**KEYCLOAK**

1. Định nghĩa

* Là 1 hệ thống quản lý mã nguồn mở hỗ trợ quản lý và xác thực tài khoản người dùng

1. Thành phần

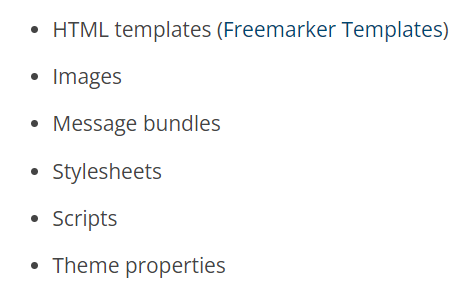
* Realms (vùng miền):
  + Là 1 tập hợp riêng biệt bao gồm users, group, role, setting. Các realms luôn tách biệt với nhau
  + Có thể tạo nhiều realms khác nhau để dễ dàng quản lý và xây dựng các services khác nhau
* Client (khách hàng):
  + Đại diện cho các ứng dụng, services có thể là API, web, app
  + Client nằm ở trong realms và dùng để xác thực quyền cụ thể cho từng client
* User (người dùng):
  + Là user
* Roles (vai trò):
  + Là quyền của cho user hoặc group có thể sử dụng và truy cập ở trong realms hoặc client
* Group (nhóm):
  + Là tập hợp các User với nhau cùng chung 1 roles để dễ dàng quản lý
  + 1 User có thể thuộc nhiều group
* Identity providers (nhà cung cấp):
  + Cho phép keycloak có thể kết nối đến các dịch vụ bên ngoài như Google, Github, Facebook, …
* Authentication flows (Luồng xác thực):
  + Xác định cách thức người dùng được xác thực khi đăng nhập vào một Client
  + Ví dụ: multi-factor authentication (MFA) hoặc one-time password (OTP)

1. Cách hoạt động

* SSO: Single Sign On
* Hệ thống nhận dạng sẽ gồm 4 nền tang:
  + Xác thực (Authentication)
  + Phân quyền (Authorization)
  + Trao đổi thông tin người dùng (User attributes exchange)
  + Quản lý người dùng (User management)
* Có 2 giao thức phổ biến cho SSO:
  + SAML(Security assertion markup language)
  + OpenID: JWT
* docker run --name keycloak -p 8000:8000 -e KEYCLOAK\_ADMIN=admin -e KEYCLOAK\_ADMIN\_PASSWORD=admin quay.io/keycloak/keycloak start-dev
* Login:
  + Sau khi user truy cập vào ứng dụng và vào trang login
  + Ứng dụng sẽ chuyển hướng sang trang keycloak và login tài khoản như bình thường
  + Sau khi nhập form và submit, keycloak sẽ hỗ trợ xác thực tài khoản dựa trên roles hoặc client
  + Nếu thành công, keycloak sẽ tạo access token, refesh token, ID token

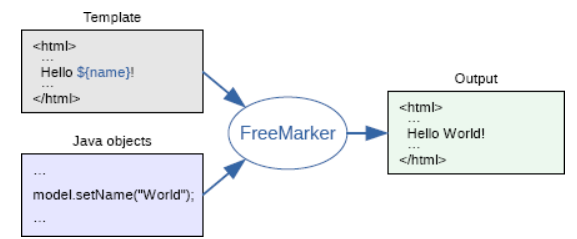
1. Custom lại theme

* File UI của keycloak sẽ hoạt động gồm

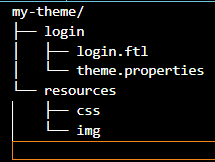


* Trong đấy file Freemarker Templates (ftl) là 1 thư viện Java dùng để tạo ra văn bản. Freemarker cho phép chèn logic code vào trong code.
* Tuy nhiên không nên tự code lại file ftl do chúng ta không biết được cách mà hệ thống keycloak gửi request từ form đến hệ thống ntn

🡺 Chúng ta sẽ sử dụng source code cũ và đè lên trên đấy code mới của chúng ta nhưng vẫn giữ nguyên cách form gửi request (github của 1 trong những theme: [KeycloakTheme](https://github.com/Halazv2/keycloak-theme/tree/master))



* Để tạo thành công nên tạo thành 1 folder được cấu tạo nhưng do lỗi nên là hiện tại chưa thể link css mà phải code css trực tiếp trong ftl



1. Google authentication

* Sử dụng tài khoản gmail, google để đăng nhập vào web app
* Để sử dụng được thì mình sẽ cần tạo project ở trên [Google Cloud](https://console.cloud.google.com/)
* Sau đấy trong mục API & Service, chọn Oauth consent sẽ có 2 lựa chọn Internal (Chỉ cho phép user thuộc cùng 1 tổ chức, công ty mới vào được), External (Cho phép tất cả user có tài khoản gg vào)
* Sau khi điền hết thông tin ở bên Oauth consent thì qua Credential và điền 1 số thông tin cơ bản ở phần Authorized redirect URLs thì điền link do keycloak tạo ra.



* Sau đó gg sẽ tạo ra Client ID và Client Secret key để chúng ta add vào keycloak là xong

