

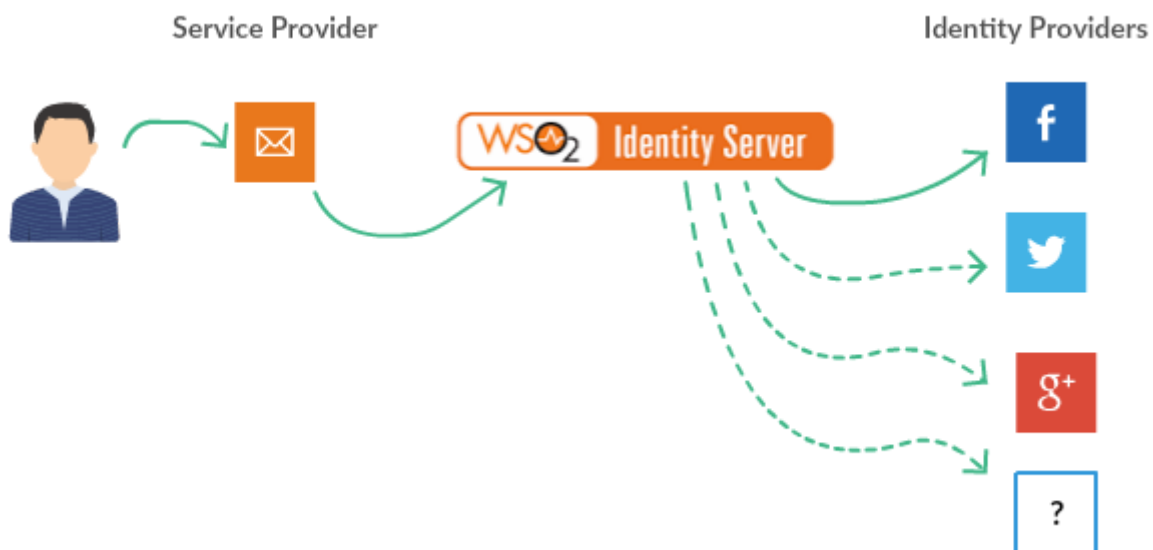
[Open in app](#)

Piraveena Paralogarajah

229 Followers

[About](#)[Follow](#)

Configuring Keycloak as a federated IDP in WSO2 Identity Server

[Piraveena Paralogarajah](#) Jun 1, 2019 · 5 min read

With WSO2 Identity Server, you can integrate different Identity Servers. So WSO2 Identity server will have trust relationships with that identity server and it will assert the identities belong to the other Identity providers.

Now we are going to configure keycloak as a federated IDP in WSO2 Identity Server.

This can be done in 4 steps:

1. Configuring WSO2 Identity Server as a service provider in Keycloak

[Open in app](#)


or, alternatively, create a certificate in Identity Server's truststore

4. Configure a Service provider in WSO2 Identity Server

1. Configuring WSO2 Identity Server as a service provider in Keycloak

- Download keycloak from this [keycloak](#) official site

Downloads

6.0.1 - Release notes

For a list of community maintained extensions check out the Extensions page.

Server

Server	Standalone server distribution	ZIP (sha1)	TAR.GZ (sha1)
Overlay "DEPRECATED"	Server add-on for WildFly. Not recommended in production. "DEPRECATED"	ZIP (sha1)	TAR.GZ (sha1)

Gatekeeper

Linux	TAR.GZ (sha1)
Windows	TAR.GZ (sha1)
Darwin	TAR.GZ (sha1)

Examples

Quickstarts distribution	GitHub	ZIP
Old examples "DEPRECATED"	ZIP (sha1)	

Client Adapters

OAUTH CONNECT	SAML 2.0
---------------	----------

- Download and Unzip, and run the command

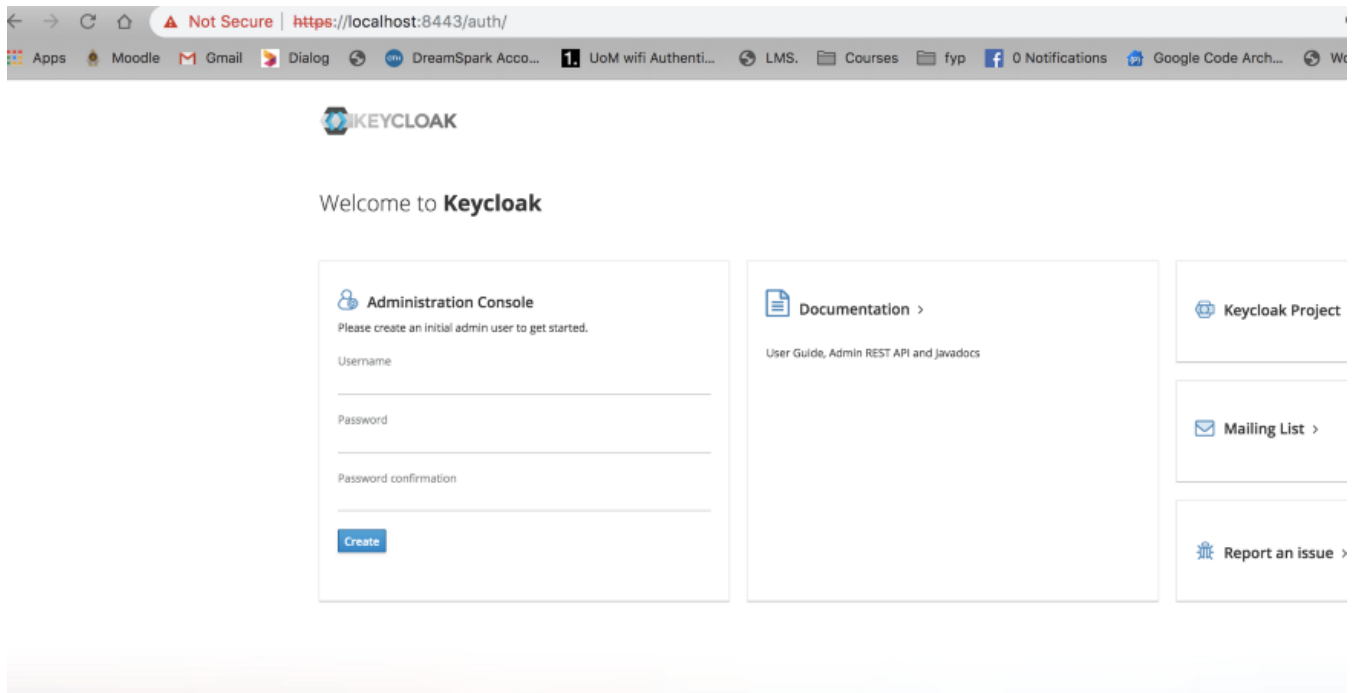
```
sh.standalone.sh
```

```

15:57:36,462 INFO [org.jboss.modules] (main) JBoss Modules version 1.9.0.Final
15:57:37,058 INFO [org.jboss.nsc] (main) JBoss NSC version 1.4.5.Final
15:57:37,069 INFO [org.jboss.threads] (main) JBoss Threads version 2.3.3.Final
15:57:37,231 INFO [org.jboss.as] (MSC service thread 1-1) WFLYSRV0049: Keycloak 6.0.1 (WildFly Core 8.0.0.Final) starting
15:57:37,980 INFO [org.wildfly.security] (ServerService Thread Pool -- 15) ELYW0001: WildFly Elytron version 1.8.0.Final
15:57:38,588 INFO [org.jboss.as.controller.management-deprecated] (Controller Boot Thread) WFLYCTL0028: Attribute 'security-realm' in the resource at address '/subsystem=undertow/server=default-server/https-listener=https' is deprecated, and may be removed in a future version. See the attribute description in the output of the read-resource-description operation to learn more about the deprecation.
15:57:38,612 INFO [org.jboss.as.controller.management-deprecated] (ServerService Thread Pool -- 75) WFLYCTL0028: Attribute 'security-realm' in the resource at address '/subsystem=undertow/server=default-server/https-listener=https' is deprecated, and may be removed in a future version. See the attribute description in the output of the read-resource-description operation to learn more about the deprecation.
15:57:38,733 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0019: Creating http management service using socket-binding (management-http)
15:57:38,755 INFO [org.wildfly] (MSC service thread 1-2) WFLY0000: Version 3.6.5.Final
15:57:38,765 INFO [org.jboss.as] (ServerService Thread Pool -- 48) WFLY0001: The node-identifier attribute on the /subsystem=transactions is set to the default value. This is a danger for environments running multiple servers. Please make sure the attribute value is unique.
15:57:38,817 INFO [org.wildfly.extension.microprofile.config.smallrye-private] (ServerService Thread Pool -- 49) WFLYCONF0001: Activating WildFly MicroProfile Config Subsystem
15:57:38,822 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 36) WFLYCLINF0001: Activating Infinispan subsystem.
15:57:38,829 INFO [org.wildfly.extension.health] (ServerService Thread Pool -- 50) WFLYHEALTH0001: Activating Eclipse MicroProfile Health Subsystem
15:57:38,832 INFO [org.wildfly.extension.io] (ServerService Thread Pool -- 35) WFLYIO0001: Worker 'default' has auto-configured to 24 core threads with 128 task threads based on your 12 available processors
15:57:38,833 INFO [org.wildfly.extension.microprofile.metrics.smallrye] (ServerService Thread Pool -- 51) WFLYMET0001: Activating Eclipse MicroProfile Metrics Subsystem
  
```

[Open in app](#)

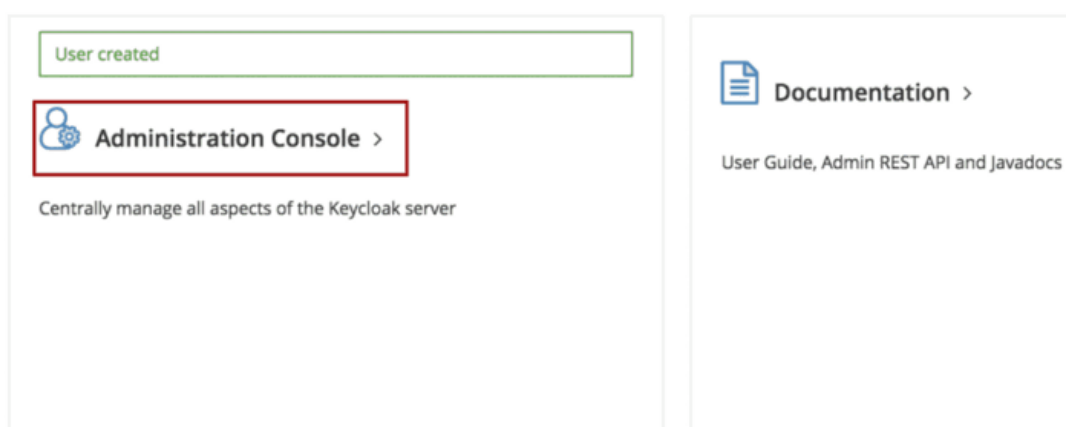
- Now you need to setup a admin user to login to admin console. For that you will have to run add-user-keycloak script. If you have local access then you can create an admin user by logging on to auth portal. Go to <https://localhost:8443/auth/> and create a admin user

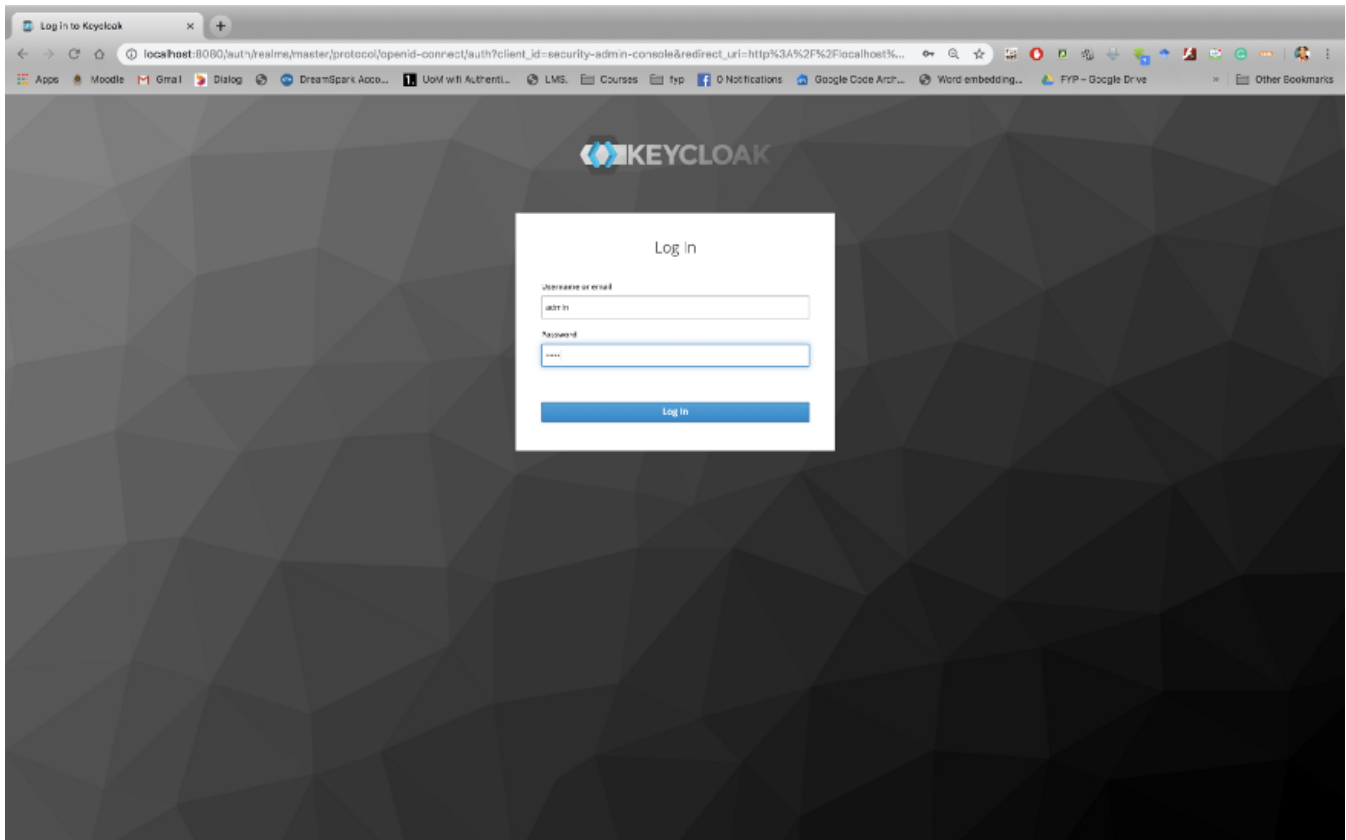


- Once the user got created, you can see a success message.
- Click on **Administration Console**. You will be prompted username / password page. Enter registered admin user's username and password'

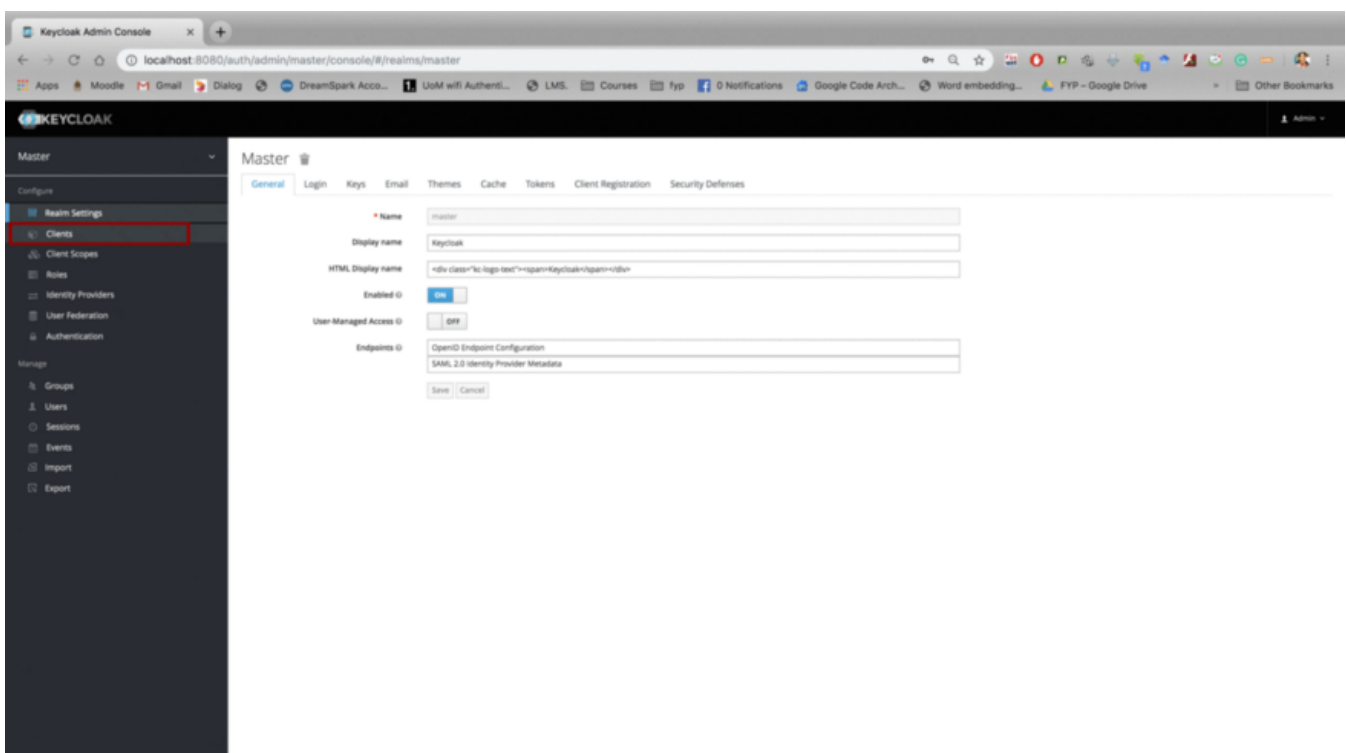


Welcome to **Keycloak**



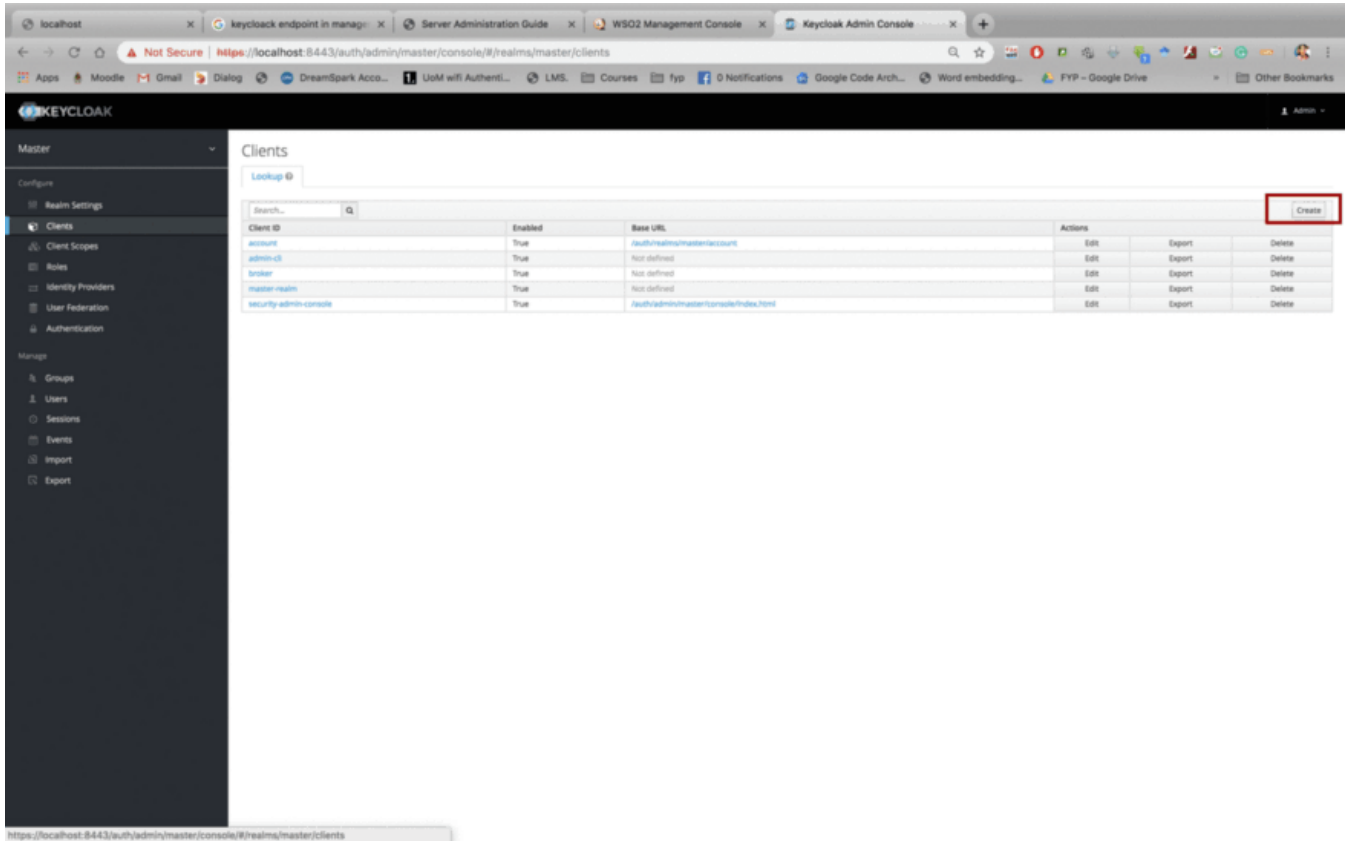
[Open in app](#)

- Now you need to configure WSO2 Identity server as a service provider in Keycloak Identity Server. Click on Clients menu.

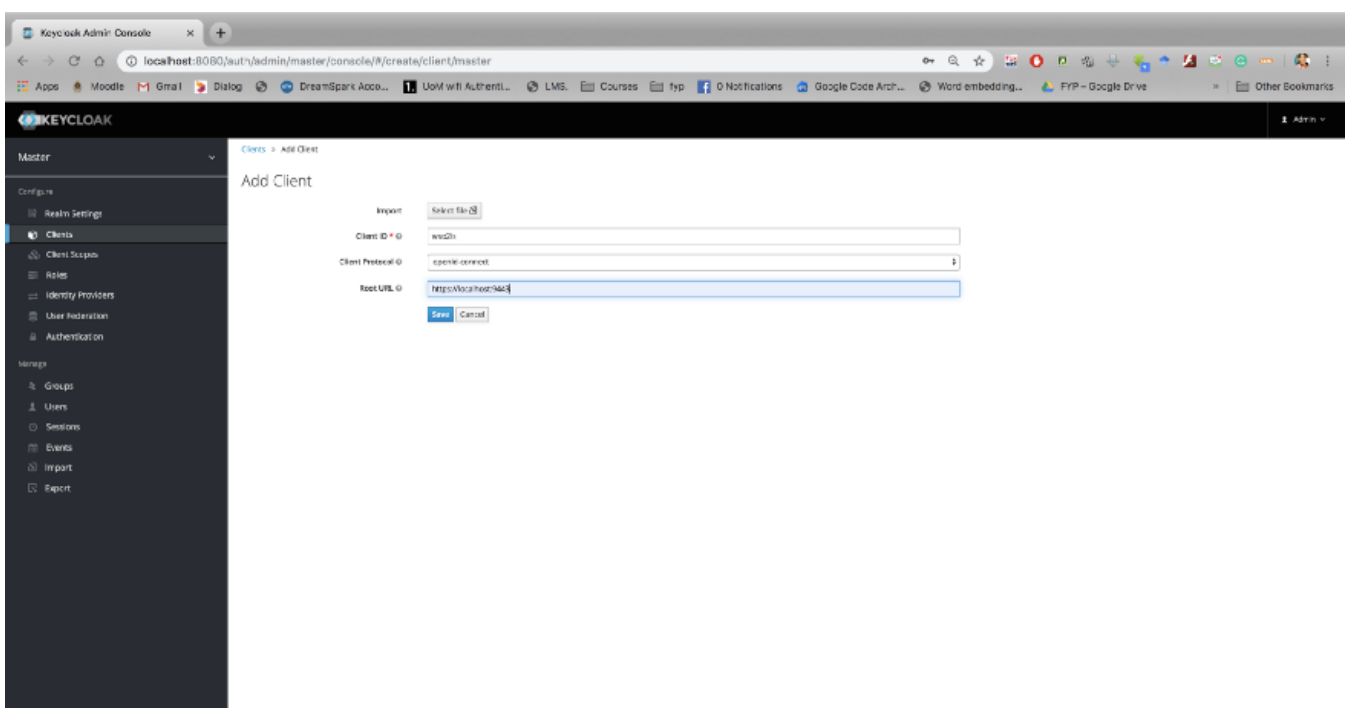


[Open in app](#)

- Now you can see some service providers already configured. Click on **Create** button and create a new service provider.



- Enter the client ID and client protocol and root URL of the service provider (Here WSO2 Identity server will act as a service provider to Keycloak Identity Server)



[Open in app](#)


- Once you create the service provider, you will be redirected to the service provider details.

The screenshot shows the Keycloak Admin Console interface. The left sidebar contains navigation options under 'Configure' (Realm Settings, Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication) and 'Manage' (Groups, Users, Sessions, Events, Import, Export). The main panel is titled 'wso2is' and shows the 'Settings' tab. The configuration fields are as follows:

- Client ID: wso2is
- Name: (empty)
- Description: (empty)
- Enabled: ☒
- Consent Required: ☐
- Login Theme: (empty)
- Client Protocol: openid-connect
- Access Type: public
- Standard Flow Enabled: ☒
- Implicit Flow Enabled: ☐
- Direct Access Grants Enabled: ☒
- Authentication Enabled: ☐
- Root URL: https://localhost:9443
- Valid Redirect URIs: https://localhost:9443/*
- Base URL: (empty)
- Admin URL: https://localhost:9443
- Web Origins: https://localhost:9443

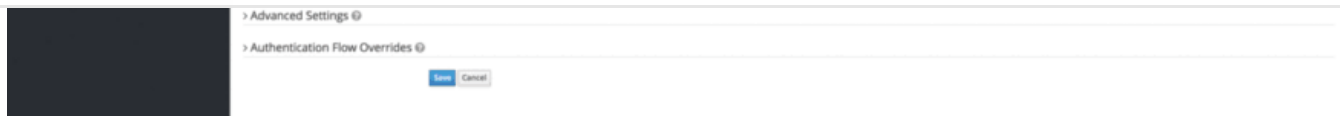
Below the configuration fields, there are expandable sections for 'Fine Grain OpenID Connect Configuration', 'OpenID Connect Compatibility Modes', 'Advanced Settings', and 'Authentication Flow Overrides'. At the bottom, there are 'Save' and 'Cancel' buttons.

- Enable all **grant types** using the management console of Keycloak. Make sure the access type is **confidential**. That will create a client secret for the service provider. And click on save button to update your changes.

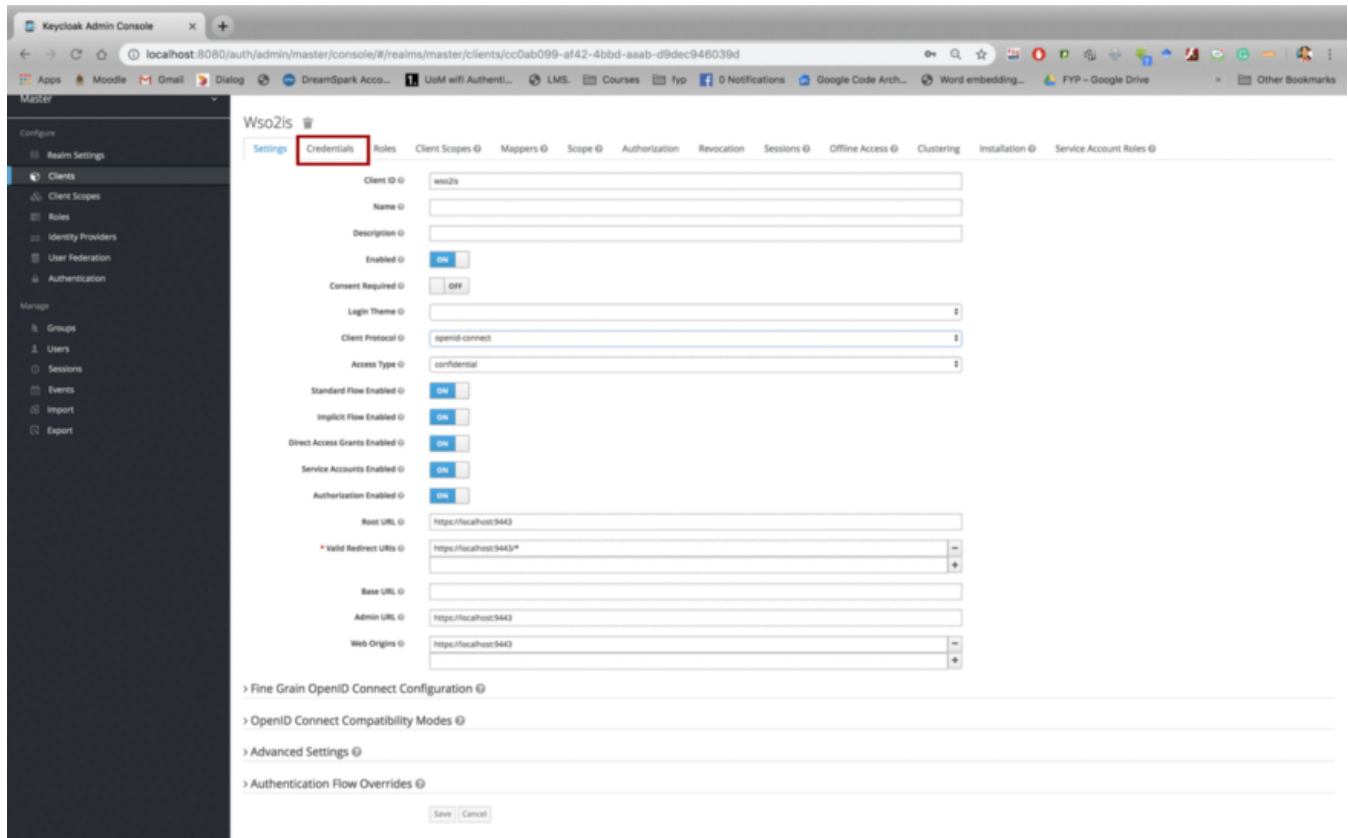
This screenshot shows the same Keycloak Admin Console configuration for the 'wso2is' client, but with changes made to the grant types and access type. The configuration fields are:

- Client ID: wso2is
- Name: (empty)
- Description: (empty)
- Enabled: ☒
- Consent Required: ☐
- Login Theme: (empty)
- Client Protocol: openid-connect
- Access Type: confidential
- Standard Flow Enabled: ☒
- Implicit Flow Enabled: ☒
- Direct Access Grants Enabled: ☒
- Service Accounts Enabled: ☒
- Authentication Enabled: ☒
- Root URL: https://localhost:9443
- Valid Redirect URIs: https://localhost:9443/*
- Base URL: (empty)
- Admin URL: https://localhost:9443

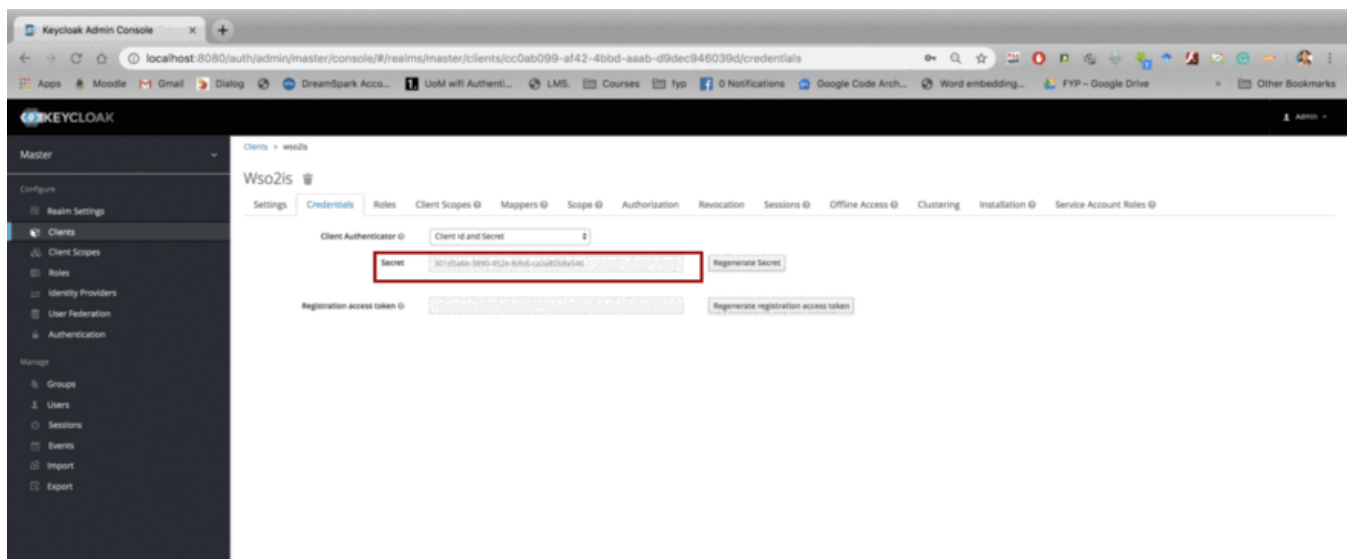
A red box highlights the 'Access Type' dropdown (set to 'confidential') and the 'Standard Flow Enabled', 'Implicit Flow Enabled', 'Direct Access Grants Enabled', 'Service Accounts Enabled', and 'Authentication Enabled' checkboxes, all of which are now checked. The 'Save' and 'Cancel' buttons are at the bottom.

[Open in app](#)


- Once you select Access Type to **confidential** new tab will be appeared as Credentials. There you can extract client secret. Now you saved the changed. Now you can see credentials tab. Click on **credential** tab.



- Now you can view the **client secret** of the service provider you created



[Open in app](#)


2. Configure Keycloak as a Federated Identity Server in WSO2 Identity Server.

- Run WSO2 Identity server.

```
sh wso2server.sh
```

- Go to management console (<https://localhost:9443/carbon>)
- Navigate to Main > Identity > **Identity Providers** and click on **Add**.

The screenshot shows the WSO2 Identity Server Management Console. The left sidebar has a menu with 'Identity Providers' highlighted and the 'Add' button next to it. The main content area displays 'WSO2 Identity Server Home' with a welcome message and a table of system information.

Server	
Host	localhost
Server URL	local://services/
Server Start Time	2019-06-01 16:36:46
System Up Time	0 day(s) 0 hr(s) 4 min(s) 43 sec(s)
Version	5.7.0
Repository Location	file://Users/piraveena/wso2is-5.7.0/full/wso2is-5.7.0.2/repository/deployment/server/
Operating System	
OS Name	Mac OS X
OS Version	10.13.6
Operating System User	
Country	UK
Home	/Users/piraveena
Name	piraveena
Timezone	Asia/Colombo
Java VM	
Java Home	/Library/Java/JavaVirtualMachines/jdk1.8.0_191.jdk/Contents/Home/jre
Java Runtime Name	java(TM) SE Runtime Environment
Java Version	1.8.0_191
Java Vendor	Oracle Corporation
Java VM Version	25.191-b12
Registry	
DBMS	H2
DBMS Version	1.3.175 (2014-01-18)
DBMS Driver	H2 JDBC Driver
DBMS Driver Version	1.3.175 (2014-01-18)
DBMS URL	jdbc:h2:./repository/database/WSO2CARBON_DB

- Now click on Add Identity provider. Add details of Identity Provider.

The screenshot shows the 'Add New Identity Provider' form. The 'Basic Information' tab is selected. The 'Identity Provider Name' field is filled with 'keycloak'.

Add New Identity Provider

Basic Information

Identity Provider Name:

Enter a unique name for this identity provider

[Open in app](#)


Home Realm Identifier:
Enter the home realm identifier for this identity provider

Choose IDP certificate type: ☒ Use IDP JWKS endpoint ☐ Upload IDP certificate

Identity Provider's JWKS Endpoint:
Enter Identity Provider's JWKS endpoint

Alias:
If the resident identity provider is known by an alias at the federated identity provider specify it

Claim Configuration

Role Configuration

Federated Authenticators

SAML2 Web SSO Configuration

OAuth2/OpenID Connect Configuration

Enable ☒ Specifies if OAuth2/OpenID Connect is enabled for this identity provider

OAuth2/OpenIDConnect Default ☒ Specifies if OpenID Connect is the default

Client id:
Enter OAuth2/OpenID Connect client identifier value

Client Secret:
Enter OAuth2/OpenID Connect client secret value [Show](#)

Authorization Endpoint URL:
Enter OAuth2/OpenID Connect authorization endpoint URL value

Token Endpoint URL:
Enter OAuth2/OpenID Connect token endpoint URL value

Callback Url:
Enter value corresponding to callback url

Userinfo Endpoint URL:
Enter value corresponding to userinfo endpoint url

OpenID Connect User ID Location: ☒ User ID found in 'sub' attribute ☐ User ID found among claims
Specifies the location to find the user identifier in the ID token assertion

Additional Query Parameters:
Additional query parameters. e.g: paramName1=value1

Enable HTTP basic auth for client authentication: ☐ Specifies that HTTP basic authentication should be used for client authentication, else client credentials will be included in the request body

WS-Federation (Passive) Configuration

Facebook Configuration

Microsoft (Hotmail, MSN, Live) Configuration

Google Configuration

SMS OTP Configuration

Twitter Configuration

Email OTP Configuration

Yahoo Configuration

IWA Kerberos Configuration

Office365 Configuration

Just-In-Time Provisioning

Outbound Provisioning Connectors

Authorization endpoint:

<https://localhost:8443/auth/realms/master/protocol/openid-connect/auth>

Token endpoint:

<https://localhost:8443/auth/realms/master/protocol/openid-connect/token>

Callback Url : <https://localhost:9443/commonauth>

UserInfo endpoint:

<https://localhost:8443/auth/realms/master/protocol/openid-connect/userinfo?schema=openid>

- You can check oidc endpoints of keycloak from here [1]

3. Import Keycloak certificate in Identity Server's truststore

[Open in app](#)

To ensure the trust relationship, the public certificate of Keycloak need to be imported in to IS truststore.

- Shutdown keycloak server and Identity Server
- Go to `<KEYCLOAK_HOME>/standalone/configuration`
- Remove existing application.keystore file and create new keystore. Here the **CN name** and the **host name** of keycloak should match.
- In my example, both are localhost.

```
keytool -genkey -alias server -keyalg RSA -keysize 2048 -validity 3650 -keystore application.keystore -dname "CN=localhost,OU=Support,O=WSO2,L=Colombo,S=Western,C=LK" -storepass password -keypass password -noprompt
```

```
keytool -export -alias server -file keycloak.crt -keystore application.keystore -storepass password -noprompt
```

- Now you can see keycloak.crt file in the folder `<KEYCLOAK_HOME>/standalone/configuration`. Now you can export the public certificate into IS truststore.
- Copy the server.crt file into the folder certificate and Go to `<IS_HOME>/repository/resources/security` and paste there. Now execute the following command and Import the server.crt to client-truststore.jks

```
keytool -import -trustcacerts -alias keycloak -file keycloak.crt -keystore client-truststore.jks -storepass wso2carbon -noprompt
```

- Restart both servers.

4. Configure a Service provider in WSO2 Identity Server

- Navigate to Menu > Identity > **Service Providers** and click on Add

[Open in app](#)

Identity Server

Users and Roles

Users

Groups

Roles

Claims

Service Providers

Identity Providers

Entitlement

Manage

Workflow Engagements

Workflow Definitions

Challenge Questions

Email Templates

Key Stores

Consent Purposes

Add New Service Provider

Select Mode

Manual Configuration

File Configuration

Basic Information

Service Provider Name*

Description

Register Cancel

- Provide service provider's **name** and **description**. Click on **Register**.

Here I'm going to register playground as a service provider.

localhost

keycloak endpoint in manager

Server Administration Guide

WSO2 Management Console

Keycloak Admin Console

Not Secure

https://localhost:9443/carbon/application/add-service-provider.jsp?region=region1&item=add_service_providers_menu

Apps

Moodle

Gmail

Dialog

DreamSpark Acco...

UoM wifi Authent...

LMS

Courses

typ

0 Notifications

Google Code Arch...

Word embedding...

FYP - Google Drive

Other Bookmarks

WSO2 Identity Server

Management Console

Signed in as admin@carbon.super | Sign-out | Docs | About

Home

Identity

Users and Roles

Users

Groups

Roles

Claims

Service Providers

Identity Providers

Entitlement

Manage

Workflow Engagements

Workflow Definitions

Challenge Questions

Email Templates

Key Stores

Consent Purposes

Add New Service Provider

Select Mode

Manual Configuration

File Configuration

Basic Information

Service Provider Name*

Description

Register Cancel

- Navigate to **Inbound Authentication Configuration > OAuth/OpenID Connect Configuration**. Click on **Configure**.

Open in app



Service Providers

Basic Information

Service Provider Name* A unique name for the service provider

Description:

Select SP Certificate Type ☒ Use SP JWKs endpoint ☐ Upload SP certificate

JWKS URI:

SaaS Application ☐ Applications are by default restricted for usage by users of the service provider's tenant. If this application is SaaS enabled it is opened up for all the users of all the tenants.

Claim Configuration

Role/Permission Configuration

Inbound Authentication Configuration

☒ SAML2 Web SSO Configuration

☒ OAuth/OpenID Connect Configuration

☒ **Configure**

☒ OpenID Configuration

☒ WS-Federation (Passive) Configuration

☒ WS-Trust Security Token Service Configuration

☒ Kerberos KDC

Local & Outbound Authentication Configuration

Inbound Provisioning Configuration

Outbound Provisioning Configuration

Register New Application

New Application

OAuth Version* ☒ 1.0L ☒ 2.0

☒ Code ☒ Implicit ☒ Password ☒ Client Credential ☒ Refresh Token ☒ SAML2 ☒ PWA-NTLM ☒ Use + to add more oauth grant types (not tested)

Allowed Grant Types

☒ Refresh Token This option will issue a new refresh token when Access Token Grant is used.

☐ PACE Mentimeter Only allow applications that have NECP Cook Challenge with them.

☐ Support PACE 'Race' Transition Algorithm Server supports PACE transition algorithm by default.

☐ Allow authentication without the client secret This option will allow the client to authenticate without a client secret.

Default URI*

User Access Token Expiry Time seconds

Application Access Token Expiry Time seconds

Refresh Token Expiry Time seconds

Id Token Expiry Time seconds

☐ Enable Audience Restriction Audience:

☐ Enable Request Object Signature Validation

☐ Enable ID Tokens Encryption

Encryption Algorithm

Encryption Method

Scope Validators ☐ Role based scope validator ☐ XACML Scope Validator ☐ JWT ☒ Default

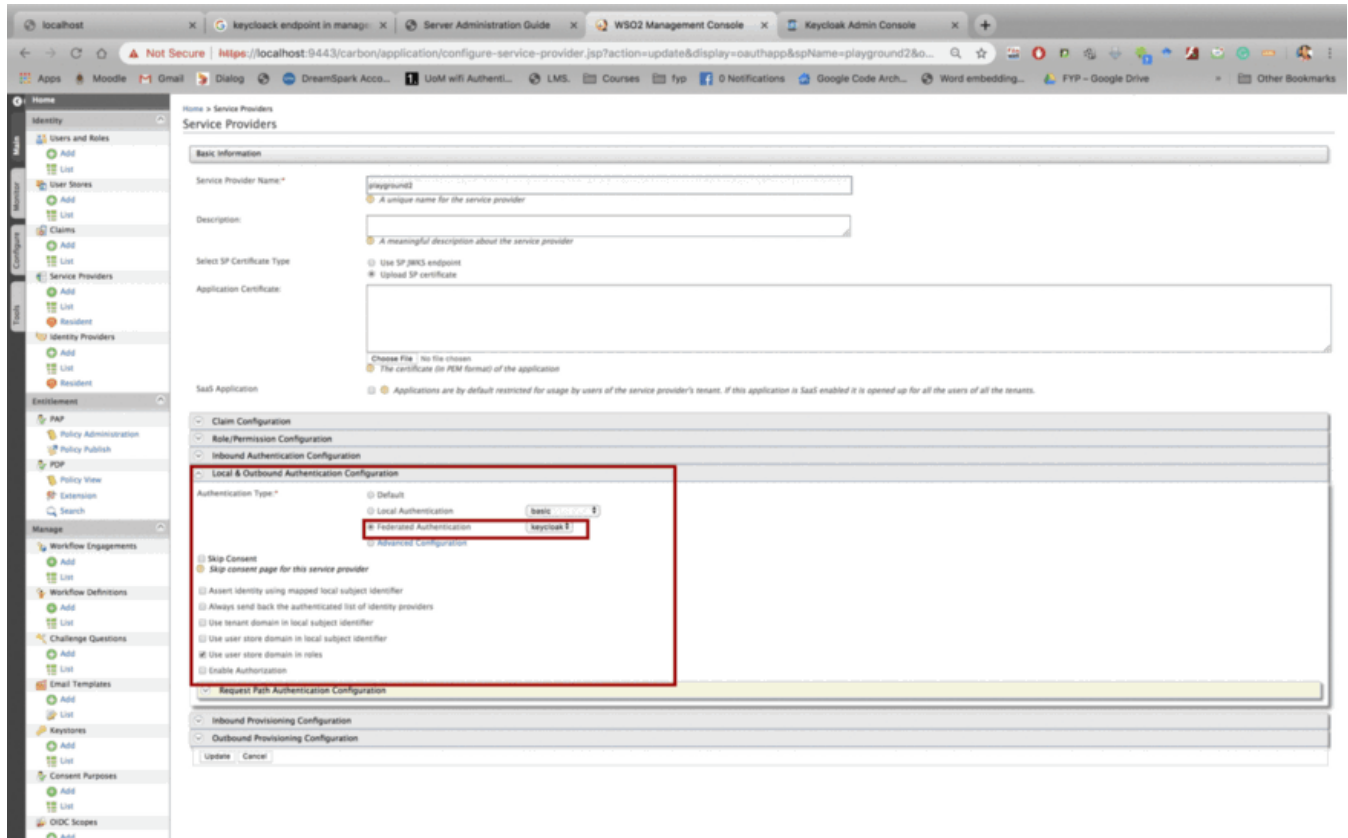
Token Issuer ☒ Default

Callback Url of Playground app :
<http://localhost:8080/playground2/oauth2client>

Open in app



- Navigate to **Local & Outbound Configuration** and open the menu. Make the **Authentication Type** as **Federated Authentication** and the IDP as **Keycloak**



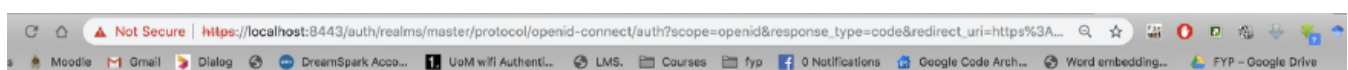
- Click on **Update**.
- Now you have configured Playground service provider with federated IDP as Keycloak.

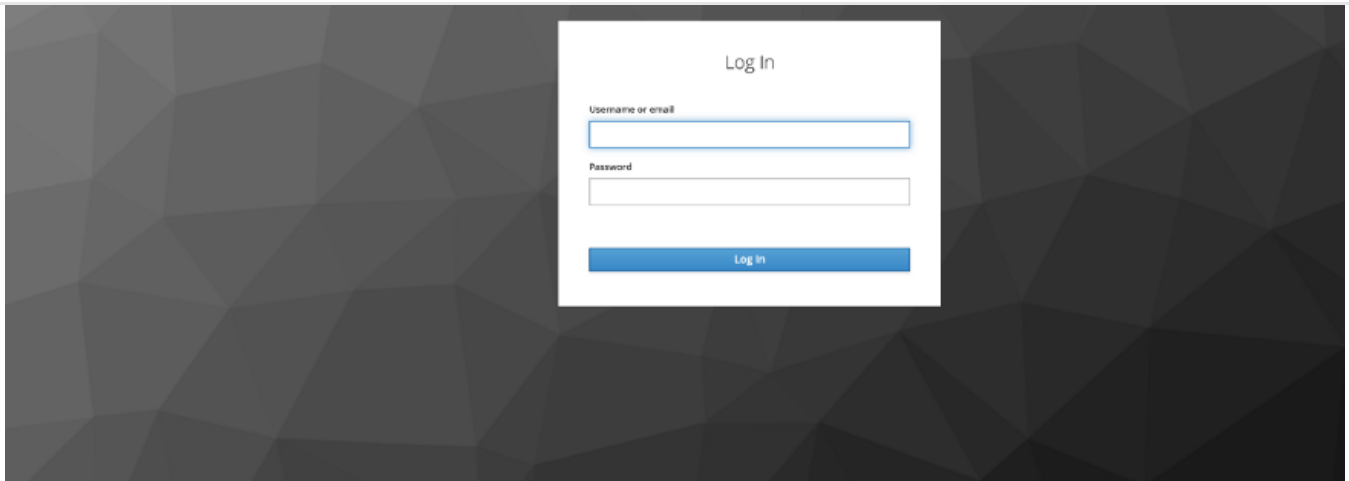
Try it out !

- Call the authorize endpoint of WSO2 Identity Server with oidc implicit flow.

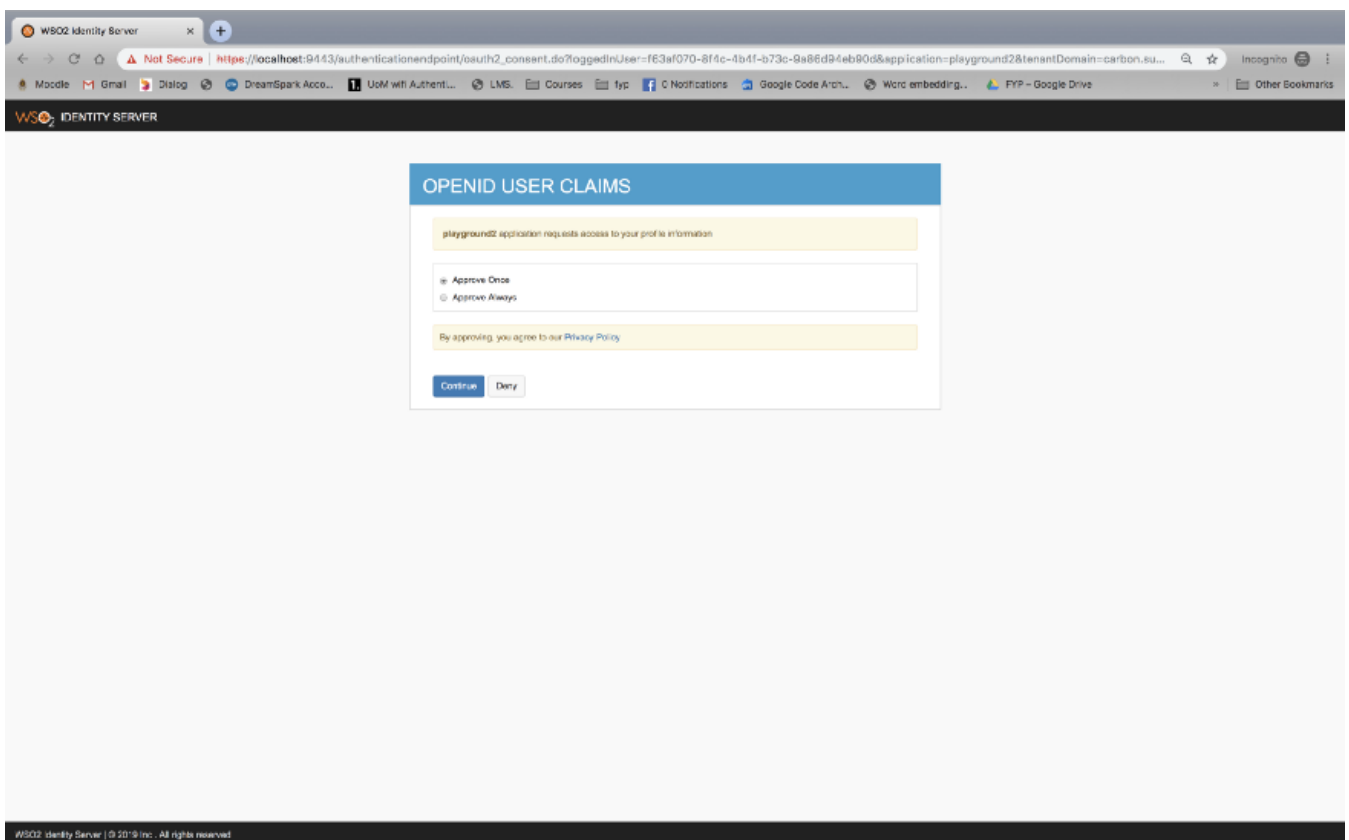
```
https://localhost:9443/oauth2/authorize?
response_type=id_token%20token&scope=openid&redirect_uri=http%3A%2F%
2Flocalhost%3A8080%2Fplayground2%2Foauth2client&client_id=
<client_id>&nonce=n-0S6_WzA2Mj
```

- You will be redirected to key cloak login page.



[Open in app](#)

- Provide the user credentials and login.
- You will be redirected to the consent page of Identity Server.



- Once you approve this you can get the access token and idtoken.

[1]https://www.keycloak.org/docs/latest/server_admin/index.html#keycloak-

[Open in app](#)

[2] <https://docs.wso2.com/display/IS570/Configuring+Federated+Authentication>

[Wso2](#) [Wso2is](#) [Identity Management](#) [Keycloak](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

