O que é Keycloak e como começar a usar

24/09/2020 Wilde Rossi Code Journey, Desenvolvimento 0 Comments

Já parou pra pensar quantas vezes desenvolvemos **funcionalidades iguais pra sistemas diferentes**? Provavelmente a feature mais replicada em toda a história da computação é a de autenticação e autorização. Pense em quantas vezes você já criou uma classe User, com atributos name, username, email, password…

Foi pra tentar tornar esse processo menos chato para você, querido desenvolvedor, que surgiu o Keycloak.

O que é Keycloak?

Patrocinado pela Red Hat, o Keycloak é um software open source (https://github.com/keycloak/keycloak) de um servidor JBoss feito para trabalhar em conjunto com sua aplicação em implementações mais comuns de autenticação e autorização. Caso as configurações padrão não te atendam, existem várias configurações e customizações que podem ser feitas para adequar o funcionamento ao seu sistema.

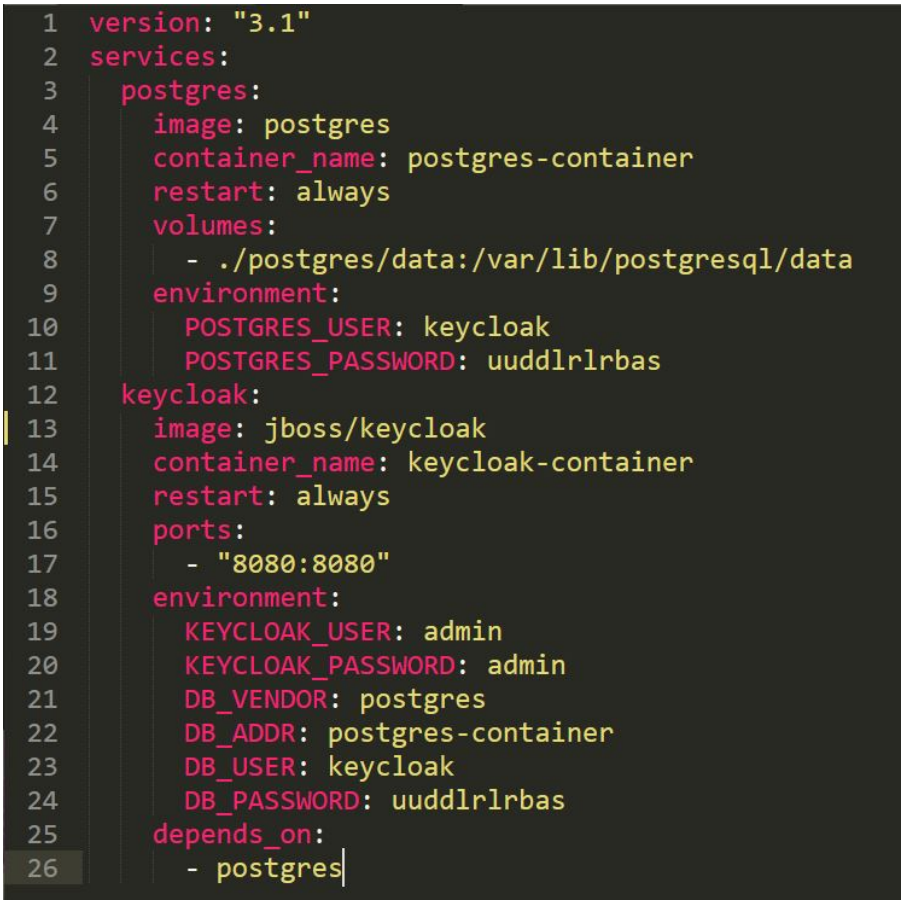
O que ele me oferece?

São oferecidas pelo Keycloak funcionalidades como:

- Criação de usuário (podendo ser criados pelo administrador do sistema, e habilitado ou não para o próprio usuário se cadastrar);
- Login, “esqueci minha senha”, login com plataformas externas como redes sociais;
- Integração dos usuários com Active Directory;
- Ativação de usuários por confirmação de e-mail;
- Obrigar aceitação de termos de uso antes de logar;
- Customização das páginas que o usuário acessa (como a própria tela de login);
- Serviços para que outras aplicações busquem dados de usuários;
- Criação de permissões que seu sistema vai usar;
- Criação de grupos de usuário;
- i18n (internationalization);
- Entre outras várias configurações e customizações mais avançadas.

Legal! Por onde começamos?

O primeiro passo pra usarmos o Keycloak é baixar e rodar a versão mais recente na sua máquina. Você pode baixar direto no site oficial. Se você é um amante de conteinerização como eu, vai saber o que fazer com esse link do Docker.



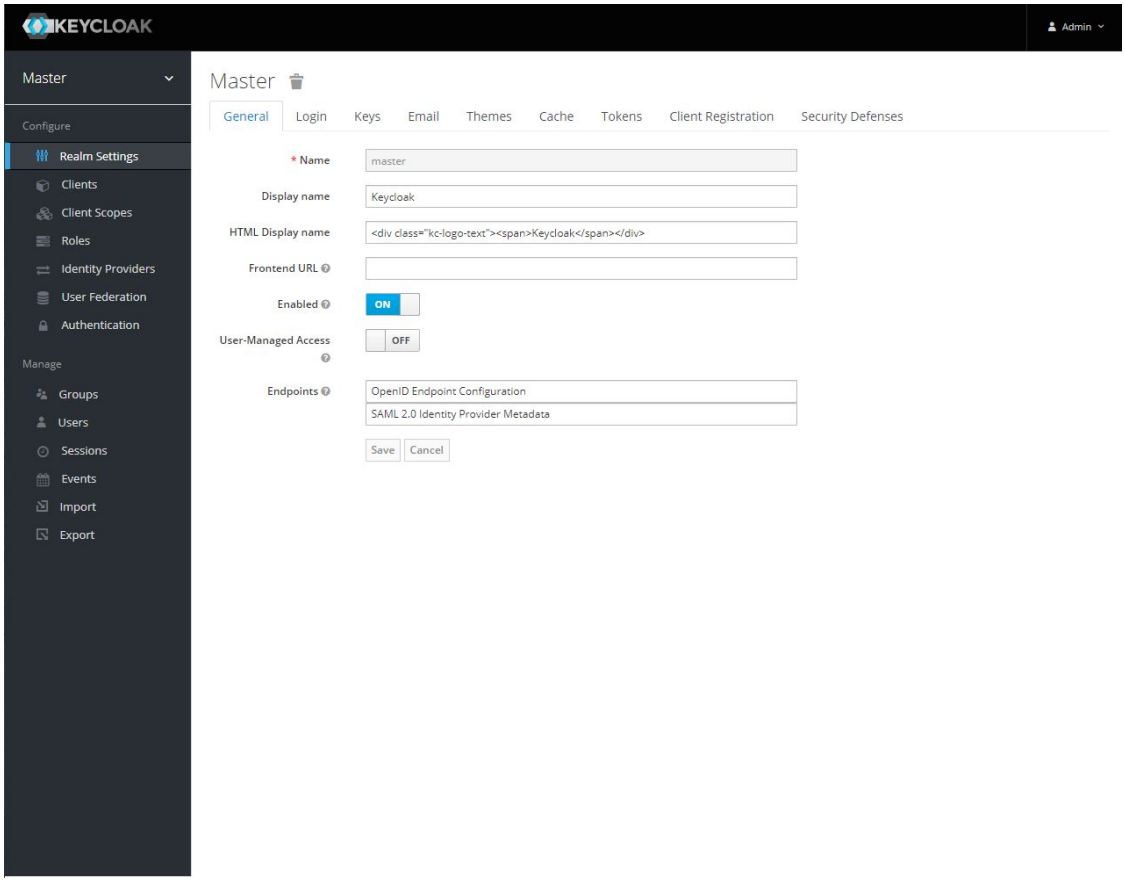
```yaml
version: "3.1"
services:
  postgres:
    image: postgres
    container_name: postgres-container
    restart: always
    volumes:
      - ./postgres/data:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: keycloak
      POSTGRES_PASSWORD: uuddlrlrbas
  keycloak:
    image: jboss/keycloak
    container_name: keycloak-container
    restart: always
    ports:
      - "8080:8080"
    environment:
      KEYCLOAK_USER: admin
      KEYCLOAK_PASSWORD: admin
      DB_VENDOR: postgres
      DB_ADDR: postgres-container
      DB_USER: keycloak
      DB_PASSWORD: uuddlrlrbas
    depends_on:
      - postgres
```

Um exemplo de docker-compose.yml para subir um Keycloak com banco de dados Postgres

Se você baixou pelo site, agora vai precisar iniciar o servidor. Para isso existe um script na pasta /bin chamado standalone.bat (para Windows), ou standalone.sh (para Linux e afins). Iniciando por aqui, o Keycloak vai gravar os dados em um banco de dados H2, então vale lembrar que para ambientes de produção o recomendado é subir via contêiner, e especificando outro banco nas variáveis de ambiente, conforme mostrado anteriormente.

Aguarde alguns instantes enquanto a aplicação se inicializa, e abra seu localhost, na porta 8080, no próprio navegador (http://localhost:8080). Agora vamos acessar a parte do Administration Console. Se você inicializou pelo docker, o login e senha será o que está definido nas variáveis de ambiente **KEYCLOAK_USER** e **KEYCLOAK_PASSWORD**.

Se você inicializou pelos scripts de standalone, vai precisar definir usuário e senha do usuário administrador. Agora que temos o usuário, vamos clicar em Administration Console, logar com esse usuário, e vamos visualizar uma tela parecida com isso.



Tela inicial do Keycloak

Primeiros conceitos para o Keycloak

Antes de ligarmos nossa aplicação no keycloak, precisamos entender alguns conceitos para não fazermos bobagem.

- Realms

Observe que próximo ao canto superior esquerdo está escrito “Master”. Ali é mostrado o realm atual em que você está. Para fins didáticos, você pode entender realm como um **balde de usuários**. Normalmente usamos o realm Master para os administradores do sistema, já que seu usuário de administrador está criado nele. É extremamente recomendado que para começarmos as configurações para nossa aplicação, criemos antes um outro realm para os usuários que vão acessar nosso sistema. A ideia por trás disso é não permitir que os administradores usem o(s) sistema(s) que o Keycloak protege, e muito menos que usuários normais tenham acesso a configurações de administrador. Para criar esse novo realm, coloque o mouse sobre o Master, e clique em “Add realm”.

- Realm Settings

Agora que criamos o realm para nossos usuários do sistema que queremos proteger, nos deparamos com uma tela com várias abas, onde podemos alterar várias configurações do nosso realm. As abas principais que você precisa conhecer daqui são General (define coisas como o nome do realm que será mostrado na tela de login), Login (define campos que serão mostrados na tela de login, e algumas regras) e Email (define as configurações do e-mail que será utilizado para fazer validações como verificação de e-mail, quando habilitado).

- Clients

Seguindo pela ordem dos menus, nos clients vamos cadastrar as aplicações que queremos proteger com o Keycloak. Não remova as aplicações que já estão cadastradas por padrão. Logo mais vamos entrar nos detalhes desse cadastro.

- Roles

São as permissões que teremos dentro do nosso sistema. Elas podem ser atribuídas a usuários ou a grupos de usuário. Dentro das nossas aplicações vamos usar os nomes cadastrados aqui para liberar acesso ou não de um determinado recurso para o usuário logado. Temos dois tipos de roles, as normais, e as composite roles. As normais servem para te apoiar a fazer de fato o controle de acesso a recursos, enquanto as composite roles são feitas para agregar outras roles, facilitando liberação de roles em lote. Também existe uma outra aba nessa tela, chamada “Default Roles”. Nessa aba são cadastradas as roles que os usuários terão por padrão, assim que forem criados.

- Identity Providers

Aqui podemos configurar login usando outras plataformas, tais como redes sociais, ou até mesmo outro keycloak. Também é necessária alguma configuração nessas outras plataformas para que o login funcione bem.

- User Federation

Serve para usarmos bases de dado externas na hora de fazer login. Você pode configurar login com um Active Directory aqui.

- Groups

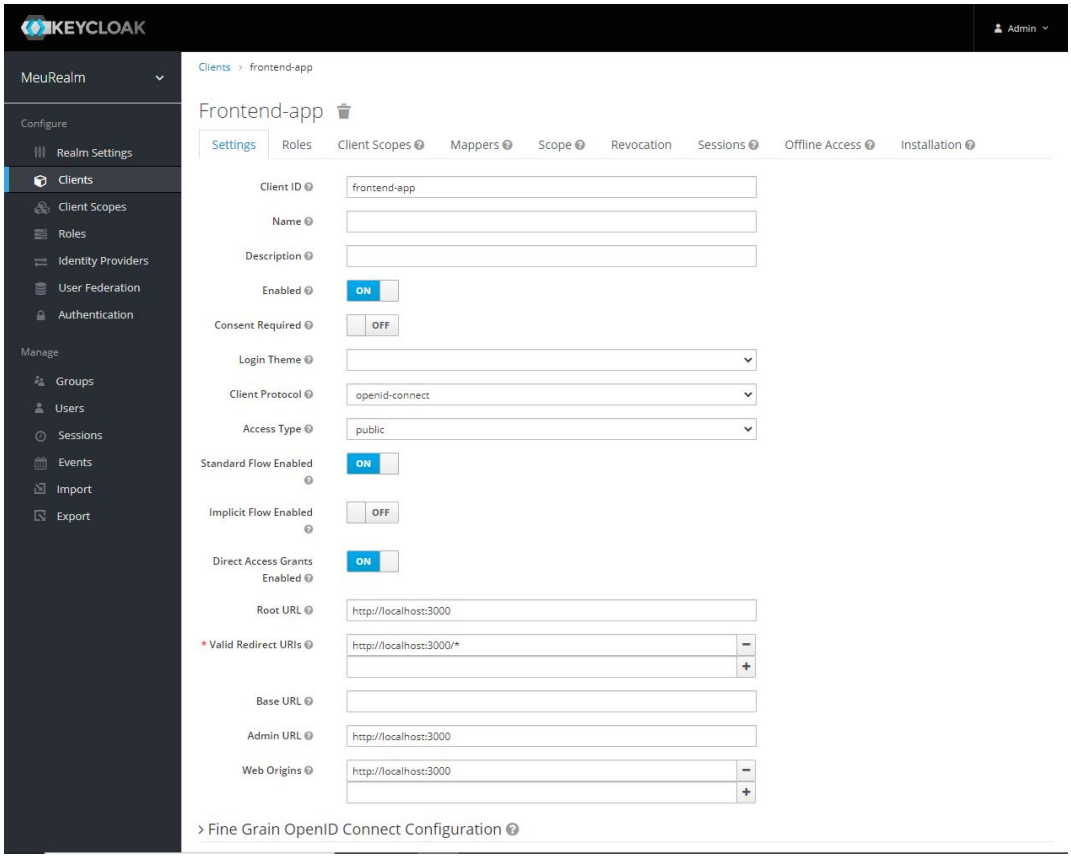
São os grupos de usuário. Você não precisa necessariamente criar grupos de usuário, mas eles podem ser úteis para separar os diferentes perfis que vão acessar suas aplicações. Por exemplo, você pode um grupo para os gerentes e outro grupo para os operacionais, atribuindo as roles necessárias para ambos. Assim, quando você criar um usuário, basta adicioná-lo ao grupo e você terá todos os acessos do grupo para o usuário.

- Users

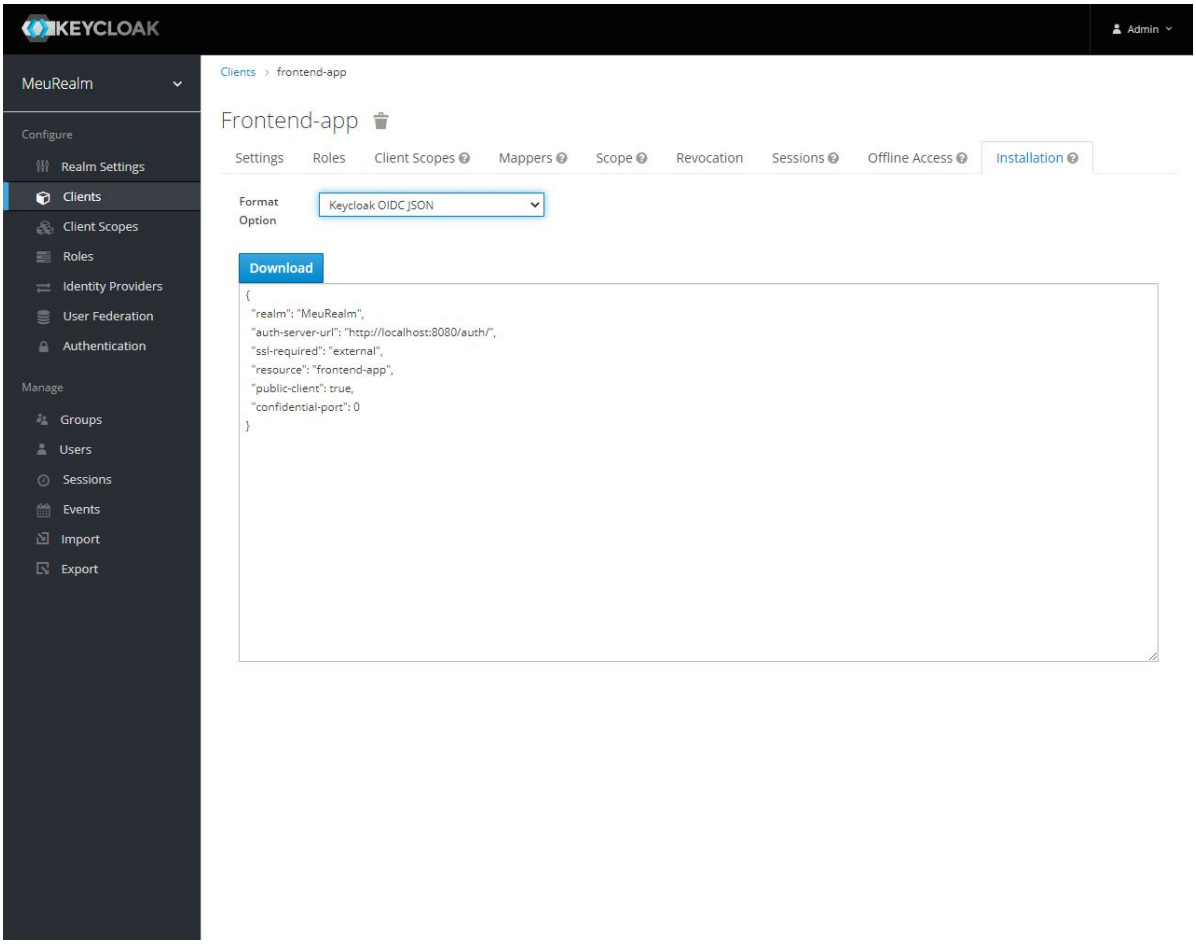
Aqui ficam todos os usuários criados no seu realm. Enquanto administrador, você pode cadastrar novos usuários, editá-los e até removê-los. Existe uma funcionalidade aqui que é excepcional, chamada “Impersonate”. Ao clicar nesse botão, você vai estar logado com aquele usuário. Isso vai te ajudar muito quando vir um chamado do suporte dizendo haver um problema com um usuário específico, e você precisa logar nele, mas não quer pedir a senha (mas cuidado, você estará logado na conta de outra pessoa, com grandes poderes vêm grandes responsabilidades).

Com essa introdução aos principais menus do painel de administração do Keycloak, estamos prontos para começar a proteger nossa aplicação.

A primeira coisa que devemos fazer é nos certificar que estamos no realm criado recentemente, e então criar um client para a aplicação que queremos proteger. Durante a criação, somente um campo Client ID é obrigatório, onde você escreverá algo que remete à sua aplicação (por exemplo, frontend-app). Se está rodando sua aplicação ainda em localhost, recomendo informar a URL para a raiz da sua aplicação no campo Root URL (no meu caso, http://localhost:3000), pois dessa forma os campos Valid Redirect URIs, e Web Origins serão preenchidos automaticamente. Esses dois campos servem, respectivamente, para definir para quais páginas o Keycloak está autorizado a fazer redirecionamentos (usamos o *wildcard* * para dar match em qualquer coisa), e definir quais endereços estão autorizados a obter recursos do Keycloak. No fim das contas, ambos são medidas de segurança para que somente páginas autorizadas consumam os serviços.



Agora que criamos o client, podemos acessar a aba “Installation” e fazer download de um arquivo que usaremos para informar à aplicação os dados de conexão. Normalmente, usamos o formato “Keycloak OIDC JSON” (para Javascript e NodeJS, esse é o formato recomendado).



A partir desse ponto, o que precisaremos fazer dependerá da linguagem do sistema que queremos proteger, então recomendo a leitura da documentação do Keycloak. Essa documentação pode ir mudando conforme o Keycloak é atualizado, mas hoje existem 9 Guides, que ensinam como fazer as coisas.

Para proteger sua aplicação, acesse o guide “Securing Applications and Services”. Logo no início existe a seção “Supported Platforms”, que mostra quais são as plataformas que já possuem adaptadores do Keycloak para serem usados. Clique em qual mais se encaixa na sua aplicação, e daí pra frente é só começar a seguir o que o guide ensina.

Uma dica de ouro, caso esteja protegendo uma aplicação Javascript (frontend), é que quando subimos o keycloak, ele mesmo já disponibiliza um adaptador Javascript, no path /auth/js/keycloak.js, então suponha que seu keycloak está rodando em localhost na porta 8080, o adaptador estará em http://localhost:8080/auth/js/keycloak.js. É interessante importar esse arquivo na sua aplicação, em vez de baixar o keycloak-js como dependência, pois dessa forma não teremos problemas com versão. Se alterarmos a versão do Keycloak, a versão do adaptador é alterada junto dessa forma.

Concluindo: os prós e contras do Keycloak

Pontos fortes:

- Gratuito;
- Conteinerizado;
- Mantido pela comunidade, portanto confiável;
- Flexível;
- Customizável;
- Atualizado constantemente;
- Compatível com diversas tecnologias;
- Documentação extensa;
- Poupa tempo de recriar a roda desenvolvendo autenticação e autorização.

Pontos fracos:

- Documentação toda em inglês;
- Customização de temas usam FTL (Freemarker Template) para customizações mais pesadas;
- Apesar do tempo poupado no desenvolvimento, você vai passar umas boas horas lendo documentação, especialmente quando precisar de customizações maiores.

Então é isso pessoal! Espero ter ajudado a entender o propósito e algumas das funcionalidades do Keycloak. Um abraço, e até a próxima!