


# Keycloak OAuth

The **Keycloak OAuth Policy** is a Keycloak-specific OAuth2 policy to regulate access to APIs. Keycloak's token format and auth mechanism facilitate excellent performance characteristics and enable users to easily tune the setup to meet their security requirements. In general, this is a good approach for achieving security without greatly impacting performance.



Do **not** use the **Keycloak OAuth** policy together with the other authentication policies [BASIC Authentication](#) and [JWT](#). The chaining of these policies does not currently work, but this may change in future versions.

Related Pages:

- [Policies](#)
  - [Assigning Policies](#)
  - [Policy Configuration](#)
- [API Management Best Practices](#)
  - [API Security: Authentication and Authorization](#)

Policy Type

Keycloak OAuth Policy

Keycloak OAuth Policy Configuration

Keycloak OAuth Policy Configuration

Require OAuth

true

Terminate request if no OAuth is provided.

Require Transport Security

true

Any request without transport security will be rejected. OAuth2 requires transport security (e.g. TLS, SSL) to provide protection against replay attacks. It is strongly recommended to enable this option.

Blacklist Unsafe Tokens

false

Any tokens used without transport security will be blacklisted in all gateways to mitigate associated security risks. Uses distributed data store to share blacklist.

Strip Tokens

false

Remove any authorization header or token query parameter before forwarding traffic to the API.

Realm

Realm name. Must be a full ISS domain path (e.g. https://mykeycloak.local/auth/realms/apimanrealm) is required.

Keycloak Realm Certificate

To validate OAuth2 requests. Must be a PEM-encoded X.509 certificate. If you leave this empty the policy will try to get the public-keys directly from your Keycloak.

Forward Authorization Roles

Forward Realm Roles

Forward Keycloak roles to the Authorization policy. You should specify your required role(s) in the Authorization policy's configuration.

Forward Realm Roles?

false

Delegate Kerberos Ticket

false

[Goto Keycloak Kerberos Credential Delegation Guide](#)  
Delegate any Kerberos ticket embedded in the Keycloak token to the API (via the authorization header).


Forward Keycloak Token Information

Fields from the token can be set as headers and forwarded to the API. All standard claims, custom claims and ID token fields are available (case sensitive). A special value of access\_token will forward the entire encoded token. Nested claims can be accessed by using javascript dot syntax (e.g. address.country, address.formatted).

+ Header


Add Policy



Cancel



Use the provided links underneath the fields **Delegate Kerberos Ticket** and **Header** to open further information on the subjects.

## Configuration Options

Option	Type	Description	Possible Values	Default
Require OAuth	Boolean	Terminate request if no OAuth token is provided. <div><div><p>Make sure that this option is true if you want to use this policy for authentication.</p></div></div>	true / false	true

<b>Require Transport Security</b>		Boolean	Any request used without transport security will be rejected. OAuth2 requires transport security (e.g. TLS, SSL) to provide protection against replay attacks.  <div> Please disable the TLS check if you are using Scheer PAS 21.1 or a newer version, because all PAS components are running behind a proxy server.</div>	true / false	false
<b>Blacklist Unsafe Tokens</b>		Boolean	Any tokens used without transport security will be blacklisted in all gateways to mitigate associated security risks. Uses distributed data store to share blacklist.	true / false	false
<b>Strip Tokens</b>		Boolean	Remove any Authorization header or token query parameter before forwarding traffic to the API.	true / false	false
<b>Realm</b>		String	Enter the realm name. It must be a full ISS domain path, for example <a href="http://scheer-keycloak:8080/pas-doc/keycloak/realms/PAS">http://scheer-keycloak:8080/pas-doc/keycloak/realms/PAS</a>	-	-
<b>Keycloak Realm Certificate</b>		String	To validate OAuth2 requests. Must be a PEM-encoded X.509 certificate. You can copy it from the Keycloak console.  If you leave this field empty, the policy will try to get the public keys directly from your Keycloak.	-	-
<b>Forward Authorization Roles</b>		Enum	Choose the type of roles to forward.  <div> It is not possible to forward realm <b>and</b> application roles, only one or the other.</div>	Forward Realm Roles  Forward Application Roles	Forward Realm Roles
<b>Forward Realm Roles?</b>		Boolean	Select whether to forward <b>realm roles</b> .	true / false	false
<b>Forward Application Roles?</b>		Boolean	Select whether to forward <b>application roles</b> .	true / false	false
<b>Application Name</b>		String	Which application roles to forward. If you choose to forward application roles, you must provide the <code>applicationName</code> .	-	-
<b>Delegate Kerberos Ticket</b>		Boolean	Delegate any Kerberos Ticket embedded in the Keycloak token to the API (via the Authorization header).	true / false	false
<b>Header</b>		Array [<forwardAuthInfo>]	Set auth information from the token into header(s).	-	-
<b>Forward AuthInfo Options</b>					
	<i>Headers</i>	String	The header value to set (to paired field).		
	<i>Field</i>	String	The token field name.		