ConRes
IT SOLUTIONS

Doing Cool Things with the Cisco Webex API

Part 2 - Programmatically Retrieving Data

About Me

Josh Kittle

Solutions Architect for Collaboration

20+ years of experience supporting Cisco Collaboration technologies

jkittle@conres.com

ConRes IT SOLUTIONS

Who is this for?

- Anybody supporting a Webex environment
- Desire to use the API to ‘Do Cool Things’
- Interest in automation / bulk provisioning changes

What will we cover?

- JSON Basics
- Creating an Integration
- OAuth Grant Process
- Using Postman to HELP YOU Write Working Code
- Retrieving Basic Data using the Requests library using Python

What is JSON?

- **JSON** (JavaScript Object Notation) is a lightweight data-interchange format.
- It is easy for humans to read and write.
- It is easy for machines to parse and generate

```
{
    hey: "guy",
    anumber: 243,
  - anobject: {
        whoa: "nuts",
      - anarray: [
            1,
            2,
            "thr<h1>ee"
        ],
        more: "stuff"
    },
    awesome: true,
    bogus: false,
    meaning: null,
    japanese: "明日がある。",
    link: http://jsonview.com,
    notLink: "http://jsonview.com is great"
}
```

ConRes IT SOLUTIONS

JSON Object Structure

- A JSON object contains zero, one, or more key-value pairs, also called properties.
- The object is surrounded by curly braces { }.
- Every key-value pair is separated by a comma. The order of the key-value pair is irrelevant.
- A key-value pair consists of a key and a value, separated by a colon (:)

```
{'grant_type': 'authorization_code',
 'client_id' : client_id,
 'client_secret' : client_secret,
 'code' : code,
 'redirect_uri' : redirect_uri}
```

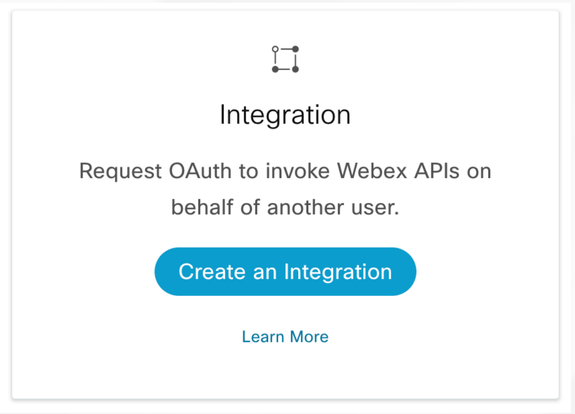
ConRes IT SOLUTIONS

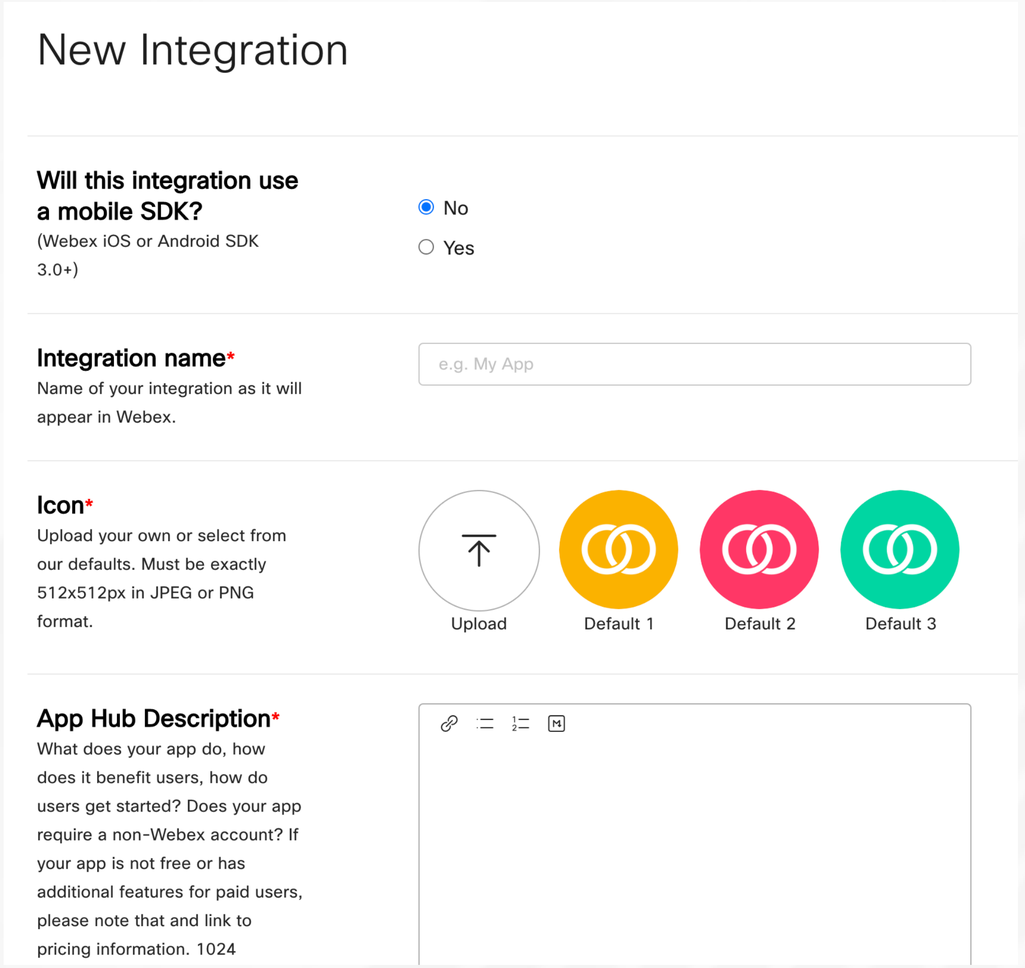
REST API Process Simplified….

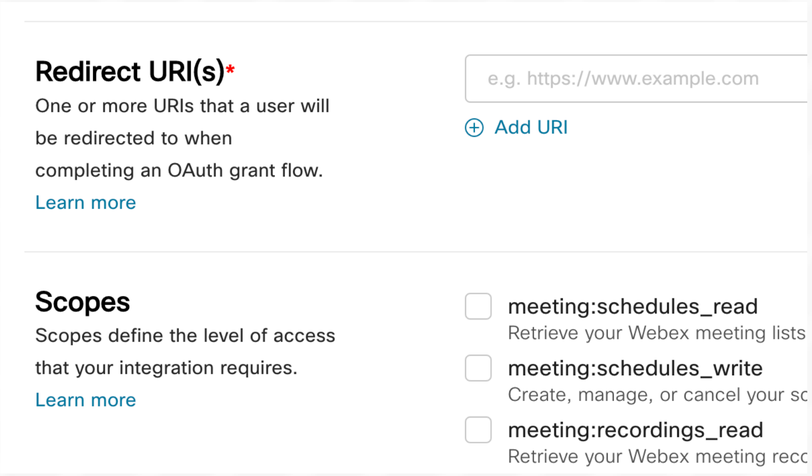
1. Register your Integration with Webex - you'll be issued 'Client Credentials'
    1. Client ID
    2. Client Secret
2. Request permission using an OAuth Grant Flow
3. Exchange the resulting authorization 'code' for an access 'token'
4. Use your 'token' to make API calls.

Creating an Integration

https://developer.webex.com/my-apps/new





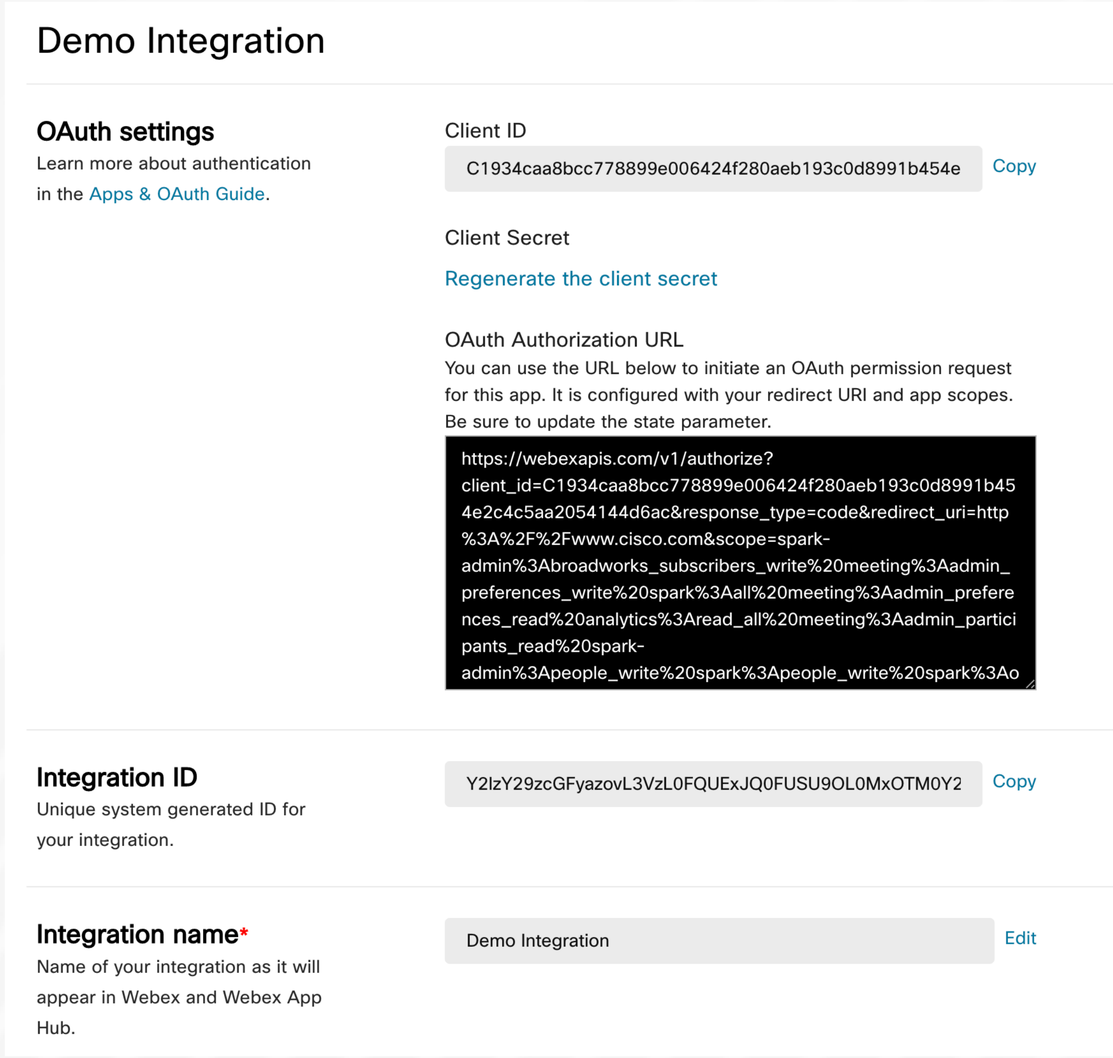


Your Completed Integration

Take note of the following values

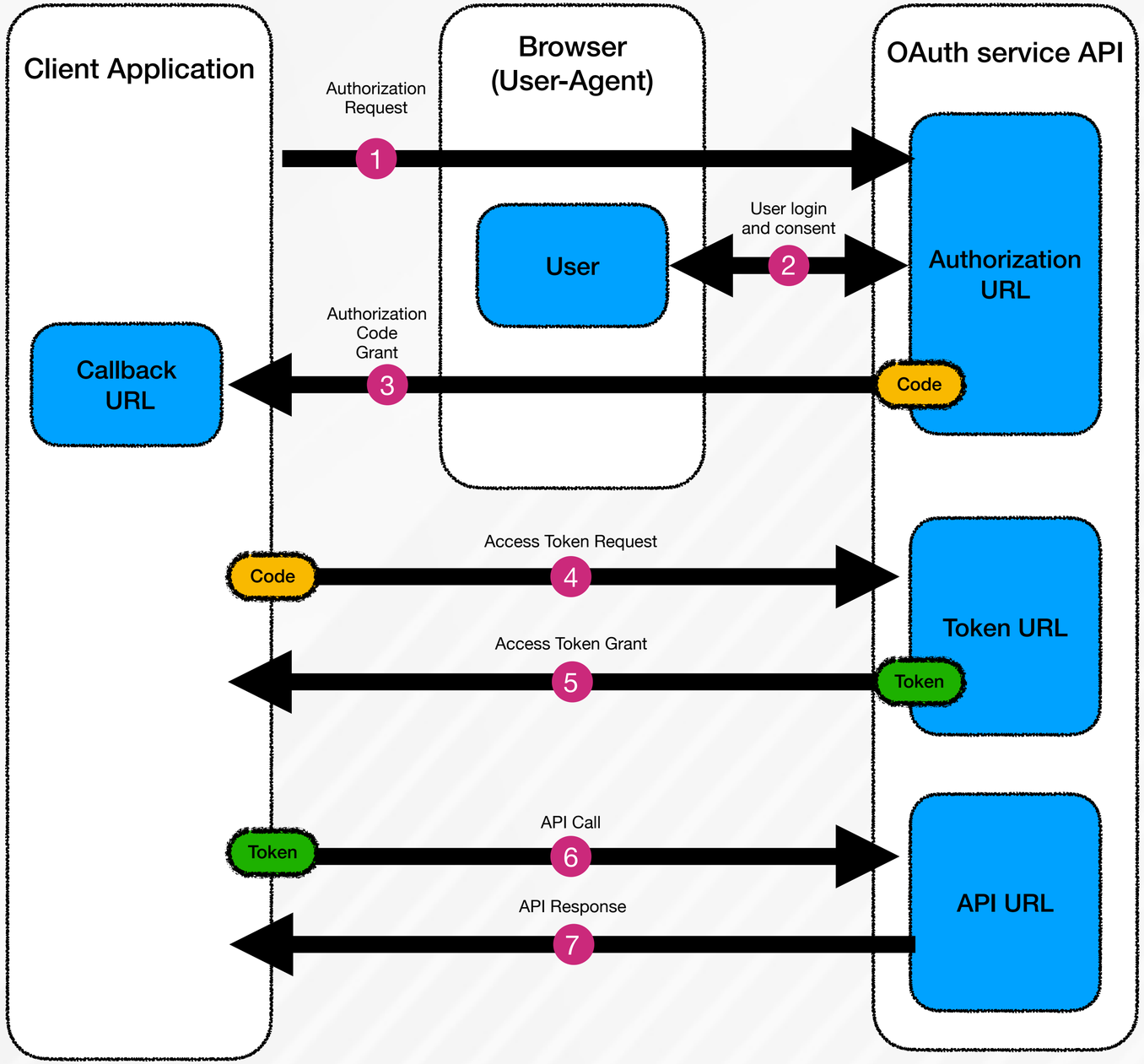
- Client ID
- Client Secret
- OAuth Authorization URL

  (Includes the callback URL)



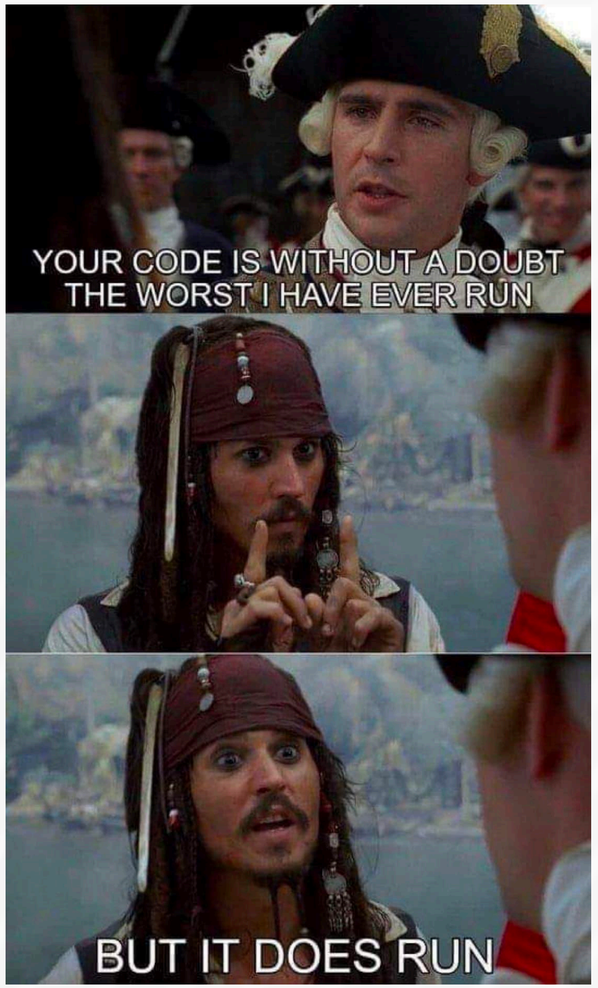
What is OAuth?

- An **open protocol** to allow **secure authorization** in a **simple** and **standard** method for web, mobile, and desktop applications



On the topic of writing code…

- We don’t need to be expert programmers to ‘get things done’
- We will share some examples that you can use to hit the ground running
- https://github.com/conresinc/cool-stuff-webex-api

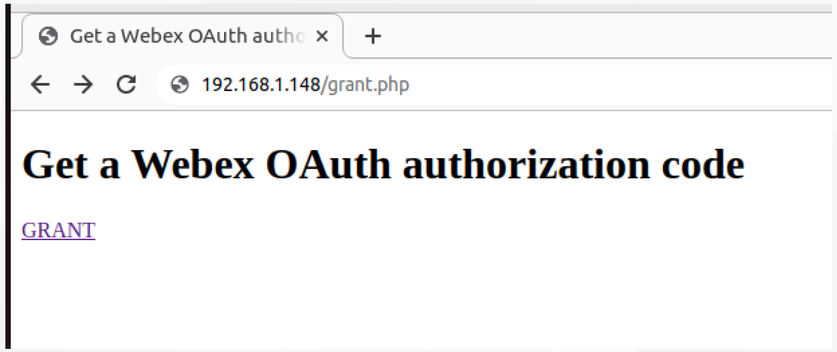


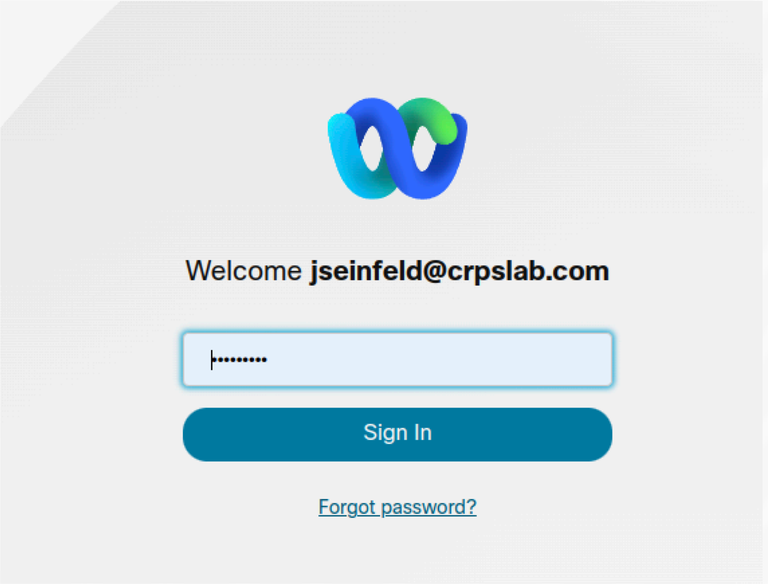
grant.php (Available on our GitHub!)

```php
<?php
$oauthAuthorizationURL ="Paste Value from OAuth Authorization URL Here in Double Quotes";
?>

<!DOCTYPE html>
<html>
  <head>
    <title>Get a Webex OAuth authorization code</title>
    <meta charset='utf-8'>
  </head>
  <body>
        <h1>Get a Webex OAuth authorization code</h1>
        </div>
        <div class='spacer-small'></div>
        <div class='center'>

<?php
echo "<a href=\"" . $oauthAuthorizationURL . "\">";
?>

          <div class='button' style='width:512px;'>GRANT</div>
        </a>
        </div>
      </div>
    </section>
  </body>
</html>
```

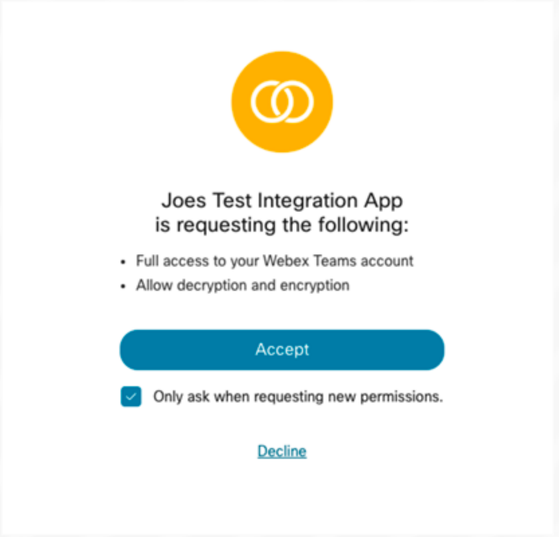
ConRes IT SOLUTIONS

What it looks like

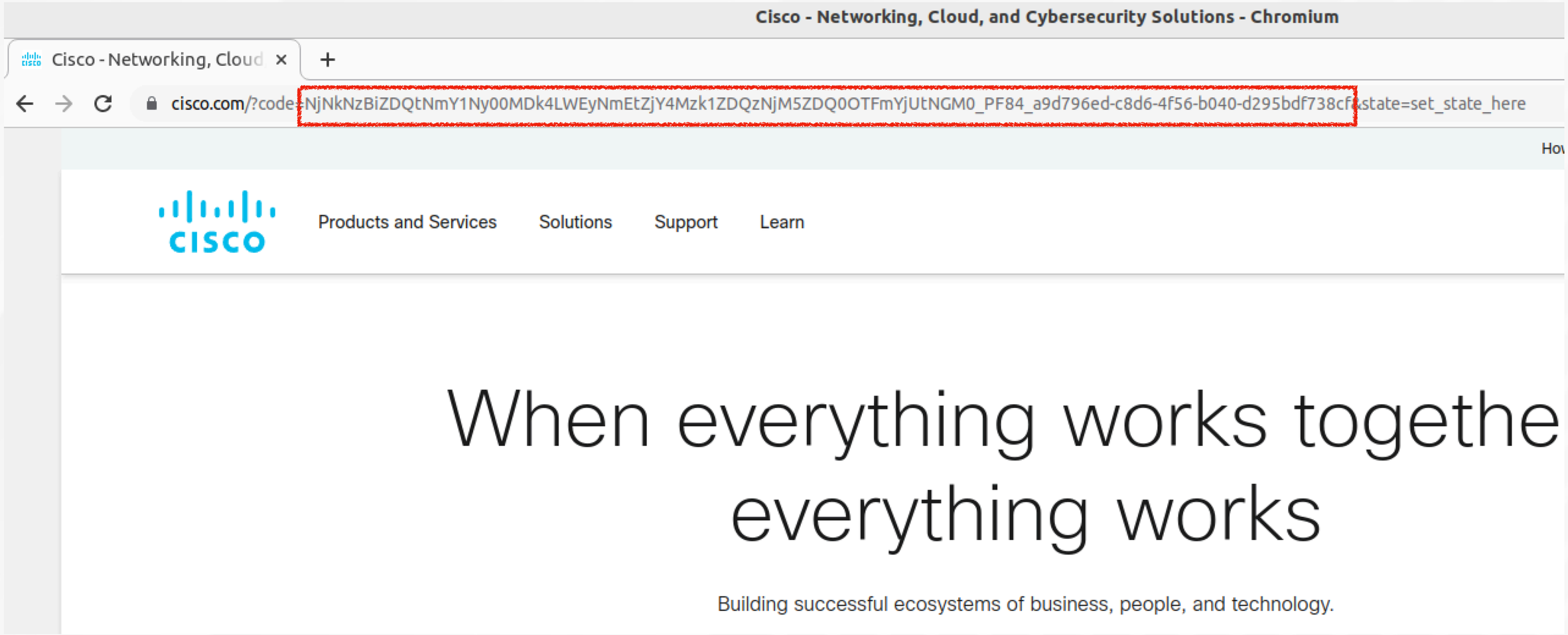




- The ‘request for access’ page can be quite lengthy, depending on the roles and permissions being granted.



Getting your authorization code



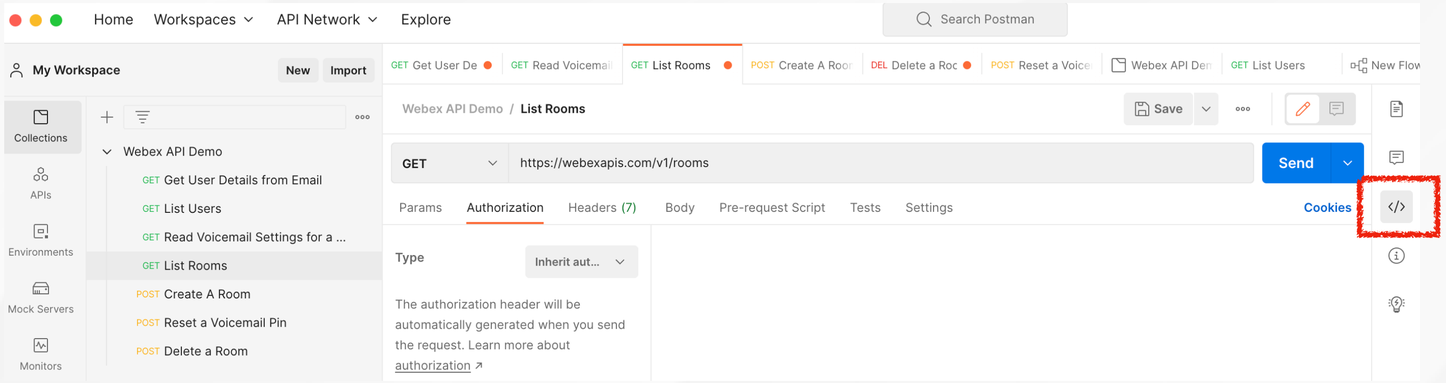
getTokenFromCode.py (Available on our GitHub!)

```python
import requests
import json

client_id = "Your Client ID in double quotes"
client_secret = "Your Client Secret in double quotes"
code = "Your OAuth Authorization Code in double quotes"
redirect_uri = 'http://www.cisco.com'

API_BASE_URL = "https://webexapis.com/v1/access_token"


def getTokenFromCode(client_id,client_secret,code,redirect_uri):
    data = {'grant_type': 'authorization_code',
            'client_id' : client_id,
            'client_secret' : client_secret,
            'code' : code,
            'redirect_uri' : redirect_uri}

    bodyPayloadText = json.dumps(data)
    API_HEADERS = { 'Content-Type' : 'application/json'}
    API_RESPONSE = requests.post(API_BASE_URL, headers=API_HEADERS, data = bodyPayloadText)
    access_token = API_RESPONSE.json()
    return access_token

results = getTokenFromCode(client_id,client_secret,code,redirect_uri)
print("Access Token is: ")
print(results['access_token'])
print("\n")
print("Complete JSON Response: \n")
print(results)

```

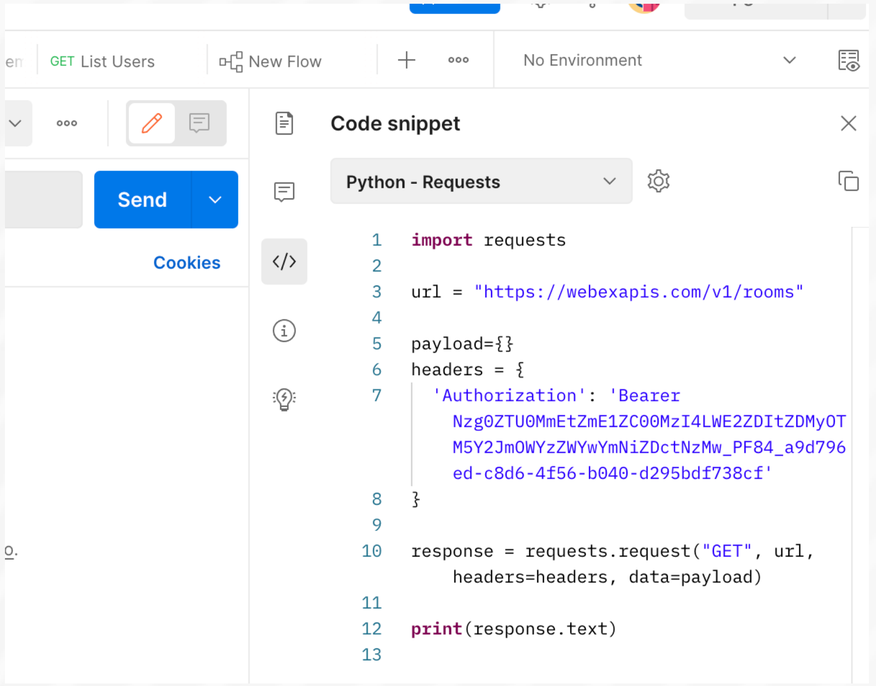
ConRes IT SOLUTIONS

Coding Tip - Don’t work too hard!

This is a Standard API Call in Postman



<- Click This Button!



- Code examples for the language of your choice will be revealed!
- In this example we are looking at a python script that’s using the requests library to make an API call

ConRes IT SOLUTIONS

Python3 Demo Code

- Obtain a user id key from an email address
- Get a list of phone numbers for a user, from the user id key
- Read the voicemail configuration for a user, from the user id key

ConRes IT SOLUTIONS

getUserIdFromEmail.py (Available on our GitHub!)

```python
import requests

email_address = "jseinfeld@crpslab.com"
bearer_token = "Enter Your Token Here in Double Quotes"

API_HEADER =   {'Authorization': 'Bearer ' + bearer_token,
                'Content-Type' : 'application/json',
                'Accept' : '*/*'}

API_RESPONSE = requests.get('https://webexapis.com/v1/people/?email=' + email_address, headers=API_HEADER, verify=True)

if API_RESPONSE.json()["items"]:
    userId = API_RESPONSE.json()["items"][0]["id"]
    print(userId)
else:
    print("No User Found")

```

ConRes IT SOLUTIONS

getPhoneNumbersFromEmail.py (Available on our GitHub!)

```python
import requests
import json

email_address = "jseinfeld@crpslab.com"
bearer_token = "Enter Your Token Here in Double Quotes"

API_HEADER =   {'Authorization': 'Bearer ' + bearer_token,
                'Content-Type' : 'application/json',
                'Accept' : '*/*'}

API_RESPONSE = requests.get('https://webexapis.com/v1/people/?email=' + email_address, headers=API_HEADER, verify=True)

if API_RESPONSE.json()["items"]:
    userId = API_RESPONSE.json()["items"][0]["id"]
    print ("The user ID for the email address " + email_address + " is :" + userId + "\n")
else:
    print("No User Found")


API_RESPONSE = requests.get('https://webexapis.com/v1/people/' + userId, headers=API_HEADER, verify=True)

results = json.loads(API_RESPONSE.text)

print("The phone number associated with the email address " + email_address + " is "+ results['phoneNumbers'][0]['value'])
```

ConRes IT SOLUTIONS