Base Oauth2 Authorization User Manual

# INTRODUCTION

This two module will implement **Oauth2 Authorization server** in Odoo and its services and **Oauth2 client service** for odoo.

## FEATURES

* Implemented in Odoov13.
* Implements the Grant Type Authorization Code with response type as “code” .
* Implements the Grant Type Implicit with response type as “token”.
* Implements the Grant Type Resource Owner Credentials.
* Implements the Grant Type Refresh Token.
* Implements the login with “access\_token”
* Implements Oauth2 client service in Odoo for the Oauth2 Authorization Server implemented with our module (both with response type code and token)

# INSTALLATION

Now simply follow these steps:

pip3 install oauthlib

For Oauth2 Authorization Server

1. Go to settings menu and click on Activate the Developer mode.
2. Now go to apps menu and Click on ‘Update Modules List’.
3. Remove the app’s filter and search for the the word ‘base\_oauth2\_authorization\_services’.
4. Now you will see the module then simply install it.

For Oauth2 Client for an Odoo client (Optional )

1. Go to settings menu and click on Activate the Developer mode.
2. Now go to apps menu and Click on ‘Update Modules List’.
3. Remove the app’s filter and search for the the word ‘oauth2\_auth\_redirect’’.
4. Now you will see the module then simply install it.

After installing you will be able to handle different functionality as mentioned in module’s work flows.

# CONFIGURATIONS FOR Oauth2 Authorization Server

**Go to Rest API Menus under Settings**

All configurations are assuming that **localhost:9879 as Oauth2 Authorization Server.**

All configurations are assuming that **127.0.0.1:9889 as Oauth2 Odoo Client.**

The endpoints should need to be same which means ipaddress, hostname, domain or port can change according to your implementations. [http://<your\_ip\_address>:<your\_port\_no>/permanent\_end\_points]

(for eg:

http://127.0.0.1:9889/auth\_oauth/oea

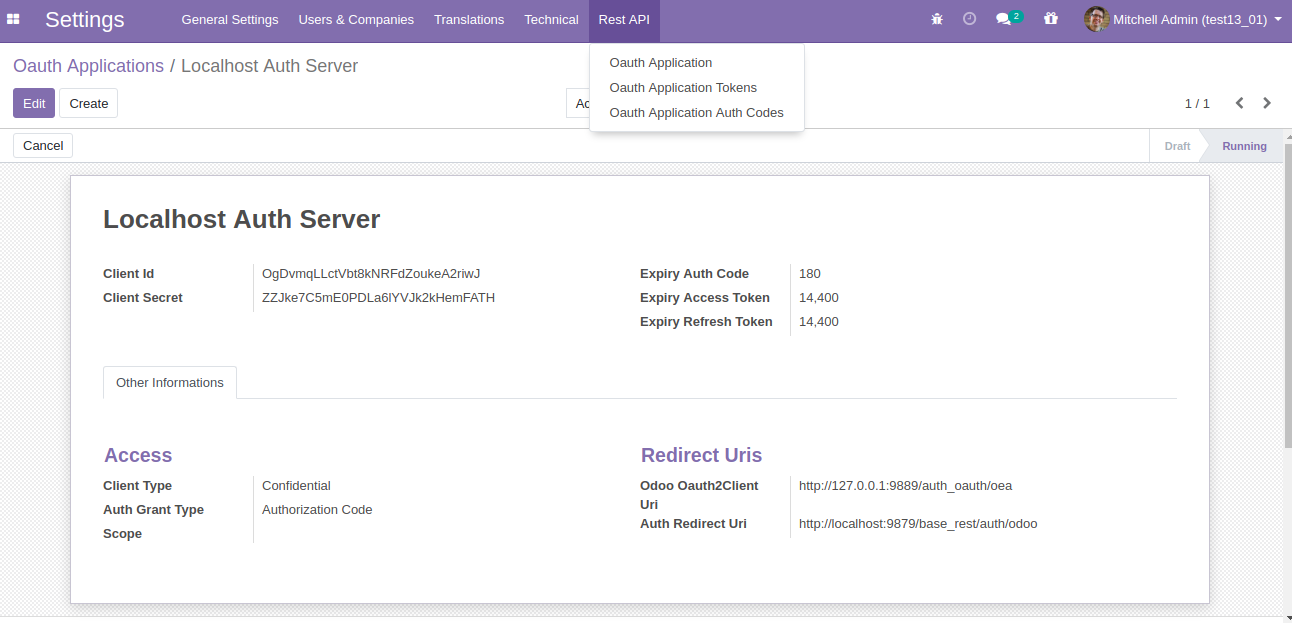
can change to

http://192.168.1.41:8879/auth\_oauth/oea

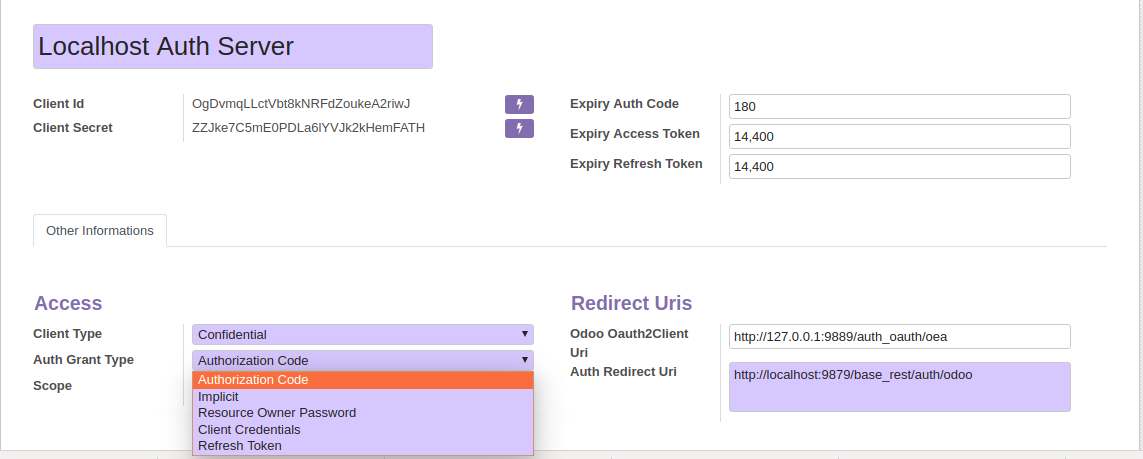
or http://localhost:9879/base\_rest/auth/odoo

can change to

http://10.0.0.11:8875/base\_rest/auth/odoo)

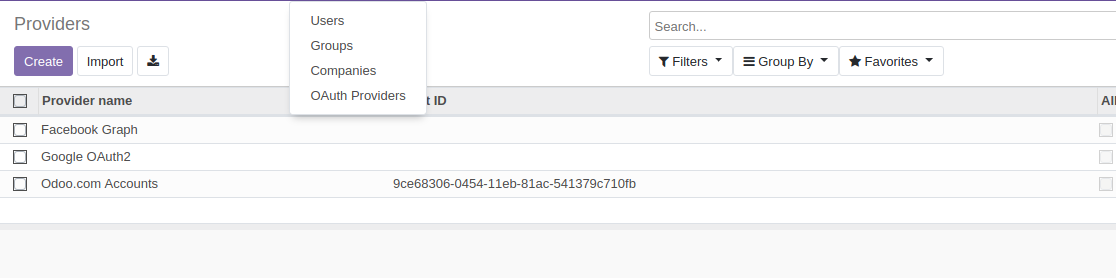
****

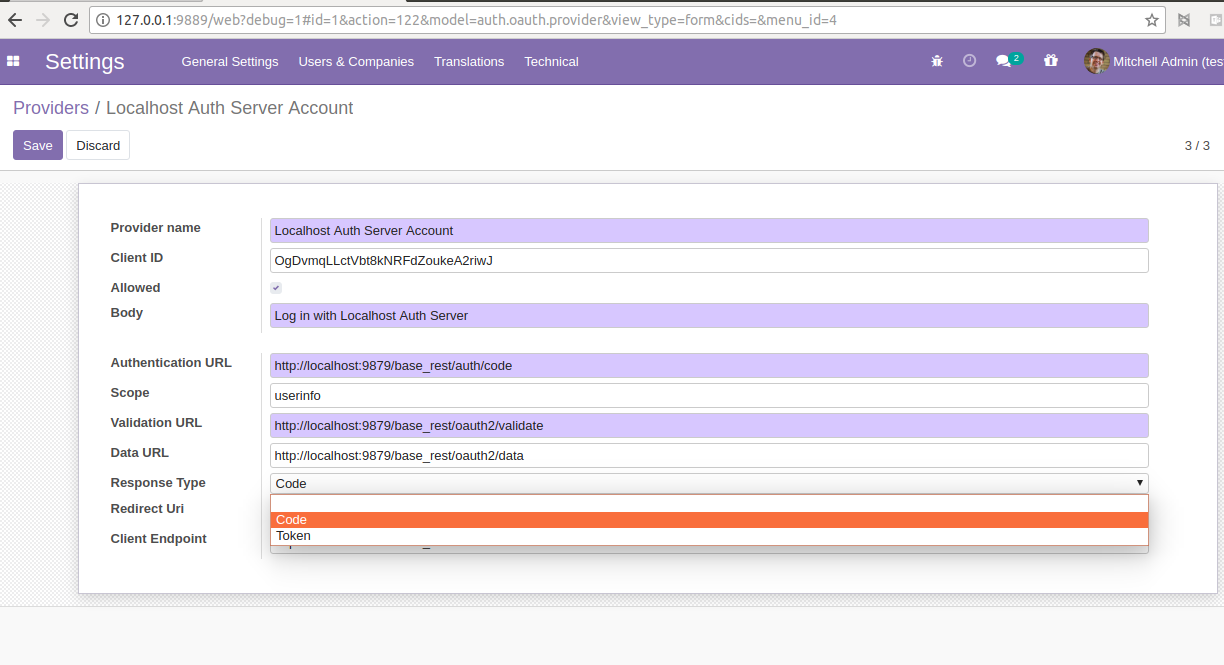
Configure the Grant type, redirect URL, Odoo , etc



For using an Odoo Oauth2 client (different server and db)

Do the following configuration

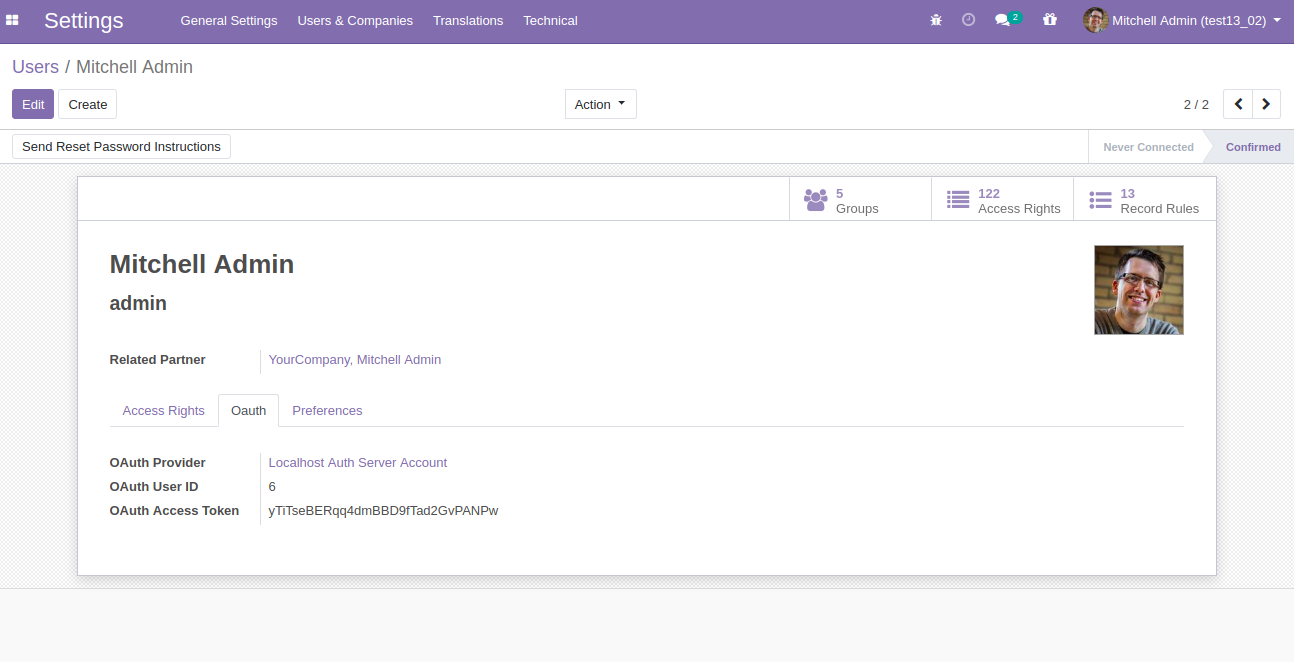
For time being choose Odoo.com Accounts and change the details.



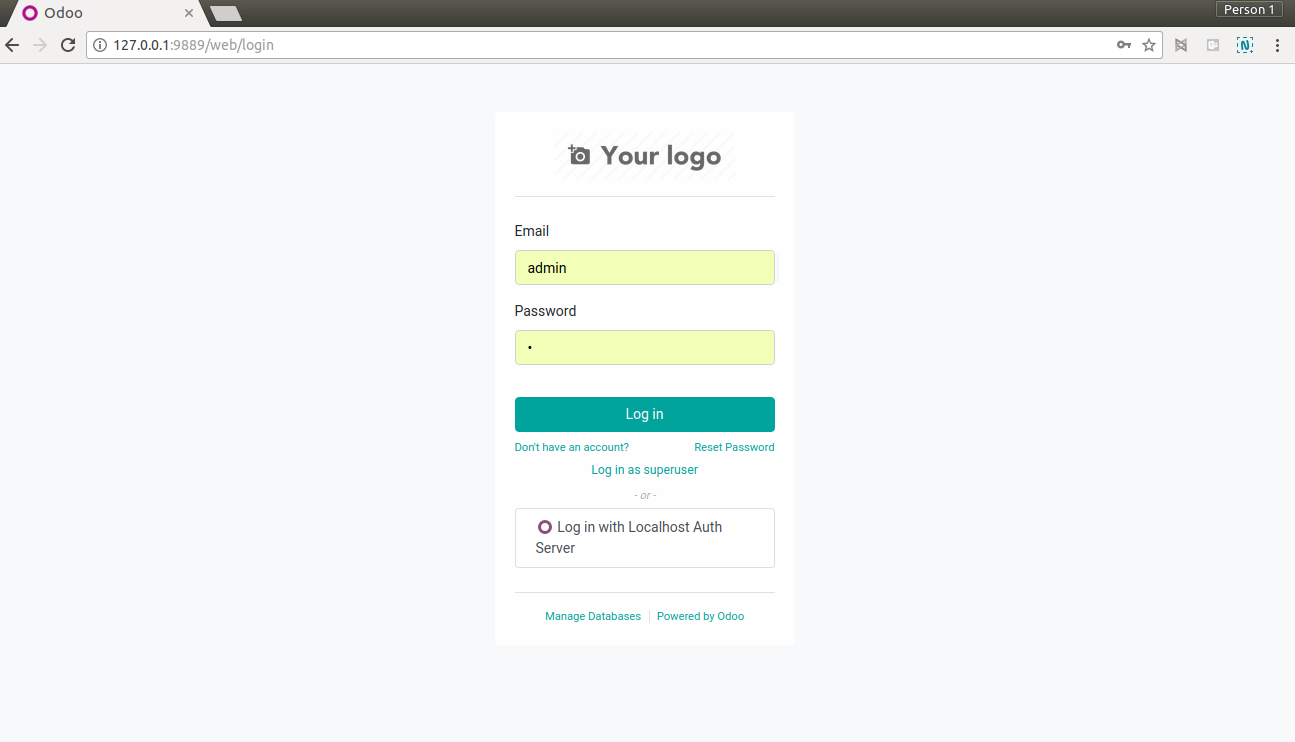
Configure the given details. Also please note that, Client ID should be obtained from Oauth2 Authorization Server and Redirect URl and client Endpoint should be same as we configured in Oauth2 Authorization Server.

**For testing as Grant Type “Authorization Code” or “ Implicit”**

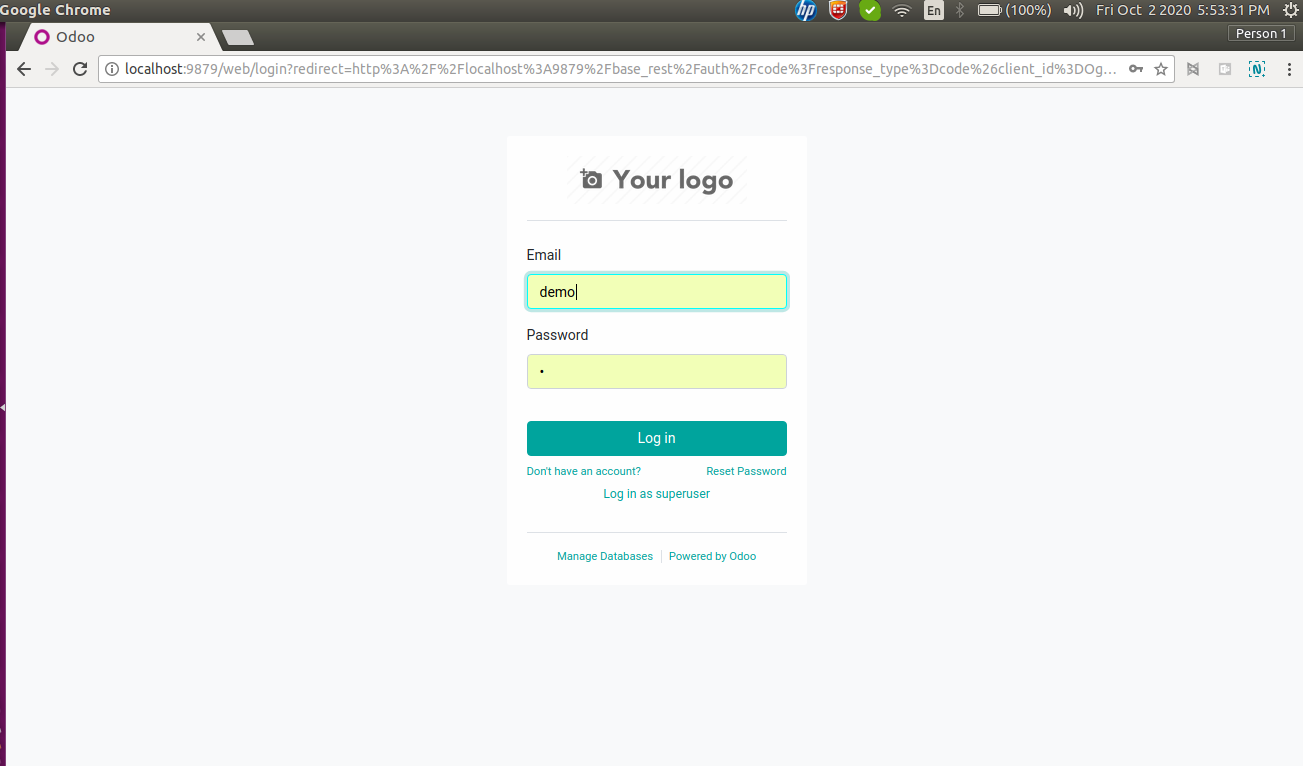
Go to Oauth2 Odoo Client server and DB instance

First need to map the users with Oauth2 Odoo client with the Authorization server user reference and logout

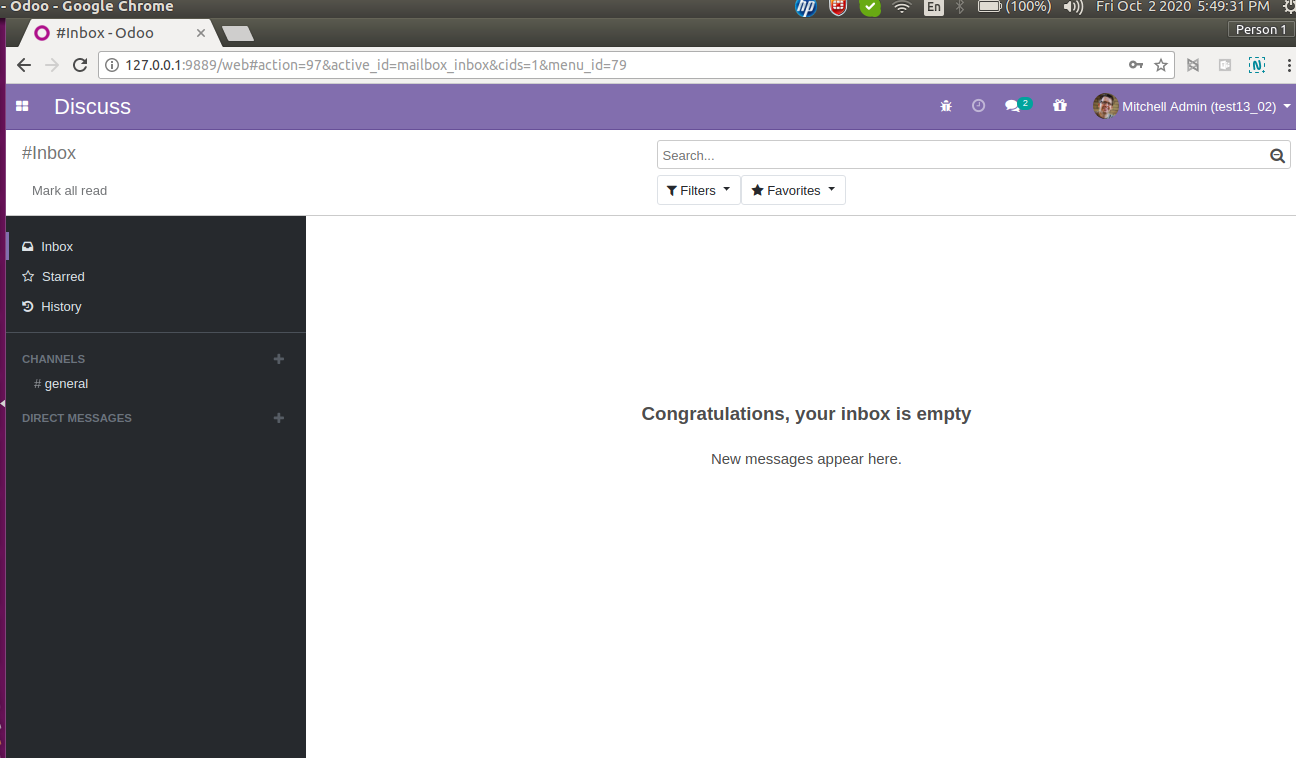
Login page of Oauth2client instance



Choose Login in with Localhost Authorization Server (here means another Odoo instance and different DB running on localhost)



After clicking that button the browser will redirect to Oauth2 Authorization server. The user need to login to **“Oauth2 Authorization server”** (here it is localhost) with the mapped user with Oauth2client (here user login called “demo” in Oauth2 server). We can see the URL is different and have the parameters from Oauth2client.



After successful login the OAuth2 Authorization Server redirected back to Oauth2 Odoo Client and their it we can successfully logined.

Some screen shots of other Grant Types.

***Please be note that, if we need error details as description, then please enable dev\_mode in odoo cofiguration file (odoo-server.conf)***

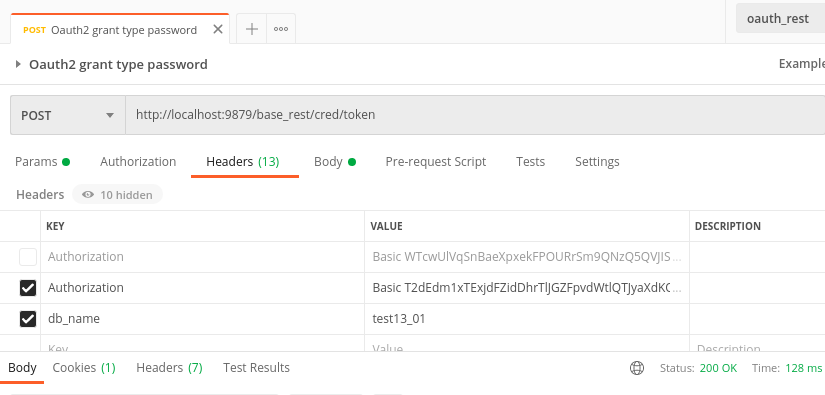
***###############################***

***[base\_rest]***

***dev\_mode=True***

***###############################***

**Grant Type Resource Owner Credentials.**



Params needed in header.

Authorization , db\_name

(Authorization Header should be the following format : Basic <BASE64({CLIENT ID}:{CLIENT SECRET})>

for eg:

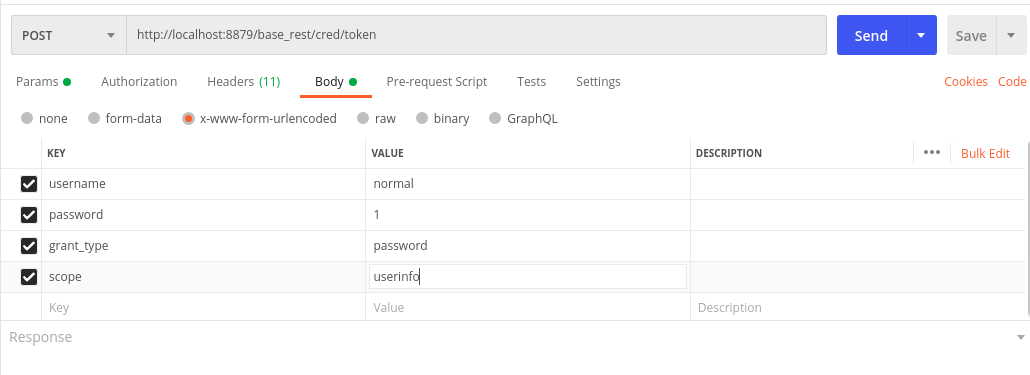
client\_id: OgDvmqLLctVbt8kNRFdZoukeA2riwJ

client\_secret: OgDvmqLLctVbt8kNRFdZoukeA2riwJ

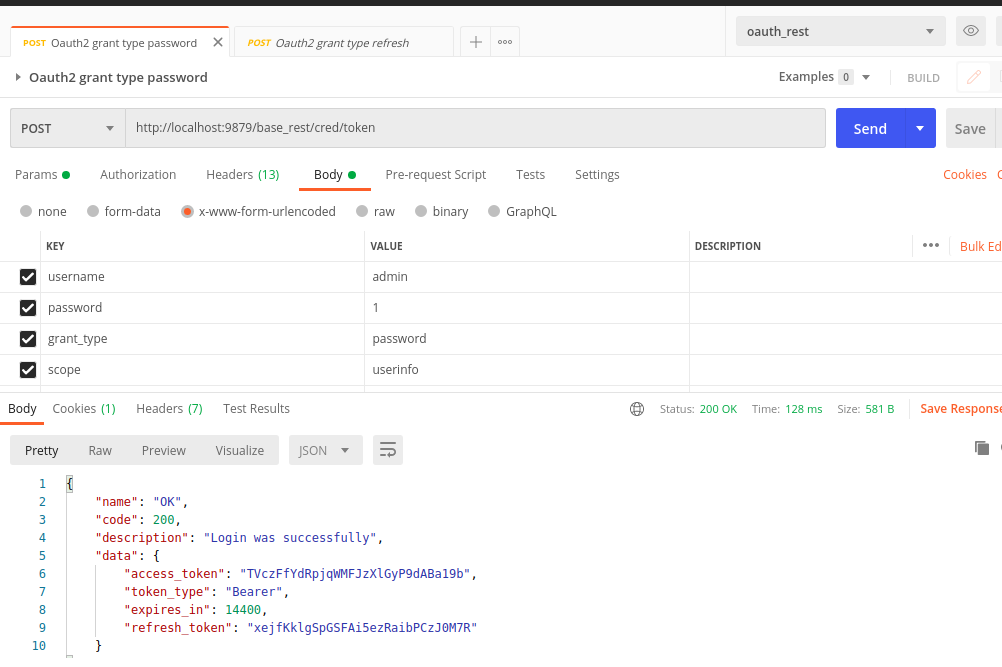
(OgDvmqLLctVbt8kNRFdZoukeA2riwJ:ZZJke7C5mE0PDLa6lYVJk2kHemFATH) should be converted to base65 encoding

T2dEdm1xTExjdFZidDhrTlJGZFpvdWtlQTJyaXdKOlpaSmtlN0M1bUUwUERMYTZsWVZKazJrSGVtRkFUSA==

Authorization: Basic T2dEdm1xTExjdFZidDhrTlJGZFpvdWtlQTJyaXdKOlpaSmtlN0M1bUUwUERMYTZsWVZKazJrSGVtRkFUSA==

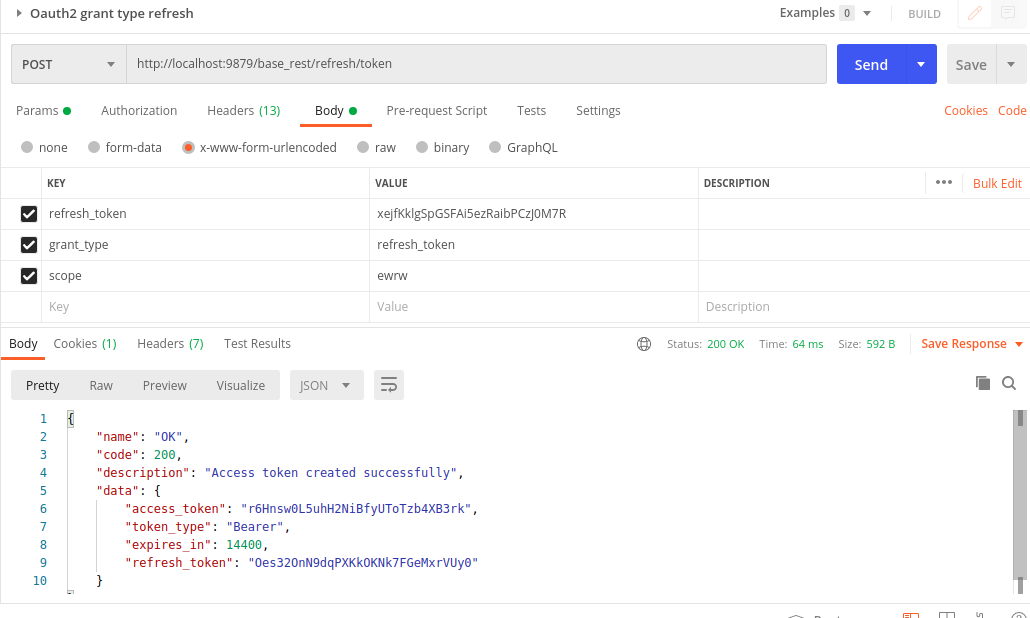


Params needed in form data.



Result Fetched.

**Grant Type Refresh Token**



Params in form data and result fetched. Header should be same as previous (resource owner grant type)

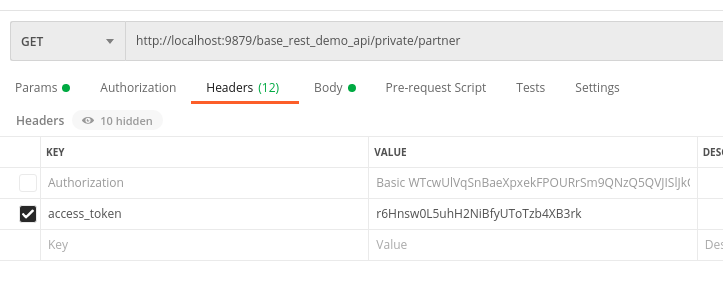
**Login with Access Token**

Sample taken from the module “base\_rest\_demo” private service.



The variable “default\_auth” needs to set to “access\_token”.

We noticed that the module “base\_rest” have some bugs. We rectified it and after that it works.

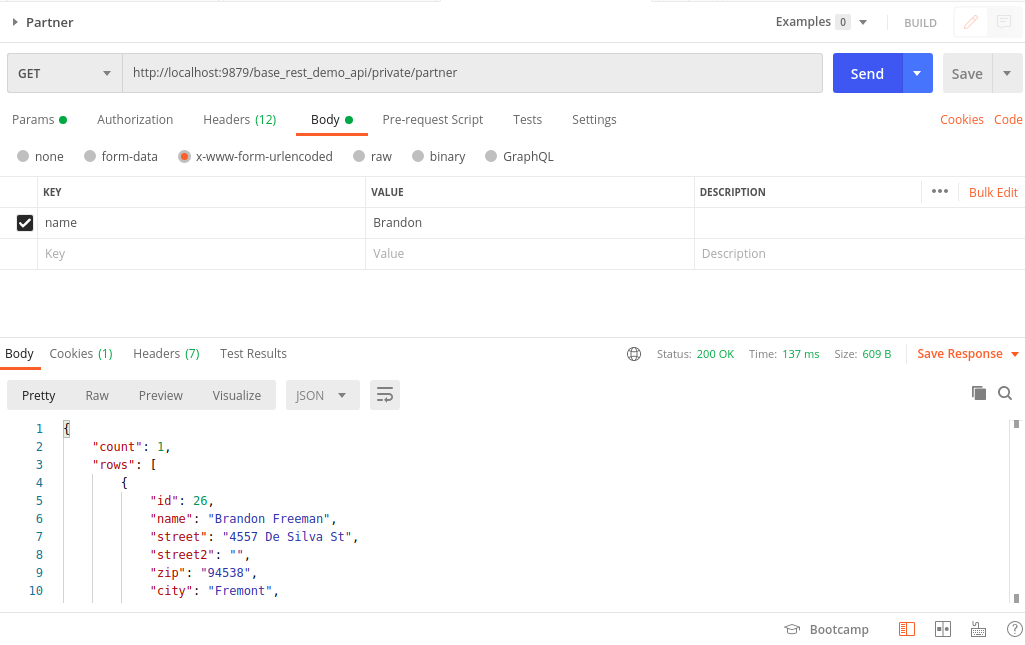


params needed in header

access\_token , db\_name

eg: access\_token: SoCjtUHs2HfpqcjAJO6D56THOayugM

db\_name: test13\_01



params needed in form data and result fetched.

**Known Issues:**

* Need to write all unit tests
* Didnt tested properly. Tested only during the development time.
* Sorry for the document as we prepared in a hurry

**Future Plans**

* Authorization Code with PKCE flow
* Need to work on client type and scope
* A generic solution for fetching the data from models