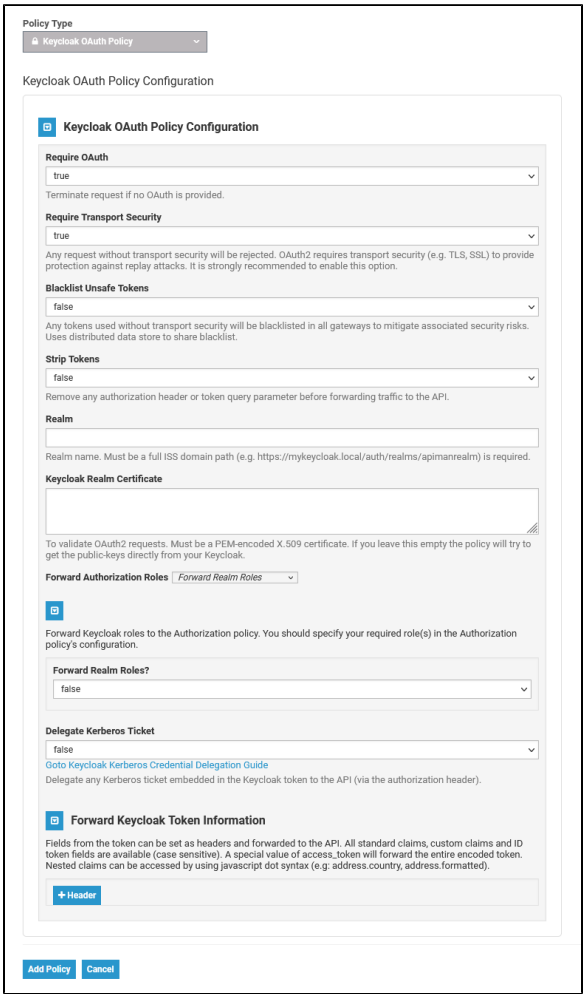
Keycloak OAuth

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.



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

Related Pages:

- Policies
  - Assigning Policies
  - Policy Configuration
- API Management Best Practices
  - API Security: Authentication and Authorization

Configuration Options

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

Require Transport Security		Boolean	Any request used without transport security will be rejected. OAuth2 requires transport security (e.g. TLS, SSL) to provide protection against replay attacks. ⚠ 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.	true / false	false
Blacklist Unsafe Tokens		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
Strip Tokens		Boolean	Remove any Authorization header or token query parameter before forwarding traffic to the API.	true / false	false
Realm		String	Enter the realm name. It must be a full ISS domain path, for example **http://scheer-keycloak:8080/pas-doc/keycloak/realms/PAS**	-	-
Keycloak Realm Certificate		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.	-	-
Forward Authorization Roles		Enum	Choose the type of roles to forward. ⓘ It is not possible to forward realm **and** application roles, only one or the other.	Forward Realm Roles Forward Application Roles	Forward Realm Roles
Forward Realm Roles?		Boolean	Select wether to forward **realm roles**.	true / false	false
Forward Application Roles?		Boolean	Select whether to forward **application roles**.	true / false	false
Application Name		String	Which application roles to forward. If you choose to forward application roles, you must provide the `applicationName`.	-	-
Delegate Kerberos Ticket		Boolean	Delegate any Kerberos Ticket embedded in the Keycloak token to the API (via the Authorization header).	true / false	false
Header		Array [<forwardAuthInfo>]	Set auth information from the token into header(s).	-	-
	Forward AuthInfo Options				
	Headers	String	The header value to set (to paired field).		
	Field	String	The token field name.		