INSIGHTS

# 10 Keycloak Connect Best Practices

Keycloak Connect is a powerful authentication and authorization service that can be used to secure applications and services. Here are 10 best practices to follow.

KI Kristin Ingber

Published Dec 22, 2022

Keycloak Connect is an open source identity and access management solution that enables organizations to secure their applications and services. It provides a unified interface for authentication, authorization, and user management.

In this article, we will discuss 10 best practices to follow when using Keycloak Connect. Following these best practices will ensure that your applications and services are secure and that your users have a seamless experience.

## 1. Configure Keycloak to use HTTPS for all communication

HTTPS is a secure protocol that encrypts data sent between the client and server, preventing malicious actors from intercepting or manipulating sensitive information. This ensures that user credentials are not exposed to potential attackers, protecting both users and Keycloak Connect applications.

To configure Keycloak to use HTTPS for all communication, you must first obtain an SSL certificate from a trusted Certificate Authority (CA). Once obtained, the certificate should be imported into the Keycloak server's keystore. After this is done, the Keycloak admin console can be used to enable HTTPS by setting the "Secure" flag in the "Web Origins" section of the realm settings. Finally, the application must be configured to use HTTPS when connecting to the Keycloak server.

## 2. Enable client registration and authentication policies

Client registration policies allow for the creation of a secure environment by ensuring that only authorized clients can access Keycloak Connect. This is done by requiring clients to register with Keycloak and provide authentication credentials, such as an API key or OAuth token. Once registered, clients are assigned roles and permissions which determine what they can do within the system.

Authentication policies ensure that all requests made to Keycloak Connect are authenticated before being processed. This helps protect against malicious actors attempting to gain unauthorized access to the system. Authentication policies also help prevent data breaches by verifying the identity of users who attempt to access sensitive information.

Then, they must add the necessary client registrations and authentication policies to the realm. Finally, they must assign roles and permissions to each client so that they have the correct level of access to the system.

## 3. Utilize the admin console for user management

The admin console provides a centralized, secure interface for managing users and their associated roles. It allows administrators to quickly create new user accounts, assign them roles, and manage existing user accounts with ease. Additionally, the admin console is designed to be intuitive and easy-to-use, making it ideal for those who are not familiar with Keycloak Connect or its underlying technology.

Using the admin console also ensures that all user management activities are performed in a secure environment. All data stored within the admin console is encrypted, ensuring that only authorized personnel can access it. Furthermore, the admin console supports role-based access control (RBAC), allowing administrators to restrict user access based on their assigned roles. This helps ensure that only those with the appropriate permissions can perform certain tasks.

## 4. Set up roles, groups, and permissions in Keycloak

Roles are used to define the type of access a user has. They can be assigned to individual users or groups, and they determine what actions a user is allowed to take within an application. For example, a role might allow a user to view certain pages but not edit them.

Groups are collections of users that share common roles and permissions. This allows administrators to easily assign roles and permissions to multiple users at once. Groups also make it easier to manage large numbers of users since changes made to one group will apply to all members of that group.

Permissions are specific rules that control how users interact with applications. Permissions can be set on individual resources (such as files) or on entire applications. For example, a permission might allow a user to read a file but not modify it.

Keycloak Connect makes it easy to set up roles, groups, and permissions in Keycloak. Administrators can create roles, assign them to users or groups, and then configure permissions for each role. This ensures that only authorized users have access to the appropriate resources. Additionally, Keycloak Connect provides an audit trail so administrators can track who accessed which resources and when.

## 5. Create a secure token service (STS) endpoint to authenticate users

The STS endpoint is a secure, dedicated URL that can be used to authenticate users. It provides an extra layer of security by ensuring that only authenticated requests are allowed access to the Keycloak Connect API. This helps protect against malicious

To create an STS endpoint, you must first configure your application to use OAuth 2.0 and OpenID Connect (OIDC). Once configured, you can then set up the STS endpoint in your application's configuration file. The endpoint should include the following information:

- Client ID – A unique identifier for the client application
- Client Secret – A secret key used to authenticate the client
- Authorization Endpoint – The URL where the user will be redirected to authorize the request
- Token Endpoint – The URL where the token will be exchanged for authentication
- User Info Endpoint – The URL where the user's profile information will be retrieved

Once the STS endpoint is created, it can be used to authenticate users when they attempt to access the Keycloak Connect API. When a user attempts to access the API, their credentials will be sent to the STS endpoint, which will validate them and return a token if successful. This token can then be used to access the API.

## 6. Implement OAuth 2.0 authorization flows

OAuth 2.0 is an open standard for authorization that provides a secure way to access resources without having to share credentials. It allows users to grant third-party applications access to their data without sharing passwords or other sensitive information. OAuth 2.0 also enables the user to control which permissions are granted and revoked, providing more granular control over how their data is used.

When using Keycloak Connect, implementing OAuth 2.0 authorization flows helps ensure that only authorized clients can access protected resources. This is done by requiring the client to authenticate itself with an access token before being allowed to make requests. The access token contains information about the client's identity and what it is allowed to do. By validating this token, Keycloak Connect can verify that the client has permission to access the requested resource.

## 7. Leverage OpenID Connect protocol for single sign-on

OpenID Connect is an authentication protocol that enables users to securely authenticate with a single identity provider, such as Keycloak. It provides a secure and easy way for applications to communicate with the identity provider in order to verify user identities and obtain access tokens. This eliminates the need for users to remember multiple usernames and passwords for different applications.

Using OpenID Connect also allows developers to easily integrate their applications with Keycloak Connect. The protocol defines how applications should interact with the identity provider, making it easier for developers to implement authentication without having to write custom code. Additionally, OpenID Connect supports various security features, such as encryption and signing of messages, which helps protect user data from malicious actors.

Access tokens are used to authenticate a user and grant access to protected resources, while refresh tokens are used to obtain new access tokens. Storing these tokens securely is important because they contain sensitive information that can be used to gain unauthorized access to the system.

The best way to store access tokens and refresh tokens is by using an encrypted database or file storage system. This ensures that only authorized users have access to the data and prevents malicious actors from gaining access to it. Additionally, it's important to use strong encryption algorithms such as AES-256 to ensure that even if someone were able to gain access to the token data, they would not be able to decrypt it. Finally, it's also important to regularly rotate the tokens to reduce the risk of them being compromised.

## 9. Monitor user activity with audit logs

Audit logs provide a detailed record of user activity, including authentication attempts, authorization decisions, and other events. This information can be used to detect suspicious behavior or unauthorized access attempts, as well as to identify potential security vulnerabilities in the system. Additionally, audit logs can help with troubleshooting issues by providing an accurate timeline of events that occurred prior to the issue.

To monitor user activity with audit logs, Keycloak Connect provides several options. The first is to enable logging for specific operations such as authentication, authorization, and token issuance. This allows administrators to track which users are accessing what resources and when. Additionally, Keycloak Connect also supports log aggregation, which allows administrators to collect and analyze log data from multiple sources in one place. Finally, Keycloak Connect offers integration with external logging solutions such as Splunk and ELK Stack, allowing administrators to store and analyze log data more efficiently.

## 10. Protect against brute force attacks by setting up rate limiting

Rate limiting is a security measure that limits the number of requests an IP address can make to a server within a certain period of time. This helps protect against brute force attacks, which are attempts to guess passwords by repeatedly trying different combinations until one works. By setting up rate limiting, Keycloak Connect can detect when someone is making too many requests and block them from accessing the system. This prevents attackers from being able to guess passwords or other sensitive information. Additionally, it also reduces the load on the server, as it will not have to process all of the malicious requests. To set up rate limiting in Keycloak Connect, administrators should configure the maximum number of requests allowed per minute for each client. They should also set up a blacklist of IP addresses that have been identified as malicious. Finally, they should monitor the logs regularly to identify any suspicious activity.

KI

## Kristin Ingber

Kristin Ingber has been working in the security industry for over 10 years. She started her career as a security guard and has since worked her way up to a management position. Kristin is a strong advocate for security training and has helped many individuals and businesses create and implement security plans.

## You may also be interested in...

INSIGHTS

### 10 SQL Server Partitioning Best Practices

MR Milton Rosco
Nov 10, 2022

INSIGHTS

### 10 React Naming Conventions Best Practices

KS Kenneth Santos
Dec 4, 2022

INSIGHTS

### 10 Web Session Timeout Best Practices

RD Robert Deauville
Nov 30, 2022

INSIGHTS

### 10 Button Widths Design System Best Practices

LM Leslie Minton
Dec 23, 2022





CLIMB