Lab - Using Python to Automate Listing Rooms

Objectives

- Prompt the users for their access token else use the hard-coded access token.
- Setup the API call by initializing variables to hold the URL and header information.
- Display a list of the user's Webex Teams rooms in JSON format.

Background / Scenario

Python can be used to perform the same API calls that can be made in Cisco Webex for Developers and Postman. This lab introduces you to using existing Python code to create a simple API call for listing Webex Teams rooms.

This program first asks users if they want to use their own access token or a generic one. If the answer is no, users are prompted to provide their own access token. If the answer is yes, then an access token value that is written in the code is used. After the access token value is established, the program sets up the API request by initializing variables that contain the required header information and the URI of the Webex Teams rooms API endpoint. These values are used to make the request with the **requests.get()** method. Finally, the response JSON that contains the list of rooms is displayed.

Required Resources

- Webex Teams account
- Webex Teams desktop application
- Python 3 with IDLE
- Python code files

Step 1: Get access token input from the user.

a. Open the file **07_list-rooms.py** in IDLE.

b. Go to **Student Step #1** in the code. In this step, you will modify the code to ask users to use whether they want to use either a hardcoded access token or to enter their own access token. To do so, create a variable called **choice** and use the **input()** method to initialize it to the value that is entered by the user at the prompt. Ask the user, **"Do you want to use the hard-coded access token? (y/n)? "**

c. Create a conditional **if-else** statement. If the value of the **choice** variable is "**n**" or "**N**", use another input statement to prompt the user to enter an access token value. Set the variable **accessToken** to the value of this **input()** method. You must add the word "Bearer" to the value of this variable. Use the following statements:

```
accessToken = input("Please enter your access token. ")
accessToken = "Bearer " + accessToken
```

d. If the user did not enter "**n**" or "**N**", we assume that they want to use the hardcoded token. Because we do not have a general use token value to supply here, use your own token value. You would not make your access token available like this in an actual application, of course. Use the following statement:

```
accessToken = "Bearer <!!!REPLACEME with Webex Teams access token!!!>"
```

Step 2: Add the URL for the Webex Teams rooms endpoint.

a. Go to Student Step #2 in the code. Within the file, locate the statement:

```
apiUri = "<!!!REPLACEME with URL for Webex Teams Rooms API!!!>"
```

b. Replace **<!!!REPLACEME wtih URL for Webex Teams Rooms API!!!>** with **https://api.ciscospark.com/v1/rooms**.

Step 3: Execute the program.

a. Save your program and then choose **Run** > **Run Module** to execute the program in IDLE.

b. Debug as necessary.

Common Python problems include:

- **Punctuation and symbols:** " ",' ', :, =,==, (), { }, [ ] are meaningful.
- **Paired symbols:** must match (open and close).
- **Indentation:** standard indentation is four (or multiples of four) spaces, control structures additional indentation.

c. Refer to the solution file **07_list-rooms_sol.py**, if necessary.

Reflection

1. Look for the section of the code that sends the API request to the URL. What are the values that are required by this method?

____

____

2. What variable contains the actual response from the API?

____

____

3. Python is well-suited to working with JSON because JSON can easily be converted into Python dictionaries and lists. What variable contains the JSON that has been converted to Python?

____

____