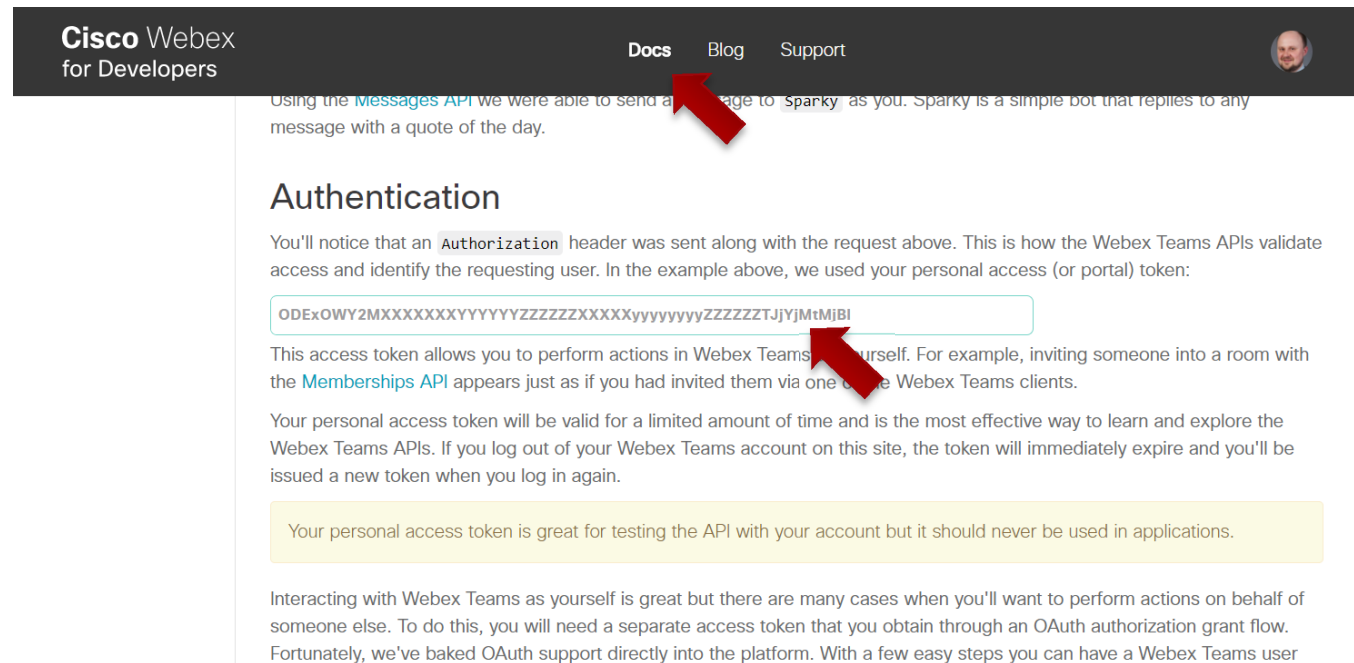


# Interact with Spark from IDLE

# Getting your access token

1. Go to <https://developer.webex.com/>
2. Log in with your Spark account
3. Go to **Docs** page
4. Scroll down the **Getting Started** page down to **Authentication** section
5. Copy your personal **access token**



Cisco Webex for Developers

Docs Blog Support

Using the [Messages API](#) we were able to send a message to "Sparky" as you. Sparky is a simple bot that replies to any message with a quote of the day.

## Authentication

You'll notice that an `Authorization` header was sent along with the request above. This is how the Webex Teams APIs validate access and identify the requesting user. In the example above, we used your personal access (or portal) token:

```
ODExOWY2MXXXXXXYYYYZZZZZZXXXXXyyyyyyyyyZZZZZZTJYjMtMjBI
```

This access token allows you to perform actions in Webex Teams as yourself. For example, inviting someone into a room with the [Memberships API](#) appears just as if you had invited them via one of the Webex Teams clients.

Your personal access token will be valid for a limited amount of time and is the most effective way to learn and explore the Webex Teams APIs. If you log out of your Webex Teams account on this site, the token will immediately expire and you'll be issued a new token when you log in again.

Your personal access token is great for testing the API with your account but it should never be used in applications.

Interacting with Webex Teams as yourself is great but there are many cases when you'll want to perform actions on behalf of someone else. To do this, you will need a separate access token that you obtain through an OAuth authorization grant flow. Fortunately, we've baked OAuth support directly into the platform. With a few easy steps you can have a Webex Teams user

# Getting the room id

1. From the **Docs** page,
2. Under **API Reference**, navigate to **Rooms – List Rooms**
3. Enable **Test Mode**
4. Click **Run**
5. Copy room **id** from the server Response

**Cisco Webex for Developers** Docs Blog Support

**WEBEX TEAMS API**

People

**Rooms**

- List Rooms
- Create a Room
- Get Room Details
- Update a Room
- Delete a Room

Memberships

Messages

Teams

Team Memberships

Webhooks

Organizations

Licenses

Roles

## List Rooms

Test Mode

List rooms.

The **title** of the room for 1-to-1 rooms will be the display name of the other person.

By default, lists rooms to which the authenticated user belongs.

Long result sets will be split into [pages](#).

**GET** `https://api.ciscospark.com/v1/rooms`

### Request Headers

Content-type	application/json; charset=utf-8
Authorization	<u>Bearer ODExOWY2MDYtYjdmZS001</u>

### Response

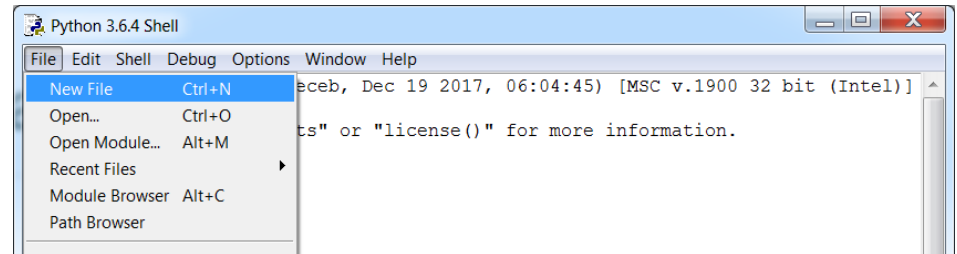
200 / success

```
{
  "items": [
    {
      "id": "Y2lZY29zcGFyYXovL3VzL1JPT0",
      "title": "Akademietag Fulda",
      "type": "group",
      "isLocked": false,
      "lastActivity": "2018-04-21T06:43",
      "teamId": "Y2lZY29zcGFyYXovL3VzL1",
      "creatorId": "Y2lZY29zcGFyYXovL3V",
      "created": "2018-04-11T07:33:23.1"
    },
    {
      "id": "Y2lZY29zcGFyYXovL3VzL1JPT0",
      "title": "Mugdha Ralegaonkar",
      "type": "direct",
      "isLocked": false,
      "lastActivity": "2018-04-05T08:32",
      "creatorId": "Y2lZY29zcGFyYXovL3V",
      "created": "2018-04-05T08:32:53.4"
    },
    {
      "id": "Y2lZY29zcGFyYXovL3VzL1JPT0",
      "title": "Kostiantyn Herasymenko",
      "type": "direct",
      "isLocked": false,
      "lastActivity": "2018-04-04T13:11",
      "creatorId": "Y2lZY29zcGFyYXovL3V"
    }
  ]
}
```

# Getting messages from the room

8. Start IDLE and create a new file

9. Use this code sample:



```
import json
import requests
```

```
accessToken = "[YOUR ACCESS TOKEN]"
roomId = "[YOUR ROOM ID]"
```

```
headers = {"Authorization": "Bearer " + accessToken,
           "Content-Type": "application/json; charset=utf-8"}
```

```
resp = requests.get("https://api.ciscospark.com/v1/messages?roomId=" + roomId + "&max=5",
                   headers=headers, verify=False)
```

```
status = resp.status_code
print("Request status: " + str(status)+"\n")
```

```
response_json = resp.json()
```

```
print("Last messages in the room: " + json.dumps(response_json, indent=4))
```



Import necessary Python modules

Specify you access token and room id

This performs the HTTP GET request

Check if response code OK (200)

Output JSON response