

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.
- Have the Grant Type Authorization Code with response type as "code" .
- Have the Grant Type Implicit with response type as "token".
- Have the Grant Type Resource Owner Credentials.
- Have the Grant Type Refresh Token.
- Have the login with "access_token"
- Can act an Odoo instance as client to Our Authorisatiion Server.

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.

(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 interface for the 'Localhost Auth Server'. The top navigation bar includes 'Settings', 'General Settings', 'Users & Companies', 'Translations', 'Technical', and 'Rest API'. The 'Rest API' menu is open, showing 'Oauth Application', 'Oauth Application Tokens', and 'Oauth Application Auth Codes'. The 'Localhost Auth Server' configuration page is displayed, showing fields for 'Client Id', 'Client Secret', 'Expiry Auth Code', 'Expiry Access Token', and 'Expiry Refresh Token'. Below these are sections for 'Access' (Client Type, Auth Grant Type, Scope) and 'Redirect Uris' (Odoo Oauth2Client Uri, Auth Redirect Uri).

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

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

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 client (enterly different server and db)

Do the following configuration

Providers

Create

Import

⤴

Users

Groups

Companies

OAuth Providers

Search...

⌵ Filters

≡ Group By

★ Favorites

<input type="checkbox"/>	Provider name	Client ID	All
<input type="checkbox"/>	Facebook Graph		<input type="checkbox"/>
<input type="checkbox"/>	Google OAuth2		<input type="checkbox"/>
<input type="checkbox"/>	Odoo.com Accounts	9ce68306-0454-11eb-81ac-541379c710fb	<input type="checkbox"/>

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

127.0.0.1:9889/web?debug=1#id=1&action=122&model=auth.oauth.provider&view_type=form&cids=&menu_id=4

Settings General Settings Users & Companies Translations Technical Mitchell Admin (tes)

Providers / Localhost Auth Server Account

Save Discard 3 / 3

Provider name	Localhost Auth Server Account
Client ID	OgDvmqLLctVbt8kNRFdZoukeA2riwJ
Allowed	<input checked="" type="checkbox"/>
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
Redirect Uri	Code
Client Endpoint	Token

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

Settings General Settings Users & Companies Translations Technical Mitchell Admin (test13_02)

Users / Mitchell Admin

Edit Create Action 2 / 2

Send Reset Password Instructions Never Connected Confirmed

5 Groups 122 Access Rights 13 Record Rules

Mitchell Admin
admin

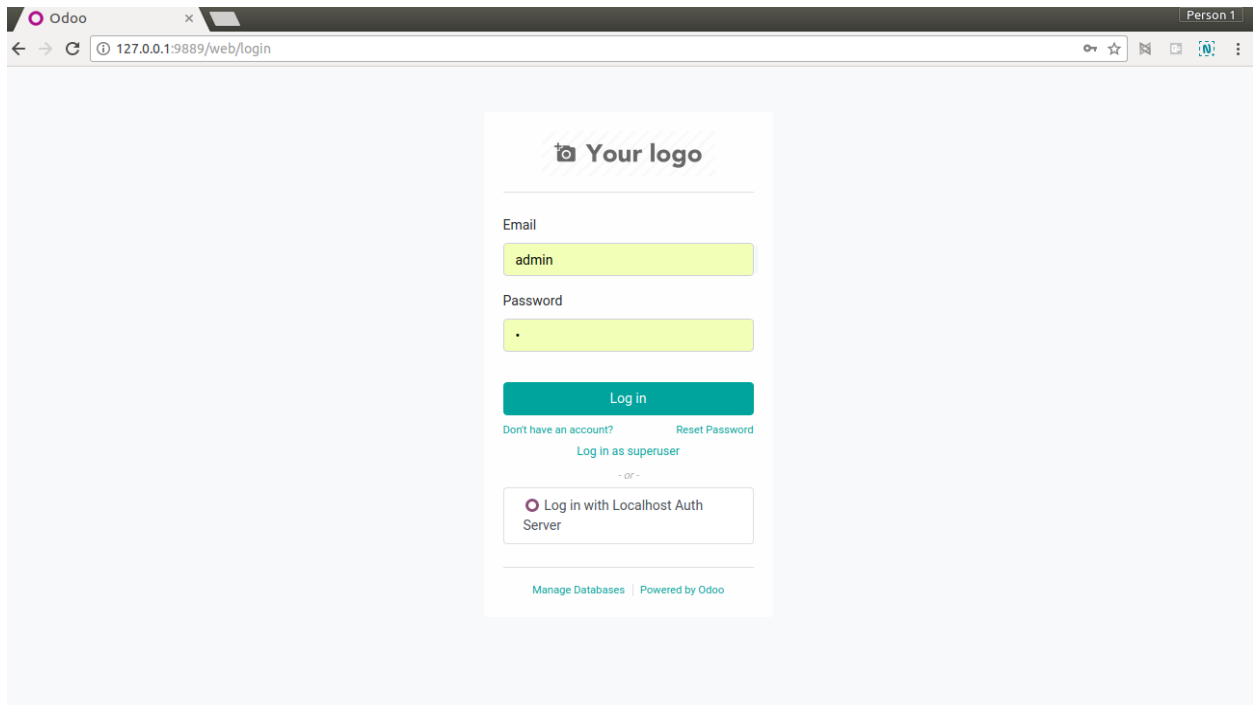
Related Partner YourCompany, Mitchell Admin

Access Rights Oauth Preferences

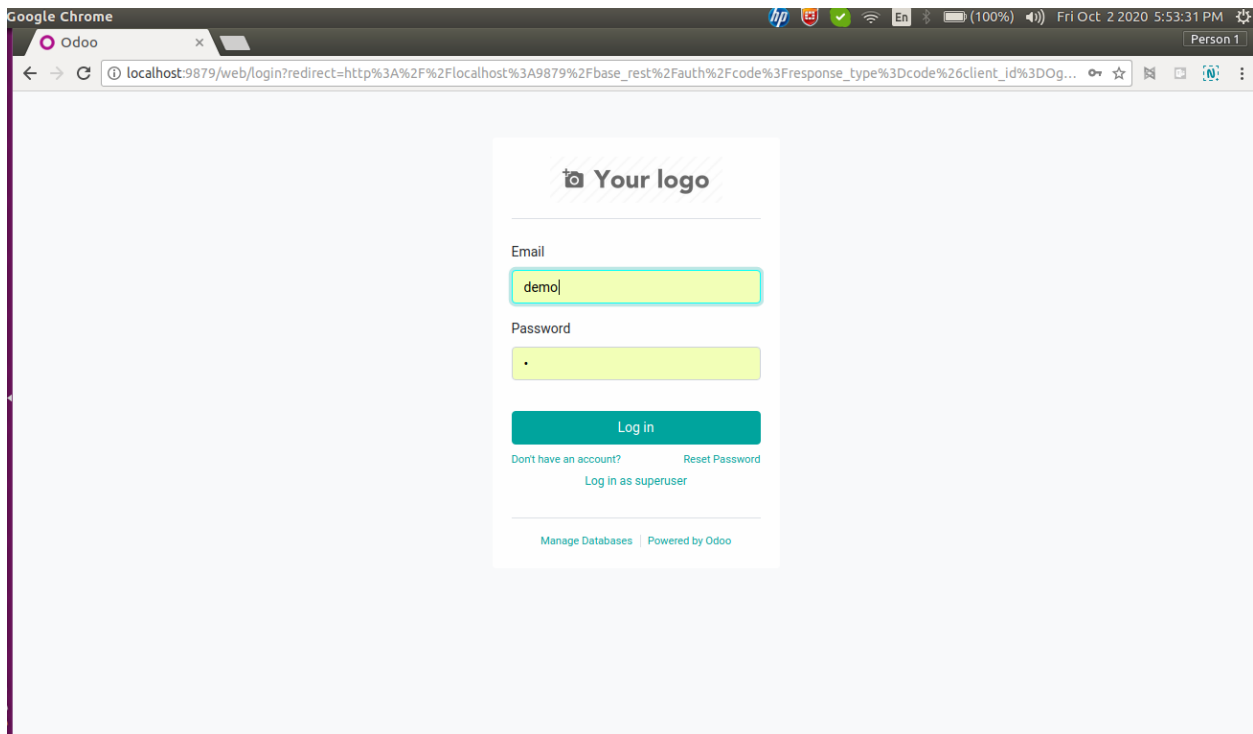
OAuth Provider Localhost Auth Server Account
OAuth User ID 6
OAuth Access Token yTITseBERqq4dmBBD9fTad2GVpANPw

First need to map the users with Oauth2 odoo client with the Authorisation server user reference

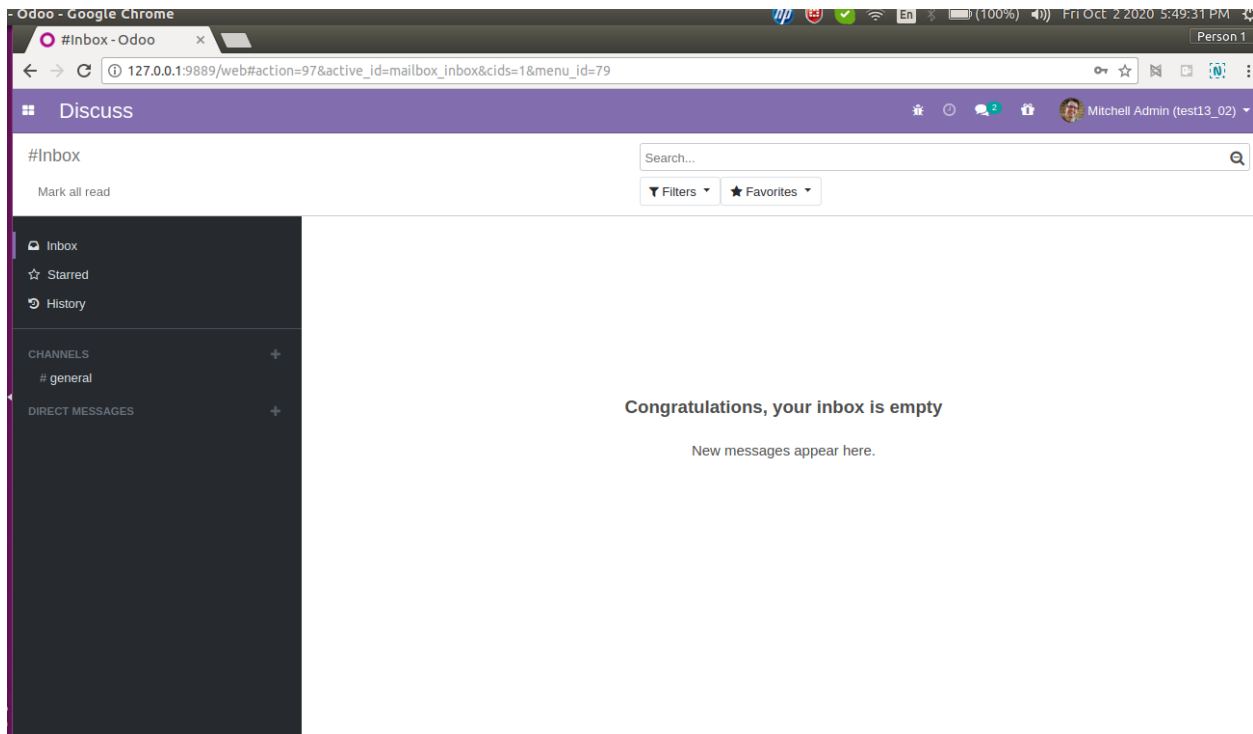
Then go to Oauth2 Odoo Client server and DB instance Login page



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 Authorisation server"** (here it is localhost) with the mapped user with OAuth2client (here user login called "demo" in OAuth2 server).



After successful login the OAuth2 Authorisation Server redirected back to OAuth2 Odoo Client and their it successfully login.

Some screen shots of other Grant Types.

Grant Type Resource Owner Credentials.

The screenshot shows a REST client interface with a tab labeled "POST OAuth2 grant type password" and a sub-tab "oauth_rest". The request URL is "http://localhost:9879/base_rest/cred/token". The "Headers" tab is selected, showing a table with 13 headers. The table has columns for KEY, VALUE, and DESCRIPTION. The headers are:

KEY	VALUE	DESCRIPTION
<input type="checkbox"/> Authorization	Basic WTcwUlVqSnBaeXpxekFPOURrSm9QNzQ5QVJIS ...	
<input checked="" type="checkbox"/> Authorization	Basic T2dEdm1xTExdFZidDhrTlJGZFpvdWtIQTJyaXdKC ...	
<input checked="" type="checkbox"/> db_name	test13_01	

At the bottom, the "Body" tab is selected, showing a table with 1 body. The table has columns for KEY, VALUE, and DESCRIPTION. The body is:

KEY	VALUE	DESCRIPTION

The status bar at the bottom shows "Status: 200 OK" and "Time: 128 ms".

Params needed in header. (Authorization Header should be the following format : Basic <BASE64({CLIENT ID}:{CLIENT SECRET})>

POST http://localhost:8879/base_rest/cred/token Send Save

Params Authorization Headers (11) **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"/> username	normal			
<input checked="" type="checkbox"/> password	1			
<input checked="" type="checkbox"/> grant_type	password			
<input checked="" type="checkbox"/> scope	userinfo			
Key	Value	Description		

Response

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 C

☐ 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": "xejFKlgSpGSFAi5ezRaibPCzJ0M7R"
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	xejfKklgSpGSFAI5ezRaibPCzj0M7R			
<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

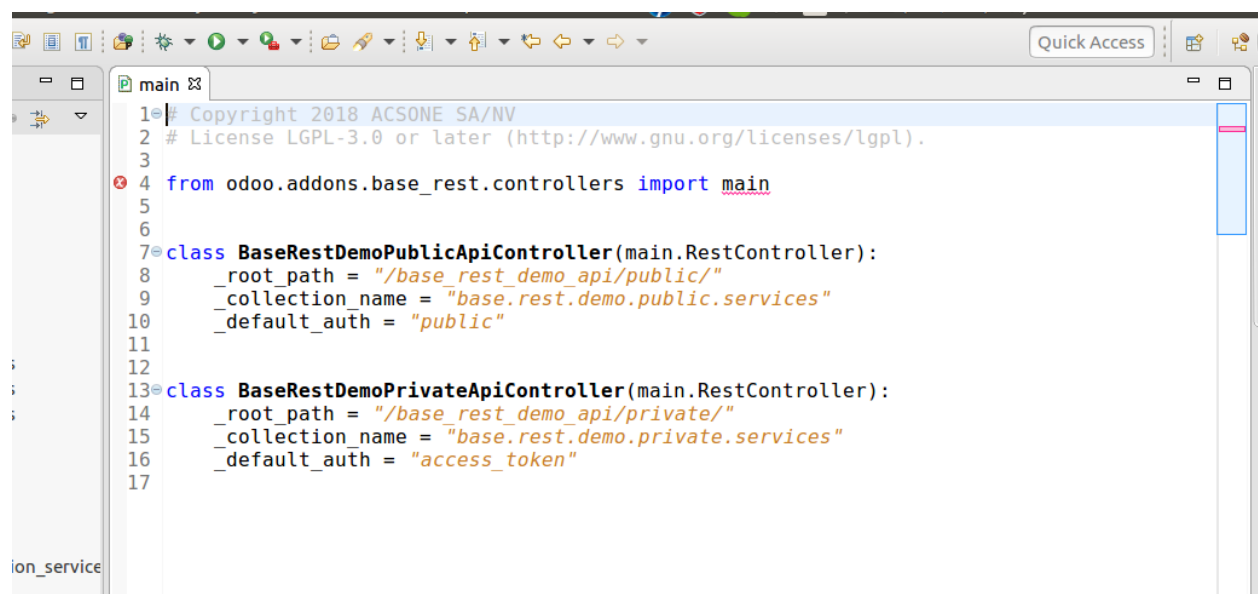
Pretty Raw Preview Visualize JSON

```
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": "0es320nN9dqPXKk0KNk7FGeMxrVUy0"
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.



```
1# Copyright 2018 ACSONE SA/NV
2# License LGPL-3.0 or later (http://www.gnu.org/licenses/lgpl).
3
4from odoo.addons.base_rest.controllers import main
5
6
7class 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
13class 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
```

lon_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.

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	DES
<input type="checkbox"/>	Authorization	Basic WTcwUlVqSnBaeXpxekFPOURrSm9QNzQ5QVJlSljk	
<input checked="" type="checkbox"/>	access_token	r6Hnsw0L5uhH2NiBfyUToTzb4XB3rk	
	Key	Value	De:

params needed in header

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 Cookies Code

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 unit tests
- ✗ Didn't test properly. Tested only during the development time.
- ✗ Sorry for the document as we prepared in a hurry

Future Plans

- ✓ Need to work on client type and scope
- ✓ A generic solution for fetching the data from models