## Lab 22: Construct a Python Script to Manage Webex Teams

### Case Study

Kimcorp is headquartered in Finland and has offices in the United States, Canada, India, Germany, and Spain. Kimcorp’s products are available in warehouses and logistics centers in more than 40 countries, and a range of Cisco collaboration solutions are helping to enhance the company's global reach.

As Kimcorp continues to expand its business across different regions of the world, it requires a unified collaboration solution that offers messaging, calling, meetings, and file-sharing capabilities on a single platform. To address this need, the company has decided to adopt Cisco Webex Teams for internal and global collaboration.

Before the deployment of Webex Teams, various departments were using different collaboration applications, making it difficult to manage communications effectively. This approach required multiple collaboration tools to be installed and maintained, resulting in inefficiencies and management challenges.

### Business Challenge

As Kimcorp adopted Cisco Webex Teams for collaboration across various departments within the organization, the company's IT support team sought to automate its deployment and user management. Manually creating individual user accounts in Webex Teams proved to be time-consuming and inefficient, particularly given the large number of departments and employees.

To address this challenge, the IT support team engaged you, a Cisco Certified DevNet Associate, to automate the user provisioning process and streamline the overall management of Cisco Webex Teams.

### Solution

You will use Python to automate and manage Cisco Webex Teams. The REST API will be utilized to provision new users and retrieve their details. To accomplish this, you will use a Webex developer account, from which you will obtain the API access key required to perform these tasks. Without the API access key, it will not be possible to execute any operations through the Python script. In this lab, we will use Webex Teams APIs to authenticate, manage people, manage rooms, manage memberships to rooms, and send a message.

|  |
| --- |
| **Note:** Before starting this lab, you will need to install the Webex Teams application and add at least one other person to your contacts list. This must be completed before proceeding with this lab. Follow the following link: **https://help.webex.com/en-us/article/nw5p67g/Webex-App-%7C-Installation-and-automatic-upgrade** to install Webex Teams. |

1. Get and test your Webex Teams access token.
2. Use a Python script to test your access token.
3. Manage rooms in Webex Teams.
4. Manage memberships in Webex Teams.
5. Manage messages in Webex Teams.

|  |  |
| --- | --- |
| **// Get and Test Your Webex Teams Access Token**  1. Turn on the Ubuntu VM and then open any web browser, go to the following link: **https://developer.webex.com/.** Log in if you already have an account. If not, go through the signup process.    2. Click on the **Documentation** dropdown. Then click on **Webex Meetings**.    3. From the left-hand side menu under **API References,** click on the **All APIs** to expand its submenu.    4. Explore all the different API calls. In this lab, we will use the API documentation for **People**, **Rooms**, **Membership**, and **Message**. Click on the **People**. Then click on the **Get My Own Details**.    5. You may test your access token using the OpenAPI documentation on the developer website. However, you will be using your token in Python programs. As a result, you should also test its functionality in a script. To access of token, click on the **user icon** in the top right corner. Copy the **Bearer** access token key by clicking on the **copy icon**. Paste this access token into the clipboard. Later, we use it in a Python script to access Webex.    6. Click on the **OK** button.    7. Scroll down to test that the access token is working properly. Here, you will not be able to insert the access token key in Authorization, as it is inserted by default. Click on the **Run** button.    8. Scroll down you will see the **Response** with your personal information. In the next step, we will extract this same information using a Python script.    9. Open Visual Studio Code. Then click on **File**. After that, click on **Open Folder…**    10. Navigate to the **devsac/labs/devnet-src/webex-teams** directory. Then click on the **Open** button to open it.    11. In the **EXPLORER** panel, you should now see all the placeholder **.py** files you will use in this lab. Click on the **authentication.py** file to open it.    12. Copy and paste the below-provided Python script into the **authentication.py.**  Replace **your\_token\_here** with your personal access token you copied in the previous step. Press **Ctrl+S** to save it.   |  | | --- | | import requests  import json  access\_token = 'your\_token\_here'  url = 'https://webexapis.com/v1/people/me'  headers = {  'Authorization': 'Bearer {}'.format(access\_token)  }  res = requests.get(url, headers=headers)  print(json.dumps(res.json(), indent=4)) |     13. Open a new **Terminal** window. First, execute the following command: **cd** **labs/devnet-src/webex-teams** to go inside the directory.    14. Execute the following command: **python3 authentication.py** to run the Python script. You should get the same output you saw in the OpenAPI documentation. |

|  |  |  |
| --- | --- | --- |
| **// Use a Python Script to Test Access Token**  1. People are registered users on Webex Teams. The API allows you to retrieve a list of individuals, create a person, retrieve an individual's details, change a person, and delete a person.  2. Go back to the **developer.webex.com** website. Under **All** **APIs** > **People**, click on the **List People**.    3. In the **Query Params,** click on the **email** check box. Enter your email address to find a specific user. Alternatively, you could use the **displayName** parameter if you know the exact name.    4. Scroll down, click on the **Run** button.    5. Scroll down you will see the **Response** with user information. In the next step, we will extract this same information using a Python script.    6. In Visual Studio Code, click on the **list-people.py** file to open it.    7. Copy and paste the below-provided code into the **list-people.py** file. Replace **your\_token\_here** with your personal access token and **user@example.com** with an actual registered Webex Team user. Press **Ctrl+S** to save it.   |  | | --- | | import requests  import json  access\_token = 'your\_token\_here'  url = 'https://webexapis.com/v1/people'  headers = {  'Authorization': 'Bearer {}'.format(access\_token),  'Content-Type': 'application/json'  }  params = {  'email': 'user@example.com'  }  res = requests.get(url, headers=headers, params=params)  print(json.dumps(res.json(), indent=4)) |     8. Execute the following command: **python3 list-people.py** to run the Python script. You should get the same output, similar to the following. The value for the **id** key will be used in the next API call.    9. Copy and paste the below-provided script in the **list-people.py** to get additional details for a person by using the value of the person **id** key in your API call. Replace **previous\_id\_here** with the value for **id** from the previous API call. Press **Ctrl+S** to save it.   |  | | --- | | person\_id = 'previous\_id\_here'  url = 'https://webexapis.com/v1/people/{}'.format(person\_id)  headers = {  'Authorization': 'Bearer {}'.format(access\_token),  'Content-Type': 'application/json'  }  res = requests.get(url, headers=headers)  print(json.dumps(res.json(), indent=4)) |     10. Execute the following command: **python3 list-people.py** to run the Python script. You will get information that is very similar to the previous step. |

|  |  |  |
| --- | --- | --- |
| **// Manage Rooms in Webex Teams**  1. Rooms, also known as spaces in the user interface, allow individuals to transmit messages and files to work virtually in shared meeting areas. In this section, you will list rooms, create rooms, and get room data.  2. Navigate back to the **developer.webex.com** website. Click on the **Documentation** dropdown. Then click on **Webex Messaging**.    4. From the left-hand side menu under **API References,** click on the **All APIs** to expand its submenu.    5. Click on the **Rooms**. Explore the various API calls you can make with the **Rooms**. Click on the **List Rooms** and explore the **Query Parameters**.    6. In Visual Studio Code, click on the **list-rooms.py** file to open it.    7. Copy and paste the below-provided code into the **list-rooms.py** file. Replace **your\_token\_here** with your personal access token and **user@example.com** with an actual registered Webex Team user. Press **Ctrl+S** to save it.   |  | | --- | | import requests  import json  access\_token = 'your\_token\_here'  url = 'https://webexapis.com/v1/rooms'  headers = {  'Authorization': 'Bearer {}'.format(access\_token),  'Content-Type': 'application/json'  }  params={'max': '100'}  res = requests.get(url, headers=headers, params=params)  print(json.dumps(res.json(), indent=4)) |     8. Execute the following command: **python3 list-rooms.py** to run the Python script. You will get information that is very similar to the previous step. Your output will differ from the following. There is only one room listed here.    9. In Visual Studio Code, click on the **create-rooms.py** file to open it.    10. Copy and paste the below-provided code in the **create-rooms.py** file. This code will create a room in Webex Teams. Replace **your\_token\_here** with your personal access token. Notice that this is a POST request and uses the **title** parameter. Press **Ctrl+S** to save it.   |  | | --- | | import requests  import json  access\_token = 'your\_token\_here'  url = 'https://webexapis.com/v1/rooms'  headers = {  'Authorization': 'Bearer {}'.format(access\_token),  'Content-Type': 'application/json'  }  params={'title': 'DevNet Associate Certification Training!'}  res = requests.post(url, headers=headers, json=params)  print(json.dumps(res.json(), indent=4)) |     11. Execute the following command: **python3 create-rooms.py** to run the Python script. You should get a response similar to the following. The room **id** and **title** are highlighted. Copy the room ID and save it in a text file for use in the rest of this lab. |

|  |  |  |
| --- | --- | --- |
| **// Manage Memberships in Webex Teams**  1. In this section, you will use the Membership API to add someone to your room.  2. Go back to the **developer.webex.com** website. Under **All** **APIs** > **Membership**, click on the **List Memberships**and explore the**Query Parameters**.    3. In Visual Studio Code, click on the **list-memberships.py** file.    4. Copy and paste the below-provided script in the **list-memberships.py** file. This code will list down the membership information that is in Webex Teams. Replace **your\_token\_here** with your personal access token. Replace **your\_room\_id** with the value you got in the previous part. Press **Ctrl+S** to save it.   |  | | --- | | import requests  import json  access\_token = 'your\_token\_here'  room\_id = 'your\_room\_id'  url = 'https://webexapis.com/v1/memberships'  headers = {  'Authorization': 'Bearer {}'.format(access\_token),  'Content-Type': 'application/json'  }  params = {'roomId': room\_id}  res = requests.get(url, headers=headers, params=params)  print(json.dumps(res.json(), indent=4)) |     5. Execute the following command: **python3 list-memberships.py** to run the Python script. You should receive something similar to the following. Unless you have added someone to the Webex Teams app, you should be the sole member.    6. Go back to the **developer.webex.com** website. Under **All** **APIs** > **Membership**, click on **Create a Membership**.Explore the **Query Parameters** that are available. In your script, you will use the required **roomID**and**personEmail** parameters.    3. In Visual Studio Code, click on the **create-membership.py** file.    4. Copy and paste the below-provided script in the **create-membership.py** file. This code will create a membership in Webex Teams. Replace **your\_token\_here** with your personal access token. Replace **your\_room\_id**with the value you got in the previous part. Replace **new-user@example.com**w with the email of the person you want to add to the room. Press **Ctrl+S** to save it.   |  | | --- | | import requests  import json  access\_token = 'your\_token\_here'  room\_id = 'your\_room\_id'  person\_email = 'new-user@example.com'  url = 'https://webexapis.com/v1/memberships'  headers = {  'Authorization': 'Bearer {}'.format(access\_token),  'Content-Type': 'application/json'  }  params = {'roomId': room\_id, 'personEmail': person\_email}  res = requests.post(url, headers=headers, json=params)  print(json.dumps(res.json(), indent=4)) |     5. Execute the following command: **python3 create-membership.py** to run the Python script. You should receive a response similar to the following. Unless you add someone to the Webex Teams app, you should be the only member. |

|  |  |
| --- | --- |
| **// Manage Messages in Webex Teams**  1. In Webex Teams, a message can include plain text, Markdown, or a file attachment. Each message is presented on its own line, complete with a timestamp and sender information. You can use the Messages API to view, create, and delete messages. In this section, you will send a message to the room you created in this lab.  2. Navigate back to the **developer.webex.com** website. Click on the **Documentation** dropdown. Then click on the **Webex Messaging**.    4. From the left-hand side menu under **API References,** click on the **All APIs** to expand its submenu.    5. Click on the **Messages**. Explore the various API calls you can make with the **Messages**. Click the POST request for **Create a Message** and explore the **Query Parameters**. Notice that for a simple text message, you can use the **text** or **markdown** parameter. In this step, you will specify a message with Markdown formatting.    6. In Visual Studio Code, click on the **creat-markdown-message.py** file to open it.    7. Copy and paste the below-provided code into the **creat-markdown-message.py** file. This code will create a message in a specified room in Webex Teams. Replace **your\_token\_here** with your personal access token. Replace **your\_room\_id** with the value you got in the previous part. Press **Ctrl+S** to save it.   |  | | --- | | import requests  import json  access\_token = 'your\_token\_here'  room\_id = 'your\_room\_id'  message = 'Hello \*\*DevNet Associates\*\*!!'  url = 'https://webexapis.com/v1/messages'  headers = {  'Authorization': 'Bearer {}'.format(access\_token),  'Content-Type': 'application/json'  }  params = {'roomId': room\_id, 'markdown': message}  res = requests.post(url, headers=headers, json=params)  print(json.dumps(res.json(), indent=4)) |     8. Execute the following command: **python3** **creat-markdown-message.py** to run the Python script. You should get a similar response. Notice that the Markdown was converted to HTML. |