

# Tutorial de Keycloak

Paulo Jerônimo

---

# Table of Contents

1. Resumo .....	1
2. Pré-requisitos .....	2
3. Baixando e iniciando o Keycloak para a execução de aplicações de demonstração .....	3
4. Testando alguns exemplos .....	4
4.1. Os exemplos "pré-configurados" .....	4
4.1.1. Importando o realm dos exemplos .....	4
4.1.2. Compilando e implantando os exemplos .....	4
4.1.3. Testando, manualmente, os exemplos customer-portal e product-portal .....	4
4.1.4. Uma aplicação escrita puramente em Javascript: customer-portal-js .....	6
4.1.5. Gerenciando a conta do usuário logado .....	7
4.2. O exemplo basic-auth .....	8
4.2.1. Compilando, implantando e testando a aplicação .....	9
5. A console de administração do Keycloak .....	10
6. Baixando e compilando os fontes do Keycloak .....	11
6.1. Utilizando uma versão específica .....	11

---

# Chapter 1. Resumo

Este tutorial apresenta um passo a passo prático para a instalação e testes do [Keycloak](http://keycloak.org)<sup>1</sup> 1.5.0.Final.

Os procedimentos descritos aqui podem ser realizados num ambiente Linux, OS X ou Windows mas, nesse último caso, você precisará instalar o [Cygwin](http://cygwin.com)<sup>2</sup> para não precisar fazer nenhuma adaptação nos comandos apresentados. A instalação do Cygwin para a execução deste tutorial deve ser realizada conforme os passos descritos na página [instalacao-cygwin.asciidoc](https://github.com/paulojeronimo/dicas-windows/blob/master/instalacao-cygwin.asciidoc)<sup>3</sup>

---

<sup>1</sup> <http://keycloak.org>

<sup>2</sup> <http://cygwin.com>

<sup>3</sup> <https://github.com/paulojeronimo/dicas-windows/blob/master/instalacao-cygwin.asciidoc>

---

## Chapter 2. Pré-requisitos

Para executar os passos deste tutorial você precisará ter instalado em tua máquina:

1. O JDK (este tutorial foi testado com a versão 1.8.0\_60);
2. O Maven (este tutorial foi testado com a versão 3.3.3);



A montagem de um ambiente Java EE pode ser realizada rapidamente no Linux, no OS X ou no Windows (+ Cygwin), através do projeto [javaee-ambiente](https://github.com/paulojeronimo/javaee-ambiente)<sup>1</sup>.

---

<sup>1</sup> <https://github.com/paulojeronimo/javaee-ambiente>

---

## Chapter 3. Baixando e iniciando o Keycloak para a execução de aplicações de demonstração

Uma forma fácil de testar e aprender os conceitos do Keycloak é baixando o [instalador de desenvolvimento](#)<sup>1</sup> (arquivo [keycloak-demo-1.5.0.Final.tar.gz](#)<sup>2</sup>). Esse arquivo contém parte dos arquivos necessários para a execução dos exemplos deste tutorial. Isso inclui um servidor Wildfly 9.0.1.Final com o servidor Keycloak e seus adaptadores já configurados.



Se você montou teu ambiente através do projeto [javaee-ambiente](#)<sup>3</sup> o download e a descompactação do instalador de desenvolvimento do keycloak podem ser realizados pelo comando `keycloak_demo install`. Dessa forma, ele ficará instalado no diretório configurado para a variável `FERRAMENTAS_DIR`.

Descompacte esse arquivo. *Você pode utilizar o comando abaixo:*

---

```
tar xvfz keycloak-demo-1.5.0.Final.tar.gz
```

---

Ajuste o valor da variável `JBOSS_HOME` da seguinte forma:

---

```
export JBOSS_HOME=$PWD/keycloak-demo-1.5.0.Final/keycloak
```

---

Inicie o Keycloak com o seguinte comando:

---

```
$JBOSS_HOME/bin/standalone.sh &
```

---

---

<sup>1</sup> <http://keycloak.github.io/docs/userguide/html/server-installation.html#d4e118>

<sup>2</sup> <http://downloads.jboss.org/keycloak/1.5.0.Final/keycloak-demo-1.5.0.Final.tar.gz>

<sup>3</sup> <https://github.com/paulojeronimo/javaee-ambiente>

---

# Chapter 4. Testando alguns exemplos

## 4.1. Os exemplos "pré-configurados"

Vá para o diretório `$JBOSS_HOME/./examples/preconfigured-demo` e leia o arquivo `README.md`. Os passos descritos a seguir são uma simplificação dos passos apresentados nele.

---

```
cd $JBOSS_HOME/./examples/preconfigured-demo
view README.md
```

---

### 4.1.1. Importando o realm dos exemplos

Abra a URL <http://localhost:8080/auth/admin/master/console/#/create/realm>.

Logue-se com o usuário `admin` e a senha `admin`. Será solicitada a troca da senha.

Clique em `Select file` e selecione o arquivo `testrealm.json` que está dentro do diretório em que você está. Em seguida, clique no botão `Upload`.

### 4.1.2. Compilando e implantando os exemplos

Abra um shell que contenha o Maven no `PATH`. Compile e implante os exemplos com os comandos a seguir. Observe o log do Wildfly enquanto o último comando é executado para averiguar a implantação dos exemplos.

---

```
mvn clean install
mvn wildfly:deploy
```

---

### 4.1.3. Testando, manualmente, os exemplos customer-portal e product-portal

Acesse a URL <http://localhost:8080/customer-portal/>. Clique em `Customer Listing` e você será redirecionado para a página de autenticação no Keycloak. Informe o usuário (`bburke@redhat.com`) e a senha (`password`).

Observe o log do Wildfly.

Observe a tela que será apresentada. Note o valor de `Servlet User Principal`.

Clique em `products`. Observe que não foi solicitada nova autenticação apesar do contexto ter sido alterado para `product-portal` (esta app também utiliza os mecanismos de segurança do Java EE).

Clique em `Product Listing`. Note que o valor do `User` é o mesmo que o apresentado na tela `customers` para o campo `Servlet User Principal`.

Retorne a tela anterior e clique em `Admin Interface` (URL: <http://localhost:8080/product-portal/admin/admin.jsp>). Note que será exibida a mensagem `Forbidden`. Isso ocorre pelo fato do usuário logado (`bburke@redhat.com`) não ter o perfil `admin` (exigido para esse acesso). Podemos saber disso observando dois arquivos: o primeiro deles é o `web.xml` que, conforme a saída do comando abaixo, exige que o usuário logado tenha o perfil `admin` para acessar qualquer URL abaixo de `/admin/`:

```
$ sed -n 9,17p product-app/src/main/webapp/WEB-INF/web.xml
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Admins</web-resource-name>
    <url-pattern>/admin/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>admin</role-name>
  </auth-constraint>
</security-constraint>
```

O segundo arquivo é o `testrealm.json`. As configurações para o usuário `bburke@redhat.com` são apresentadas na saída do comando a seguir:

```
$ sed -n 15,29p testrealm.json
{
  "username" : "bburke@redhat.com",
  "enabled": true,
  "email" : "bburke@redhat.com",
  "firstName": "Bill",
  "lastName": "Burke",
  "credentials" : [
    { "type" : "password",
      "value" : "password" }
  ],
  "realmRoles": [ "user" ],
  "clientRoles": {
    "account": [ "manage-account" ]
  }
}
```

```
},
```

---

Notamos que, nessa saída, o usuário em questão não possui a role `admin` (as roles que ele possui estão definidas para o valor de `realmRoles`).

Retorne a URL <http://localhost:8080/customer-portal/customers/view.jsp>.

Clique em `manage.acct`. Navegue pelos links.

Em `Account` edite os campos `Email`, `Fist name` e `Last name` informando teus próprios dados e clique em `Save`.

Em `Password`, altere a senha e clique em `Save`.

Em `Sessions`, clique em `Log out all sessions`.

Efetue o `Log in` informando teu `email` e senha. Após o logon, note que o `Username` permanece `bburke@redhat.com`.

Observe o log do Wildfly.

#### 4.1.4. Uma aplicação escrita puramente em Javascript: `customer-portal-js`

Em `Applications` clique em `customer-portal-js`. Em seguida, em `Customer Listing`. Note a alteração do valor dos campos `Email` e `Full Name`, `Fist` e `Last`, conforme os dados que você editou no passo anterior.

A aplicação <http://localhost:8080/customer-portal-js> é uma aplicação totalmente escrita em Javascript. Para ver sua estrutura, sem os arquivos gerados na construção da aplicação, execute:

---

```
$ (cd customer-app-js; mvn clean)
$ tree customer-app-js/
customer-app-js/
|-- pom.xml
`-- src
    |-- main
        |-- webapp
            |-- customers
            |   |-- view.html
            |   |-- index.html
            |-- keycloak.json
```



4 directories, 4 files

---

Clique em `logout`.

### 4.1.5. Gerenciando a conta do usuário logado

Efetue o `Log in` informando `admin` e `password`.

Clique em `products` e, em seguida, em `Admin Interface`. Note que, agora, o acesso a tela de administração não exibe a mensagem `Forbidden` liberando a visualização como deveria ser. Isso ocorre pelo fato do usuário logado (`admin`) ter o perfil `admin` (exigido para esse acesso). Detalhe: essa configuração é visível no `testrealm.json`. *Observe a saída do comando abaixo. Ela imprime as linhas relativas a configuração do usuário `admin` no arquivo `testrealm.json`:*

---

```
$ sed -n 60,74p testrealm.json
{
  "username" : "admin",
  "enabled": true,
  "email" : "admin@admin.com",
  "firstName": "Admin",
  "lastName": "Burke",
  "credentials" : [
    { "type" : "password",
      "value" : "password" }
  ],
  "realmRoles": [ "user", "admin" ],
  "clientRoles": {
    "realm-management": [ "realm-admin" ]
  }
},
```

---

Vá para a URL <http://localhost:8080/customer-portal/>. Clique em `Customer Admin Interface`. Note que, agora, também é obtido o acesso a esse link.

Volte para a tela anterior. Clique em `Customer Listing` e, em seguida, em `manage acct` (URL: <http://localhost:8080/auth/realms/demo/account?referrer=customer-portal>). Note que o usuário logado (`admin`) não tem permissão de acesso. A explicação para disso está logo abaixo.

Volte para a tela anterior e clique em `logout`. Note que você estará na página `Customer Portal`. Clique em `Customer Listing`.

Efetue o `Log in` como `bburque@redhat.com` (utilize a nova senha que você criou).

Clique em `manage acct`. *Note que é possível que esse usuário gerencie sua conta, o que não ocorre para o usuário `admin`. A explicação está nas roles definidas para esse usuário: ele possui a role `manage-account` definida para o acesso ao cliente `account` (associado a URL base `/auth/realms/demo/account`). O mesmo não ocorre para o usuário `admin`.*

Em `Applications`, clique em `angular-product`.

Clique em `Reload` para exibir a lista de produtos.

Clique em `Sign Out` para voltar a tela de autenticação.

Observe que qualquer tentativa de acesso a URLs protegidas pelo Keycloak (como, por exemplo, <http://localhost:8080/angular-product/>) será redirecionada a tela de autenticação provida pelo Keycloak.

Acesse a tela de administração de usuários do `realm` (<http://localhost:8080/auth/admin/master/console/#/realms/demo/users>). Clique em `View all users` e, em seguida, no usuário `admin`. Vá para a aba `Role Mappings`. Em `Client Roles` selecione `account`. Em `Available Roles` selecione `manager-account` e clique no botão `Add selected`. Fazendo isso o usuário `admin` poderá gerenciar sua conta acessando [manage acct](http://localhost:8080/auth/realms/demo/account?referrer=customer-portal)<sup>1</sup>. Verifique!

## 4.2. O exemplo basic-auth

A estrutura do exemplo `basic-auth` pode ser observada pela seguinte saída:

```
$ tree
.
|-- basicauthrealm.json
|-- pom.xml
|-- README.md
`-- src
    |-- main
        |-- java
            |-- org
                |-- keycloak
                    |-- example
```

---

<sup>1</sup> <http://localhost:8080/auth/realms/demo/account?referrer=customer-portal>

```
|           |-- basicauth
|           |-- BasicAuthService.java
|           |-- BasicAuthServiceApplication.java
|-- webapp
    |-- WEB-INF
        |-- keycloak.json
        |-- web.xml
```

9 directories, 7 files

---

### 4.2.1. Compilando, implantando e testando a aplicação

Acesse a interface administrativa do Keycloak e importe o arquivo `basicauthrealm.json`.

Compile e implante a aplicação:

---

```
mvn clean package wildfly:jboss
```

---

Teste a aplicação:

---

```
curl http://admin:password@localhost:8080/basicauth/service/echo?
value=hello
```

---

Observe, na interface administrativa do Keycloak, a existência de uma sessão.

---

## Chapter 5. A console de administração do Keycloak

A URL <http://localhost:8080/auth/admin/index.html> possibilita o acesso a interface de administração do Keycloak.



Você se lembra que trocou a senha para o usuário admin no primeiro acesso a essa interface?

---

# Chapter 6. Baixando e compilando os fontes do Keycloak

## 6.1. Utilizando uma versão específica

Para gerar a versão 1.5.0.Final:

```
git clone https://github.com/keycloak/keycloak
cd keycloak
git tag
git checkout 1.5.0.Final
mvn install
```

Mais detalhes sobre como contribuir na solução de um bug, gerar um release, etc, podem ser obtidos na página [HackingOnKeycloak.md](https://github.com/keycloak/keycloak/blob/master/misc/HackingOnKeycloak.md)<sup>1</sup>.

---

<sup>1</sup> <https://github.com/keycloak/keycloak/blob/master/misc/HackingOnKeycloak.md>