Introduction to Cisco Spark Apps

Duration: 15 minutes

Objectives

In this lab, we will take a few steps back to present Cisco Spark's extensibility "Big Picture",

As we introduce Cisco Spark API, you will use the interactive documentation to experience Cisco Spark programmability. We'll then go through a few use-cases, and drill into the fundamentals of Cisco Spark Apps.

After this lab, you should have enough knowledge to easily navigate among Cisco Spark Learning Labs, as well as broader technical resources such as "Spark for Developers" and DevNet 's "Cisco Spark Community of Interest".

How to setup your own computer

This lab can be completed from any platform with an HTML5-compatible browser.

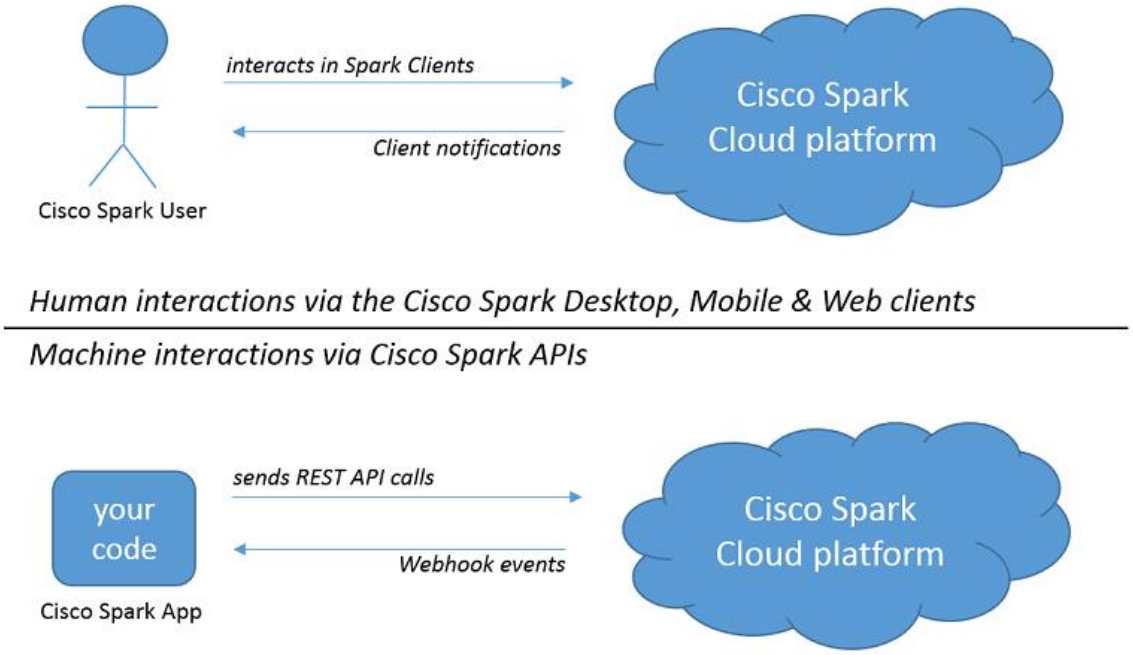
Pre-requisites

You will need a Cisco Spark user account to complete this lab. If you're not a Cisco Spark user yet, click to sign up.

Step 1: about Cisco Spark's extensibility

Extending Cisco Spark lets you create applications that:

- automate tasks: such as creating rooms, posting messages, adding participants to an existing room,
- take actions as events happen in Cisco Spark: such as participants being added to rooms, or new messages being created.



In order to automate tasks, custom applications leverage the Cisco Spark REST API. The API lets you interact with Cisco Spark's main concepts:

- Rooms: create, update or delete Rooms,
- Teams: create, update or delete Teams,
- People: look for Cisco Spark users,
- Messages: create or delete Messages.
- Memberships & Team Memberships: add, remove participants in Rooms & Teams, and promote participants as moderators,

Moreover, your application can register Webhooks to be notified when a new message is posted.

Note that the documentation pointed above lets you interact with the Cisco Spark API straight from your Web browser. As we leverage the interactive documentation extensively along these labs, we will give it a try right now. Moreover, it will be a great opportunity to make these extensibility concepts very concrete to you!

Step 2: Meet the Cisco Spark API documentation

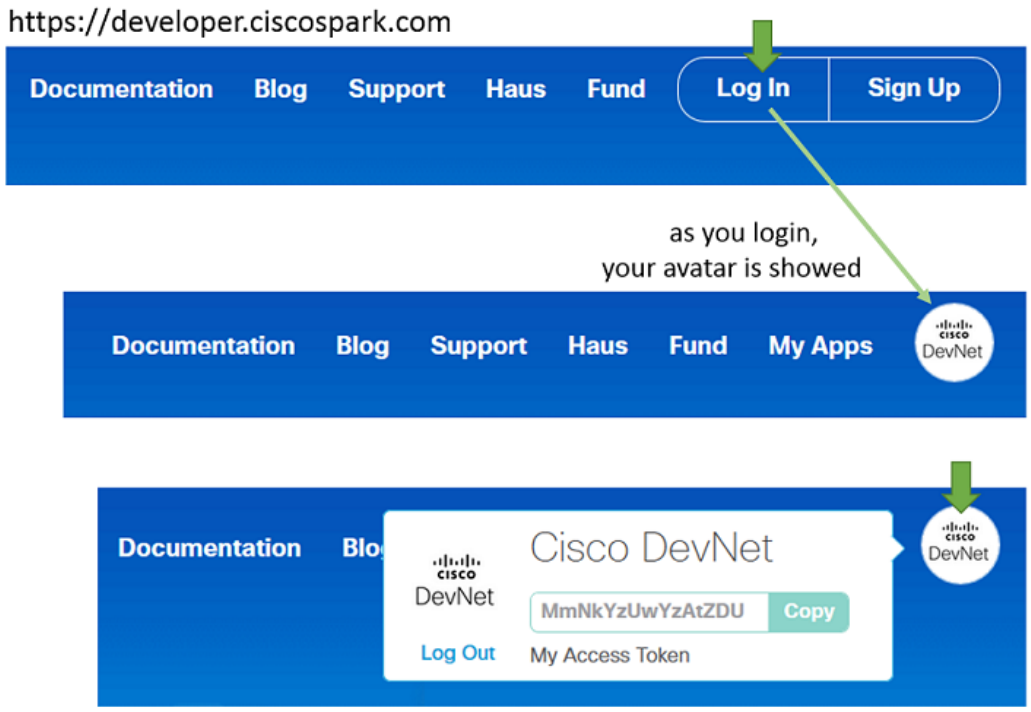
"Spark for Developers" is the home of Cisco Spark REST API documentation. It can be reached at https://developer.ciscospark.com.

Open "Spark for Developers".

Click "Log In" in the upper right corner and enter your Cisco Spark credentials. If you don't already have a Cisco Spark account, click "Sign up".

Once logged in you'll see your avatar displayed in the upper right corner.

Click on your picture to reveal your personal secret. This secret is referred as the "Cisco Spark API Access Token", or "Developer Access Token" when we refer to your personal secret.



Each Cisco Spark API Access Token is unique and acts under the identity of its owner. As such your Developer Access Token is "another you".

If one of your access tokens ever gets revealed, contact the Cisco Spark API support to have it revoked. Simply copy paste the token you want to revoke, and email it to devsupport@ciscospark.com.

Click on Documentation, and look for the "API Reference" entry on the left page.

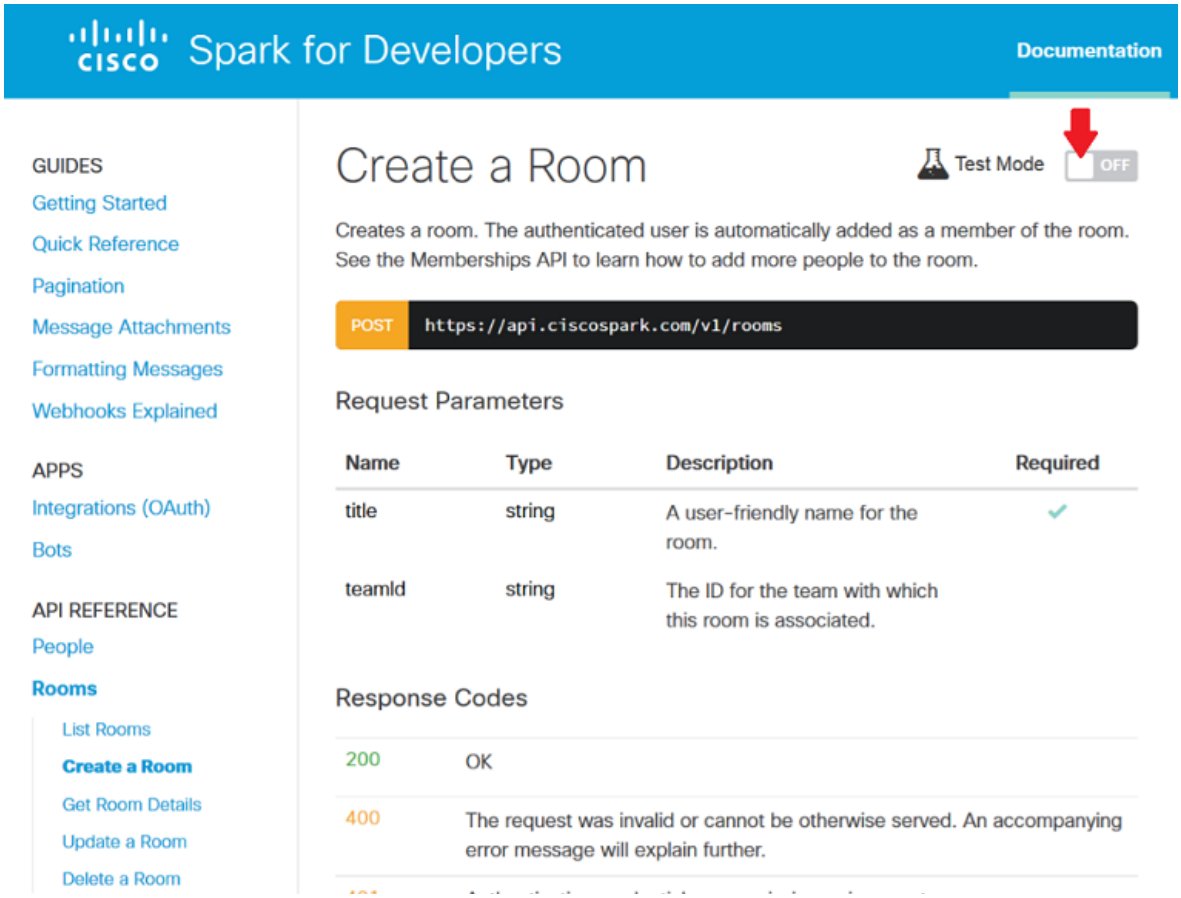
Discover the various Cisco Spark concepts that you can interact with from the API: People, Rooms, Memberships, Messages, Teams… In the next step, you'll learn to create a Room straight from the documentation.

Step 3: Your first request to the Cisco Spark API

Choose the Rooms resource under the API Reference, and click “POST” to access the wizard to create a new room. These actions should drive you to this page.

First of all, we will activate the documentation's interactive mode.

Click the “Test mode” toggle as displayed by the red arrow in the figure below.

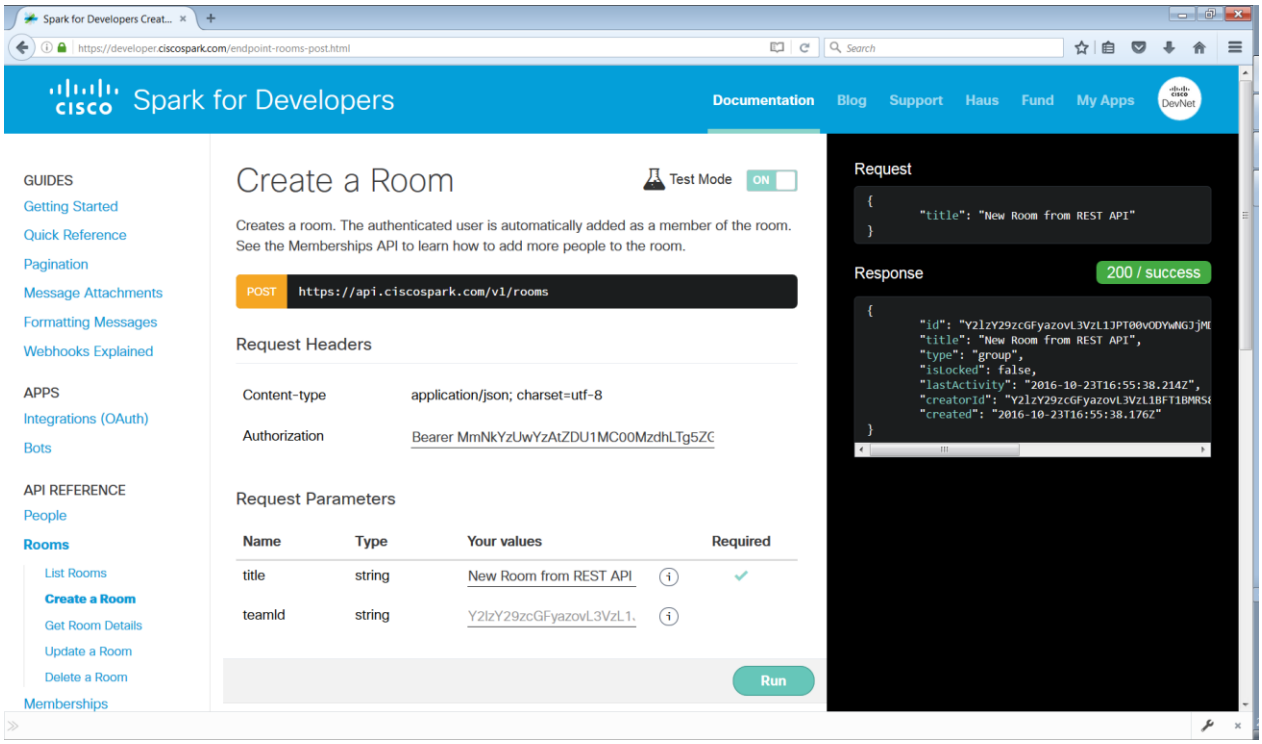


You are now able to edit the title of the Room.

Specify a title of your choice and click Run.

The right pane now displays the request sent to, and the response received from the Cisco Spark API.

Note that the request is executed **on your behalf** because your personal token got injected in the “Authorization” Request Headers. This happened as you toggled to "Test mode".



Let's take a minute to examine the response.

First, notice that a "200 / success" is displayed in green: this is referred as the HTTP status code in the REST API terminology. It gives you instant information about the success or failure of your API call. The main HTTP status codes are "2xx" for success, "4xx" for client calls error and "5xx" for server errors. If you scroll down the page, the documentation shows the various HTTP status code you may encounter.

Second, notice that the response is filled up with information about the room that just got created. This JSON payload mentions an "id" property. This is the unique identifier of the room which just got created.

JSON payload mentions an "id" property. This is the unique identifier of the Room which just got created.

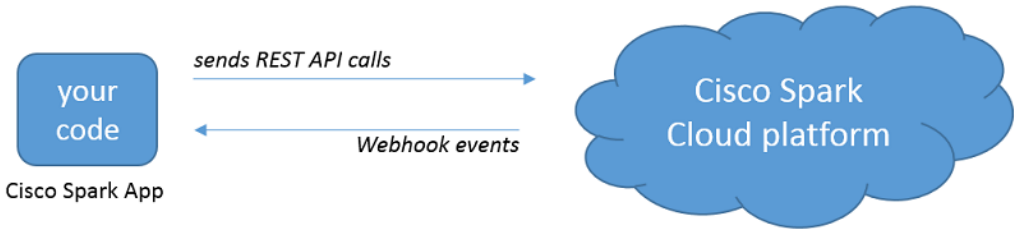
Now, open a Cisco Spark client, and see it by yourself: the new Room showed up!

In future labs, we'll spend more time on the various REST API calls proposed by Cisco Spark.

For now, let's go through some of the potential of Cisco Spark extensibility: meet the Cisco Spark Apps...

Step 4: Cisco Spark Apps use cases

As we experienced in steps 2 and 3, Cisco Spark Apps are nothing more than code that interacts with Cisco Spark in an automated and pre-defined way.



Let's take a few minutes to imagine the different types of applications we could build:

I. a Cisco Spark App could listen to Events happening in Cisco Spark such as a new message with a specific mention, or a new participant added to a Room. As these events happen, your code could take action accordingly in order to send a SMS to raise an alert, or parse a Cisco Spark message in order to insert data into an enterprise back-end for example.

This type of application is referred as a "Controller" in Cisco Spark terminology. If a Cisco Spark App simply listens and takes no further action, we can also name it "Watcher".

II. another Cisco Spark App could send messages to a "Business Activity" Room as event happen in a Business Process, such as a new customer registers or a command is cancelled. This "Business Activity" Room will have been pre-populated with Human Agents that could check if further action needs to be taken.

Such an application is referred as a "Notifier" in Cisco Spark terminology.

III. a third Cisco Spark App could ask a customer a set of questions in order to pre-populate a form, find a schedule for a doctor appointment, or initiate a loan simulation by asking your customer very specific questions.

These applications are referred as "Interactive Assistants" in Cisco Spark terminology.

More globally in this industry, these interactions are referred as "Bots" or "Chat Bots".

Yet, when extending Cisco Spark, we prefer to speak of Applications rather than Bots. Indeed, as you will see in next chapter, Cisco Spark Bots are a concept of its own within Cisco spark, which ties to "Bot Accounts" and specific interactions in Group rooms.

How to create your own Cisco Spark App

Depending on your technical background or business goals, you are given several options to create your own Cisco Spark apps:

A. either via assembly through a visual integration tool (think of this as a "Lego" game): the approach here is to create an application with little to no coding.

In practice, you will assemble from existing building-blocks that do the actual job of connecting together to various back-end services.

Note that several 3rd party vendors propose Cloud Services, such as IFTTT, Zapier, Built.io, Gupshup, Stamplay, Workato...

These services will not only free you from the complexity of deploying your Cisco Spark app, but also provide its own set of extra benefits such as pre-configured integrations, natural language processing, voice synthesis or recognition, image analysis...

B. or by writing and deploying your own code: the Cisco Spark API lets you create an application in any language, and the community has been working on frameworks and examples to help you on this journey.

Note that deployment of your Spark App will be a dedicated activity if you want to deliver more than a prototype, with the traditional tasks around availability and security (DNS, load-balancing, firewalls, traffic monitoring, rate limitations...).

In the upcoming labs, you'll have the opportunity to either code or leverage 3rd party tools in order to create your own Cisco Spark Apps.

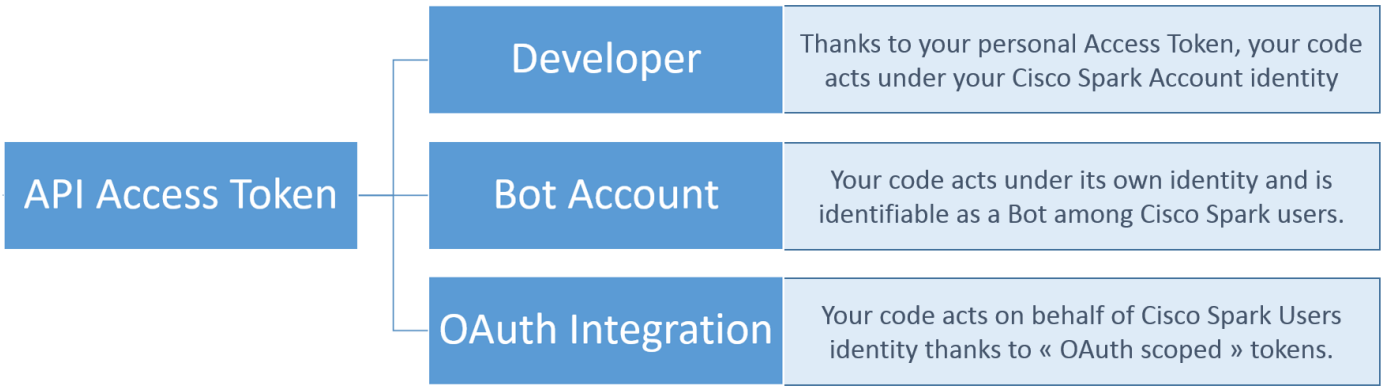
Step 5: Choosing an identity for your Cisco Spark App

As you experimented in step 3, code acts on your behalf when your personal Access Token is transmitted to Cisco Spark.

There are some situations though, where you want your application to execute under the behalf of other Cisco Spark Users. Let's illustrate this.

Suppose you're building an Out Of Office Assistant that automatically responds to messages on behalf of pre-registered Users. This Spark App will need to access all messages of these pre-registered users, so that it can check for mentions, and respond back under their identities if mentioned. Users of this Spark App will have granted the authorization for your Cisco Spark App to access their messages.

Such an application is referred to as a Cisco Spark Integration. Spark integrations leverage the OAuth Grant Flow protocol to be issued access tokens that can act under Cisco Spark users' identities. These tokens are scoped: limited to the set of authorizations granted by the Cisco Spark users themselves.



Now, suppose we want the Out Of Office Assistant to behave a little differently.

Imagine for a minute that any user could invite the assistant into Cisco Spark rooms and interact as such: "@OOF is Mandy on vacation?", or "@OOF note that I'll be out of office for the next 2 weeks".
Then, it is more like if your application would be receiving messages and answering questions under its own identity, and not on behalf of someone else.

This is where Bot Accounts come into play.

Bots are dedicated Cisco Spark accounts that act under their own identity. They can be added to Rooms like any other Cisco Spark user. Some restrictions apply though: main of which happens in Group Rooms, in which bots MUST be mentioned in order to receive messages.

In the upcoming labs, you'll have the opportunity to run, extend and create your own Cisco Spark Bots and Integrations.

Your takeaways

Cisco Spark exposes a REST API that lets you create various kind of applications: notifiers, watchers, controllers or interactive assistants.

Cisco Spark Apps are nothing more than code that interacts with Cisco Spark through an API access token. From this access token, your applications are given an identity and specific authorizations as they interact with Cisco Spark.

You can create Cisco Spark Apps either by writing code, or leveraging 3rd party Cloud Integration Services.