

Cisco Spaces SDK Integration

This chapter provides information on the integration of **Cisco Spaces Software Development Kit** (SDK).

- Cisco Spaces SDK Integration, on page 1
- Integrating Cisco Spaces, on page 1

Cisco Spaces SDK Integration

The **Cisco Spaces Software Development Kit** (SDK) leverages OpenRoaming technology to attach users, seamlessly and securely, to Wi-Fi networks, without the need for user interaction. The Cisco Spaces SDK allows an iOS or Android application developer to configure iOS and Android devices with an identity of choice that can be verified with the back-end system. The Cisco Spaces SDK also allows the developer to add more information about the users, and engage with them, directly on their device, through the iOS and Android notification framework.

The SDK configuration section is accessible through **Menu** icon ( ) > **Configure** > **Cisco Spaces SDK**. This allows the customers to register their native app (iOS and/or Android) with Cisco Spaces.

Integrating Cisco Spaces

The Cisco Spaces Software Development Kit (SDK) is a platform-independent SDK (Software Development Kit), across iOS and Android platforms, that Cisco Spaces customers can use to enable Apps to provision and manage OpenRoaming user profiles in a mobile device. For more information, see Cisco Spaces SDK Developer Documentation.

Before you begin

Following Google's deprecation of API key-based functionality, a new method using Firebase JSON through Google Cloud Messaging (GCM) is introduced in the Cisco Spaces Software Development Kit (SDK) (**Cisco Spaces Dashboard** > **Configure** > **Cisco Spaces SDK**).

Before this, in Cisco Spaces Dashboard, you had to configure the API Key, but now it has changed to Firebase JSON.

Cisco Spaces utilizes this to trigger push notifications on Android devices using the SDK.

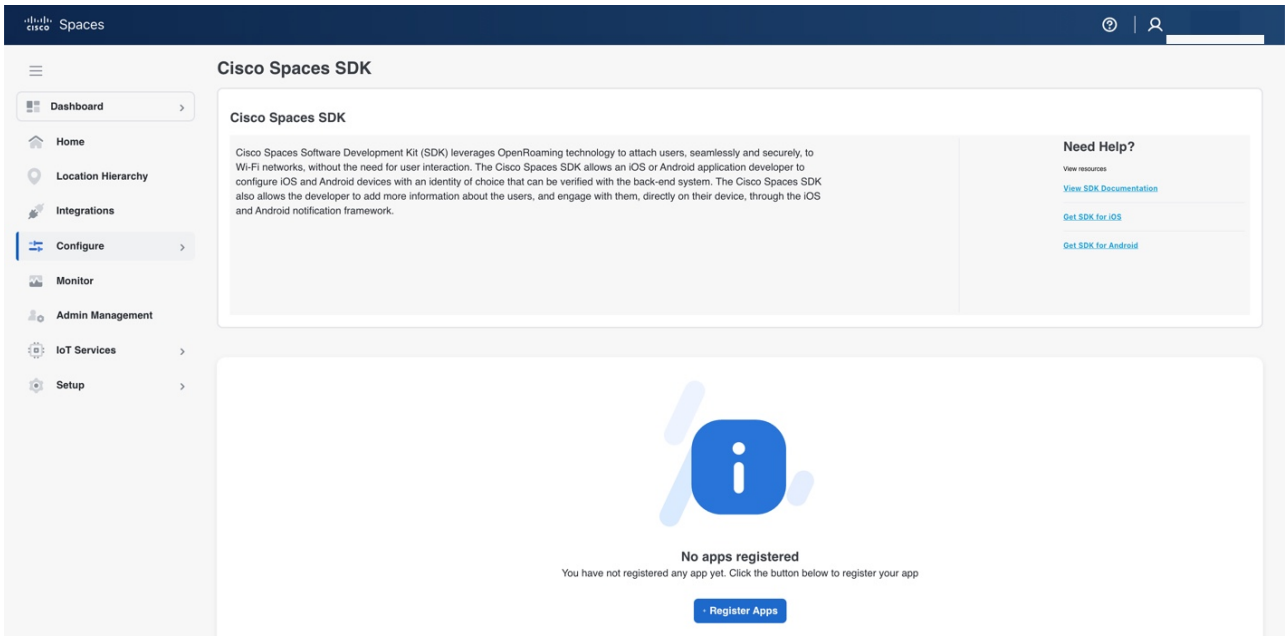
Procedure

Step 1 Log in to Cisco Spaces.

The Cisco Spaces **Home** window is displayed.

Step 2 From the top-left corner, click the **Menu** icon ( ) and choose **Configure** > **Cisco Spaces SDK**.

Figure 1: Cisco Spaces SDK



The **Cisco Spaces SDK** window is displayed.

Step 3 Click **Register App**.

Step 4 To choose the platform for the new app, check either the **iOS** or **Android** check box or both.

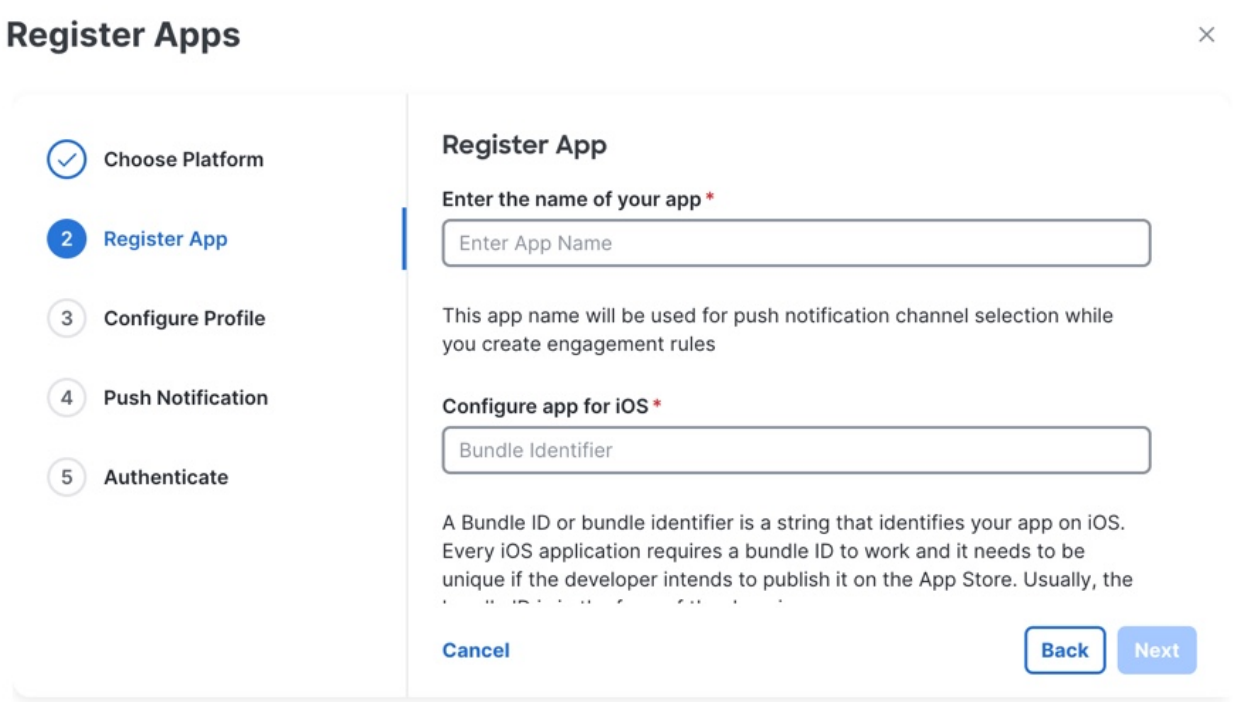
If you select both platforms, the subsequent windows display parameters for both platforms.

Step 5 Click **Next**.

To successfully register a new app, enter information in the following sections:

- Register App
- Configure Profile
- Push Notification
- Authenticate

Figure 2: Register App Sections



Step 6 In the **Register App** section, enter the following:

- **App Name**: Enter the name of the application.
- **Bundle Identifier**: Enter the Bundle ID or bundle identifier string that identifies your app on the iOS platform.

Note Every iOS application requires a bundle ID to work and must be unique if the developer intends to publish it on the App Store. The bundle ID is in the format `domain.your-company.app-name`.

- **Package Name**: Enter the unique package name to identify an Android app.

Note The package name of an app is in the format `domain.your-company.app-name`. However, you can choose to enter any name.

Step 7 Click **Next**.

Step 8 In the **Configure Profile** section, enter the following:

- **Displayed Operator Name**: Enter a valid identifier. This name is displayed as the Wi-Fi SSID name on the user's android or iOS device.
- **Domain**: Enter a unique domain name for the profile.
  - The domain name is available in the OpenRoaming app. For more information see, Configure Network Controller.
  - You should add this domain name in the controller configuration.
  - If you have more than one app, then each app must have a unique domain name.
- **Roaming Consortium OIs**: From the **Roaming Consortium OIs** drop-down list, check the check box next to the array of Roaming Consortium Organization (RCO) identifiers. This is optional.

Step 9 To enable push notifications for iOS, check the **Enable Push Notification for iOS** check box.

a) Enter the iOS App ID.
b) Click **Upload** to browse and upload the APNS P12 and Certificate.
c) Enter the APNS certificate password.

Step 10 To enable push notifications for Android, check the **Enable Push Notification for Android** check box.

a) Enter the Android App ID.
b) Enter the API key.

Step 11 Click **Next**.

Step 12 To support Apple sign-in as the user identity for the new mobile app, check **Enable Apple Sign In** in the **Authenticate** section.

a) In the **Enter Client ID** field, enter the apple account sign-in client ID.
b) In the **Enter Secret Key** field, enter the secret key for Apple account.

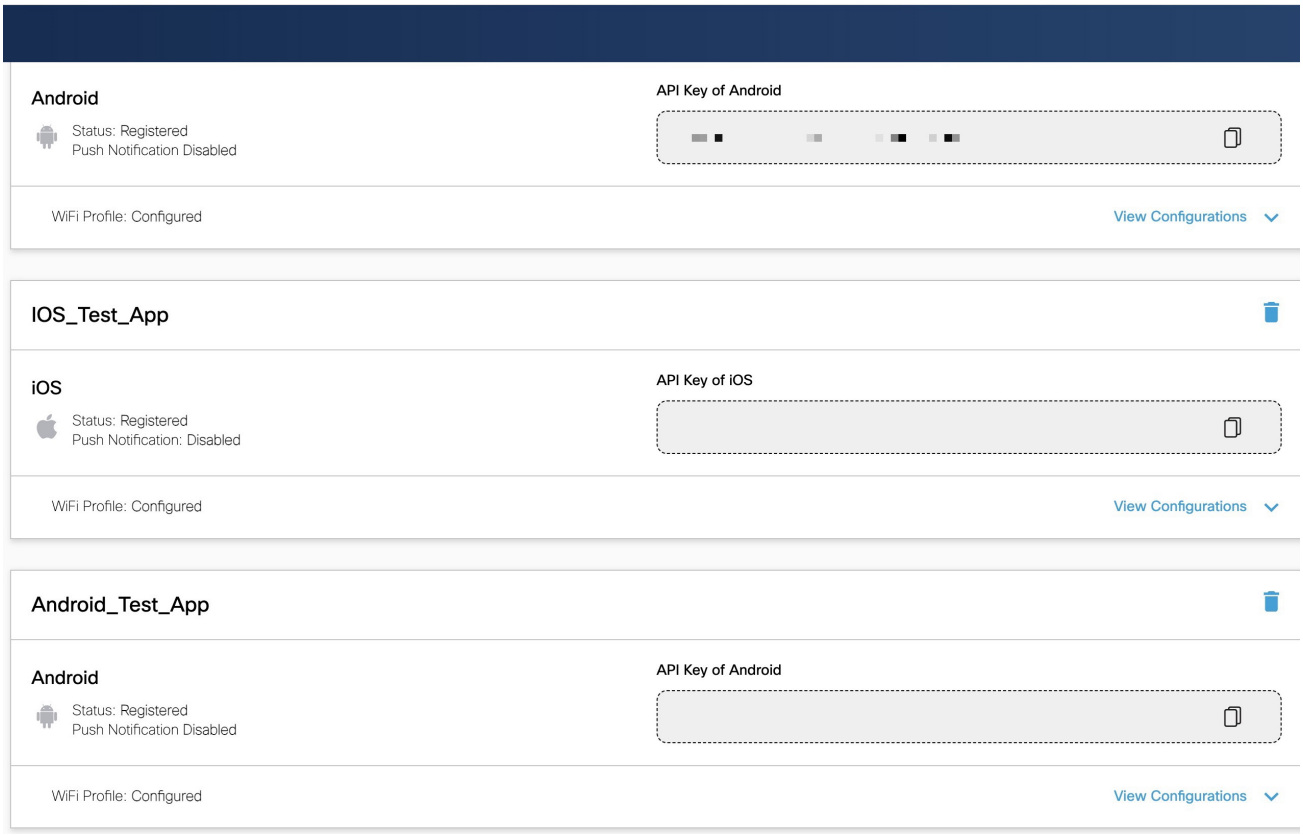
Step 13 To support Google sign-in as the user identity for the new mobile app, check **Enable Google Sign In** in the **Authenticate** section.

a) In the **Enter Client ID** field, enter the google account sign-in client ID.
b) In the **Enter Secret Key** field, enter the secret key for Google account.

Step 14 Click **Register App** to complete the app registration.

The registered apps are displayed.

Figure 3: Registered Android and iOS Apps



You can click:

- **View Configurations**: To view the application configuration details.
- **Delete** icon: To delete the registered application.

Step 15 (Optional) Click **Edit** to update push notifications for iOS and Android platforms.

Step 16 (Optional) Click **Update**.