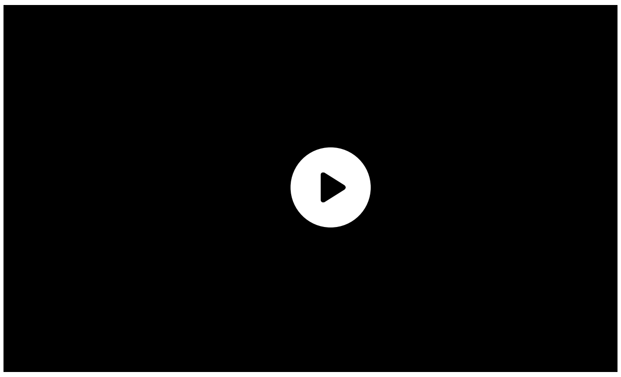
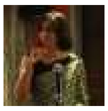
(/)

Customizing the Login Page for Keycloak

FEATURED VIDEOS



Last updated: May 11, 2024



Written by: Sampada Wagde (https://www.baeldung.com/author/sampadawagde)

Reviewed by: Michal Aibin (https://www.baeldung.com/editor/michal-author)

Spring (https://www.baeldung.com/category/spring) +

Keycloak (https://www.baeldung.com/tag/keycloak)

OAuth (https://www.baeldung.com/tag/oauth)

1. Overview

Keycloak (https://www.keycloak.org/) is a third-party authorization server used to manage our web or mobile applications' authentication and authorization requirements. It uses a default login page to sign-in users on our app's behalf.

In this tutorial, we'll focus on **how we can customize the login page for our Keycloak server** so that we can have a different look and feel for it. We'll see this for both standalone as well as embedded servers.

We'll build on top of customizing themes for the Keycloak tutorial (/spring-keycloak-custom-themes) to do that.

2. Customizing a Standalone Keycloak Server

Continuing with our example of the *custom* (/spring-keycloak-custom-themes#default-themes) theme, let's see the standalone server first.

2.1. Admin Console Settings

To start the server, let's navigate to the directory where our Keycloak distribution is kept, and run this command from its *bin* folder:

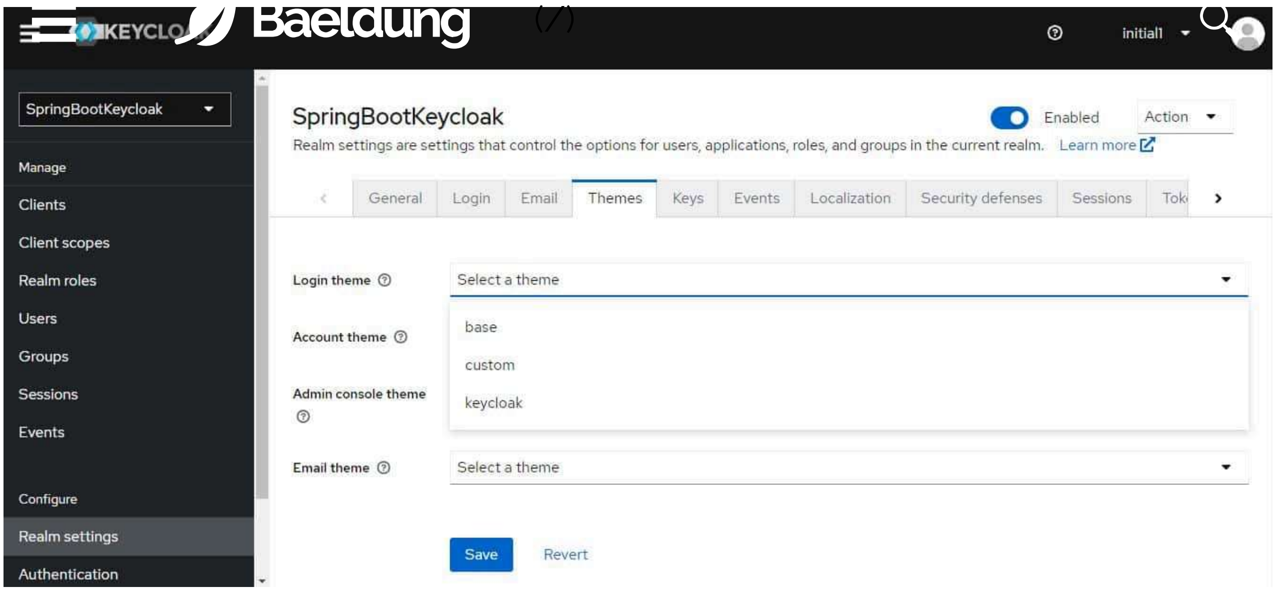
```
kc.[sh|bat] start-dev --spi-theme-static-max-age=-1 --spi-theme-cache-
themes=false --spi-theme-cache-templates=false
```

Once the server is started, we only need to refresh the page to see our changes reflected, thanks to the above command.

Now let's create a new folder, named *login*, inside the *themes/custom* directory. To keep things simple, we'll first copy all the contents of the *themes/keycloak/login* directory here. This is the default login page theme.

Then, we'll go to the admin console (http://localhost:8080/admin/master/console), key-in the *initial1/ zaq1!QAZ* credentials and go to the *Themes* tab for our realm:

Baeldung (/)



SpringBootKeycloak

Realm settings are settings that control the options for users, applications, roles, and groups in the current realm. Learn more

Login theme: Select a theme (base, custom, keycloak)

Account theme

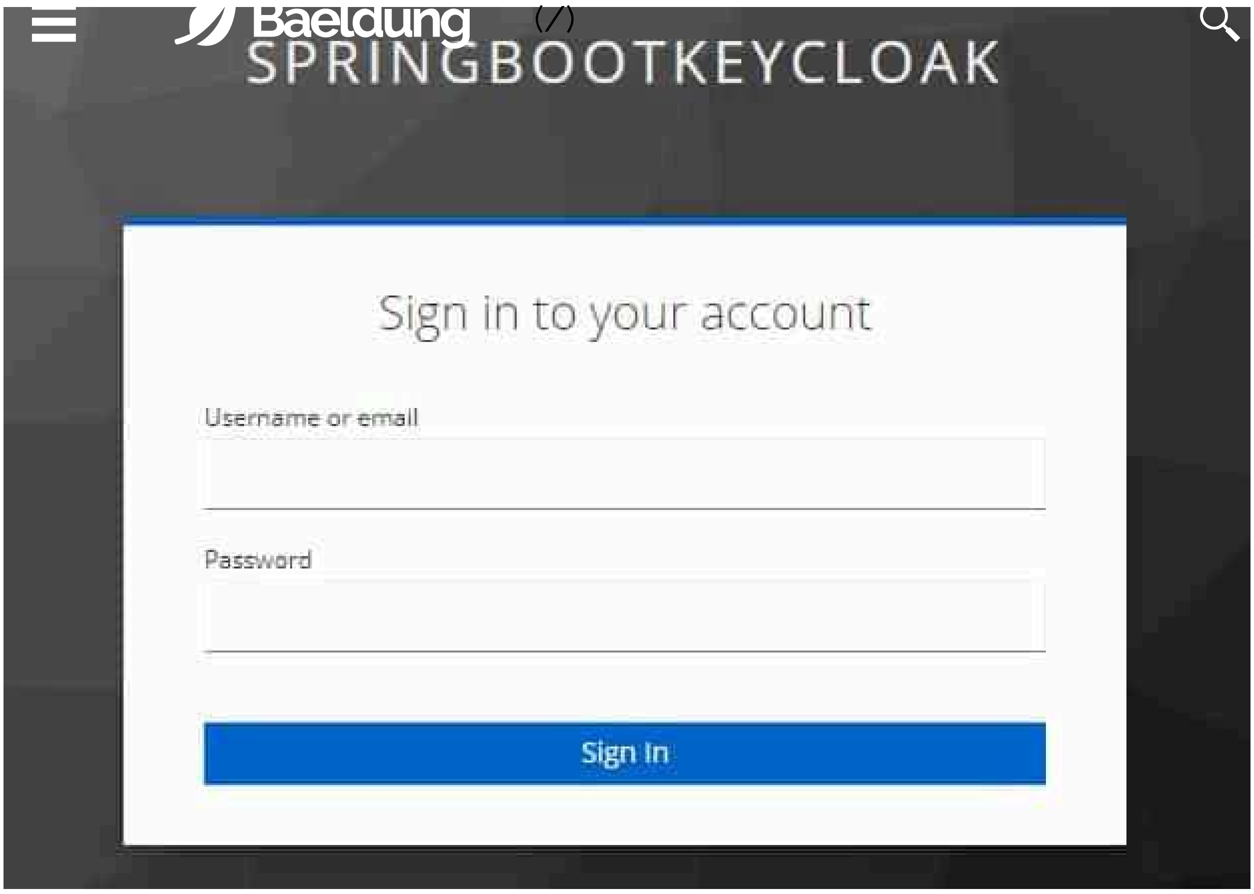
Admin console theme

Email theme: Select a theme

(/wp-content/uploads/2020/08/keycloak-themes-1.jpg)
We'll select *custom* for the *Login Theme* and save our changes.

With that set, we can now try some customizations. But before that, let's have a look at the default login page (http://localhost:8080/realms/SpringBootKeycloak/protocol/openid-connect/auth?response_type=code&client_id=login-app&scope=openid&redirect_uri=http://localhost:8081/):

(https://ads.freestar.com/?utm_campaign=branding&utm_medium=banner&utm_source=
ntent=baeldung_leaderboard_mid_2)



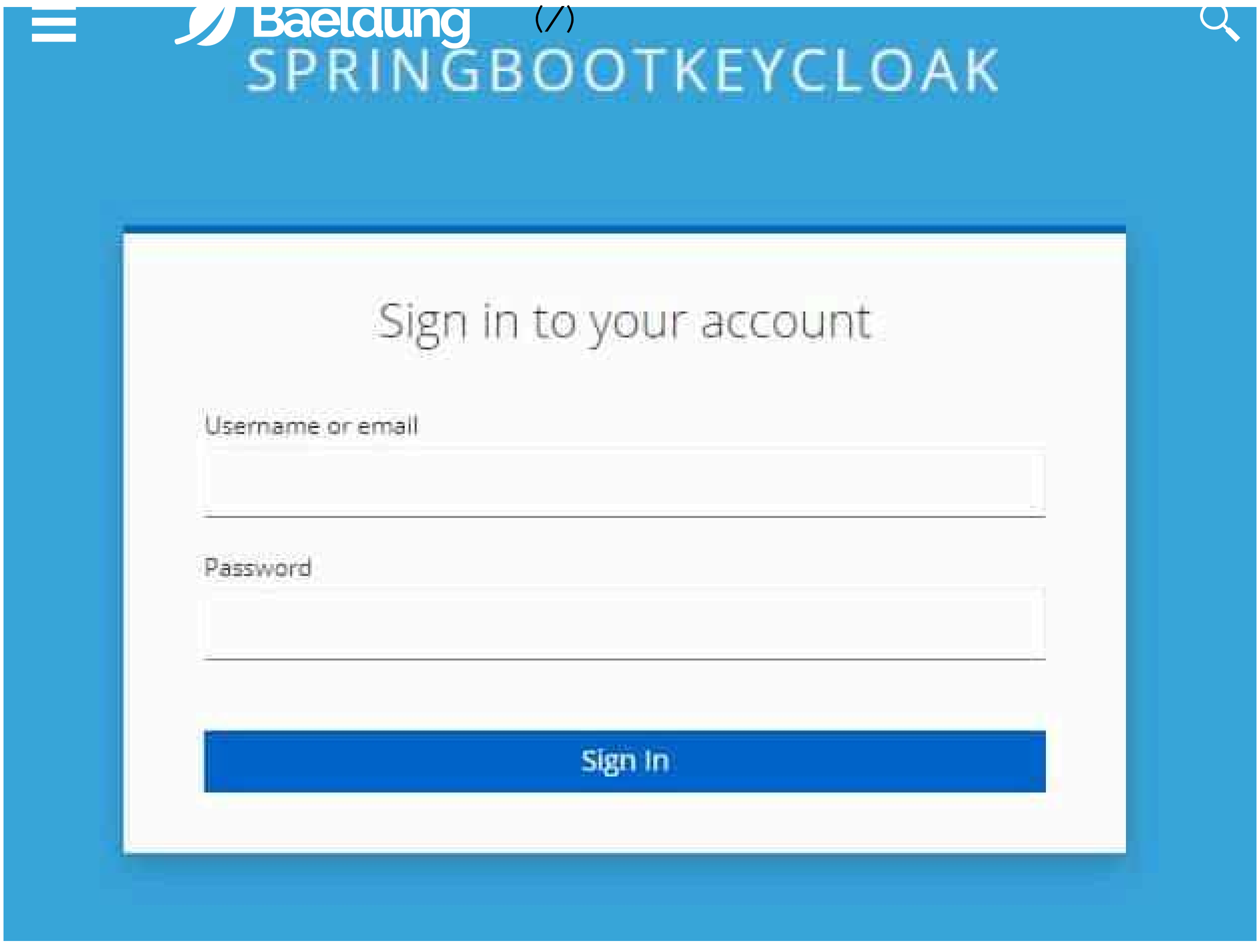
(/wp-content/uploads/2020/08/keycloak-default_login_page.jpg)

2.2. Adding Customizations

Now let's say we need to change the background. For that, we'll open *login/resources/css/login.css* and change the class definition:

```
.login-pf body {
    background: #39a5dc;
    background-size: cover;
    height: 100%;
}
```

To see the effect, let's refresh the page:



(/wp-content/uploads/2020/08/keycloak-login_page_bg_change.jpg)
Next, let's try to change the labels for the username and password.

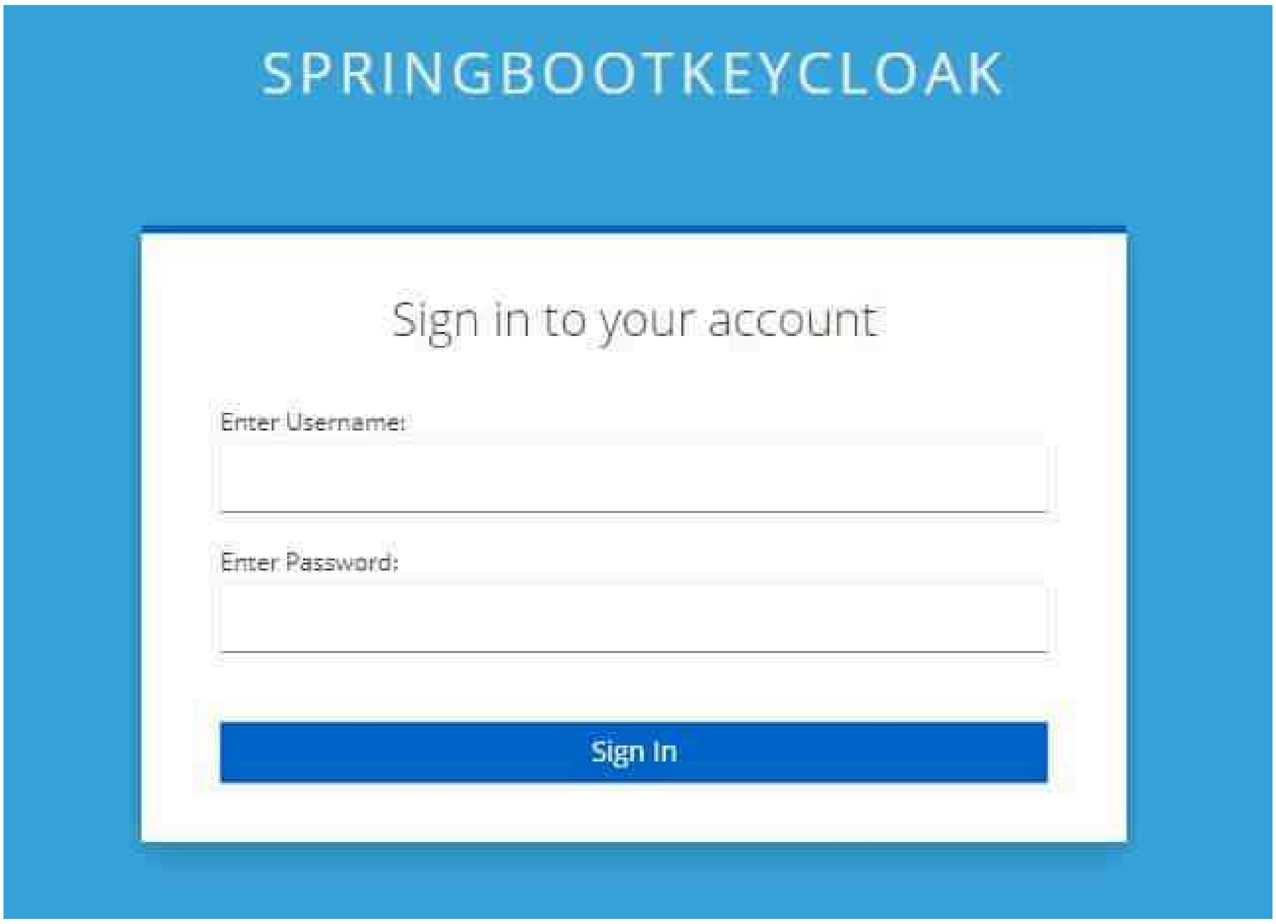
To achieve that, we need to create a new file, *messages_en.properties* in the *theme/login/messages* folder. This file overrides the default message bundle being used for the given properties:

```
usernameOrEmail=Enter Username:
password=Enter Password:
```

To test, again refresh the page:

(/)

(https://ads.freestar.com/?
utm_campaign=branding&utm_medium=lazyLoad&utm_source=baeldung.co
rd_mid_3)



(/wp-content/uploads/2020/08/login_page_label_change.jpg)
Suppose we want to change the entire HTML or a part of it, we'll need to override the freemarker templates that Keycloak uses by default. The default templates for the login page are kept in the *base/login* directory.

Let's say we want *WELCOME TO BAELDUNG* to be displayed in place of *SPRINGBOOTKEYCLOAK*.

For that, we'll need to copy *base/login/template.ftl* to our *custom/login* folder.

In the copied file, change the line:

```
<div id="kc-header-wrapper"   (/)
class="${properties.kcHeaderWrapperClass!}">
    ${kcSanitize(msg("loginTitleHtml",(realm.displayNameHtml!'')))?
no_esc}
</div>
```

To:

```
<div id="kc-header-wrapper"
class="${properties.kcHeaderWrapperClass!}">
    WELCOME TO BAELDUNG
</div>
```

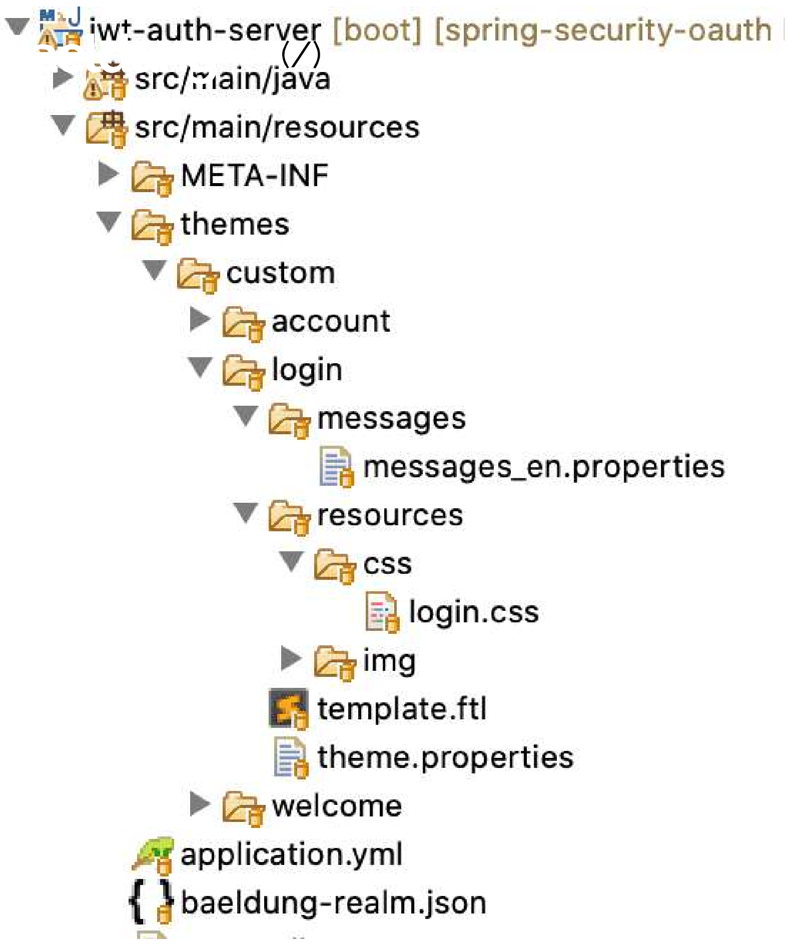
The login page would now display our custom message instead of the realm name.

3. Customizing an Embedded Keycloak Server

The first step here is to add all the artifacts we changed for the standalone server to the source code of our embedded authorization server.

So, let's create a new folder *login* inside *src/main/resources/themes/custom* with these contents:

(/)



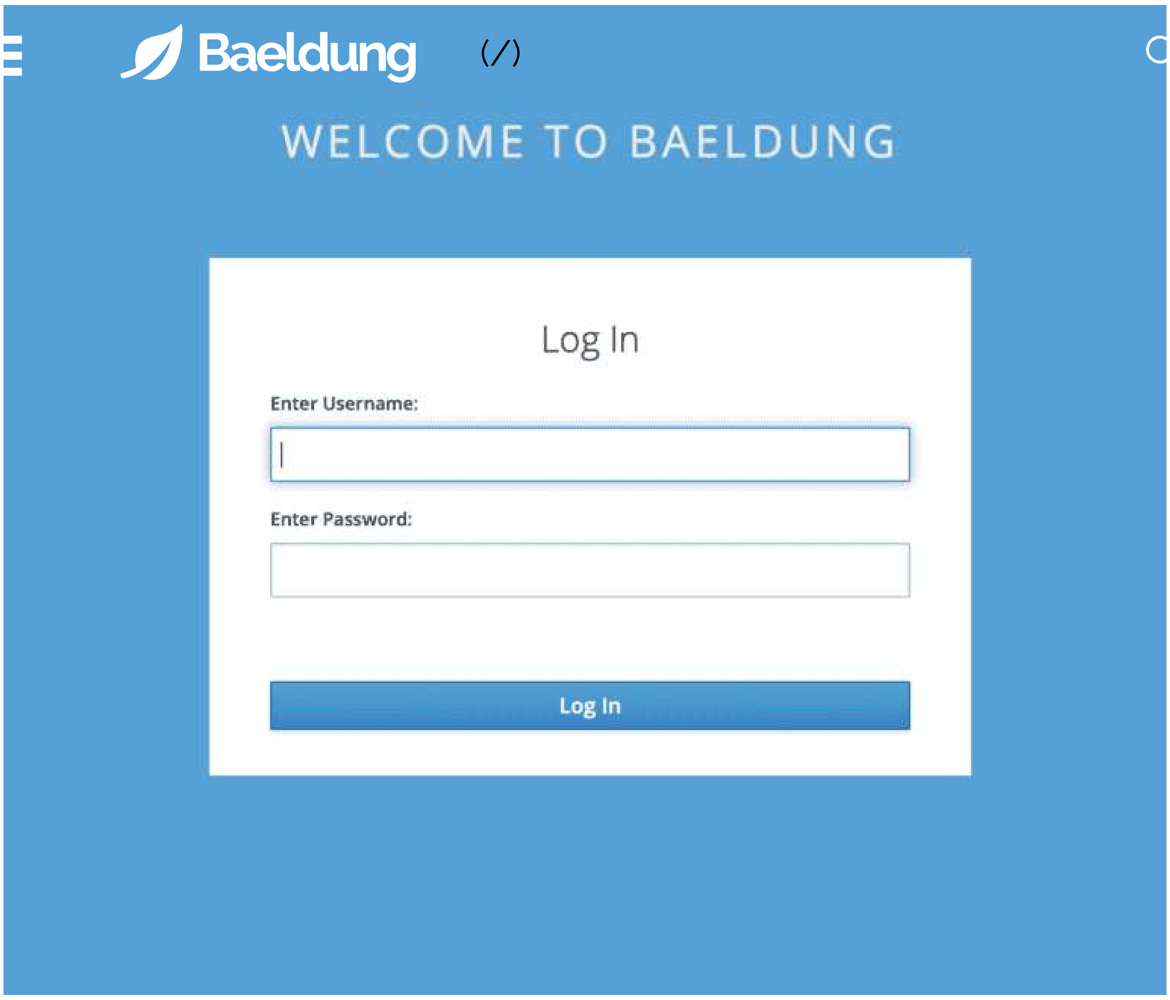
```
▼ iwt-auth-server [boot] [spring-security-oauth
  ▶ src/main/java
  ▼ src/main/resources
    ▶ META-INF
    ▼ themes
      ▼ custom
        ▶ account
        ▼ login
          ▼ messages
              messages_en.properties
          ▼ resources
            ▼ css
                login.css
            ▶ img
            template.ftl
            theme.properties
        ▶ welcome
    application.yml
    baeldung-realm.json
```

(/wp-content/uploads/2020/08/folder_Structure.png)
Now all we need to do is to add instruction in our realm definition file, *baeldung-realm.json* so that *custom* is used for our login theme type:

```
"loginTheme": "custom",
```

We've already redirected to the *custom* theme directory (/spring-keycloak-custom-themes#redirection) so that our server knows from where to pick up the theme files for the login page.

For testing, let's hit the login page (http://localhost:8083/auth/realms/baeldung/protocol/openid-connect/auth?response_type=code&&scope=openid%20write%20read&client_id=newClient&redirect_uri=http://localhost:8089/):



(/wp-content/uploads/2020/08/loginpage_embedded.png)
As we can see, all the customizations done earlier for the standalone server, such as the background, label names, and page title, are appearing here.

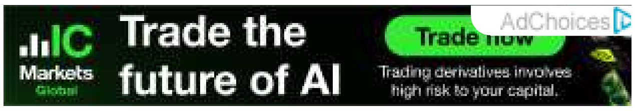
4. Bypassing Keycloak Login Page

Technically, we can completely bypass the Keycloak login page by using the password or direct access grant (https://oauth.net/2/grant-types/password/) flow. However, **it's strongly recommended that this grant type shouldn't be used at all.**

In this case, there is no intermediary step of getting an authorization code, and then receiving the access token in exchange. Instead, we can directly send the user credentials via a REST API call and get the access token in response.

This effectively means that we can use our login page to collect the user's id and password, and along with the client id and secret, send it to Keycloak in a POST to its *token* endpoint.

(/)



(https://ads.freestar.com/?utm_campaign=branding&utm_medium=lazyLoad&utm_source=baeldung.com&utm_content=baeldung_inconte nt_2)

But again, since Keycloak provides a rich feature set of login options – such as remember me, password reset, and MFA – to name a few, there is little reason to bypass it.

5. Conclusion

In this tutorial, **we learned how to change the default login page for Keycloak and add our customizations**.

We saw this for both a standalone and an embedded instance.

Lastly, we briefly went over how to bypass Keycloak's login page entirely and why not to do that.

As always, the source code is available over on GitHub. For the standalone server, it's on the tutorials GitHub (https://github.com/eugenp/tutorials/tree/master/spring-boot-modules/spring-boot-keycloak), and for the embedded instance, on the OAuth GitHub (https://github.com/Baeldung/spring-security-oauth/tree/master/oauth-jwt).

(/)

Learning to build your API
with Spring?

Download the E-book (/rest-api-spring-guide)

(https://ads.freestar.com/?
tm_campaign=branding&utm_medium=lazyLoad&utm_source=baeldung.con
d_btf_2)

COURSES

ALL COURSES (/COURSES/ALL-COURSES)
ALL BULK COURSES (/COURSES/ALL-BULK-COURSES)

ALL BULK TEAM COURSES (/COURSES/ALL-BULK-TEAM-COURSES)
THE COURSES PLATFORM (HTTPS://COURSES.BAELDUNG.COM)

(/)

SERIES

JAVA “BACK TO BASICS” TUTORIAL (/JAVA-TUTORIAL)
JACKSON JSON TUTORIAL (/JACKSON)
APACHE HTTPCLIENT TUTORIAL (/HTTPCLIENT-GUIDE)
REST WITH SPRING TUTORIAL (/REST-WITH-SPRING-SERIES)
SPRING PERSISTENCE TUTORIAL (/PERSISTENCE-WITH-SPRING-SERIES)
SECURITY WITH SPRING (/SECURITY-SPRING)
SPRING REACTIVE TUTORIALS (/SPRING-REACTIVE-GUIDE)
JAVA ARRAY (HTTPS://WWW.BAELDUNG.COM/CATEGORY/JAVA/JAVA-ARRAY)

ABOUT

ABOUT BAELDUNG (/ABOUT)
THE FULL ARCHIVE (/FULL_ARCHIVE)
EDITORS (/EDITORS)
OUR PARTNERS (/PARTNERS/)
PARTNER WITH BAELDUNG (/PARTNERS/WORK-WITH-US)
EBOOKS (/LIBRARY/)

TERMS OF SERVICE (/TERMS-OF-SERVICE)
PRIVACY POLICY (/PRIVACY-POLICY)
COMPANY INFO (/BAELDUNG-COMPANY-INFO)
CONTACT (/CONTACT)