

Keycloak Document

Once you've installed Keycloak, you'll need an administrator account to serve as a super admin. This super admin account comes with full permissions to manage Keycloak. You can use this account to access the Keycloak Admin Console, where you can create realms, manage users, and register applications that will be secured by Keycloak.

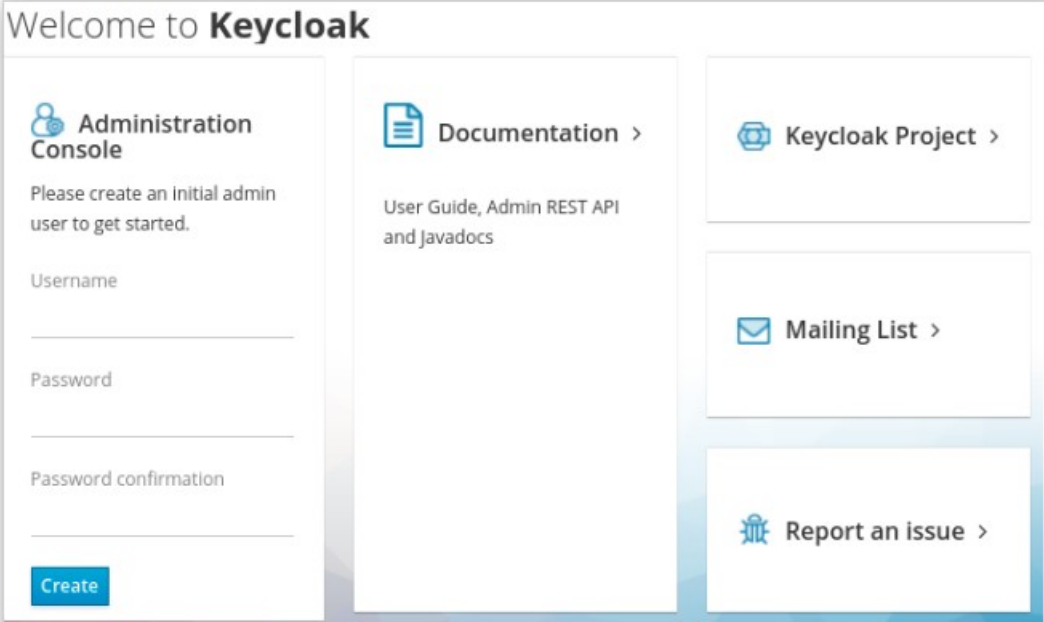
1. Creating a Administrator account on the local host

If your server is accessible from the `localhost`, then perform these steps.

Procedure

1. In a web browser, go to the <http://localhost:8080> URL.
2. Enter username , password and password confirmation

Welcome page

The image shows the Keycloak Welcome page. At the top, it says "Welcome to Keycloak". Below this, there are four main sections. On the left, the "Administration Console" section prompts the user to "Please create an initial admin user to get started." and provides input fields for "Username", "Password", and "Password confirmation", followed by a blue "Create" button. In the center, the "Documentation" section includes a document icon and links to the "User Guide, Admin REST API and javadocs". On the right, there are three more sections: "Keycloak Project" with a gear icon, "Mailing List" with an envelope icon, and "Report an issue" with a bug icon. Each of these three sections has a right-pointing arrow next to its title.

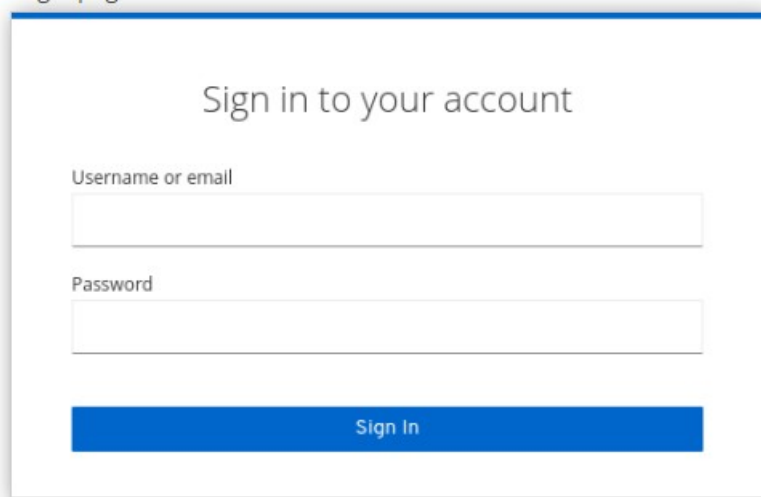
- 3 . Click on create
4. Click on Administrator console(This will redirect on login page)

2. Configuring realms

Once you have an administrative account for the Admin Console, you can configure realms. A realm is a space where you manage objects, including users, applications, roles, and groups. A user belongs to and logs into a realm. One Keycloak deployment can define, store, and manage as many realms as there is space for in the database.

(2.1) Logging here by

Login page

A screenshot of the Keycloak login page. It features a white background with a blue border. At the top, the text "Sign in to your account" is centered. Below this, there are two input fields: "Username or email" and "Password". At the bottom, there is a blue button labeled "Sign In".

Sign in to your account

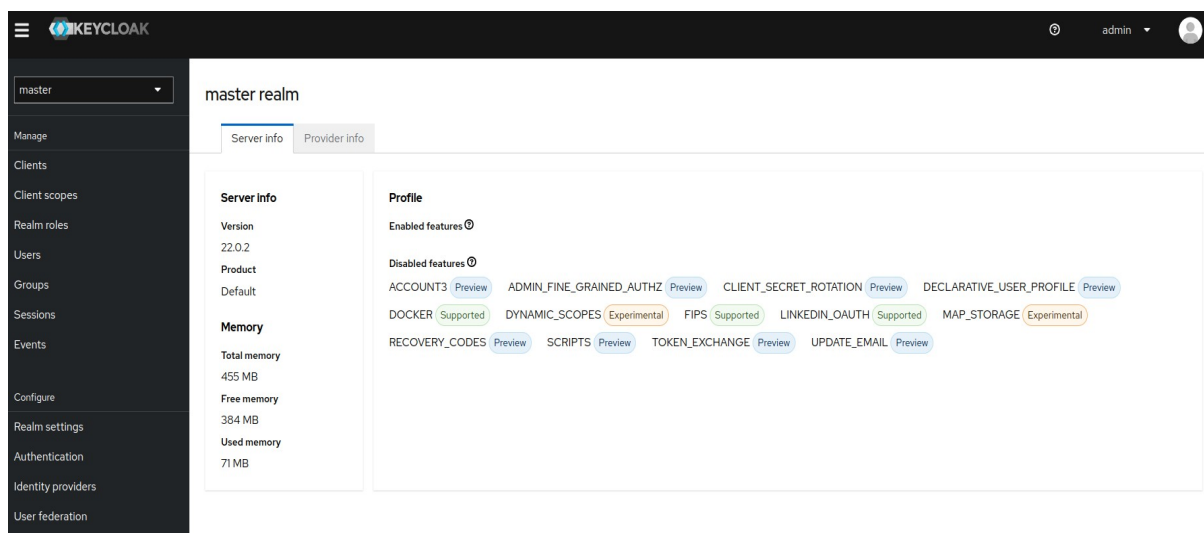
Username or email

Password

Sign In

Enter the username and password you created on the Welcome Page or the `add-user-keycloak` script in the `bin` directory. This action displays the Admin Console.

A page similar to the following is displayed.

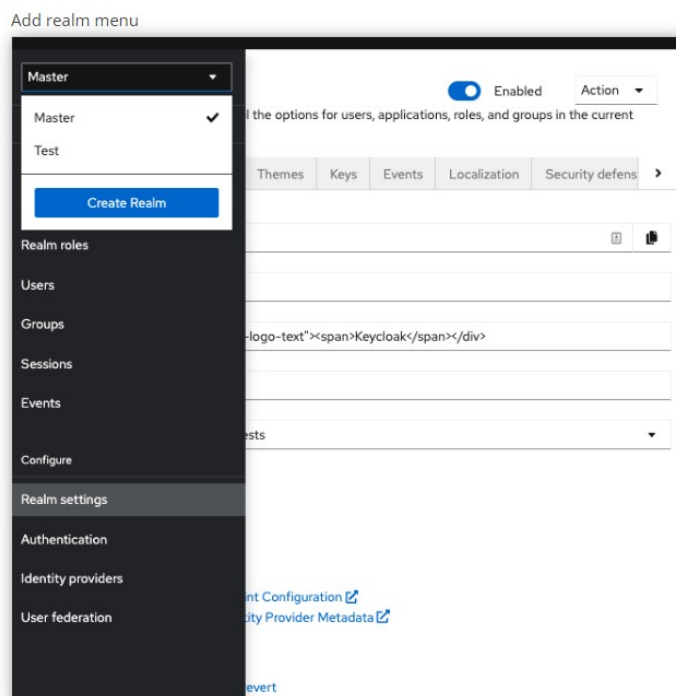


3. Creating a realm

You create a realm to provide a management space where you can create users and give them permissions to use applications.

(3.1) Point to the top of the left pane.

(3.2) Click **Create Realm**.



(3.3) Enter the name for the realm.

(3.4) Click **Create**.

Create realm

Create realm

A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and can only manage and authenticate the users that they control.

Resource file

Drag a file here or browse to upload

Browse...Clear

1

Upload a JSON file

Realm name *

Enabled

☒ On

Create

Cancel

The current realm is now set to the realm you just created. You can switch between realms by clicking the realm name in the menu.

4. Creating a client application

The first step to enable Keycloak Authorization Services is to create the client application that you want to turn into a resource server.

Procedure

(4.1) Select Client menu

(4.2) Click on Create client

Clients

Hello-world-authz

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Clients

Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list

Initial access token

Q Search for client

→

Create client

Import client

1 - 7

<

>

Client ID	Type	Description	Home URL
account	OpenID Connect	—	http://localhost:8180/realms/hello-world-authz/account/
account-console	OpenID Connect	—	http://localhost:8180/realms/hello-world-authz/account/
admin-cli	OpenID Connect	—	—
app-authz-vanilla	OpenID Connect	—	—
broker	OpenID Connect	—	—
realm-management	OpenID Connect	—	—
security-admin-console	OpenID Connect	—	http://localhost:8180/admin/hello-world-authz/console/

1 - 7<>

On this page

Add Client

The screenshot shows the 'Create client' page in Keycloak. The left sidebar has a dropdown menu with 'Hello-world-authz' selected. Below it are links for 'Manage', 'Clients', 'Client scopes', 'Realm roles', 'Users', 'Groups', 'Sessions', and 'Events'. The main content area is titled 'Create client' with a subtitle 'Clients are applications and services that can request authentication of a user.' Below this is a progress indicator with '1 General Settings' highlighted. The form fields are: 'Client type' (OpenID Connect), 'Client ID' (my-resource-server), 'Name' (empty), and 'Description' (empty).

Type the **Client ID** of the client. For example, *my-resource-server*.

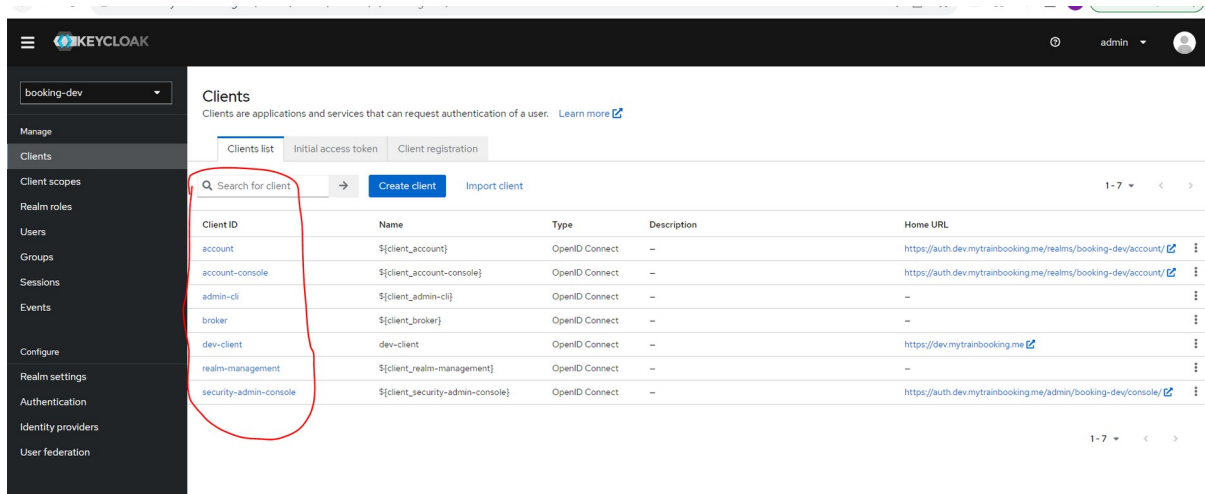
(4.4) Click **Next**.

(4.5) Toggle **Client authentication** to ON.

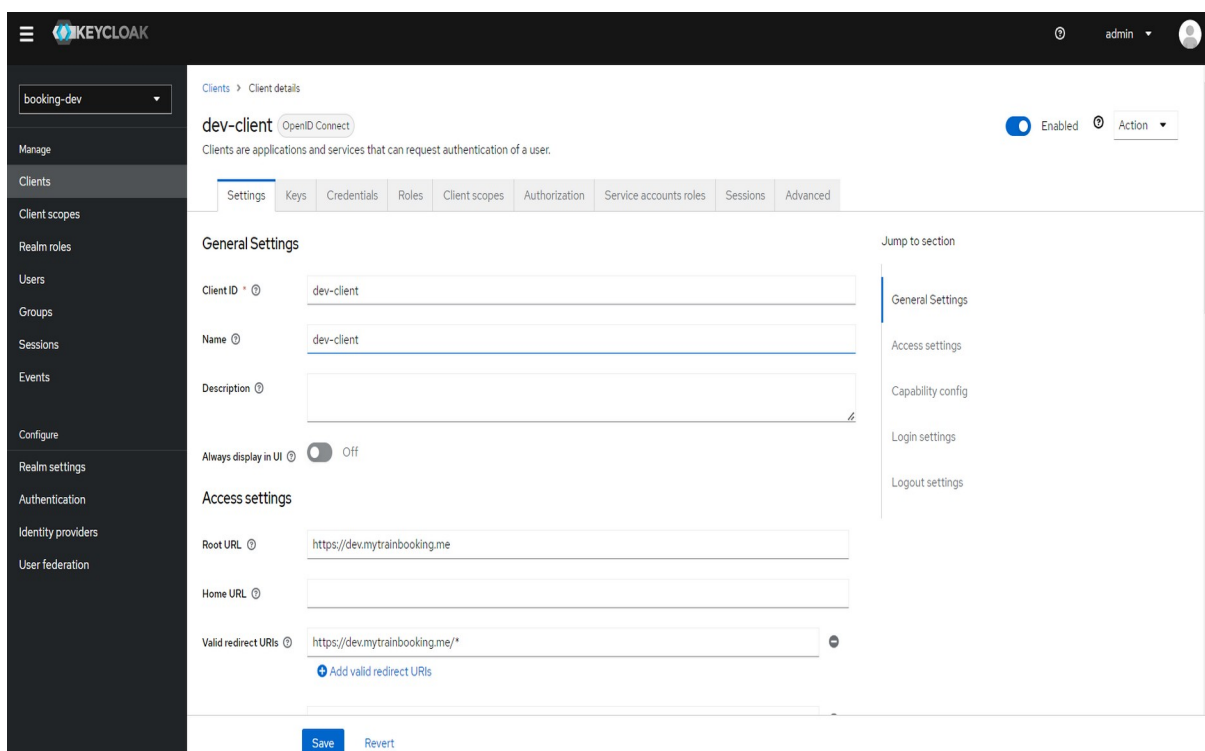
(4.6) Click **Save**.

The screenshot shows the 'Create client' page in Keycloak, Step 2: Capability config. The left sidebar has a dropdown menu with 'booking-dev' selected. Below it are links for 'Manage', 'Clients', 'Client scopes', 'Realm roles', 'Users', 'Groups', 'Sessions', 'Events', 'Configure', 'Realm settings', 'Authentication', 'Identity providers', and 'User federation'. The main content area is titled 'Create client' with a subtitle 'Clients are applications and services that can request authentication of a user.' Below this is a progress indicator with '2 Capability config' highlighted. The form fields are: 'Client authentication' (On), 'Authorization' (On), 'Authentication flow' (Standard flow, Direct access grants, Implicit flow, OAuth 2.0 Device Authorization Grant, OIDC CIBA Grant), and 'Service accounts roles' (checked). At the bottom are buttons for 'Next', 'Back', and 'Cancel'.

Now all clients will be show .We can see a page similar to the following is displayed

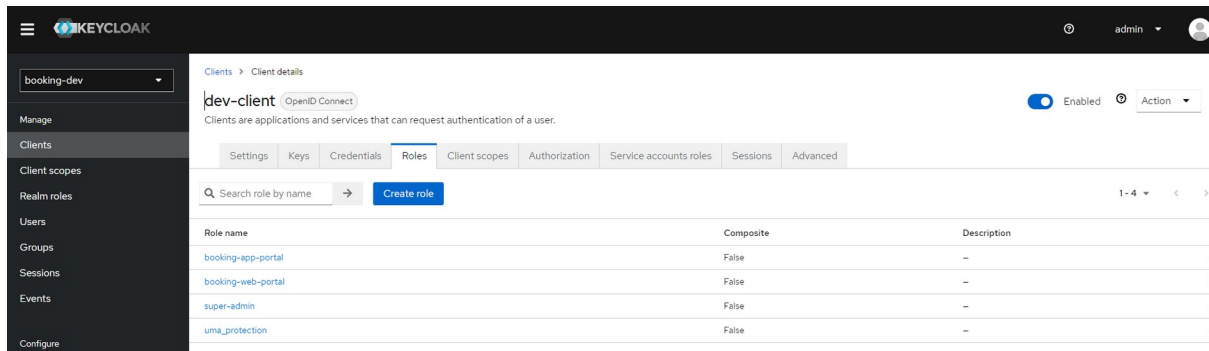


Inside Client Section select your created client, As I am selecting dev-Client then a page similar to the following is displayed



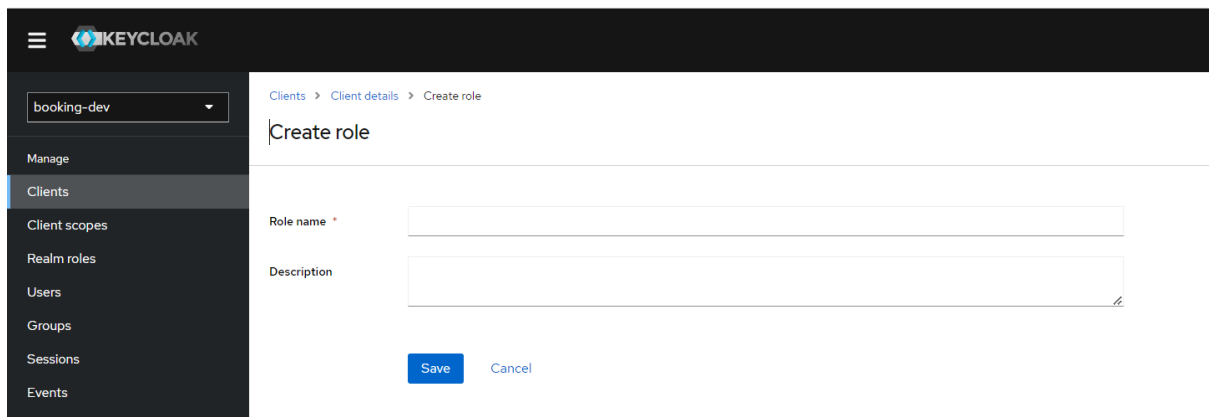
5. Role Creation for client

A Roles tab is displayed for this client. Click the **ROLES** tab and a page similar to the following is displayed:



(5.1) Click on Create role

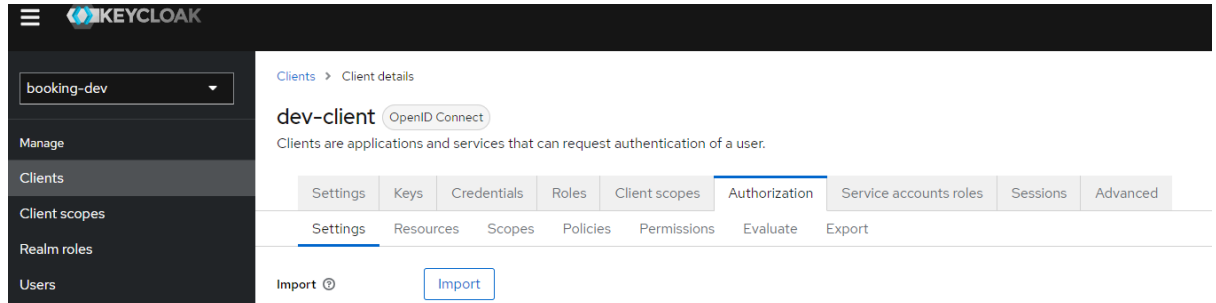
Web page



Enter Role name and Description then click on save .
(you can repeat this process for create multiple role)

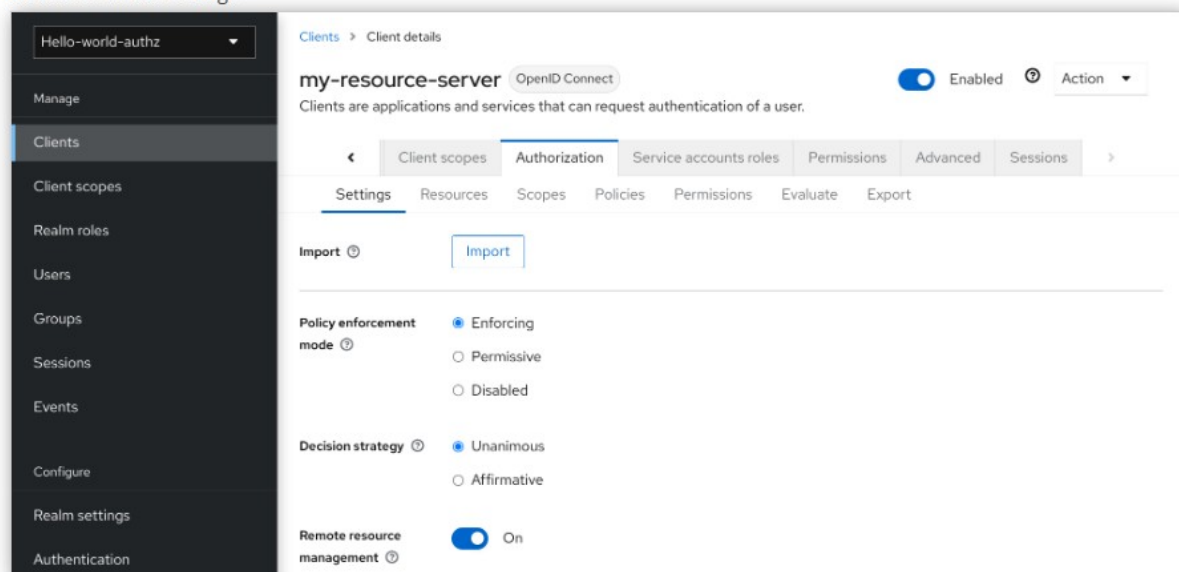
6. Set Authorization for Selected client

(6.1) Click on **Authorization** tab



A Authorization tab is displayed for this client . Authorization page similar to the following is displayed. Authorization tab contain multiple tab also.

Resource server settings



The Authorization tab contains additional sub-tabs covering the different steps that you must follow to actually protect your application's resources. Each tab is covered separately by a specific topic. Here is a quick description about each one:

- **Settings**

General settings for your resource server. For more details about this page see the [Resource Server Settings](#) section.

- **Resource**

From this page, you can manage your application's [resources](#).

- **Authorization Scopes**

From this page, you can manage [scopes](#).

- **Policies**

From this page, you can manage [authorization policies](#) and define the conditions that must be met to grant a permission.

- **Permissions**

From this page, you can manage the [permissions](#) for your protected resources and scopes by linking them with the policies you created.

- **Evaluate**

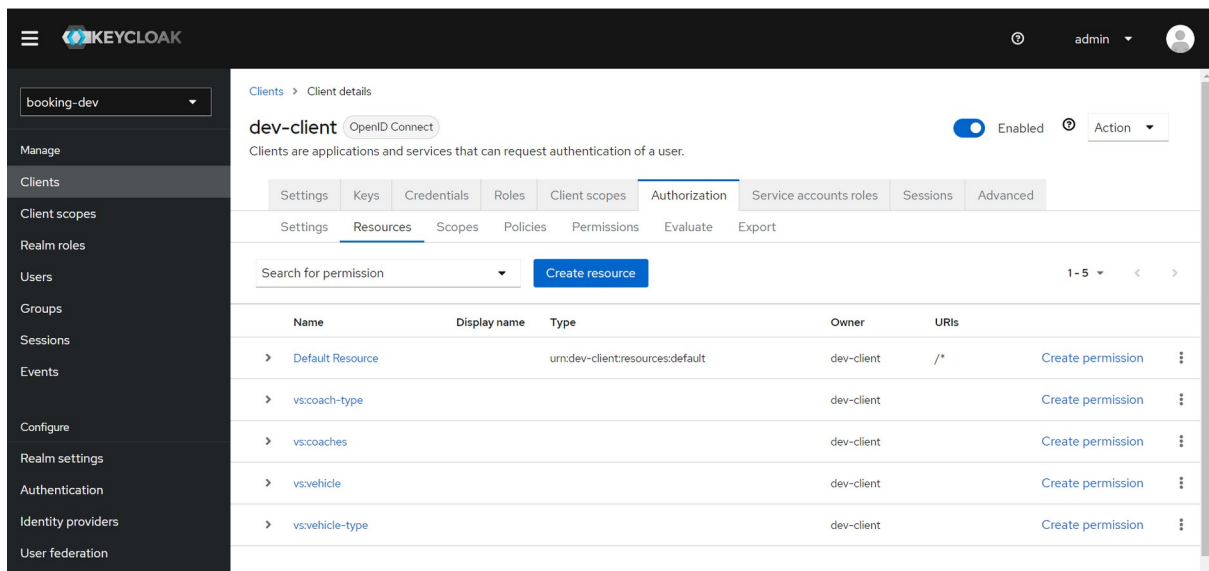
From this page, you can [simulate authorization requests](#) and view the result of the evaluation of the permissions and authorization policies you have defined.

- **Export Settings**

From this page, you can [export](#) the authorization settings to a JSON file.

7. Create a resource

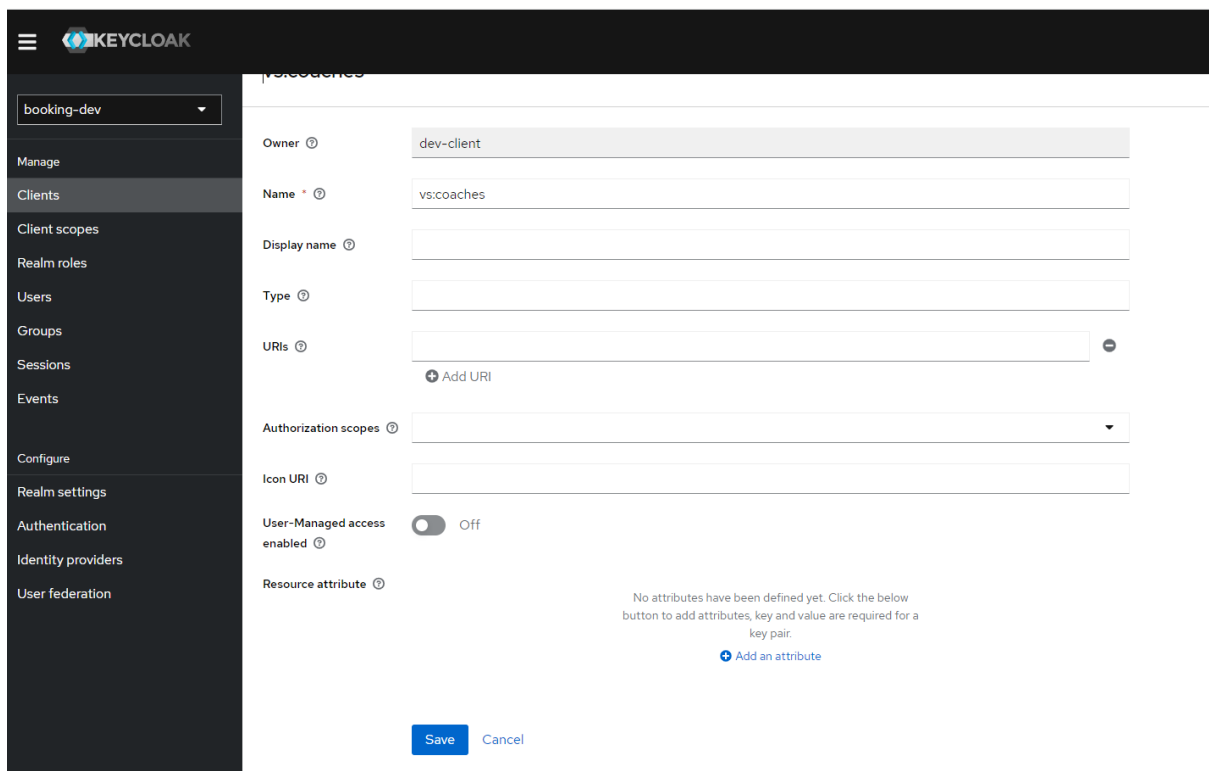
(7.1) Click on Create resource



The screenshot shows the Keycloak Admin Console interface. On the left is a sidebar with navigation options: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Clients > Client details' for 'dev-client'. It includes tabs for Settings, Keys, Credentials, Roles, Client scopes, Authorization (selected), Service accounts roles, Sessions, and Advanced. Under the 'Authorization' tab, there are sub-tabs: Settings, Resources (selected), Scopes, Policies, Permissions, Evaluate, and Export. A search bar for permissions and a 'Create resource' button are present. Below is a table of resources:

Name	Display name	Type	Owner	URIs	
> Default Resource		urn:dev-client:resources:default	dev-client	/*	Create permission ⋮
> vs:coach-type			dev-client		Create permission ⋮
> vs:coaches			dev-client		Create permission ⋮
> vs:vehicle			dev-client		Create permission ⋮
> vs:vehicle-type			dev-client		Create permission ⋮

After clicked on Create resource ,A page similar to the following is displayed.



The screenshot shows the 'Create Resource' form in the Keycloak Admin Console. The sidebar is the same as in the previous screenshot. The main content area is titled 'Resources'. The form fields are:

- Owner: dev-client
- Name: vs:coaches
- Display name: (empty)
- Type: (empty)
- URIs: (empty) with an 'Add URI' button
- Authorization scopes: (empty)
- Icon URI: (empty)
- User-Managed access enabled: Off
- Resource attribute: (empty)

At the bottom, there is a message: 'No attributes have been defined yet. Click the below button to add attributes, key and value are required for a key pair.' with an 'Add an attribute' button. Below the message are 'Save' and 'Cancel' buttons.

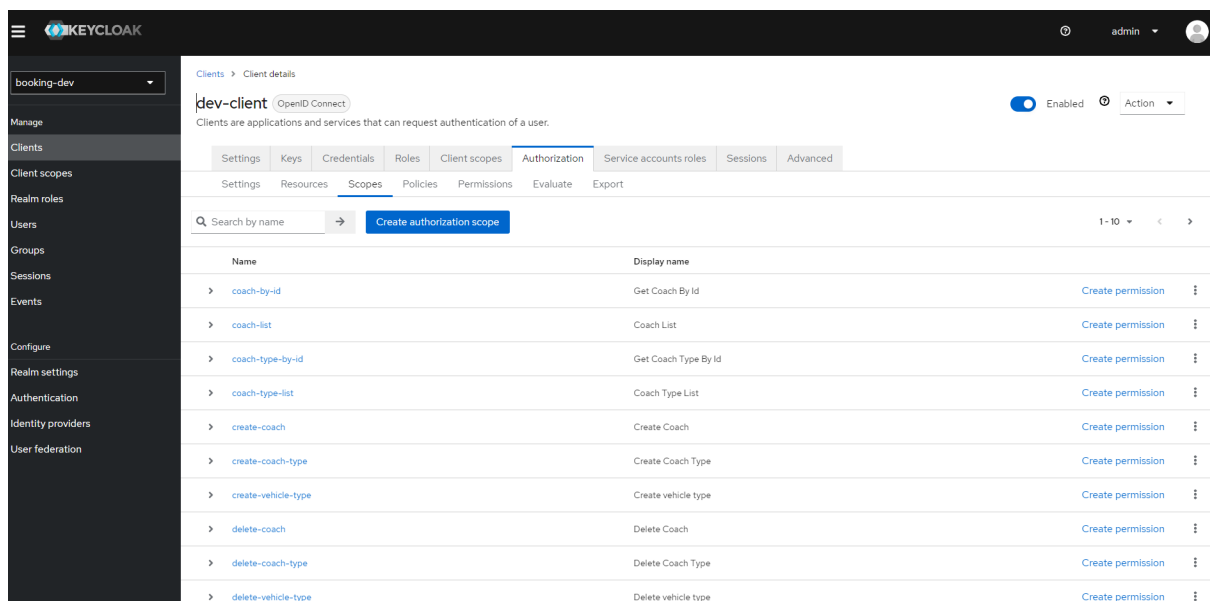
(7.2) Enter the name of resource

(7.3) select required authorization scope (Frist we need to create scope ,I mentioned below how to create resource)

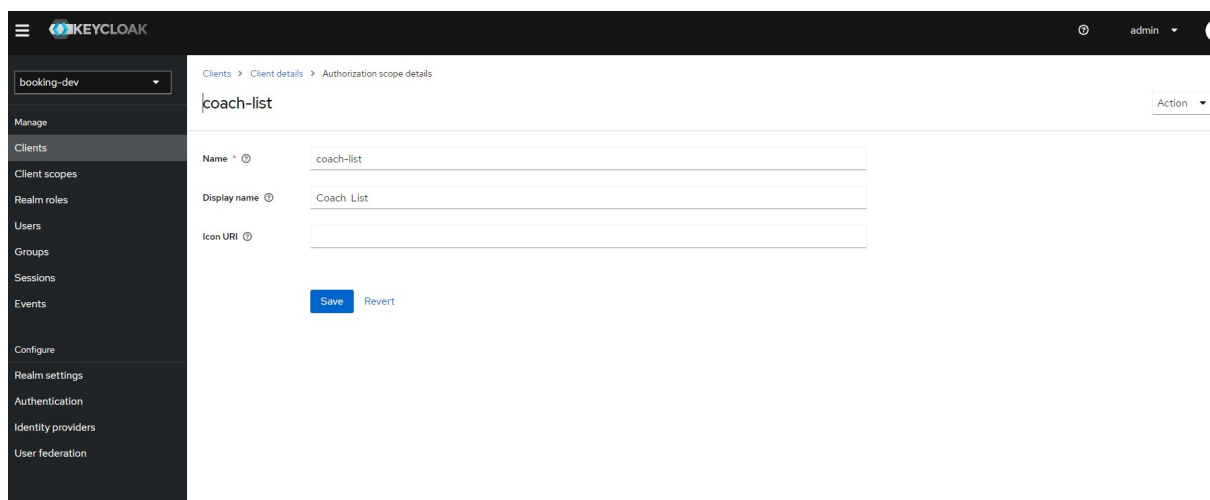
(7.4) click on save

8. Creating Scope

(8.1) Click on Create authentication scope



After clicked on A page similar to the following is displayed.

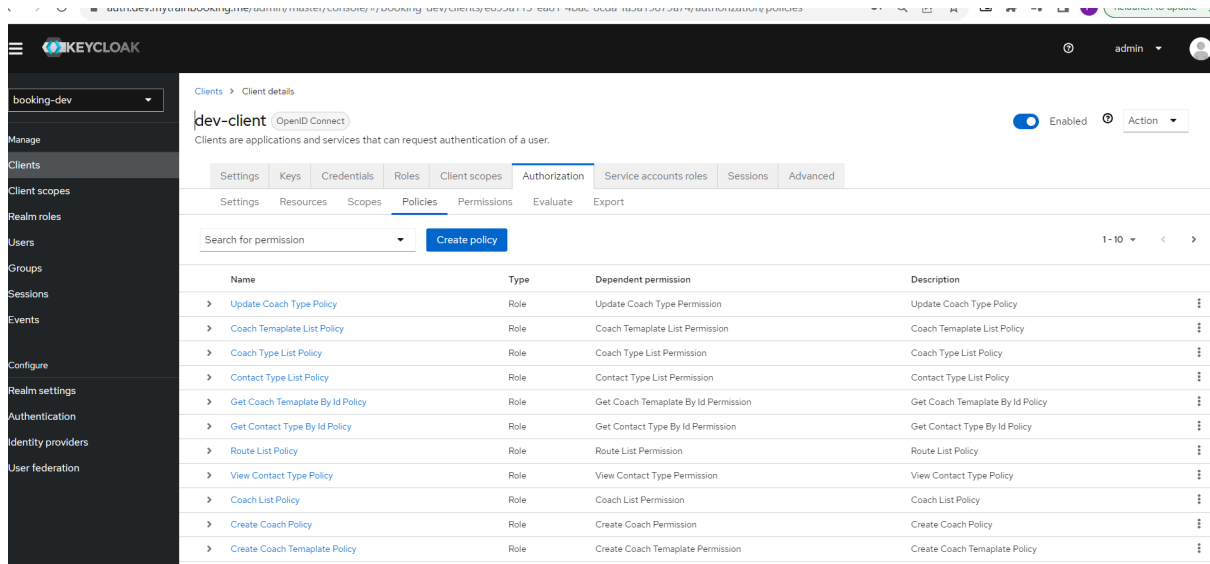


(8.2) Enter the Name and Display Name of the Scope.

9. Creating policy for resource and scope

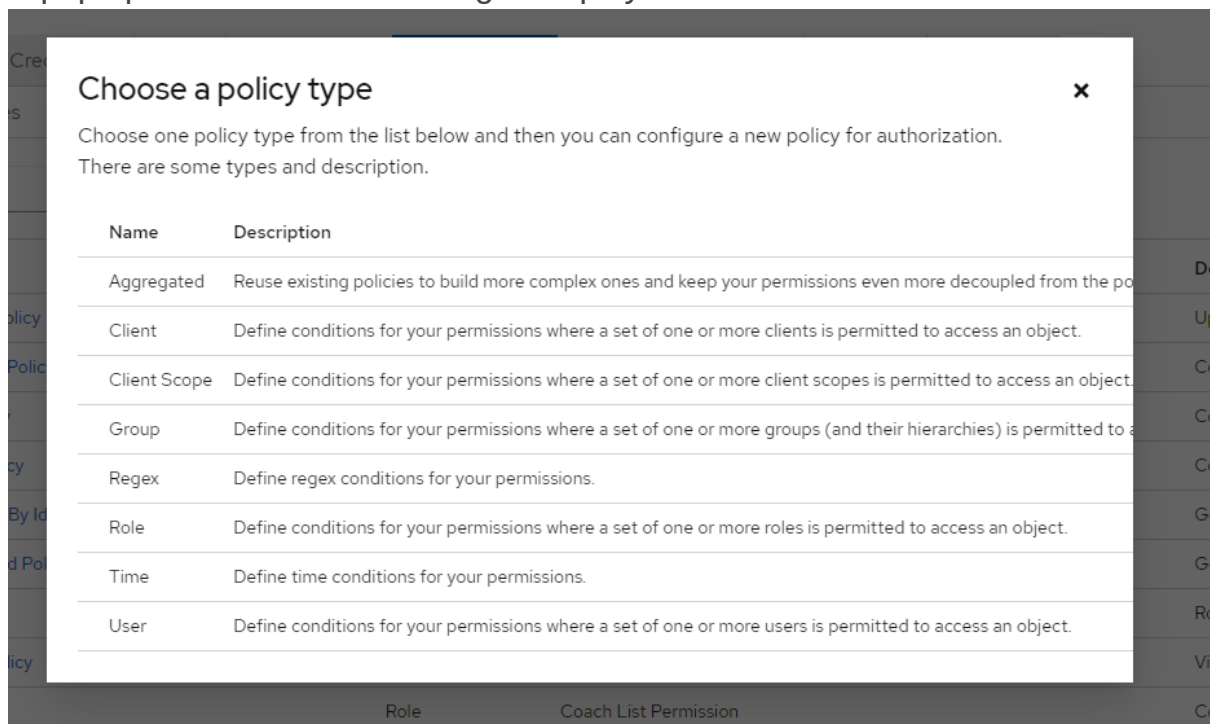
(9.1) Click on Policies tab

A page similar to the following is displayed



(9.2) Click on Create policy

A pop up similar to the following is displayed



You can create multiple type of policy

We are creating Role-based policy

Select Role

A page similar to the following is displayed

The screenshot shows the Keycloak administration interface for updating a policy. The breadcrumb trail is 'Clients > Client details > Policy details'. The page title is 'Update Coach Type Policy'. The left sidebar shows the 'Manage' section with 'Clients' selected. The main form has the following fields:

- Name:** Update Coach Type Policy
- Description:** Update Coach Type Policy
- Roles:** A table with two columns: 'Roles' and 'Required'. The 'Roles' column contains 'dev-client' and 'super-admin'. The 'Required' column has a checkbox that is currently unchecked.
- Logic:** Radio buttons for 'Positive' (selected) and 'Negative'.

At the bottom of the form are 'Save' and 'Cancel' buttons.

*When creating a role-based policy, you can specify a specific role as **Required**. When you do that, the policy will grant access only if the user requesting access has been granted all the required roles. Both realm and client roles can be configured as such.*

Configuration

- **Name**

A human-readable and unique string describing the policy. A best practice is to use names that are closely related to your business and security requirements, so you can identify them more easily.

- **Description**

A string containing details about this policy.

- **Realm Roles**

Specifies which **realm** roles are permitted by this policy.

- **Client Roles**

Specifies which **client** roles are permitted by this policy. To enable this field must first select a **client**.

- **Logic**

The logic of this policy to apply after the other conditions have been evaluated.

Positive and negative logic

Policies can be configured with positive or negative logic. Briefly, you can use this option to define whether the policy result should be kept as it is or be negated.

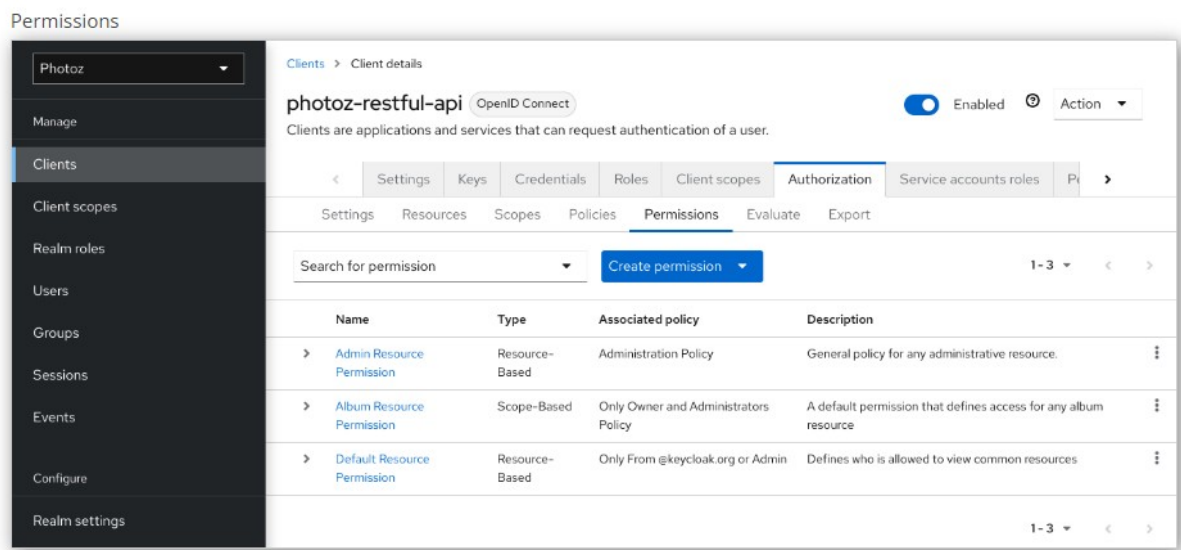
For example, suppose you want to create a policy where only users **not** granted with a specific role should be given access. In this case, you can create a role-based policy using that role and set its **Logic** field to **Negative**. If you keep **Positive**, which is the default behavior, the policy result will be kept as it is.

11. Managing permissions

A permission associates the object being protected and the policies that must be evaluated to decide whether access should be granted.

After creating the resources you want to protect and the policies you want to use to protect these resources, you can start managing permissions. To manage permissions, click the **Permissions** tab when editing a resource server.

(11.1) Click on Permissions tab



Permissions can be created to protect two main types of objects:

- **Resources**
- **Scopes**

To create a permission, select the permission type you want to create from the item list in the upper right corner of the permission listing. The following sections describe these two types of objects in more detail.

Creating scope-based permissions

A scope-based permission defines a set of one or more scopes to protect using a set of one or more authorization policies. Unlike resource-based permissions, you can use this permission type to create permissions not only for a resource, but also for the scopes associated with it, providing more granularity when defining the permissions that govern your resources and the actions that can be performed on them. To create a new scope-based permission, select **Create scope-based permission** from the **Create permission** dropdown.

Add Scope Permission

The screenshot shows a web interface for creating a scope-based permission. On the left is a dark sidebar with a menu: 'Photoz' (selected), 'Manage', 'Clients', 'Client scopes', 'Realm roles', 'Users', 'Groups', 'Sessions', 'Events', 'Configure', 'Realm settings', 'Authentication', and 'Identity providers'. The main content area has a breadcrumb 'Clients > Client details > Create permission' and the title 'Create scope-based permission'. The form includes: a 'Name' text input; a 'Description' text input; a toggle for 'Apply to resource type' (currently 'Off'); three dropdown menus for 'Resources', 'Authorization scopes', and 'Policies'; and a 'Decision strategy' section with three radio buttons: 'Unanimous' (selected), 'Affirmative', and 'Consensus'.

Configuration

- **Name**

A human-readable and unique string describing the permission. A best practice is to use names that are closely related to your business and security requirements, so you can identify them more easily.

- **Description**

A string containing details about this permission.

- **Resource**

Restricts the scopes to those associated with the selected resource. If none is selected, all scopes are available.

- **Scopes**

Defines a set of one or more scopes to protect.

- **Policy**

Defines a set of one or more policies to associate with a permission. To associate a policy you can either select an existing policy or create a new one by selecting the type of the policy you want to create.

12. Evaluating and testing policies

When designing your policies, you can simulate authorization requests to test how your policies are being evaluated.

You can access the Policy Evaluation Tool by clicking the **Evaluate** tab when editing a resource server. There you can specify different inputs to simulate real authorization requests and test the effect of your policies.

Policy evaluation tool

The screenshot shows the 'Policy evaluation tool' interface. On the left is a dark sidebar with a 'Photoz' dropdown and a menu containing: Manage, Clients (selected), Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main panel is titled 'Clients > Client details' and shows the 'photoz-restful-api' client, which is 'Enabled'. Below this is a description: 'Clients are applications and services that can request authentication of a user.' A series of tabs are visible: Settings, Keys, Credentials, Roles, Client scopes, Authorization (selected), and Service accounts roles. Below the tabs is another set of tabs: Settings, Resources, Scopes, Policies, Permissions, Evaluate (selected), and Export. The 'Evaluate' tab contains two sections: 'Identity Information' and 'permissions'. The 'Identity Information' section has three dropdowns: 'Client' (set to 'photoz-restful-api'), 'User' (set to 'Select a user'), and 'Roles' (empty). The 'permissions' section has a toggle for 'Apply to Resource Type' (set to 'Off') and a table for 'Resources and Authentication Scopes'. The table has two columns: 'Key' and 'Value'. The 'Key' column has a dropdown set to 'Select or type a key', and the 'Value' column has a text input set to 'Select or type a key'. There is a '+ addAttributeText' link below the table.

Providing identity information

The **Identity Information** filters can be used to specify the user requesting permissions.

Providing contextual information

The **Contextual Information** filters can be used to define additional attributes to the evaluation context, so that policies can obtain these same attributes.

Providing the permissions

The **Permissions** filters can be used to build an authorization request. You can request permissions for a set of one or more resources and scopes. If you want to simulate authorization requests based on all protected

resources and scopes, click-> **Add** without specifying any Resources or Scopes. When you've specified your desired values, click -> **Evaluate**.