🏠 / Cisco Platforms and Development / Cisco SDKs

Cisco SDKs

8.2.1

What is an SDK?   🔖

SDK stands for Software Development Kit. Typically an SDK contains a set of software development tools integrated for developing applications for a specific device or system.

From an end user's point of view, an SDK is quite different from an API. Most SDKs are a package, integrated with libraries, documents, code examples, etc. Most SDKs require installation. In comparison, an API is essentially a documented set of URIs. Developers require only a reference guide and a resource address to get started.

For example, if you want to develop a mapping application for an iPhone, you need to download and install the iOS SDK. In contrast, to try the Google Map API, you need only an API reference and require initial authentication credentials.

An SDK can provide simpler authentication methods as well as enabling token refreshes as part of the package. SDKs often help with pagination or rate limiting constraints on responses for a particular API. Also, it can be easier to read examples in a programming language you are already familiar with, so code examples in an SDK can be helpful.

Cisco SDKs

Cisco provides a wide range of SDKs on different Cisco platforms. This list is not an exhaustive list:

- **Webex Teams Android SDK –** Use the Webex Teams Android SDK to customize your app and to access powerful Webex Teams collaboration features without making your users leave the mobile app. See Webex Teams Android SDK.
- **Jabber Web SDK –** This SDK is used for developing web applications on Cisco Unified Communications, including voice and video, IM and Presence, voice messaging, and conferencing. See Jabber Web SDK.
- **Jabber Guest SDK for Web –** The Cisco Jabber Guest SDK for Web is primarily a call widget that is embedded in an iframe within another web page. See Jabber Guest SDK for Web.
- **Jabber Guest SDK for iOS –** The Jabber Guest SDK for iOS coordinates and simplifies the implementation, use, and quality of two-way video calls from within an application. See Cisco Jabber Guest for iOS.

☰   ılıılı   DevNet Associate   v1.0
     CISCO

aspects of establishing and maintaining the two-way video call within your application. See Cisco Jabber Guest for Android.

- **Cisco DNA Center Multivendor SDK –** For partners and customers who have a mix of Cisco and non-Cisco devices in their network, this SDK builds support directly in Cisco DNA Center. See Cisco DNA Center Multivendor SDK.
- **UCS Python SDK –** The Cisco UCS Python SDK is a Python module used to automate all aspects of Cisco UCS management, including server, network, storage, and hypervisor management. See Official Documentation for the UCS Python SDK.
- **Cisco APIC Python SDK (Cobra SDK) –** The Cobra SDK is a native Python language binding for the APIC REST API to interact with the APIC controller. The installation of Cobra requires installing two .egg files: `acicobra.egg` and `acimodel.egg`. See Installing the Cisco APIC Python SDK.
- **Cisco IMC Python SDK –** Cisco IMC Python SDK is a Python module supporting the automation of all Cisco IMC management servers (C-Series and E-Series). See Instructions for installing IMC Python SDK and Official Documentation for the IMC Python SDK.
- **Cisco Instant Connect SDK –** Cisco Instant Connect has an Android, Windows, and Apple iOS Software Development Kit, providing tools for partners to embed Push-To-Talk in mobile or desktop applications. See Cisco Instant Connect SDK Downloads and Introducing Cisco Instant Connect.
- **Webex Teams Python SDK example –** You can work with the Webex Teams APIs using a familiar language, in this case Python, with the `webexteamssdk`. This SDK is available on GitHub. This SDK handles pagination for you automatically, simplifies authentication, provides built-in error reporting, and manages file attachments. Here is a simple Python example to retrieve a user's account information:

```
from webexteamssdk import WebexTeamsAPI, ApiError
access_token = 'jY1...e10f'
api = WebexTeamsAPI(access_token=access_token)
try:
    me = api.people.me()
    print(me.emails)
except ApiError as e:
    print(e)
```

The output you'd see from this script is a line with the email address or addresses for the account related to the `access_token`. If something goes wrong, the `try... except` block returns a readable error.

A couple of notes about this `webexteamssdk` example:

- The SDK needs a Webex Teams access token to make API calls. You can either set it with a `WEBEX_TEAMS_ACCESS_TOKEN` environment variable, or by putting the `access_token` argument into the function call. This example uses an `access_token` variable and then passes it in as an argument.
- The SDK provides error reporting with the `ApiError` exception. Notice that `ApiError` imported in the first line.

The SDK gives you access to precise parts of the return body, such as `me.emails`. In the example above it returns only the `emails` data from the response JSON.