

OpenID-Connect example

Identity Provider

Identification

IdP type :
publicID :
Name :
Organization :
contact :

OpenID Connect

OpenIDConnect_ID

OpenIDConnect_Test

Soffid

pgarcia@soffid.com

Login rules

User regular expression :
Login hint script :
Identity provisioning script :

Regular expression to detect users of this identity provider

loginHint

Script to bind or register a new identity. Return the user name of the owner identity for the authenticated acc

Service configuration

Metadata :
oAuth key :
oAuth secret :

```
{
  "authorization_endpoint": "https://server/oauth2/auth",
  "token_endpoint": "https://server/oauth2/token",
  "userinfo_endpoint": "https://server/oauth2/userinfo",
  "scopes_supported": [ "openid", "email", "profile" ],
  "display": "page"
}
```

oAuth key

oAuth secret

Undo

Apply changes

Service Provider

Identification

Type :
publicID :
Name :

OpenID Connect

openidlab

OpenID Connect tenant

OpenID authorization flow

Implicit :
Authorization code :
User's password :
User's password + Client credentials :
Client id :
Client secret :
Response URL :
Logout response URL :
oAuth Session timeout (secs) :
Allowed scopes

Yes

No

Yes

No

Yes

No

Yes

No

tenant

Client secret

https://localhost/return

Logout response URL

oAuth Session timeout (secs)

Scope name	Required roles
Filter	Filter
api.person.read	
api.person.write	
openid	

Note that the scope 'openid' will always be accepted.
A scope with no roles will be granted always.
A scope with roles will be granted if the identified user has the required role.
Add the scope * to allow any scope

Login rules

Allow impersonations :
UID Script :
Ask for consent :
Roles required to login :
System where an enabled account is required :

Target application URL

Script to compute the user name to pass to the target application

Yes

No

Roles required to login

Undo

Apply changes

Revision #3

Created 25 August 2022 06:18:56 by pgarcia@soffid.com

Updated 25 August 2022 06:24:13 by pgarcia@soffid.com