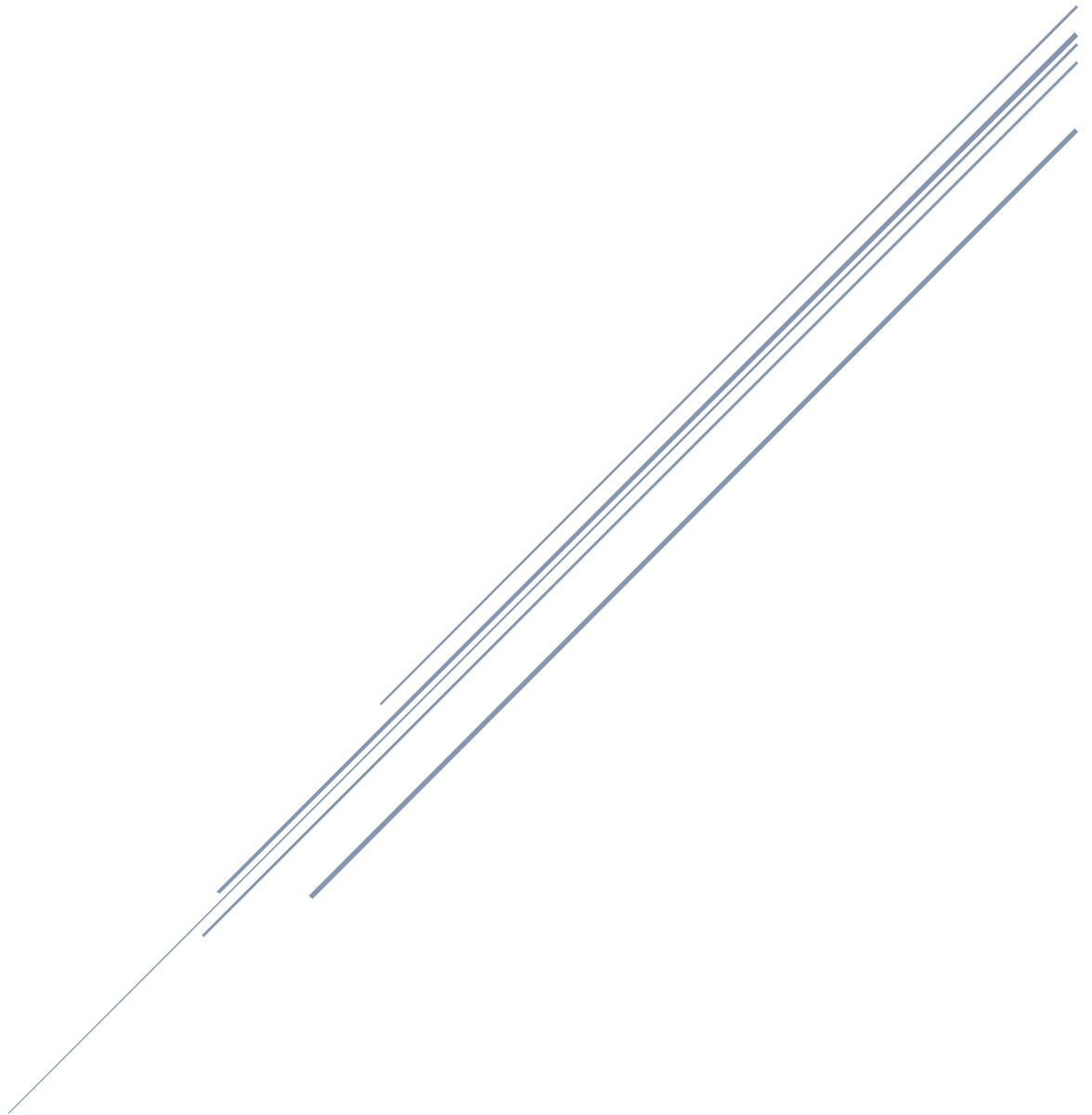


JSON API DOCUMENTATION

SuiteCRM API V8



In this documentation, it is expected that you already have a running SuiteCRM instance on your local VM machine.

SuiteCRM API version 8 exposes a set of resources, to be consumed by clients who wish to harness the powerful CRM functionality provided by SuiteCRM.

The API framework employs a Restful design to facilitate the JSON API 1.0 standard messages over HTTPS. It includes meta objects to provide functionality which is not yet defined in the JSON API 1.0 standard. The SuiteCRM API is secured by the OAuth 2 Server provided in SuiteCRM.

API Authentication using Postman

Please do the following to avoid authentication issues before calling the actual endpoints.

Required tools:

- WinSCP / Putty Terminal
- Postman 9.8.2

Generate private and public key for OAUTH2

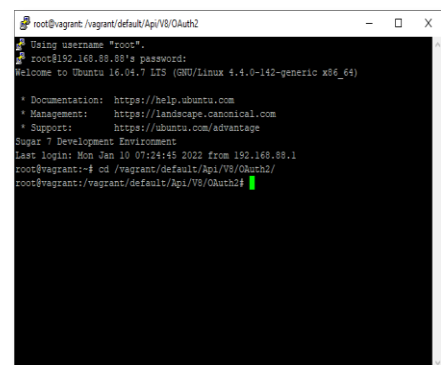
SuiteCRM API uses OAuth2 protocol, which needs private and public keys.

1. Open a terminal and go to “{{suitecrm.root}}/Api/V8/OAuth2”

In this documentation, the directory of my SuiteCRM instance project is “/vagrant/default/”

Open your Putty terminal and enter

```
cd /vagrant/default/Api/V8/OAuth2
```



```
root@vagrant: /vagrant/default/Api/V8/OAuth2
Using username "root".
root@192.168.88.88's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Super 7 Development Environment
Last login: Mon Jan 10 07:14:45 2022 from 192.168.88.1
root@vagrant:~# cd /vagrant/default/Api/V8/OAuth2/
root@vagrant:/vagrant/default/Api/V8/OAuth2#
```

2. Generate a private key using the script

```
openssl genrsa -out private.key 2048
```

3. Generate a public key using the script

```
openssl rsa -in private.key -pubout -out public.key
```

4. The permission of the key files must be above 600 so we have to set it

```
sudo chmod 600 p*.key
```

5. Also, we have to ensure that the files are owned by PHP

```
ls -l | grep http
```

-this will display active users in your vm

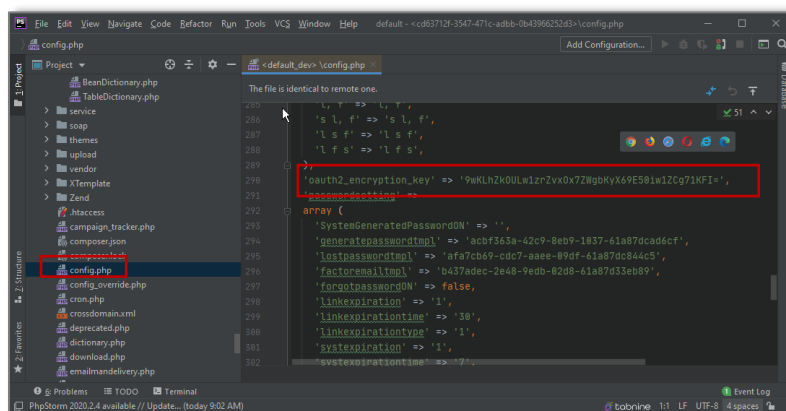
```
sudo chown vagrant:vagrant p*.key
```

-setting the ownership to vagrant

6. Generate the OAUTH2 Encryption Key

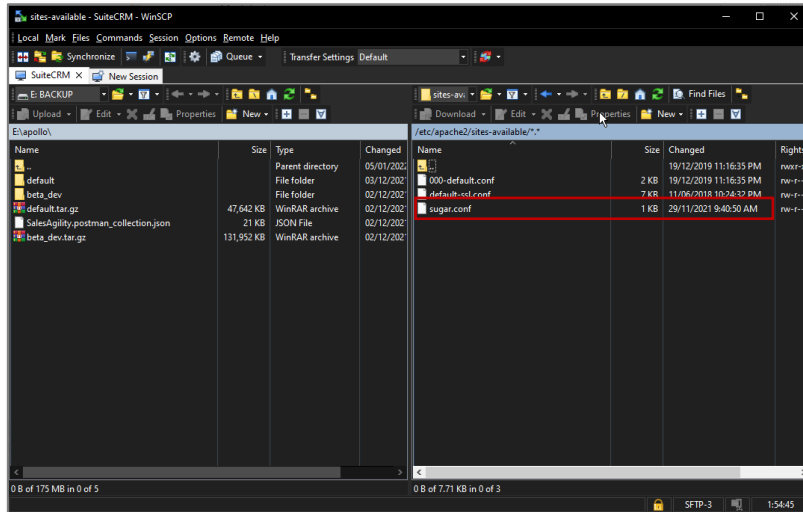
OAuth2's AuthorizationServer needs to set an encryption key for security reasons. This key has been generated during the SuiteCRM installation and stored in the config.php under "oauth2_encryption_key". If you would like to change its value you may generate a new one by using the script and storing the output in the config.php file

```
php -r 'echo base64_encode(random_bytes(32)), PHP_EOL;
```

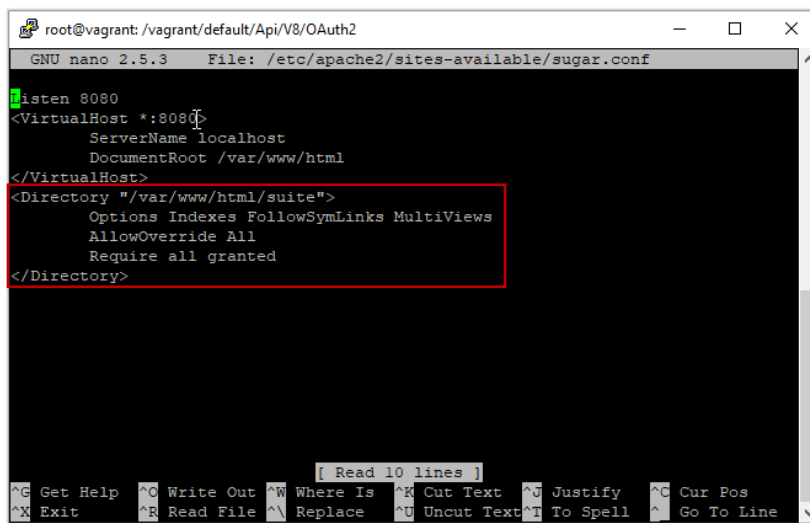


7. Verify if rewrite module is installed and activated

In this documentation, I can check and verify this by checking apache2 sugar.conf file and the directory of this file is “/etc/apache2/sites-available”.



nano /etc/apache2/sites-available/sugar.conf

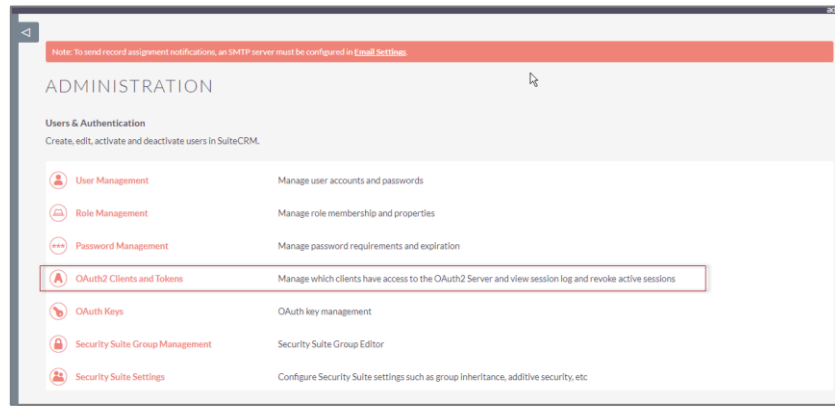


It is necessary to verify if ‘mod_rewrite’ module of Apache server is enabled. Make sure to enable and activate it. This process depends the VM configuration you have on your local instance.

Also, for the SuiteCRM location ‘/var/www/html/suite’ one should change the **AllowOverride** directive inside Directory directive from **None** to **All** to assure that rewrite rules of .htaccess work.

Configure Grant Types

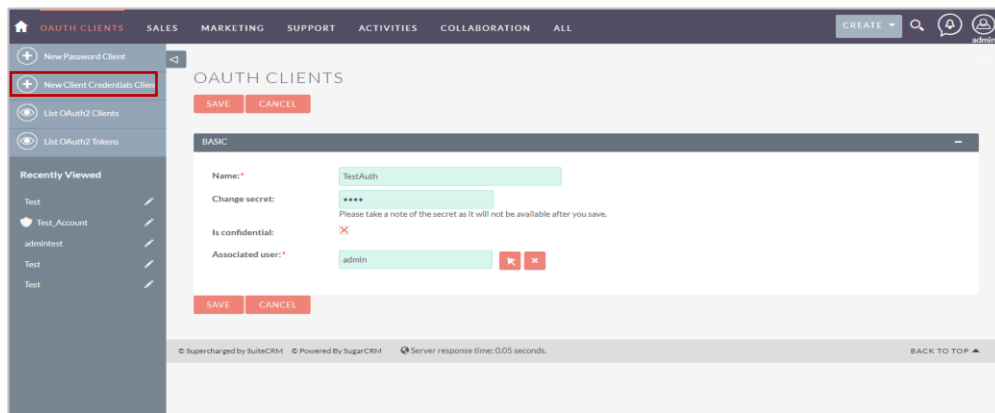
Before you can consume the API, you must first configure SuiteCRM to grant access to a client. SuiteCRM 7.10 provides administrative panel, through which you can add clients and revoke tokens. To configure the grant types, select the admin panel, and then select **OAuth2 Client and Tokens**.



Client Credentials Grant

A client credentials grant is the simplest of all of the grant types, which is used to authenticate a machine or service.

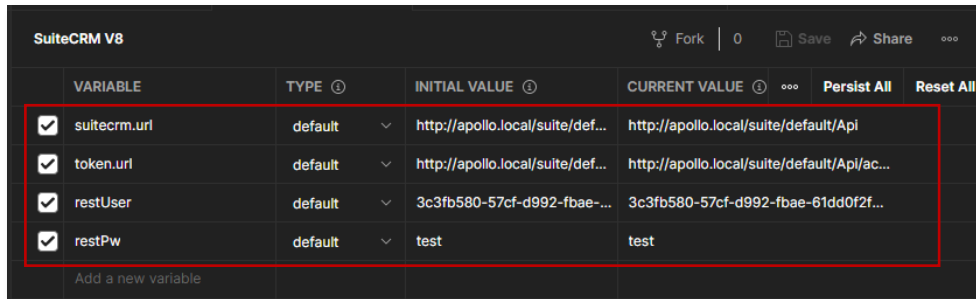
In this documentation, we will only be using the Client Credentials Grant. To do this, click *New Client Credentials Client* on the left portion of the screen as shown below.



Field	Description
Name	This makes it easy to identify the client
Secret	Defines the client_secret which is posted to the server during authentication
Is Confidential	Keeping client password confidential to the world
Associated User	Limits the client access to CRM, associating client with security privileges

Create environment variables using Postman

Open your postman application and set your environment variables that will be used.

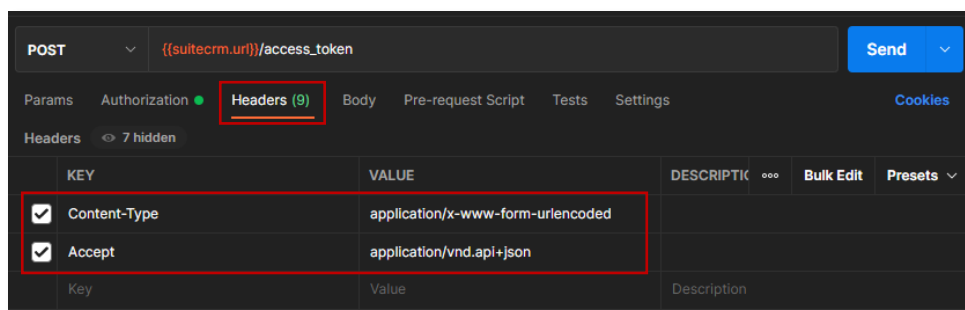


Variable	Description format
suitecrm.url	http://{{IP address}}/{{your instance}}/Api
token.url	http://{{IP address}}/{{your instance}}/Api/access_token
restUser	ID of the Client Credentials Client created earlier
restPW	Secret of the Client Credentials Client created earlier

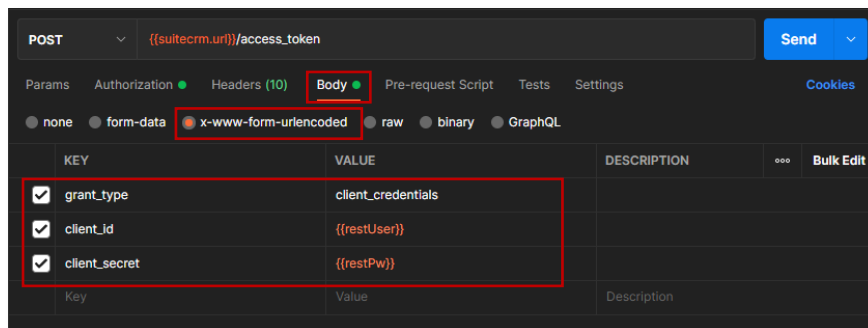
Variable	Values
suitecrm.url	http://apollo.local/suite/default/Api
token.url	http://apollo.local/suite/default/Api /access_token
restUser	3c3fb580-57cf-d992-fbae-61dd0f2f799d
restPW	test

Create a request

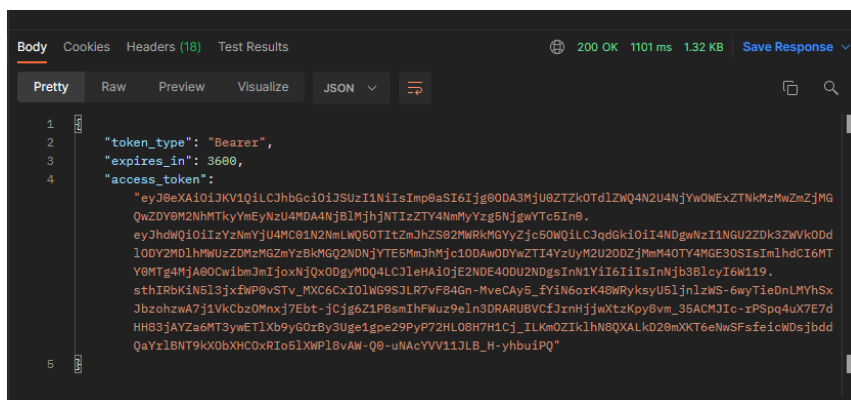
Create a new POST request to {{suitecrm.url}}/access_token
Add the following key value pair to the Headers tab



Add the following key and value pair to the body.
Make sure the *x-www-form-urlencoded* is selected and click 'Send'



You will have this response once authentication is successful



Import a Collection File to Postman

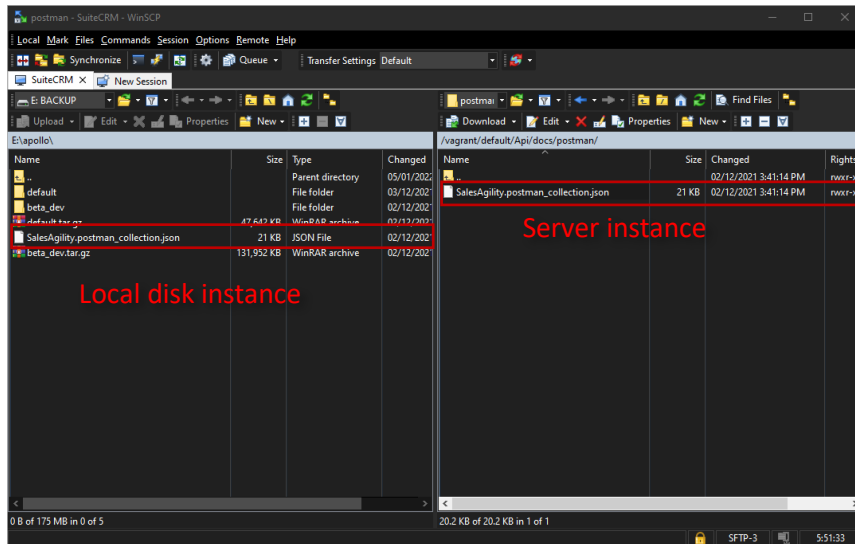
Using your WinSCP, download the .json collection file from your local SuiteCRM instance. You can get the collection file in its default directory:

`{{IP Address}}/{{your instance}}/Api/docs/postman/`

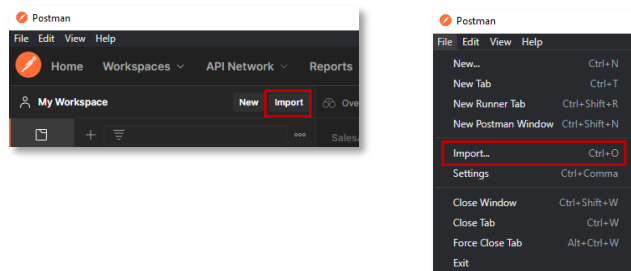
In this documentation, I setup my SuiteCRM in stance in this directory

```
/vagrant/default/Api/docs/postman/
```

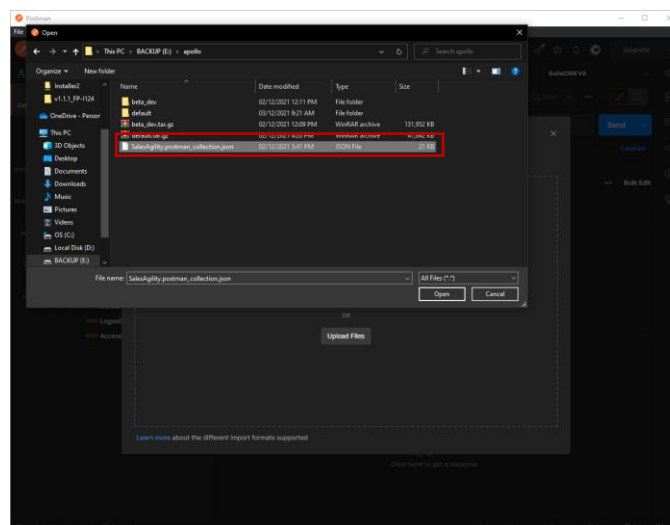
Download the SalesAgility.postman_collection.json to your local disk directory



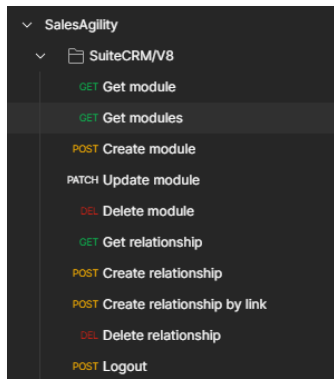
On your Postman application, click the 'Import' button or you can click 'File' then select 'Import'.



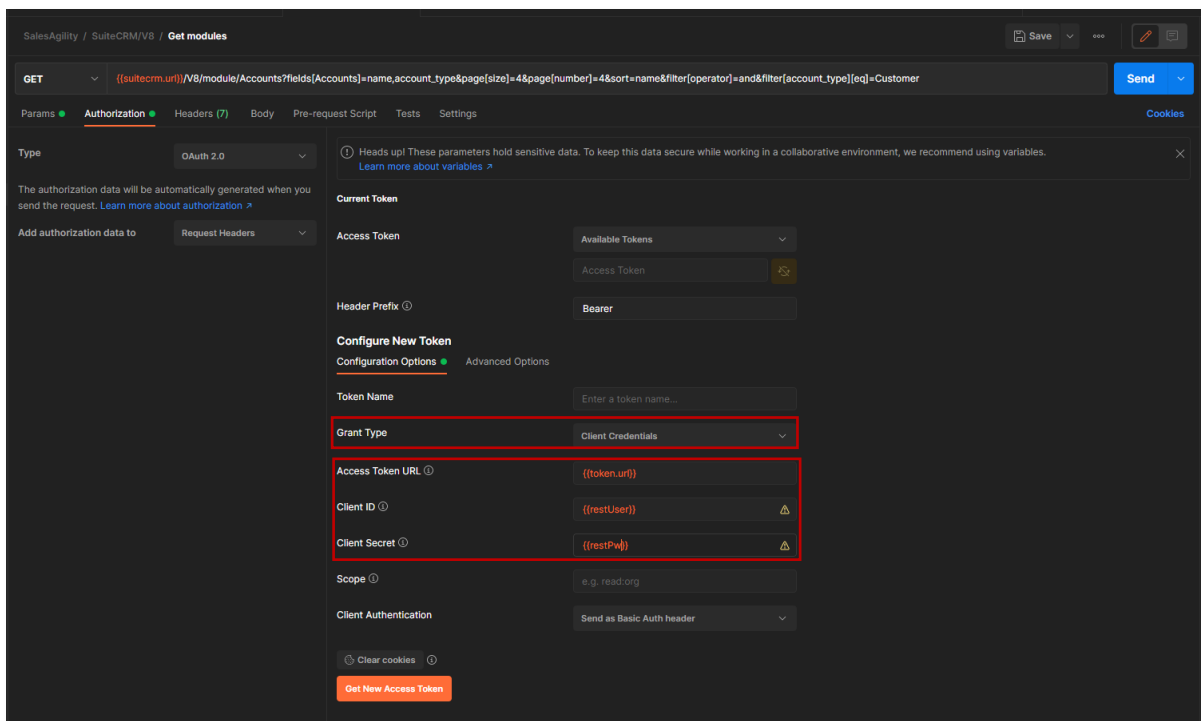
Locate and Import the SalesAgility.postman_collection.json file you downloaded to your local disk



After importing the collection file, you will have something this...



Select the 'Get modules' request and on the **Authorization** tab, select the **OAuth 2.0** as shown below



Select 'Client Credentials' for the **Grant Type** value and use your environment variables as shown above. Once done, click the 'Get New Access Token' button.

The image below shows the confirmation that you successfully got a new token. Click the 'Use Token' button and click the 'Send' button. Check the response, you should get 'Request successful' message.

