

# OpenID-Connect example

## Identity Provider

### Identification

IdP type :

publicID :

Name :

Organization :

contact :

### Service configuration

Metadata : 

```
{
  "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 :

### Login rules

User regular expression :

Login hint script :

Identity provisioning script :

[Undo](#) [Apply changes](#)

## Service Provider

### Identification

Type :

publicID :

Name :

### Login rules

Allow impersonations :

UID Script :

Ask for consent :  Yes  No

Roles required to login :

System where an enabled account is required :

### OpenID authorization flow

Implicit :

Authorization code :

User's password :

User's password + Client credentials :  Yes  No

Client id :

Client secret :

Response URL :

Logout response URL :

oAuth Session timeout (secs) :

Scope name	Required roles
<input checked="" type="checkbox"/> Filter	Filter
<input type="checkbox"/> api.person.read	
<input type="checkbox"/> api.person.write	
<input type="checkbox"/> openid	

Displayed rows: 3

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

[Undo](#) [Apply changes](#)

Revision #3

Created 25 August 2022 06:18:56

Updated 25 August 2022 06:24:13