

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)

5. Go to settings menu and click on Activate the Developer mode.
6. Now go to apps menu and Click on 'Update Modules List'.
7. Remove the app's filter and search for the the word 'oauth2_auth_redirect'.
8. 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)

The screenshot shows the Odoo Settings application with the 'Rest API' menu selected. The 'Oauth Applications / Localhost Auth Server' configuration page is displayed. The page includes a 'Create' button and a 'Cancel' button. The configuration details are as follows:

Localhost Auth Server	
Client Id	OgDvmqLLctVbt8kNRfdZoukeA2riwJ
Client Secret	ZZJke7C5mE0PDLa6iYVJk2kHemFATH
Expiry Auth Code	180
Expiry Access Token	14,400
Expiry Refresh Token	14,400

Other Informations

Access	
Client Type	Confidential
Auth Grant Type	Authorization Code
Scope	

Redirect Uris	
Odoo Oauth2Client Uri	http://127.0.0.1:9889/auth_oauth/oea
Auth Redirect Uri	http://localhost:9879/base_rest/auth/odoo

Configure the Grant type, redirect URL, Odoo , etc

Localhost Auth Server

Client Id

OgDvmqLLctVbt8kNRFdZoukeA2riwJ

Client Secret

ZZJke7C5mE0PDLa6lYVJk2kHemFATH

Expiry Auth Code

180

Expiry Access Token

14,400

Expiry Refresh Token

14,400

Other Informations

Access

Client Type

Confidential

Auth Grant Type

Authorization Code

Scope

Authorization Code

Implicit

Resource Owner Password

Client Credentials

Refresh Token

Redirect Uris

Odoo OAuth2Client Uri

http://127.0.0.1:9889/auth_oauth/oea

Auth Redirect Uri

http://localhost:9879/base_rest/auth/odoo

For using an Odoo OAuth2 client (different server and db)

Do the following configuration

Providers			Search...	
<div>Create Import</div>			<div> <div>Filters</div> <div>Group By</div> <div>Favorites</div> </div>	
<input type="checkbox"/>	Provider name	ID	All	
<input type="checkbox"/>	Facebook Graph			
<input type="checkbox"/>	Google OAuth2			
<input type="checkbox"/>	Odoo.com Accounts	9ce68306-0454-11eb-81ac-541379c710fb		

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

The screenshot shows the 'Providers / Localhost Auth Server Account' configuration page in Odoo. The page has a purple header with 'Settings' and navigation links: 'General Settings', 'Users & Companies', 'Translations', and 'Technical'. The user 'Mitchell Admin (test13_02)' is logged in. Below the header, there are 'Save' and 'Discard' buttons. The main form contains the following fields:

- Provider name:** Localhost Auth Server Account
- Client ID:** OgDvmqLLctVbt8kNRFdZoukeA2riwJ
- Allowed:** ☒
- Body:** Log in with Localhost Auth Server
- Authentication URL:** http://localhost:9879/base_rest/auth/code
- Scope:** userinfo
- Validation URL:** http://localhost:9879/base_rest/oauth2/validate
- Data URL:** http://localhost:9879/base_rest/oauth2/data
- Response Type:** Code (selected from a dropdown menu)
- Redirect Uri:** Code (selected from a dropdown menu)
- Client Endpoint:** Token (selected from a dropdown menu)

Configure the given details. Also please note that, Client ID should be obtained from OAuth2 Authorization Server and Redirect URI and client Endpoint should be same as we configured in OAuth2 Authorization Server.

For testing as Grant Type "Authorization Code" or " Implicit"

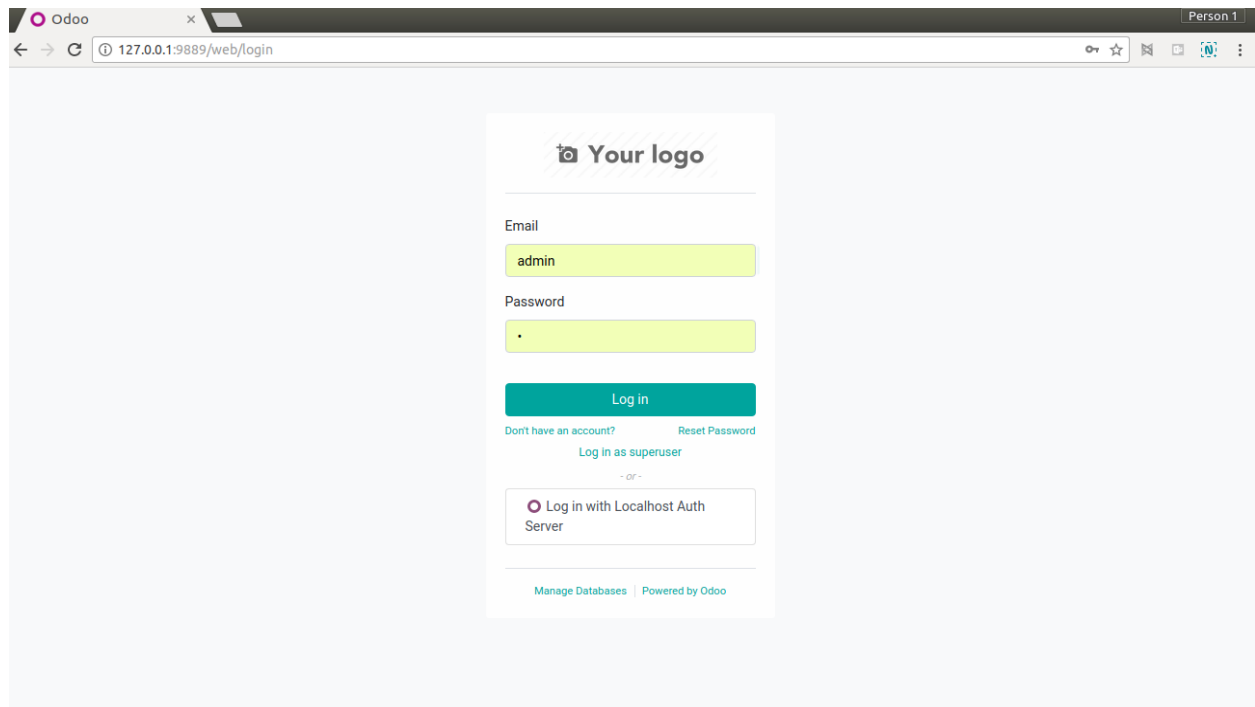
Go to OAuth2 Odoo Client server and DB instance

The screenshot shows the 'Users / Mitchell Admin' profile page in Odoo. The page has a purple header with 'Settings' and navigation links: 'General Settings', 'Users & Companies', 'Translations', and 'Technical'. The user 'Mitchell Admin (test13_02)' is logged in. Below the header, there are 'Edit' and 'Create' buttons. The main content area shows the user's profile information:

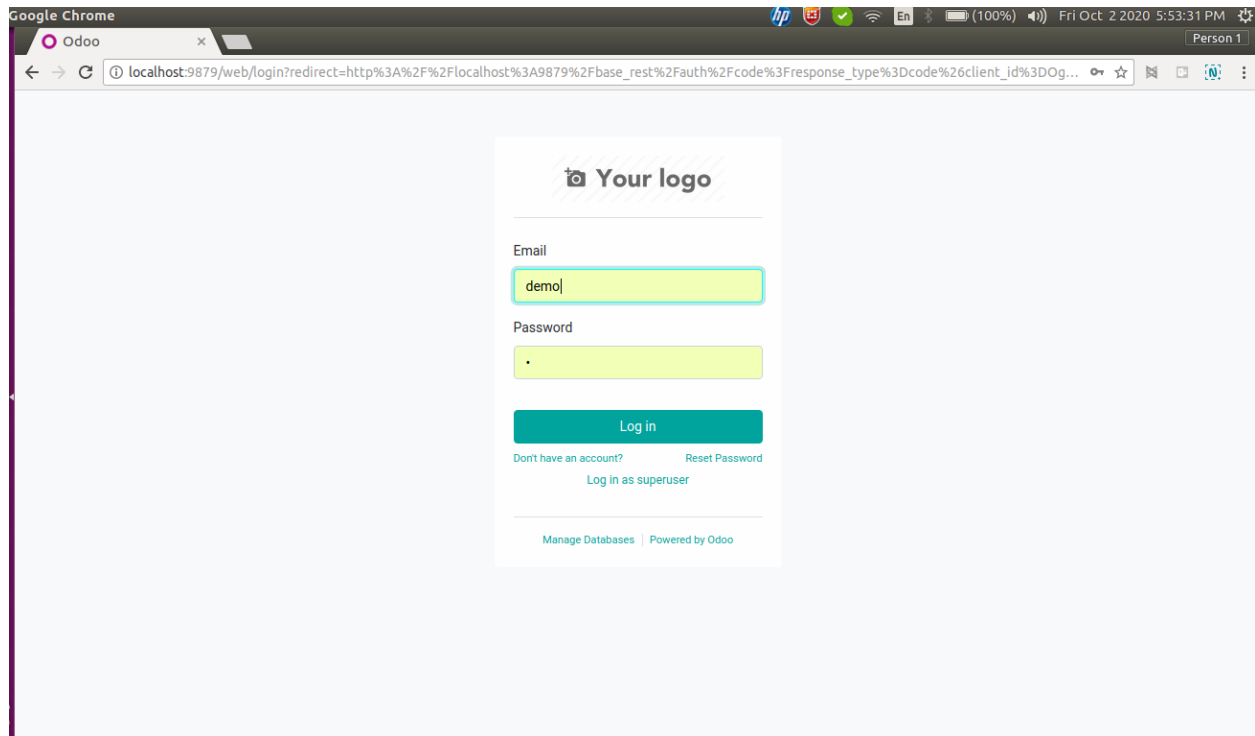
- Name:** Mitchell Admin
- Username:** admin
- Related Partner:** YourCompany, Mitchell Admin
- Access Rights:** 122
- Record Rules:** 13
- OAuth Provider:** Localhost Auth Server Account
- OAuth User ID:** 6
- OAuth Access Token:** yTITseBERqq4dmBBD9fTad2GvPANPw

First need to map the users with OAuth2 Odoo client with the Authorization server user reference and logout

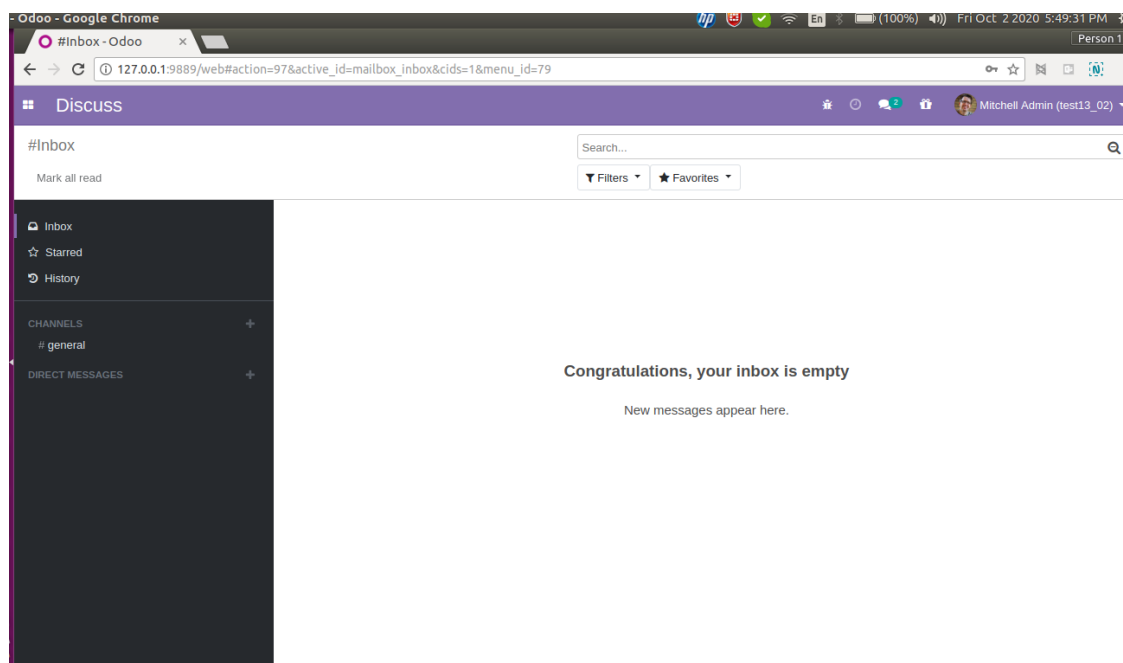
Login page of Oauth2client instance



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 then 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.

The screenshot shows a REST client interface with a POST request to `http://localhost:9879/base_rest/cred/token`. The request has three headers: `Authorization` (unchecked), `Authorization` (checked, value: `Basic T2dEdm1xTExdFZidDhrTlJGZFpvdWtQTjyaXdkKC...`), and `db_name` (checked, value: `test13_01`). The status is 200 OK and the time is 128 ms.

KEY	VALUE	DESCRIPTION
<input type="checkbox"/> Authorization	Basic WTcwUlVqSnBaeXpxekFPOURrSm9QNzQ5QVJJS ...	
<input checked="" type="checkbox"/> Authorization	Basic T2dEdm1xTExdFZidDhrTlJGZFpvdWtQTjyaXdkKC...	
<input checked="" type="checkbox"/> db_name	test13_01	

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

T2dEdm1xTExjdFZidDhrTIJGZFpvdWtIQTJyaXdKOlpaSmtlN0M1bUUwUERMvYTsWVZKazJrSGVtRkFUSA==

Authorization: Basic

T2dEdm1xTExjdFZidDhrTIJGZFpvdWtIQTJyaXdKOlpaSmtlN0M1bUUwUERMvYTsWVZKazJrSGVtRkFUSA==

The screenshot shows a REST client interface with a POST request to `http://localhost:8879/base_rest/cred/token`. The 'Body' tab is selected, and the 'x-www-form-urlencoded' radio button is chosen. The form data is as follows:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	normal	
<input checked="" type="checkbox"/> password	1	
<input checked="" type="checkbox"/> grant_type	password	
<input checked="" type="checkbox"/> scope	userinfo	
Key	Value	Description

Below the table, there is a 'Response' section with a dropdown arrow.

Params needed in form data.

POST Oauth2 grant type password X POST Oauth2 grant type refresh + ... oauth_rest

▶ Oauth2 grant type password Examples 0 BUILD

POST http://localhost:9879/base_rest/cred/token Send Save

Params Authorization Headers (13) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Ed
<input checked="" type="checkbox"/>	username	admin			
<input checked="" type="checkbox"/>	password	1			
<input checked="" type="checkbox"/>	grant_type	password			
<input checked="" type="checkbox"/>	scope	userinfo			

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 128 ms Size: 581 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "name": "OK",
3   "code": 200,
4   "description": "Login was successfully",
5   "data": {
6     "access_token": "TVczFfYdRpjqWMFJzXlGyP9dABa19b",
7     "token_type": "Bearer",
8     "expires_in": 14400,
9     "refresh_token": "xejfkKlgSpGSFAi5ezRaibPCzJ0M7R"
10  }
```

Result Fetched.

Grant Type Refresh Token

OAuth2 grant type refresh

Examples 0 | BUILD

POST http://localhost:9879/base_rest/refresh/token Send Save

Params Authorization Headers (13) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	refresh_token	xejfkKlgSpGSFAi5ezRaibPCzJOM7R			
<input checked="" type="checkbox"/>	grant_type	refresh_token			
<input checked="" type="checkbox"/>	scope	ewrw			
	Key	Value	Description		

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 64 ms Size: 592 B Save Response

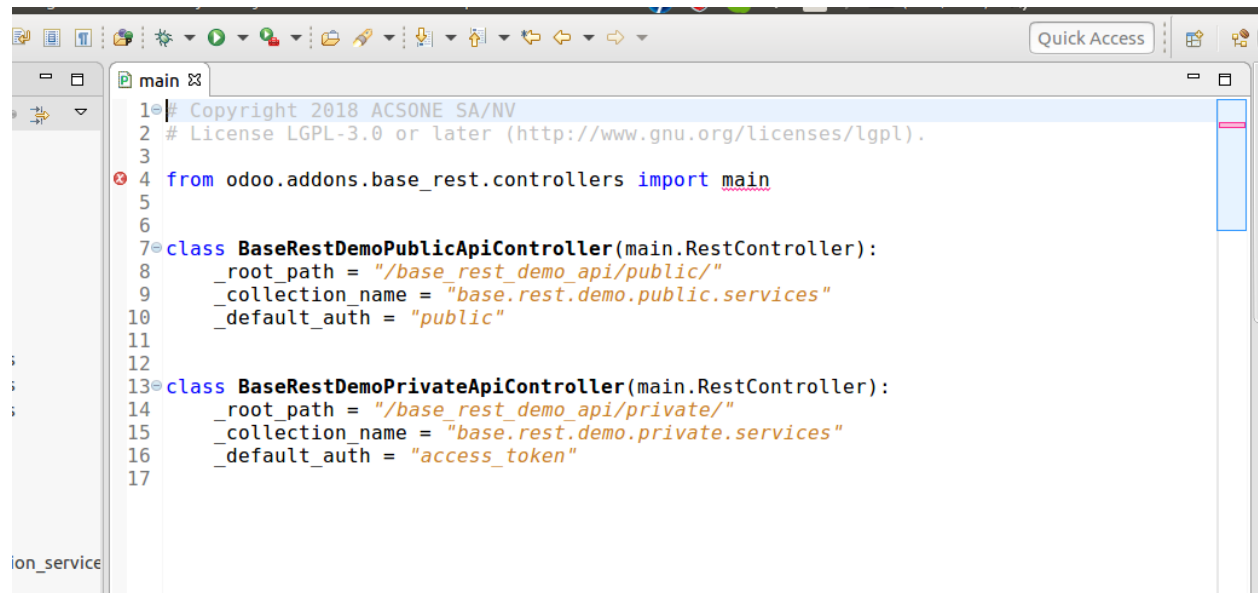
Pretty Raw Preview Visualize JSON

```
1 {
2   "name": "OK",
3   "code": 200,
4   "description": "Access token created successfully",
5   "data": {
6     "access_token": "r6Hnsw0L5uhH2NiBfyUToTzb4XB3rk",
7     "token_type": "Bearer",
8     "expires_in": 14400,
9     "refresh_token": "0es320nN9dqPXXk0KNk7FGeMxrVUy0"
10  }
```

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.

A screenshot of a code editor window. The editor has a toolbar at the top with various icons for file operations and development. Below the toolbar, there's a tab labeled 'main'. The code is written in Python and is part of an Odoo module. It includes a copyright notice for 2018 ACSONE SA/NV, licensed under LGPL-3.0. The code defines two classes: 'BaseRestDemoPublicApiController' and 'BaseRestDemoPrivateApiController', both inheriting from 'main.RestController'. The public controller has attributes for root_path, collection_name, and default_auth set to 'public'. The private controller has similar attributes but with default_auth set to 'access_token'.

```
1 # Copyright 2018 ACSONE SA/NV
2 # License LGPL-3.0 or later (http://www.gnu.org/licenses/lgpl).
3
4 from odoo.addons.base_rest.controllers import main
5
6
7 class BaseRestDemoPublicApiController(main.RestController):
8     _root_path = "/base_rest_demo_api/public/"
9     _collection_name = "base.rest.demo.public.services"
10    _default_auth = "public"
11
12
13 class BaseRestDemoPrivateApiController(main.RestController):
14     _root_path = "/base_rest_demo_api/private/"
15     _collection_name = "base.rest.demo.private.services"
16     _default_auth = "access_token"
17
```

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.

GET

http://localhost:9879/base_rest_demo_api/private/partner

Params

Authorization

Headers (12)

Body

Pre-request Script

Tests

Settings

Headers

10 hidden

	KEY	VALUE	DESCRIPTION
<input type="checkbox"/>	Authorization	Basic WTcwUlVqSnBaeXpxekFPOURrSm9QNzQ5QVJlSlJk	
<input checked="" type="checkbox"/>	access_token	r6Hnsw0L5uhH2NiBfyUToTzb4XB3rk	
	Key	Value	Description

params needed in header

access_token , db_name

eg: access_token: SoCjtUHs2HfpqcjAJO6D56TH0ayugM

db_name: test13_01

Partner

Examples 0

BUILD

GET

http://localhost:9879/base_rest_demo_api/private/partner

Send

Save

Params

Authorization

Headers (12)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/>	name	Brandon			
	Key	Value	Description		

Body

Cookies (1)

Headers (7)

Test Results

Status: 200 OK

Time: 137 ms

Size: 609 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1 {
2   "count": 1,
3   "rows": [
4     {
5       "id": 26,
6       "name": "Brandon Freeman",
7       "street": "4557 De Silva St",
8       "street2": "",
9       "zip": "94538",
10      "city": "Fremont",

```

Bootcamp

params needed in form data and result fetched.

Known Issues:

- ✗ Need to write all unit tests
- ✗ Didn't test 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