Keycloak Project Structure

Open-source identity and access management. A mature, modular Java application leveraging Quarkus for cloud-native performance and SPIs for extreme extensibility.

Updated 2025-12-30

#keycloak #java #quarkus #oidc #saml #auth #iam

PNG PDF Copy Prompt

Project Directory

```
keycloak/
  services/ Core Auth & Tok…
    src/main/java/
    pom.xml
  quarkus/ Quarkus-based D…
    runtime/
    server/
  server-spi/ Service Provide…
  model/ Persistence & S…
    jpa/ Relational DB s…
    infinispan/ Distributed cac…
  js/ Modern Admin UI…
    apps/admin-ui/
    package.json
  federation/ LDAP & Kerberos…
  pom.xml Root Maven Conf…
```

Repository Info

Repository - keycloak/keycloak

Stars - 22k+

License - Apache-2.0

Last Analyzed - December 2025

Tech Stack

Language - Java 21+

Runtime - Quarkus

Frontend - React (Admin UI)

Database - Postgres / MySQL / Oracle

Build Tool - Maven

Caching - Infinispan

Architecture Notes

Keycloak is built on a highly modular **SPI (Service Provider Interface)** architecture. Almost every feature—from authentication flows to database storage—is a 'Provider' that implements a specific 'Interface'. This allows developers to plug in custom logic without modifying the core. Recently, Keycloak migrated its runtime from WildFly to **Quarkus**, making it significantly more efficient in containerized environments.

Key Directories

services/ - Contains the implementation of OIDC, SAML, and the actual authentication state machine.

server-spi/ - Defines the contracts for all extensions. If you want to write a custom Keycloak plugin, you'll be implementing interfaces defined here.

quarkus/server/ - The entry point for the modern distribution. It configures the Quarkus application and bundles all modules into the final executable.

Why This Structure?

Keycloak is the enterprise standard for open-source IAM. Its codebase is a prime example of how to design a deeply extensible system using Java's modularity patterns.