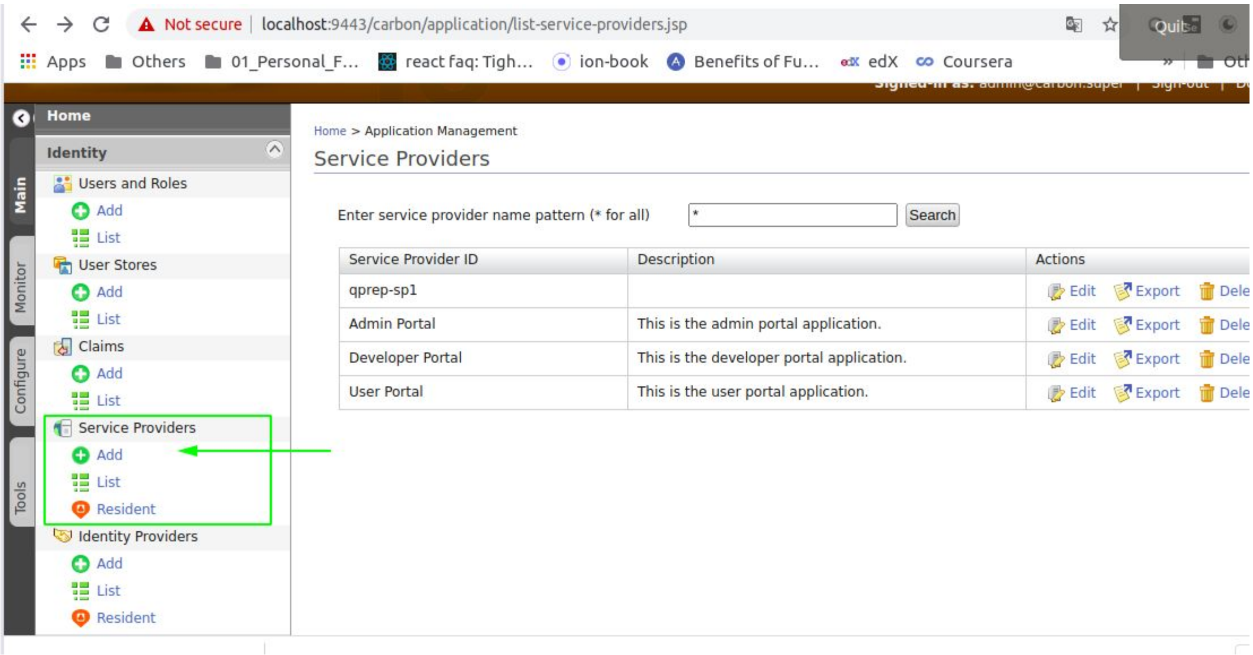
ServiceProvider Registration

Identity Server configuration (wso2-is v5.11.0)

In side menu of wso2 identity server administration console, select ServiceProvider/add option

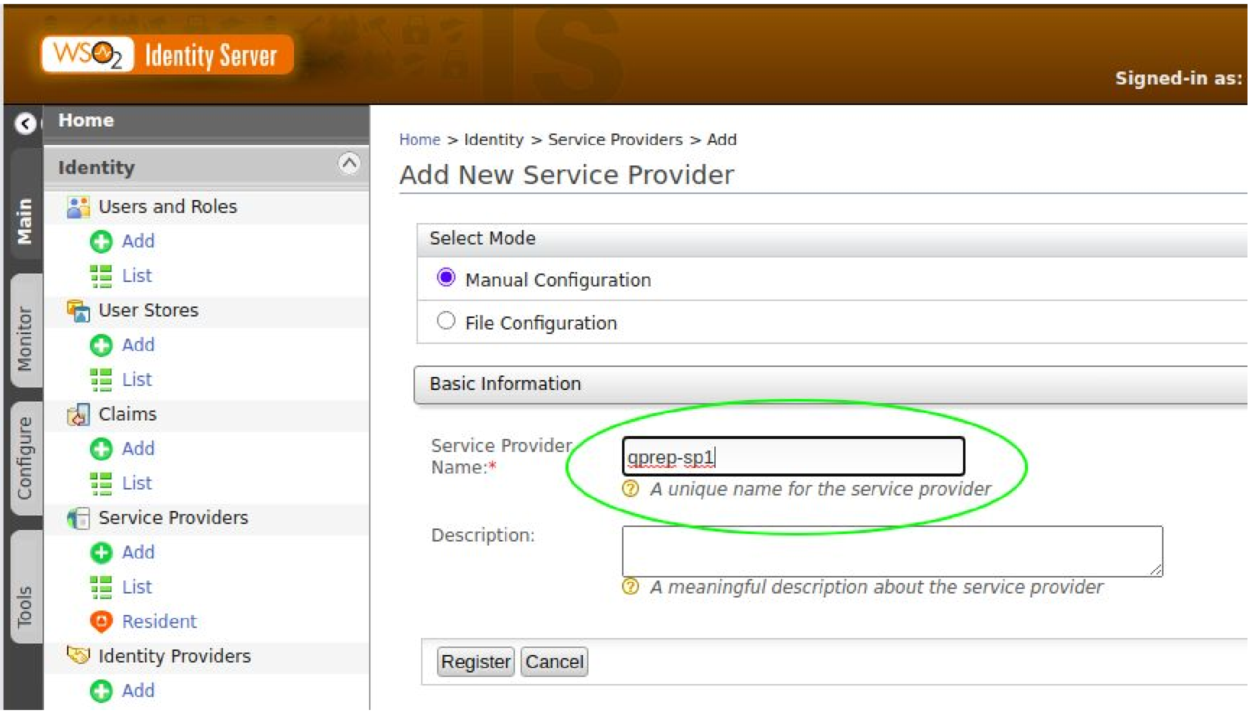


Service Providers

Enter service provider name pattern (* for all) * Search

Service Provider ID	Description	Actions
qprep-sp1		Edit Export Dele
Admin Portal	This is the admin portal application.	Edit Export Dele
Developer Portal	This is the developer portal application.	Edit Export Dele
User Portal	This is the user portal application.	Edit Export Dele

Add a brand new sp declaration



Add New Service Provider

Select Mode

● Manual Configuration

○ File Configuration

Basic Information

Service Provider Name:* qprep-sp1

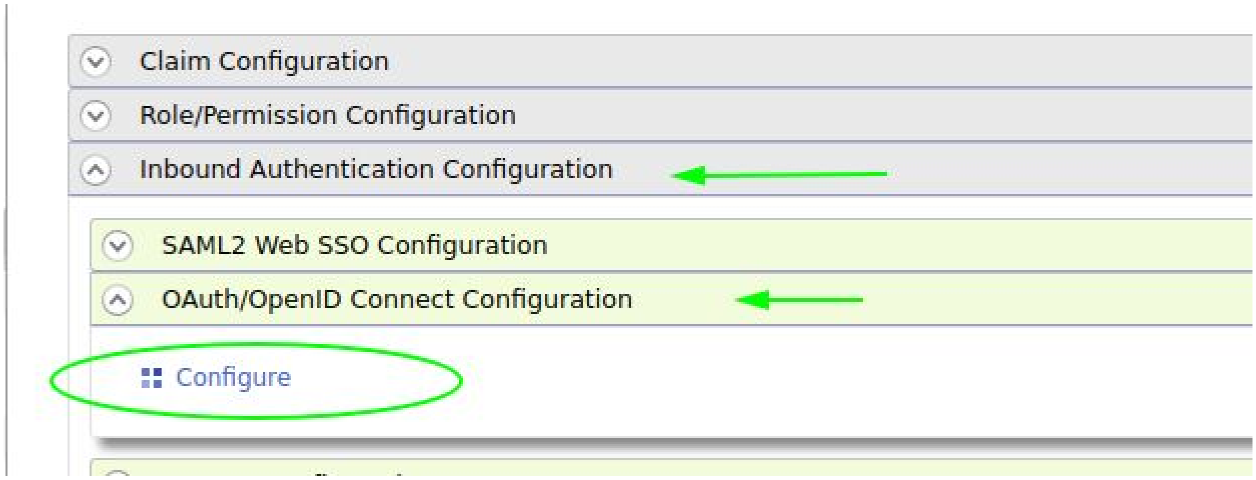
A unique name for the service provider

Description:

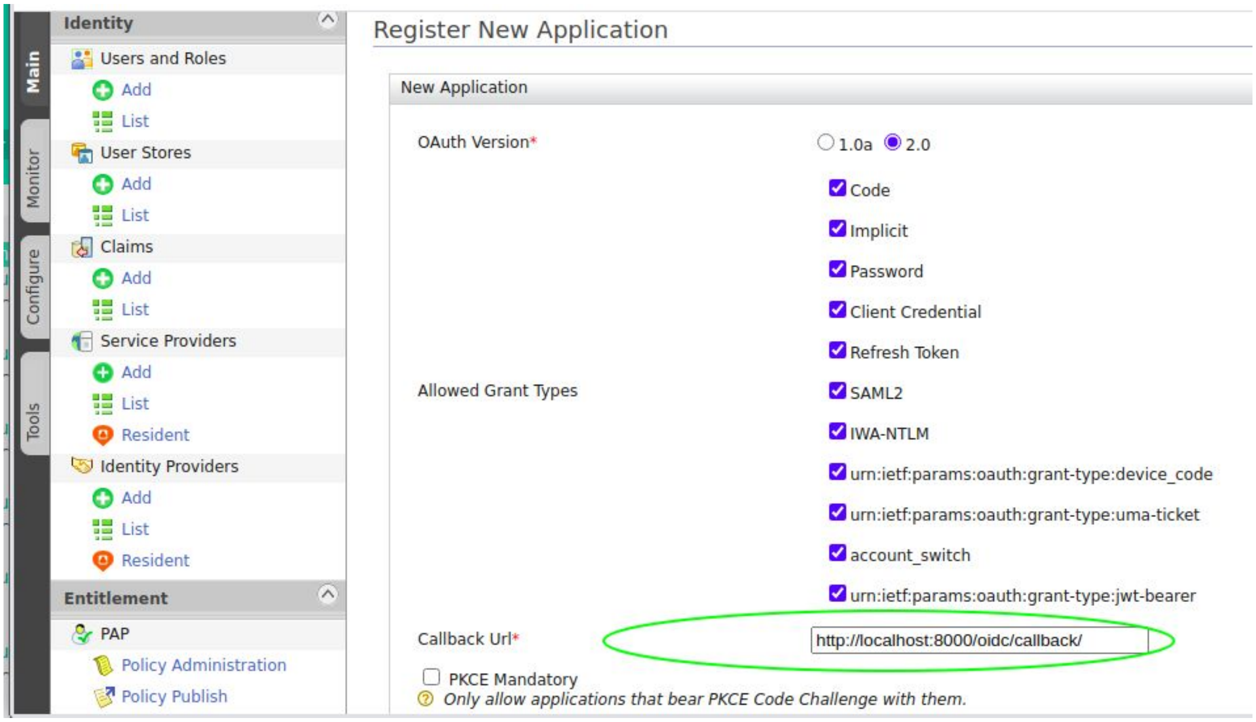
A meaningful description about the service provider

Register Cancel

Go to the next tab, where is requested information of Oauth/OpenIdConnect (expand the options to go to details). Then click on "configure" option


Claim Configuration
Role/Permission Configuration
Inbound Authentication Configuration
SAML2 Web SSO Configuration
OAuth/OpenID Connect Configuration
Configure

A new dialog opens, that let we can register a new application
Fill the callback url field



Register New Application

New Application

OAuth Version* ○ 1.0a ● 2.0

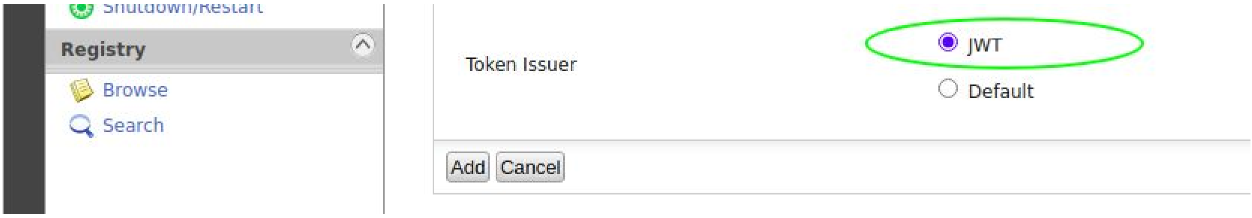
Allowed Grant Types
- ☑ Code
- ☑ Implicit
- ☑ Password
- ☑ Client Credential
- ☑ Refresh Token
- ☑ SAML2
- ☑ IWA-NTLM
- ☑ urn:ietf:params:oauth:grant-type:device_code
- ☑ urn:ietf:params:oauth:grant-type:uma-ticket
- ☑ account_switch
- ☑ urn:ietf:params:oauth:grant-type:jwt-bearer

Callback Url* http://localhost:8000/oidc/callback/

□ PKCE Mandatory

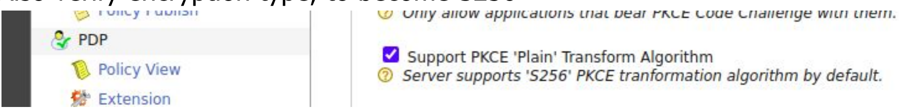
Only allow applications that bear PKCE Code Challenge with them.

And in the issuer token field, select JWT



Token Issuer ● JWT ○ Default

Verify that option "Use SP JWKS endpoint" is selected (if it is visible. It could change in future versions)
Also verify encryption type, to become S256

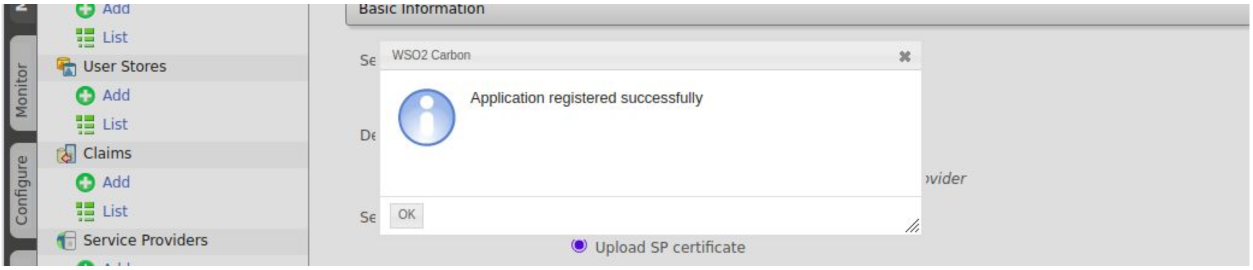


Only allow applications that bear PKCE Code Challenge with them.

☑ Support PKCE 'Plain' Transform Algorithm

Server supports 'S256' PKCE tranformation algorithm by default.

Click in "add" button



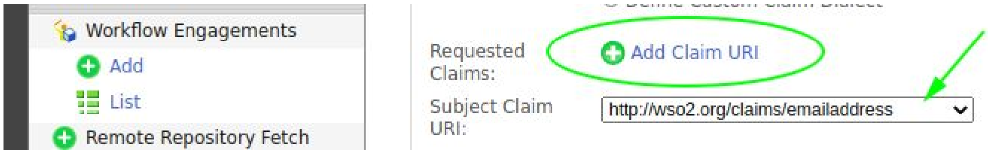
WSO2 Carbon

Application registered successfully

● Upload SP certificate

So now we have created the sp.
Now, we must create the claims.
For both django application and mozilla-django-oidc, name and email are required, so we aregoing to include both fields inside the claim usage of this service provider.
So we go to the claims section, and select:

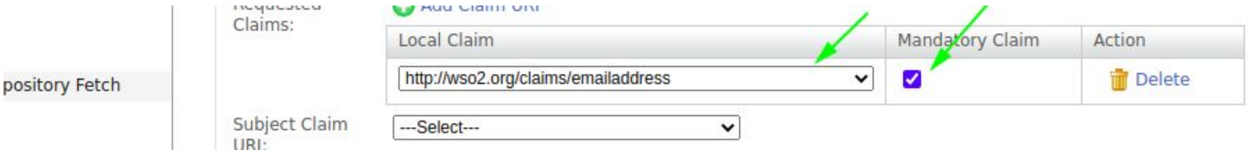
- http://wso2.org/claims/emailaddress . Then click on "dd claim URI" option.
- http://wso2.org/claims/username. Then click on "dd claim URI" option



Requested Claims:

Subject Claim URI: http://wso2.org/claims/emailaddress

When an additional panel is displayed we mark those claims as required

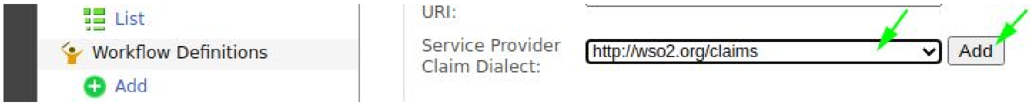


Requested Claims:

Local Claim	Mandatory Claim	Action
http://wso2.org/claims/emailaddress	☑	Delete

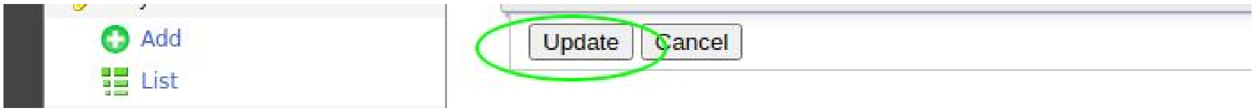
Subject Claim URI: ---Select---

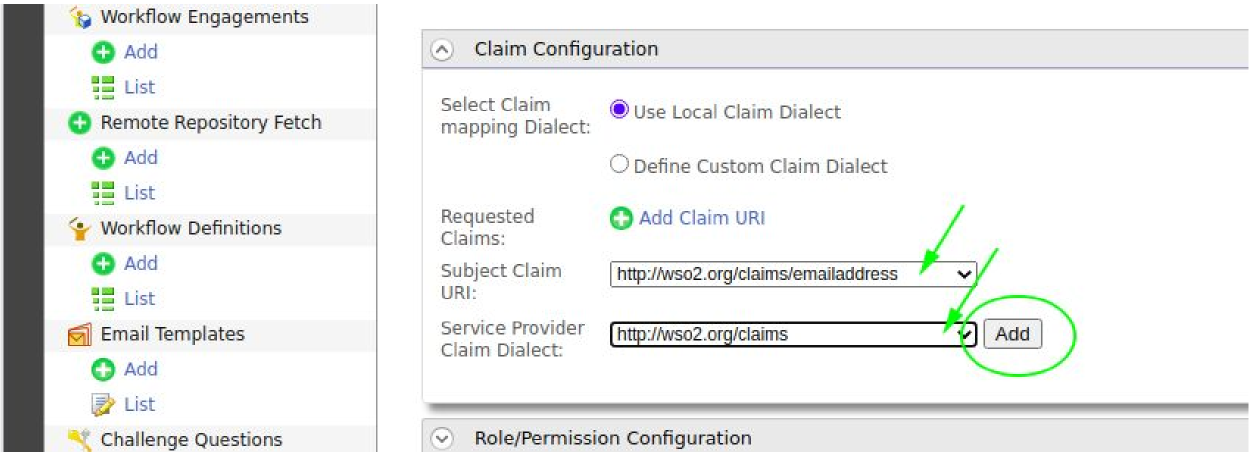
We also select the claim dialect. In ths case is http://wso2.org/claims



Service Provider Claim Dialect: http://wso2.org/claims

Then save the configuration for the service provider





Claim Configuration

Select Claim mapping Dialect: ● Use Local Claim Dialect ○ Define Custom Claim Dialect

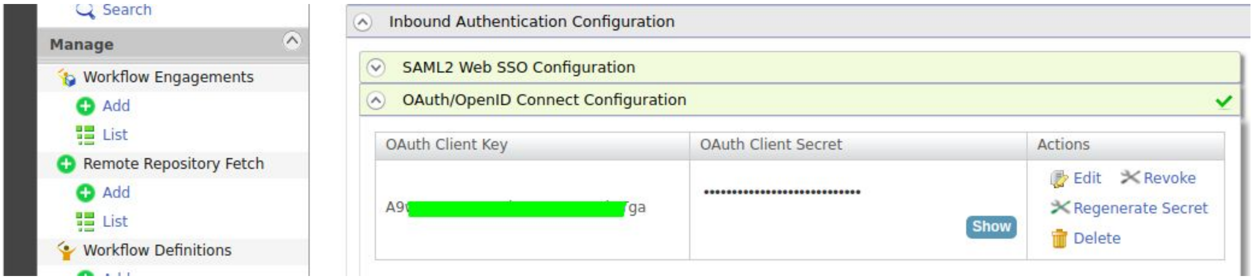
Requested Claims:

Subject Claim URI: http://wso2.org/claims/emailaddress

Service Provider Claim Dialect: http://wso2.org/claims

Role/Permission Configuration

For the oauth2 protocol, client requires a key and a secret, that you can get in the bottom panel of the configuration, with this appearance:



Inbound Authentication Configuration

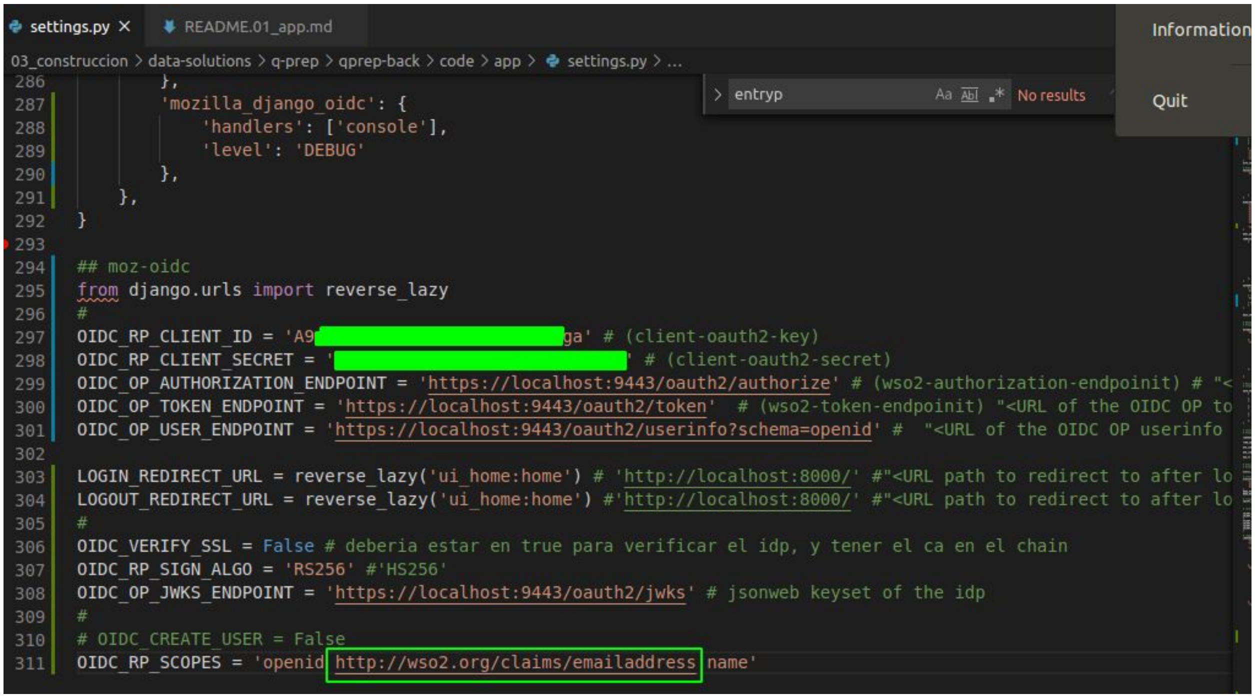
SAML2 Web SSO Configuration

OAuth/OpenID Connect Configuration ✓

OAuth Client Key	OAuth Client Secret	Actions
A9[redacted]ga	••••••••••••••••••••••••••••• Show	Edit Revoke Regenerate Secret Delete

Service Provider Config (django with mozilla-django-oidc)

This is the configuration variables used for the test. Don't forget to not have this variables in the source code. As an alternative you can reference environment variables (os.environ).



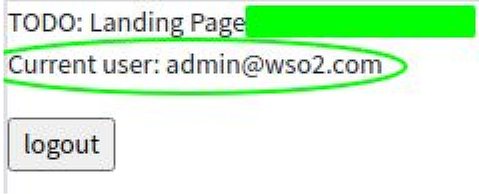
settings.py ×
README.01_app.md
03_construccion > data-solutions > q-prep > qprep-back > code > app > settings.py > ...
entryp
Aa
No results
Information
Quit

```
            },
            'mozilla_django_oidc': {
                'handlers': ['console'],
                'level': 'DEBUG',
            },
        },
}

## moz-oidc
from django.urls import reverse_lazy
#
OIDC_RP_CLIENT_ID = 'A9[redacted]ga' # (client-oauth2-key)
OIDC_RP_CLIENT_SECRET = '[redacted]' # (client-oauth2-secret)
OIDC_OP_AUTHORIZATION_ENDPOINT = 'https://localhost:9443/oauth2/authorize' # (wso2-authorization-endpoinit) # "<
OIDC_OP_TOKEN_ENDPOINT = 'https://localhost:9443/oauth2/token'  # (wso2-token-endpoinit) "<URL of the OIDC OP to
OIDC_OP_USER_ENDPOINT = 'https://localhost:9443/oauth2/userinfo?schema=openid' #  "<URL of the OIDC OP userinfo

LOGIN_REDIRECT_URL = reverse_lazy('ui_home:home') # 'http://localhost:8000/' #"<URL path to redirect to after lo
LOGOUT_REDIRECT_URL = reverse_lazy('ui_home:home') #'http://localhost:8000/' #"<URL path to redirect to after lo
#
OIDC_VERIFY_SSL = False # deberia estar en true para verificar el idp, y tener el ca en el chain
OIDC_RP_SIGN_ALGO = 'RS256' #'HS256'
OIDC_OP_JWKS_ENDPOINT = 'https://localhost:9443/oauth2/jwks' # jsonweb keyset of the idp
#
# OIDC_CREATE_USER = False
OIDC_RP_SCOPES = 'openid http://wso2.org/claims/emailaddress name'
```

Then execute the SP validating that you can get the user/email :



TODO: Landing Page[redacted]

Current user: admin@wso2.com

Then execute the logout procedure.

Adicionales

Contenido

Control de Cambios

Fecha	Versión	Comentarios
2020.06.16	1.0	Creación inicial