**Downloading a File from a Spark Room**

This example uses a bot once again to download a file in a room. That means, a webhook is required in order for the bot to retrieve the message, because a bot’s permissions won’t allow a retrieval of all messages in a room. If you’re using a real person’s account token, or Oauth, you can list the messages in a room with this function:

<https://developer.ciscospark.com/resource-messages.html>

For help creating a webhook, check out this more complete bot demo:

<https://developer.ciscospark.com/blog/blog-details-8110.html>

In this link <https://developer.ciscospark.com/webhooks-explained.html>, the section titled “Handling Requests from Spark” describes the format of webhook messages. When a webhook that has your bot mentioned, along with an attachment is received, the JSON string that is delivered to your server will contain a “files” key, under the “data” element, in addition to the other attributes, such as personEmail.

The “files” attribute is a list of URLs that can be used to retrieve the file. The URLs will need to be called like any other Spark API request, and contain the bot’s authentication token, because who can view the file is limited to who is in the room. A bot’s permission is limited to the files in which it is also mentioned in the message containing the file attachment.

The below code receives a webhook and, if the data contains a files key, attempts to download the files directly to the source folder from where it is run. The “Content-Disposition” header in the GET response is used to determine the file’s name and file type to be saved. The file data is wholly the body of the GET response. There is no base 64 or other special encoding of the file data, so it can be written directly to the destination. Downloading a file from a Spark room is easy!

from itty import \*

import urllib2

import json

def **sendSparkGET**(url):

request = urllib2.Request(url,

headers={*"Accept"* : *"application/json"*,

*"Content-Type"*:*"application/json"*})

request.add\_header(*"Authorization"*, *"Bearer "*+bearer)

contents = urllib2.urlopen(request)

return contents

*@post*(*'/'*)

def **index**(request):

webhook = json.loads(request.body)

if webhook[*'data'*].has\_key(*'files'*):

for file\_url in webhook[*'data'*][*'files'*]:

response = sendSparkGET(file\_url)

content\_disp = response.headers.get(*'Content-Disposition'*, None)

if content\_disp is not None:

filename = content\_disp.split(*"filename="*)[1]

filename = filename.replace(*'"'*, *''*)

with open(filename, *'w'*) as f:

f.write(response.read())

print *'Saved-'*, filename

else:

print *"Cannot save file- no Content-Disposition header received."*

else:

print *"No files attached to retrieve!"*

return *"true"*

####CHANGE THIS VALUE#####

bearer = *"BOT\_TOKEN\_HERE"*

run\_itty(server=*'wsgiref'*, host=*'0.0.0.0'*, port=10002)

The full code can be found here:

<https://github.com/ciscospark/SparkDownloadFile>

As always, If you have any questions, please contact [devsupport@ciscospark.com](mailto:devsupport@ciscospark.com) 24/7/365 - we’re happy to help!

Taylor Hanson, Customer Support Engineer II