

## Overview

OAuth 2.0 is a token-based mechanism, using which your apps can securely access Knox cloud services APIs without exposing their credentials. To begin using Knox cloud services APIs, you need to generate an access token. The workflow for generating the token depends on whether you're a UEM partner, customer, or a Managed Service Provider (MSP).

The two principal flows are:

- Authorization code flow for UEM partners.
- Client credentials flow for customers and MSPs.

### Authorization code flow for UEM partners

This flow applies to you if you're a UEM partner, looking to integrate Knox cloud services features into your platform, and you want to programmatically access your customer's Knox cloud services on their behalf. Here's how it works.

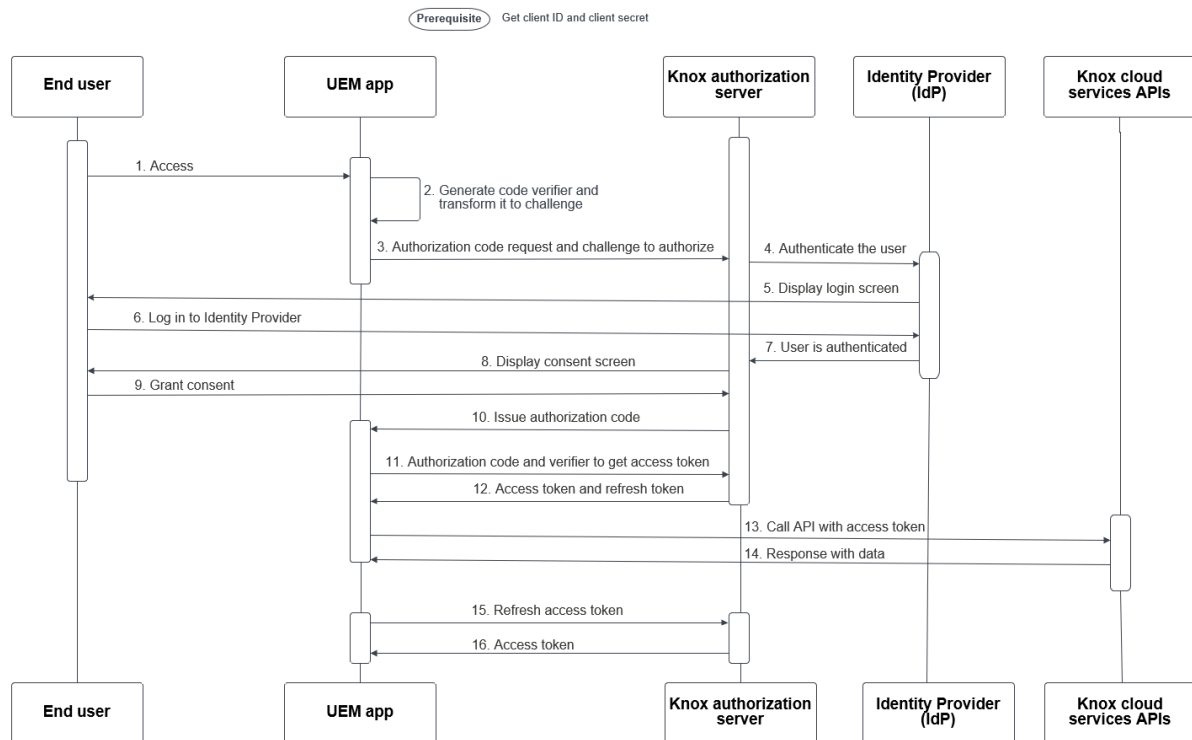
### Before you begin

[Register your app](#) to generate its client ID and client secret.

### Step-by-step authorization code flow

1. **Access request** — The end user initiates access to the UEM app.
2. **PKCE challenge generation** — The UEM app generates a code verifier, which is a random string and transforms it into a code challenge using a hashing algorithm.
3. **Authorization code request** — The UEM app sends an authorization code request to the Knox authorization server, including the client ID, the client secret, and the code challenge.
4. **User authentication** — The Knox authorization server redirects the request to the Identity Provider (IdP) for user authentication.
5. **Login screen display** — The IdP displays the login screen to the end user.
6. **User login** — The end user logs in by providing their credentials to the IdP.
7. **User authentication by IdP** — The IdP authenticates the user's credentials.

8. **Consent screen** — The IdP displays a consent screen to the end user, asking for permission to allow the UEM app to access the required resources, which are determined by scopes. For more information, see [Scopes for Knox cloud services APIs](#).
9. **Grant consent** — The end user grants consent.
10. **Authorization code issuance** — The Knox authorization server issues an authorization code and sends it back to the UEM app.
11. **Access token request** — The UEM app sends the authorization code and the code verifier to the Knox authorization server to request access token.
12. **Access token issuance** — The Knox authorization server validates the authorization code and the code verifier. If the credentials are valid, this proves that UEM app is authorized to make calls on behalf of the end user. The Knox authorization server then issues an access token and a refresh token to the UEM app.
13. **API call with access token** — The UEM app uses the access token to call the Knox cloud services APIs to access the resources. The access token must be active when you make this API call.
14. **Data response** — The Knox cloud services APIs respond with the requested data.
15. **Access token refresh** — When the access token expires, the UEM app requests a new access token from the Knox authorization server using the refresh token.
16. **New access token** — The Knox authorization server issues a new access token to the UEM app.



For more information, see [Knox OAuth 2.0 Authentication for UEMs](#).

## Client credentials flow for customers and MSPs

This flow applies to you if you're a Knox cloud services customer looking to programmatically access your Knox cloud services features, or you're an MSP looking to programmatically access your managed customer's Knox cloud services. Here's how it works.

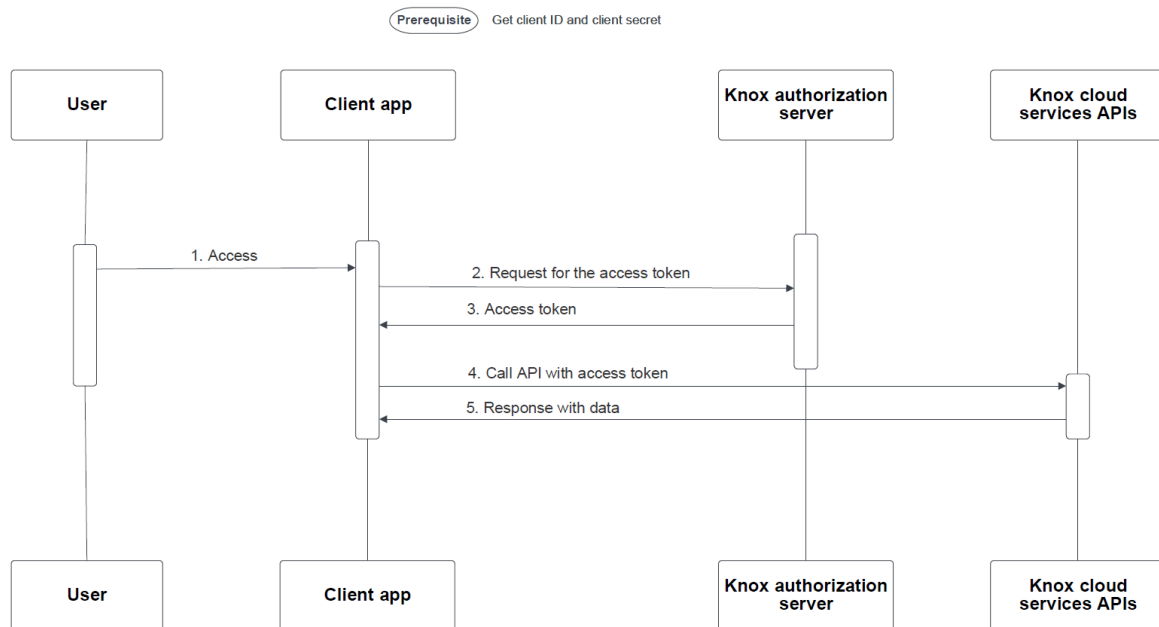
### Before you begin

[Register your app](#) to generate its client ID and client secret.

### Step-by-step client credentials flow

1. **Access request** — The user initiates access to the client app.
2. **Access token request** — The client app requests an access token from the Knox authorization server using its client ID and client secret.
3. **Access token issuance** — The Knox authorization server authenticates the client application using the provided client credentials. If the credentials are valid, the authorization server issues an access token to the client app.

4. **API call with access token** — The client app uses the access token to call the Knox cloud services APIs to access the resources. The access token must be active when you make this API call.
5. **Data response** — The Knox cloud services APIs respond with the requested data.



For more information, see [Knox OAuth Authentication 2.0 for Customers and MSPs](#).

## Register new app

This topic covers how your app can access Knox cloud services APIs, either for yourself or on the behalf of your end customers. To connect an app to Knox cloud services using APIs, you need an access token. And, to generate an access token, you must first obtain a client ID and a client secret from the Knox Developer portal for each of your apps, the following section guides you on this.

You can also register your app through client management operations in [Knox OAuth 2.0 Authentication API](#), contact [Support](#) for more information on this.

## Supported services

The Knox cloud services which currently support OAuth 2.0 are:

- Knox Asset Intelligence
- Knox Configure
- Knox Deployment Program
- Knox E-FOTA
- Knox Manage (exclusive to MSP partners)
- Knox Mobile Enrollment
- Knox MSP Portal
- Knox Webhook Notification

## Before you begin

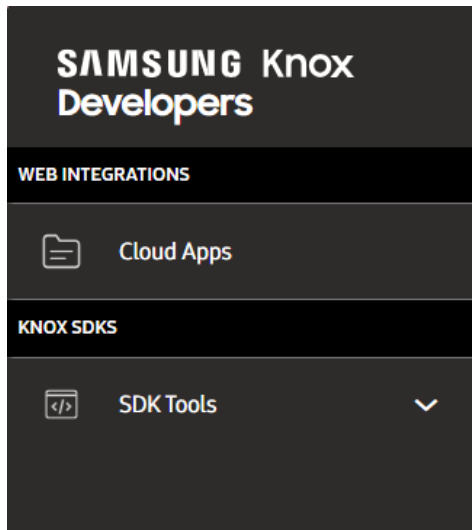
Ensure you can view the Cloud Apps menu on the navigation pane of [Knox Developer portal](#). If you're not able to view the menu, please email [knoxapi@samsungknox.com](mailto:knoxapi@samsungknox.com) with the subject 'Cloud Apps feature', and include the following details :

- Company name
- Country
- Email address associated with your Samsung Knox account
- List of Knox cloud services APIs you want to use
- Description of your use cases

You can also create a [Support](#) ticket to request access to the Cloud Apps menu.

## Generate a client ID and client secret

1. Sign in to the [Knox Developer Portal](#).
2. Go to **WEB INTEGRATIONS** > **Cloud Apps** from the landing page, or go to **Cloud Apps** on the navigation pane.



3. Click **REGISTER NEW APP**.

**You don't have any apps yet.**

Start adding your first app.

**REGISTER NEW APP**

4. The first time you register an app, the **Knox Cloud API License Agreement** appears. Review the agreement, confirm you've read it, and click **ACCEPT** to consent.

## KNOX CLOUD API LICENSE AGREEMENT



This Knox Cloud API License Agreement (the "Agreement") is between Samsung Electronics Co., Ltd. ("Samsung") and the legal entity on behalf of which you are legally authorized to act for ("Licensee") (each a "Party" or jointly the "Parties"). In the event Licensee is acting on behalf of a Customer (as defined below), Licensee shall be responsible for all actions of the Customer it is acting on behalf of with regard to this Agreement.

IT IS IMPORTANT THAT YOU READ CAREFULLY AND UNDERSTAND THIS AGREEMENT. BY CLICKING THE "ACCEPT" BUTTON OR ACCESSING, DOWNLOADING, INSTALLING OR USING ANY ASPECT OF THE SDK (DEFINED BELOW), YOU AGREE AS A LEGALLY AUTHORIZED REPRESENTATIVE OF LICENSEE THAT LICENSEE CONSENTS TO BE BOUND BY AND BECOME A PARTY TO THIS AGREEMENT. IF YOU DO NOT AGREE THAT LICENSEE CONSENTS TO BE BOUND BY ALL THE TERMS OF THIS AGREEMENT, DO NOT CLICK ON THE "ACCEPT" BUTTON OR ACCESS, DOWNLOAD, INSTALL OR USE THE SDK.

YOU AGREE THAT SAMSUNG AND ITS LICENSORS MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE SDK, IF ANY, AT ANY TIME WITHOUT NOTICE, AND FURTHER AGREE THAT SAMSUNG MAY REVISE THIS AGREEMENT, INCLUDING ANY OPERATING RULES, POLICIES AND PROCEDURES, INCLUDING THE SAMSUNG PRIVACY POLICY AT ANY TIME WITHOUT NOTICE BY UPDATING THIS POSTING. YOUR CONTINUED USE OF THE SITE AFTER SUCH MODIFICATIONS HAVE BEEN MADE CONSTITUTES YOUR ACCEPTANCE OF SUCH REVISED AGREEMENT.

### 1 Definitions

- 1.1 "Activate" means to authorize a Web Service to make valid, authorized Knox Cloud API calls for the Knox Cloud Service after Samsung checks the validity of Web Service with the API Access Token, and, in case of a Premium Knox Cloud Service, the Service License Key(s). For the avoidance of doubt, Licensee or Customer, as the case may be (hereinafter marked as "Licensee/or Customer"), shall register for the Knox Cloud Service as a subscriber of the Knox Cloud Service for the purpose of direct use or distribution of the Knox Cloud Service to its End User(s) at a corporate level and shall obtain written approval from Samsung on the registration request, and, in case of a Premium Knox Cloud Service, purchase valid Service License Key(s) for it and its End User(s) to use the Premium Knox Cloud Service.
- 1.2 "Activation" means a process to Activate.
- 1.3 "Affiliate" means any entity controlling, controlled by or under common control with a Party hereto, where "control" means the direct or indirect ownership of more than fifty percent (50%) of such entity's capital or equivalent voting rights including all legal entities, companies, corporations, firms, partnerships or other entities that are controlled by a Party to this Agreement.
- 1.4 "Web Service" means a software program developed and owned by Licensee using the SDK for operation of the

☒ I have read and I agree to the Knox Cloud API License Agreement\*

CANCEL

ACCEPT

5. Enter basic details about your app like **App name** and **App description**. App name is required.

#### APP INFORMATION

App name \*

App description

Provide a brief description of your app (i.e, integration with Samsung Knox)

Up to 3900 characters, remaining: 3900

If you choose the **Authorization code** grant type in the next step, then the **App name** you specify is included in the consent screen that is shown to your customer.

6. Select the appropriate **GRANT TYPES**. You can select multiple options, depending on your use case.

#### GRANT TYPES

Select one or both grant types

- ☐ **Client credentials**  
Choose this grant type if your app is accessing your own Knox cloud services.
- ☐ **Authorization code**  
Choose this grant type if your app is accessing Knox cloud services on behalf of your customer. You'll see a preview of the consent screen that your customer will see.

- a. Select **Client credentials** if your app needs access to your own Knox cloud services. This is a two-legged OAuth 2.0 flow.
- b. Select **Authorization code** if you are a Unified Endpoint Management (UEM) partner and your app requires access to Knox cloud services on behalf of a customer. This is a three-legged OAuth 2.0 flow.

Also provide the following details which are displayed on the consent screen shown to customers:

- **Company name** — Enter the name of your company, it is a required field.
- **Home page** — Enter the home page link of your company or app.
- **Terms of service link** — Enter the link where your customers can go to view the terms of service of your company.
- **Privacy policy link** — Enter the link where your customers can go to view the privacy policy link of your company.
- **Support email address** — Enter the support email address of your company.
- **Company logo** — Upload your company logo in .jpg or .png format. Ensure that the file size is less than 200 MB.
- **REDIRECT URLs** — Enter the URLs to redirect your customers to your app after they grant consent.



## CONSENT SCREEN

The information you provide will be shown to customers on the consent screen.

Company name \*

Provide your company name.

Home page

Provide your company or app home page.

Terms of service link

Provide an app terms of service link.

Privacy policy link

Provide an app privacy policy link.

Support email address

Provide a support email address.

Company logo

CHOOSE FILE

No file chosen.

Allowed file types: .jpg, .png

Files must be less than 200 MB


## REDIRECT URLS

Specify a URL to redirect customers back to the app with after they grant consent.

If you add more than one redirect URL, you'll need to specify which one to use when making an API call to request an authorization code.

URL 1 \*

https://www.example.com

 ADD URL

If you enter multiple redirect URLs using the **ADD URL** option, make sure to specify a preferred URL in your API call to [get authorization code](#).

- **EXPIRATION TIME** — Depending on the grant type selected, modify the following expiration times, if required.

## EXPIRATION TIMES

Based on security and usability considerations, this section is pre-filled with our recommended values.[Learn more](#)

Authorization code expiration (Must be between 1 and 5 minutes)

Access token expiration (Must be between 1 and 60 minutes)

Refresh token expiration (Must be between 60 minutes and 90 days)

days | 

[CANCEL](#)

**CONTINUE**

- **Authorization code expiration** — Enter a value between one and five minutes.
- **Access token expiration** — Enter a value between one minute to 60 minutes.
- **Refresh token expiration** — Enter a value between 60 minutes to 90 days.

You can preview the consent screen that is shown to customers, with the details you specified.

**SAMSUNG Knox**

[Company name] test want to access your account

Your company Logo

The following permissions will be granted to **test** for your organization:

[Service name]

- 
-

If you click **Allow**, you consent to this app accessing your data per the [Samsung Knox Product Terms & Conditions](#). If you click **Deny**, your data won't be shared with this app.

DENY

ALLOW

7. Click **CONTINUE** after you finish providing your app information.
8. Click **ADD SCOPE** to list the scopes that your app needs. Scopes define the permissions available to an app.

## SCOPE

Scopes allow you to define and limit the permissions that your app will request.

ADD SCOPE

9. On the **Add Scope** page, specify the following:
  - a. Select the Knox cloud services required for your app. By default, all the available services are shown.

## ADD SCOPE

### Services

Knox Asset Intelligence x  
Knox Device Management Service x Knox E-FOTA x  
Knox Mobile Enrollment x Knox MSP x Others x  
SOC x

🔍 Search by scope, user-facing description

<input type="checkbox"/>	SCOPE	USER-FACING DESCRIPTION
	Knox Asset Intelligence	
<input type="checkbox"/>	kai <small>All</small>	Manage all Knox Asset Intelligence functionality, including devices, diagnostics logs, and network settings
<input type="checkbox"/>	kai.app	Subscribe to Knox Asset Intelligence notifications about app usage and abnormal app events
<input type="checkbox"/>	kai.battery	Subscribe to Knox Asset Intelligence notifications about battery-related events

- b. Select the required scopes from the options available and click **ADD**.  
Scopes are displayed based on the services you selected. After you've added the required scopes, click cross on the top-left corner.

## ADD SCOPE

### Services

Knox Asset Intelligence x

Search by scope, user-facing description

	SCOPE	USER-FACING DESCRIPTION
Knox Asset Intelligence		
	<input type="checkbox"/> kai All	Manage all Knox Asset Intelligence functionality, including devices, diagnostics logs, and network settings
	<input checked="" type="checkbox"/> kai.app	Subscribe to Knox Asset Intelligence notifications about app usage and abnormal app events
	<input checked="" type="checkbox"/> kai.battery	Subscribe to Knox Asset Intelligence notifications about battery-related events

10. If you select the parent scope of a Knox cloud service, its sub-scopes are automatically selected.

11. If you want to delete any added scopes, select them and click **REMOVE**.

### SCOPES YOU'VE ADDED

Search by service, scope, user-facing description

REMOVE

	SCOPE ↓↑	USER-FACING DESCRIPTION
Knox Asset Intelligence		
	<input checked="" type="checkbox"/> kai.app	Subscribe to Knox Asset Intelligence notifications about app usage and abnormal app events
	<input type="checkbox"/> kai.battery	Subscribe to Knox Asset Intelligence notifications about battery-related events

CANCEL

BACK

CONTINUE

12. Click **CONTINUE**. Summary of the new app is shown, it includes the app information you've entered, the scopes you've selected, and the consent screen preview.

## SUMMARY

Review summary of your app before submission.

BASIC INFORMATION		EDIT
App name	Sample App	
App description	This is a sample app.	
Grant types	Client Credentials, Authorization Code	
Company name	Sample company name	
Home Page	https://sampleHomePage.com	
Terms of service link	https://sampleTermsOfService.com	
Privacy policy link	https://samplePrivacyPolicy.com	
Support email address	sampleSupportEmail@email.com	
Redirect URLs	https://sampleRedirectURL1.com https://sampleRedirectURL2.com	

SCOPE			EDIT
Service	Scope	User-facing description	
Knox Asset Intelligence	kai.app	Subscribe to Knox Asset Intelligence notifications about app usage and abnormal app events	
Knox Asset Intelligence	kai.battery	Subscribe to Knox Asset Intelligence notifications about battery-related events	

**CREDENTIALS**

You'll see the client ID and secret after you click SUBMIT.

Customer consent screen preview (authorization code):

**SAMSUNG Knox**

Sample company name Sample App want to access your account

Sample App Logo

The following permissions will be granted to Sample App for your organization:

[Service name]

•

•

If you click **Allow**, you consent to this app accessing your data per the [Samsung Knox Product Terms & Conditions](#). If you click **Deny**, your data won't be shared with this app.

Please also review the following information from Sample company name: [Terms of Service](#), [Privacy policy](#). For support using the app, contact [sampleSupportEmail@email.com](mailto:sampleSupportEmail@email.com)

DENY ALLOW

🔗 FULL SCREEN PREVIEW

CANCEL

BACK

SUBMIT

13. Go to **CONSENT SCREEN > FULL SCREEN PREVIEW** to see how the consent screen is presented to your end customers. The full preview shows additional details such as scopes and its descriptions.

14. Click **SUBMIT**. A client ID and client secret is issued for your app.

## App submitted!

Make sure you save the client ID and secret in a safe place.

For security reasons, we only store your client secret until your session ends. If you lose your client secret afterwards, you can rotate it to get a new one.

### Your Client ID

4b0040c-8281b70f-c542g-5d01e

### Your Client Secret

\*\*\*\*\*

BACK TO APP LIST

15. Copy and save your client secret. The client secret is displayed only till the current session is active, and will not be available after you exit the session. If you forget to copy it, see how to view or rotate the client secret in [Manage registered apps](#).

## For customers and MSPs

### Step 1 — Get access token

You can call the [POST /oauth2/token](#) operation, using your app's client ID and client secret, to get the access token.

The expiration period for an access token is 10 minutes.

The following request format is used to get an access token using client credentials:

```
curl --location 'https://api.samsungknox.com/ams/v1/oauth2/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id={OAUTH2_CLIENT_ID}' \
--data-urlencode 'client_secret={OAUTH2_CLIENT_SECRET}' \
--data-urlencode 'scope={Required Scope}'
```

For example:

```
curl --location 'https://api.samsungknox.com/ams/v1/oauth2/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=a1bC...d2efg' \
--data-urlencode 'client_secret=ABC...DeFG' \
--data-urlencode 'scope=kai'
```

You receive the access token in the following format. The value of the `expires_in` key denotes the number of seconds for which the access token is valid. You need an active access token every time you make a Knox cloud services API call.

```
{
  "access_token": "abc123...abc123",
  "scope": "kai",
  "expires_in": 599,
  "token_type": "Bearer"
}
```

## Step 2 — Call Knox cloud services APIs with access token

You can use the access token obtained in [Step 1 – Get access token](#) to call Knox cloud services APIs.

### As a customer

After obtaining an access token using Knox OAuth 2.0 Authentication, you can make Knox cloud service API requests.

For example, a cURL request for Knox Asset Intelligence service:

```
curl --location 'https://api.samsungknox.com/kai/v1/settings' \  
      --header 'Authorization: Bearer aB1cDE.....fG2itijk'
```

A sample response for the Knox Asset Intelligence request above:

```
{  
  "battery":  
    { "batteryLevelThresholds": [] },  
  "enrollment":  
    { "allowEnrolledToKnoxConfigure": false }  
}
```

### As an MSP

After obtaining an access token using Knox OAuth 2.0 Authentication, when you make Knox cloud service API requests, include the managed customer ID value in the x-wsm-managed-tenantid header. The managed customer ID is available on the **Customers** page of the Knox MSP Portal, and you can also retrieve it using [Get customer information](#) endpoints.

For example, a cURL request for Knox Asset Intelligence service:

```
curl --location 'https://api.samsungknox.com/kai/v1/settings' \  
      --header 'Authorization: Bearer aB1cDE.....fG2itijk' \  
      --header 'x-wsm-managed-tenantid: 1123123123'
```

A sample response for the Knox Asset Intelligence request above:

```
{  
  "battery":  
    { "batteryLevelThresholds": [] },
```

**"enrollment":**

```
{ "allowEnrolledToKnoxConfigure": false }  
}
```

Knox cloud services API requests are only available for your managed customers. If the request is for a non-managed customer, it fails.

### **Supported services**

The Knox cloud services that support OAuth 2.0 are:

- Knox Asset Intelligence
- Knox Configure
- Knox Deployment Program
- Knox E-FOTA
- Knox Manage (exclusively for MSP partners)
- Knox Mobile Enrollment
- Knox MSP Portal
- Knox Webhook Notification



# For UEM partners

## Step 1 — Get authorization code

The authorization code grant type lets your registered app interact with the end user's agent, typically a web browser, and receive incoming requests from the Knox authorization server through redirection.

When the end user provides consent for use of requested Knox cloud services APIs, the Knox authorization server issues an authorization code that is exchanged for the access and refresh tokens. These tokens are used to authenticate the Knox cloud services API requests, which your app makes on the behalf of your end user.

1. First, your app directs the user-agent to the [GET /oauth2/authorize](#) operation with the request body parameters as shown in the following sample:
2. `https://api.samsungknox.com/ams/v1/oauth2/authorize?`
3. `response_type=code&`
4. `client_id=example-client&`
5. `scope=kai&`
6. `redirect_uri=https://example-app.com/redirect&`
7. `code_challenge=1234abcd123abCd1a2BCdaBCdab3cDABcdaBcDABcD&`
8. `code_challenge_method=S256&`
9. `state=abcde`

The authorization code request sample must:

- Include your app's **client\_id**.
- Include the **scope** to define the scope of the access request. See [scopes for Knox cloud services APIs](#) for the list of available scopes.
- Include the redirect URL (**redirect\_uri**) that you specified when creating your app.
- Include a **code\_challenge** as per the PKCE protocol. It's required to use this, to prevent you from engaging in insecure end-user flows.
  1. First, generate a code verifier, according to the Proof Key for Code Exchange (PKCE) protocol, it is a high-entropy cryptographic random string with a length between 43 and 128 characters. It can contain letters, digits, underscores, periods, hyphens, or tildes.

2. Hash the generated code verifier using SHA-256 code challenge method.

The PKCE is a security enhancement for the OAuth 2.0 authorization framework. Ensure that your app stores the code verifier in the backend only, and never sends or exposes it through the frontend or web browser. The best practices to securely store the PKCE code verifier in your app are as follows:

- **Secure database:** Use a secure database with robust credentials.
- **Secure transmission:** Transmit the code verifier over a secure, encrypted channel.
- **Rotation or deletion:** Promptly delete or replace the code verifier after using it one-time only.

The following are some examples of how to generate the code verifier and code challenge:

Bash example

```
#!/bin/bash
```

```
# Generate random code verifier
```

```
code_verifier=$(openssl rand -hex 32)
```

```
# Hash and base64 url encode code verifier
```

```
code_challenge=$(echo -n "$code_verifier" | openssl dgst -sha256 -binary | base64  
| tr '/+' '_-' | tr -d '=')
```

```
echo "Code Verifier: $code_verifier"
```

```
echo "Code Challenge: $code_challenge"
```

○

Java example

```
import java.security.MessageDigest;  
import java.security.NoSuchAlgorithmException;  
import java.security.SecureRandom;  
import java.util.Base64;
```

```
public class PKCEGenerator {  
    public static void main(String[] args) throws NoSuchAlgorithmException {
```

```
        // Generate a secure random code verifier
```

```
        SecureRandom sr = new SecureRandom();
```

```
        byte[] code = new byte[32];
```

```
        sr.nextBytes(code);
```

```
        String codeVerifier =
```

```
Base64.getUrlEncoder().withoutPadding().encodeToString(code);
```

```
        // Generate a code challenge from the code verifier
```

```
        MessageDigest md = MessageDigest.getInstance("SHA-256");
```

```
        byte[] digest = md.digest(codeVerifier.getBytes());
```

```
        String codeChallenge =
```

```
Base64.getUrlEncoder().withoutPadding().encodeToString(digest);
```

```
System.out.println("Code Verifier: " + codeVerifier);  
System.out.println("Code Challenge: " + codeChallenge);  
}  
}
```

- Specify **code\_challenge\_method**. Knox authorization server supports only SHA-256: **code\_challenge\_method=S256**.
- The **state** parameter, specifies an opaque value, and is a random string used by the client to maintain the state between the request and callback. The authorization server includes this value when redirecting the user-agent back to the client. It's mandatory to use this parameter, it provides protection against certain attacks, such as Cross-Site Request Forgery (CSRF).

10. Knox authorization server authenticates your end user, using Samsung Account or other identity provider, and obtains their consent for you to manage their Knox cloud services resources.

When the end user provides consent, they're redirected to your app's redirect URL with the authorization code and the state parameter. Ensure that the value of **state** parameter is what you initially specified in the authorization code request.

Redirect URL example:

<https://example-app.com/redirect?code=abcdefghijkl&state=abcde>

For more information, see [Knox OAuth 2.0 Authentication API](#).

## Step 2 — Get access and refresh tokens

You can call the [POST /oauth2/token](#) operation to get access and refresh tokens in exchange for the authorization code. The following are the default expiration times for these tokens:

- Access token - Short lived, 10 minutes.
- Refresh token - Long lived, 90 days.
- Authorization code - Short lived, 1 minute.

To get a new refresh token before its expiry, see [Step 4 – Refresh access token](#) for more information.

The following request sample shows how you can get access and refresh tokens using the authorization code:

```
curl --location 'https://api.samsungknox.com/ams/v1/oauth2/token' \  
  --header 'Content-Type: application/x-www-form-urlencoded' \  
  --data-urlencode 'grant_type=authorization_code' \  
  --data-urlencode 'client_id={OAUTH2_CLIENT_ID}' \  
  --data-urlencode 'client_secret={OAUTH2_CLIENT_SECRET}' \  
  --data-urlencode 'redirect_uri={REDIRECT_URL_USED_IN_AUTHZ_CODE_REQ}' \  
  --data-urlencode 'code_verifier={PKCE_CODE_VERIFIER}' \  
  --data-urlencode 'code={AUTHORIZATION_CODE}'
```

You receive the access and refresh tokens in the below format. The value of `expires_in` denotes the number of seconds that the access token is valid for. You need an active access token every time you make Knox cloud services API call.

```
{  
  "access_token": "abCdeFg..HiJKlM2o",  
  "refresh_token": "aBCl1..DE2fgHI",  
  "scope": "kai",
```

```
"expires_in": 599,  
"token_type": "Bearer"  
}
```

### Step 3 — Call Knox cloud services APIs with access token

Access token obtained in [Step 2 – Get access and refresh tokens](#) can be used to call Knox cloud services APIs on behalf of an end user, who has provided the authorization consent.

For example, a cURL request for Knox Asset Intelligence:

```
curl --location 'https://api.samsungknox.com/kai/v1/settings' \  
--header 'Authorization: Bearer aB1c....D2fgh'
```

A sample response for the Knox Asset Intelligence request above:

```
{  
  "battery":  
    { "batteryLevelThresholds": [] },  
  "enrollment":  
    { "allowEnrolledToKnoxConfigure": false }  
}
```

### End user with Knox MSP account

As a Knox MSP account holder, when you make Knox cloud service API requests, include your managed customer ID in x-wsm-managed-tenantid header. Your managed customer ID is available on the **Customers** page of the Knox MSP Portal, and you can also retrieve it using [Get customer information](#) endpoints.

For example, a cURL request for Knox Asset Intelligence:

```
curl --location 'https://api.samsungknox.com/kai/v1/settings' \  
--header 'Authorization: Bearer aB1c....D2fgh' \  
--header 'x-wsm-managed-tenantid: 1123123123'
```

A sample response for the Knox Asset Intelligence request above::

```
{  
  "battery":
```

```
{ "batteryLevelThresholds": [] },  
"enrollment":  
  { "allowEnrolledToKnoxConfigure": false }  
}
```

Knox cloud services API requests only available for your managed customers. If the request is for a non-managed customer, it fails.

## Supported services

The Knox cloud services which currently support OAuth 2.0 are:

- Knox Asset Intelligence
- Knox Configure
- Knox Deployment Program
- Knox E-FOTA
- Knox Manage (exclusive to MSP partners)
- Knox Mobile Enrollment
- Knox MSP Portal
- Knox Webhook Notification

## Step 4 — Refresh access token

In case your access token expires, then you can use the refresh token obtained in [Step 2 –Get access and refresh tokens](#) to get another access token to call the Knox cloud services.

Knox authorization server also issues a new refresh token as a result of this request, and the old refresh token is discarded.

When the access token expires, you can request a new access token using the current refresh token:

```
curl --location 'https://api.samsungknox.com/ams/v1/oauth2/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'client_id={OAUTH2_CLIENT_ID}' \
--data-urlencode 'client_secret={OAUTH2_CLIENT_SECRET}' \
--data-urlencode 'refresh_token={CURRENT_REFRESH_TOKEN}'
```

You receive new access and refresh tokens in the below format:

```
{
  "access_token": "abCdeFg..HiJKlM2o",
  "refresh_token": "aBCl1..DE2fgHI",
  "scope": "kai",
  "token_type": "Bearer",
  "expires_in": 599
}
```

Make sure to store the new refresh token for future use.

## Revoke access token

If you no longer require access to end user's Knox cloud services resources, you can revoke your access or refresh tokens. For more information, see the [POST /oauth2/revoke](#) operation.

For example:

```
curl --location 'https://api.samsungknox.com/ams/v1/oauth2/revoke' \
--header 'Accept: application/json' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=a1b23456c7' \
--data-urlencode 'client_secret=Ab1...DEfG' \
--data-urlencode 'token=A1B...cDEF'
```

## Manage registered apps

### Get the client ID and client secret for a registered app

To get the client ID and client secret for a registered app:

1. In the [Knox Developer Portal](#), go to **WEB INTEGRATIONS > Cloud Apps** from the landing page, or click **Cloud Apps** on the Knox Developer Portal navigation pane.
2. Search for your app, or locate it in the **APP NAME** column,
  - You can view the client ID in the **CLIENT ID** column.
  - You can view the client secret in the **CLIENT SECRET** column.

You can view the client secret value only if it's created in the current session.

If created in any of the previous sessions, you can't view the client secret value because it isn't stored for security reasons. In this case, you have to [rotate the client secret](#) to get a new one.

### Rotate the client secret for your registered app


To rotate the client secret for a registered app:

1. In the [Knox Developer Portal](#), go to **WEB INTEGRATIONS > Cloud Apps** from the landing page, or go to **Cloud Apps** on the navigation pane.
2. Follow one of the options:
  - If the client secret wasn't created or rotated in the current session — find your registered app in the **APP NAME** column, and click on the gear icon in the corresponding **CLIENT SECRET** field.

This method doesn't work if the client secret is created or rotated in the current session.

### Cloud Apps

Apps that can access Knox cloud services through APIs are listed here. [Learn more](#)

Search for full email (case sensitive), app name, or client ID							REGISTER NEW APP
APP NAME	CLIENT ID	CLIENT SECRET	CREATED BY	CREATED ON	MODIFIED ON		
Sample App		*****	r.	@samsung.com	06/14/2024	06/19/2024	

- If the client secret is created or rotated in the current session — click the name of your registered app in the APP NAME column. App Details page opens.



## < App Details

### CREDENTIALS

Client ID

rlggMl-00bmeXvg5bPwklhNm4H0tEaylJm0

Secret ⓘ [Rotate Secret](#)

\*\*\*\*\* Created on Jun 13, 2024 at 5:54PM

3. Click **ROTATE SECRET** to get a new client secret for your app.

This action can't be undone.

### Rotate Secret



⚠ Warning: this action cannot be undone.

To generate a new secret for your app, click **ROTATE SECRET**.

CANCEL

ROTATE SECRET

4. The **Client Credentials** page opens, where you can view and copy the new rotated client secret from the **Your Client Secret** field.

### Client Credentials

Make sure you save the client ID and secret in a safe place.

For security reasons, we only store your client secret until your session ends. If you lose your client secret afterwards, you can rotate it to get a new one.

Your Client ID

rlggMl-00bmeXvg5bPwklhNm4H0tEaylJm0

Your Client Secret

\*\*\*\*\*

BACK

## Update a registered app

To modify the attributes of your registered app:

1. In the [Knox Developer portal](#), go to **WEB INTEGRATIONS > Cloud Apps** from the landing page, or go to **Cloud Apps** on the navigation pane.
2. In the **APP NAME** column, click an app to modify its attributes.
3. In the **App details** page, modify the attributes. You can modify **BASIC INFORMATION, GRANT TYPES, CONSENT SCREEN, EXPRATION TIME,** and **SCOPE**.
4. Click **SAVE CHANGES**.

## Delete a registered app

To delete your registered app:

1. In the [Knox Developer portal](#), go to **WEB INTEGRATIONS > Cloud Apps** from the landing page, or click on **Cloud Apps** in the navigation pane.
2. In the **APP NAME** column, click the app you want to delete.
3. Scroll down on the **App details** page, and click **DELETE APP**.
4. **Delete confirmation** pop-up appears. To confirm deletion, type the name of the app you want to delete and click **DELETE APP**.

This action can't be undone.

### Discard changes?



 **Warning: this action cannot be undone.**

If you delete this app, all data associated with Sample App will be permanently erased from the Knox developer portal.

Type Sample App to confirm

CANCEL

DELETE APP

## Scopes for Knox cloud services APIs

OAuth 2.0 scopes are required to access the Knox cloud services APIs. These scopes are service-specific, and the level of access you require for your app determines which scopes to select. For more information about the scope requirements, you can refer to the API specifications of its associated Knox cloud service.

The following scopes are available:

### Knox Configure

Scope name	Description	Applicable API endpoints
kc	Manage all Knox Configure functionality.	All endpoints in the <a href="#">Knox Configure API v2.0</a> .
kc.devices	Manage all functionality related to Knox Configure devices.	All endpoints in the <a href="#">Knox Configure API v2.0</a> .
kc.devices:view	View Knox Configure devices and their information.	POST /kc/v2/devices/getDevices
		POST /kc/v2/devices/getDeviceLogs
kc.devices:manage	Send commands to devices and delete them from Knox Configure.	POST /kc/v2/devices/sendCommand
		POST /kc/v2/devices/bulkDelete

For detailed API specification, see [Knox Deployment Program API reference](#).

### Knox Deployment Program

Scope name	Description	Applicable API endpoints
kdp	Manage all Knox Deployment Program functionality.	All endpoints in <a href="#">Knox Deployment Program API</a> .
kdp.devices	Upload and delete devices, and get device information.	PUT /kcs/v1/rp/devices/upload
		GET /kcs/v1/rp/devices/status
		GET /kcs/v1/rp/devices

		PUT /kcs/v1/rp/devices/delete
kdp.devices:view	View devices and device statuses.	GET /kcs/v1/rp/devices/status
		GET /kcs/v1/rp/devices
kdp.devices:manage	Upload and delete devices.	PUT /kcs/v1/rp/devices/upload
		PUT /kcs/v1/rp/devices/delete
kdp.profilealias	View all Knox Mobile Enrollment profile aliases.	GET /kcs/v1/rp/profilealias/kme
kdp.customers	View all Knox Deployment Program customers.	GET /kcs/v1/rp/customers/list

For detailed API specification, see [Knox Deployment Program API reference](#).

## Knox E-FOTA

Scope	Description	Applicable API endpoints
ke	Manage all Knox E-FOTA functionality.	All endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.campaign	View, assign, create, delete, and cancel Knox E-FOTA campaigns.	All campaign endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.campaign:view	View Knox E-FOTA campaign information.	GET /campaigns
		GET /campaigns/{campaignId}
		GET /campaignSchemas
ke.campaign:assign	Assign or unassign devices from a Knox E-FOTA campaign.	POST /campaigns/{campaignId}/bulkAssign
		POST /campaigns/{campaignId}/bulkUnassign
ke.campaign:manage		POST /campaigns

	Create and edit Knox E-FOTA campaigns.	PUT /campaigns/{campaignId}
ke.campaign:delete	Delete and cancel Knox E-FOTA campaigns.	DELETE /campaigns/{campaignId}
		PUT /campaigns/{campaignId}/cancel
ke.devices	View, upload, manage, and delete Knox E-FOTA devices.	All device endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.devices:view	View Knox E-FOTA device information.	GET /devices/models
		GET /devices/salesCodes
		GET /devices/aggregateDeviceUpdateStatus
		GET /devices/csc
		POST /devices/getDevices
ke.devices:manage	Upload, refresh, and unenroll Knox E-FOTA devices.	POST /devices/bulkUpload
		POST /devices/bulkRefresh
		POST /devices/bulkUnenroll
ke.devices:delete	Delete devices from Knox E-FOTA.	POST /devices/bulkDelete
ke.licenses	View, register, and delete Knox E-FOTA licenses.	All license endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.licenses:view	View Knox E-FOTA license information.	GET /licenses
ke.licenses:manage	Register commercial licenses and generate trial licenses for Knox E-FOTA.	POST /licenses
		POST /trialLicenses
ke.licenses:delete	Delete Knox E-FOTA licenses.	DELETE /licenses/{licenseId}

ke.fota:view	View Knox E-FOTA firmware information.	GET /fota
ke.privacyPolicy	View and manage the skip privacy policy setting.	All privacy policy endpoints in the <a href="#">Knox E-FOTA API</a>
ke.privacyPolicy:view	View the skip privacy policy setting.	GET /privacyPolicy
ke.privacyPolicy:manage	Manage the skip privacy policy setting.	PUT /privacyPolicy

### Knox Manage (exclusive to MSP partners)

Scope name	Description	Applicable API endpoints
km	Manage all Knox Manage functionality.	All endpoints in <a href="#">Knox Manage API</a> .
km.group	View, create, and add users to groups.	POST /km/v1/group/insertGroup
		POST /km/v1/group/insertGroupUnits
		GET /km/v1/group/selectGroups
km.group:view	View groups and their information.	GET /km/v1/group/selectGroups
km.group:manage	Update groups and their information.	POST /km/v1/group/insertGroup
		POST /km/v1/group/insertGroupUnits
km.user	View and create users, and request enrollment through email.	POST /km/v1/user/createUser
		POST /km/v1/user/requestEnrollment
		GET /km/v1/user/selectUserWithID
km.user:view	View users.	GET /km/v1/user/selectUserWithID
km.user:manage	Create users and request enrollment through email.	POST /km/v1/user/createUser
		POST /km/v1/user/requestEnrollment
km.profile	Assign profiles to groups.	POST /km/v1/mdm/profileServiceWrapper/assign

For detailed API specification, see [Knox Manage API reference](#).

The Knox Manage scopes are intended exclusively for use by MSP partners, direct use by end customers is currently not supported.

## Knox Mobile Enrollment

Scope name	Description	Applicable API endpoints
kme	Manage all Knox Mobile Enrollment functionality, including profiles, devices, and resellers.	All endpoints in <a href="#">Knox Mobile Enrollment API</a> .
kme.profiles	Manage all functionality related to Knox Mobile Enrollment profiles.	All profile management endpoints in <a href="#">Knox Mobile Enrollment API</a> .
kme.profiles:view	View Knox Mobile Enrollment profiles.	GET /kcs/v1/kme/profiles/list
		GET /kcs/v1/kme/profiles/status
		GET /kcs/v1/kme/profiles/{id}/get
kme.profiles:manage	Create, update, and delete Knox Mobile Enrollment profiles.	POST /kcs/v1/kme/profiles/create
		POST /kcs/v1/kme/profiles/createAsync/
		PUT /kcs/v1/kme/profiles/{profileId}
		DELETE /kcs/v1/kme/profiles/{id}
kme.devices	Manage all functionality related to devices in Knox Mobile Enrollment.	PUT /kcs/v1/kme/devices/assignProfile
		POST /kcs/v1/kme/devices/delete
		GET /kcs/v1/kme/devices/list
		PUT /kcs/v1/kme/devices/unassignProfile
		PUT /kcs/v2/kme/devices/unassignProfile
		POST /kcs/v1/kme/devices/uploads/approvals
		GET /kcs/v1/kme/devices/uploads/list
kme.devices:view		GET /kcs/v1/kme/devices/list

	View Knox Mobile Enrollment device information.	GET /kcs/v1/kme/devices/uploads/list
kme.devices:manage	Assign, unassign, and delete profiles, as well as approve device uploads.	PUT /kcs/v1/kme/devices/assignProfile
		POST /kcs/v1/kme/devices/delete
		PUT /kcs/v1/kme/devices/unassignProfile
		PUT /kcs/v2/kme/devices/unassignProfile
		POST /kcs/v1/kme/devices/uploads/approvals
kme.reseller	Manage all functionality related to Knox Mobile Enrollment resellers.	POST /kcs/v1/kme/reseller/approvals
		POST /kcs/v1/kme/reseller/profilealias
		PUT /kcs/v1/kme/reseller/profilealias/{profileAliasId}
		DELETE /kcs/v1/kme/reseller/profilealias/{profileAliasId}

For detailed API specification, see [Knox Mobile Enrollment API reference](#).

## Knox MSP Portal

Scope name	Description	Applicable API endpoints
msp	Manage all Knox MSP Portal functionality.	POST /msp/v1/managedCustomers
		POST /msp/v1/managedCustomers/link
		POST /msp/v1/managedCustomers/delink
		GET /msp/v1/managedCustomers
		GET /msp/v1/managedCustomers/{customerId}
		PUT /msp/v1/managedCustomers/{customerId}
		GET /msp/v1/profiles
		POST /msp/v1/profiles/copy
		POST /msp/v1/profiles/overwrite



msp.customers	Add and delink MSP customers, as well as view and edit their information.	POST /msp/v1/managedCustomers
		POST /msp/v1/managedCustomers/link
		POST /msp/v1/managedCustomers/delink
		GET /msp/v1/managedCustomers
		GET /msp/v1/managedCustomers/{customerId}
		PUT /msp/v1/managedCustomers/{customerId}
msp.profiles	Copy existing profiles to managed customers.	GET /msp/v1/profiles
		POST /msp/v1/profiles/copy
		POST /msp/v1/profiles/overwrite

For detailed API specification, see [Knox MSP Portal API reference](#).

## Knox E-FOTA

Scope	Description	Applicable API endpoints
ke	Manage all Knox E-FOTA functionality.	All endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.campaign	View, assign, create, delete, and cancel Knox E-FOTA campaigns.	All campaign endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.campaign:view	View Knox E-FOTA campaign information.	GET /campaigns
		GET /campaigns/{campaignId}
		GET /campaignSchemas
ke.campaign:assign	Assign or unassign devices from a Knox E-FOTA campaign.	POST /campaigns/{campaignId}/bulkAssign
		POST /campaigns/{campaignId}/bulkUnassign
ke.campaign:manage	Create and edit Knox E-FOTA campaigns.	POST /campaigns
		PUT /campaigns/{campaignId}

ke.campaign:delete	Delete and cancel Knox E-FOTA campaigns.	DELETE /campaigns/{campaignId}
		PUT /campaigns/{campaignId}/cancel
ke.devices	View, upload, manage, and delete Knox E-FOTA devices.	All device endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.devices:view	View Knox E-FOTA device information.	GET /devices/models
		GET /devices/salesCodes
		GET /devices/aggregateDeviceUpdateStatus
		GET /devices/csc
		POST /devices/getDevices
ke.devices:manage	Upload, refresh, and unenroll Knox E-FOTA devices.	POST /devices/bulkUpload
		POST /devices/bulkRefresh
		POST /devices/bulkUnenroll
ke.devices:delete	Delete devices from Knox E-FOTA.	POST /devices/bulkDelete
ke.licenses	View, register, and delete Knox E-FOTA licenses.	All license endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.licenses:view	View Knox E-FOTA license information.	GET /licenses
ke.licenses:manage	Register commercial licenses and generate trial licenses for Knox E-FOTA.	POST /licenses
		POST /trialLicenses
ke.licenses:delete	Delete Knox E-FOTA licenses.	DELETE /licenses/{licenseId}
ke.fota:view	View Knox E-FOTA firmware information.	GET /fota

ke.privacyPolicy	View and manage the skip privacy policy setting.	All privacy policy endpoints in the <a href="#">Knox E-FOTA API</a> .
ke.privacyPolicy:view	View the skip privacy policy setting.	GET /privacyPolicy
ke.privacyPolicy:manage	Manage the skip privacy policy setting.	PUT /privacyPolicy

### Knox OAuth 2.0 Authentication — Authorization server operations

Scope name	Description	Applicable API endpoints
email	Use your email address to verify your identity.	POST /ams/v1/oauth2/token
openid	Use your Samsung Knox user identifier.	POST /ams/v1/oauth2/token

For detailed API specification, see [Knox OAuth 2.0 Authentication API reference](#).