Azure
搜索资源、服务和文档(G+/)
Copilot
主页 › Snowball Technology | 应用注册
Sample
搜索
概述
快速入门
集成助手
诊断并解决问题
管理
品牌打造和属性
Authentication (Preview)
证书和密码
令牌配置
API 权限
公开 API
应用角色
所有者
角色和管理员
清单
删除
终结点
预览功能
概要
显示名称 : Sample
应用程序(客户端) ID : 47cd29e2-
e4c5a1da35c4
对象 ID : 93d7c9af-
db2e3869824e
目录(租户) ID : ca4cd70c-
71347e8ee307
受支持的帐户类型 : 所有 Microsoft 帐户用户
自 2020 年 6 月 30 日起，我们将不再向 Azure Active Directory 身份
要升级到 Microsoft 身份验证库(MSAL)和 Microsoft Graph。 了解更
自 2020 年 11 月 9 日起，最终用户将不再能够同意未经认证的发布者
入门
文档
Microsoft 标识平台是身份验证服务、
终结点
机构 URL (仅此组织目录中的帐户)
https://login.microsoftonline.com/common
机构 URL (任何组织目录中的帐户)
https://login.microsoftonline.com/organizations
机构 URL (任何组织目录中的帐户和个人 Microsoft 帐户)
https://login.microsoftonline.com/common
机构 URL (仅个人 Microsoft 帐户)
https://login.microsoftonline.com/consumers
OAuth 2.0 授权终结点(v2)
https://login.microsoftonline.com/common/oauth2/v2.0/authorize
OAuth 2.0 令牌终结点(v2)
https://login.microsoftonline.com/common/oauth2/v2.0/token
OAuth 2.0 授权终结点(v1)
https://login.microsoftonline.com/common/oauth2/authorize
OAuth 2.0 令牌终结点(v1)
https://login.microsoftonline.com/common/oauth2/token
SAML-P 登录终结点
https://login.microsoftonline.com/ca4cd70c-
71347e8ee307/saml2

Make Security Simple

上海雪球数智科技有限公司
Shanghai Snowball Digital Intelligence Technology Co., Ltd

www.snowballtech.com
info@snowballtech.com
上海市浦东新区御桥路 968 号御桥智慧湾 T2-19 楼

访问官网
www.snowballtech.com

© 2026 上海雪球数智科技有限公司 版权所有
本文档仅供参考，不构成任何形式的明示或暗示。
如需引用或使用，请联系上海雪球数智科技有限公司获得授权。