

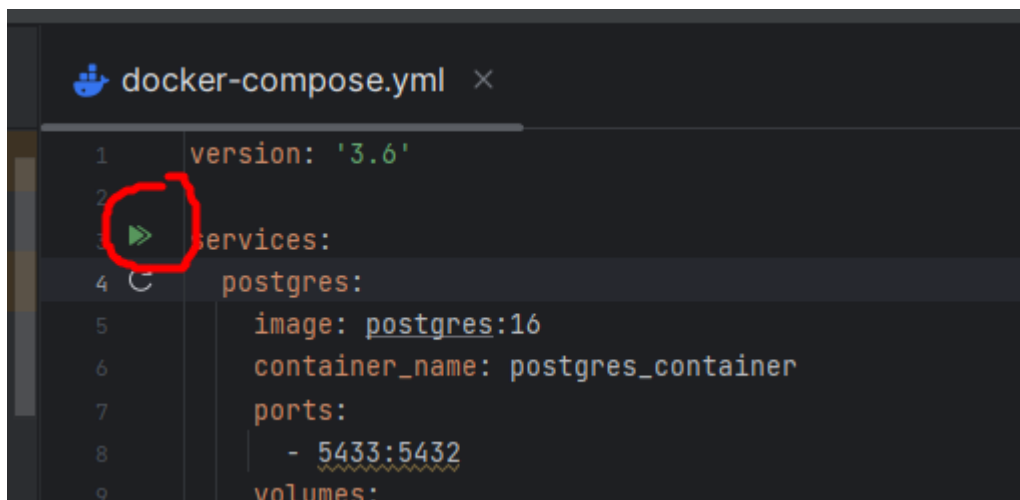
## Library App: initial setup instruction

- 1) Clone frontend and backend parts:

git clone <https://github.com/Svts3/LibraryApp.git>

git clone <https://github.com/TopWorkOut/LibraryProjectFrontEndPart.git>

- 2) Start Docker Desktop and run docker-compose.yml from backend project root



Then 3 containers were created

Containers [Give feedback](#)

Container CPU usage ⓘ  
1.32% / 1000% (10 cores allocated)

Container memory usage ⓘ  
1.24GB / 7.27GB










Show charts ▾

🔍

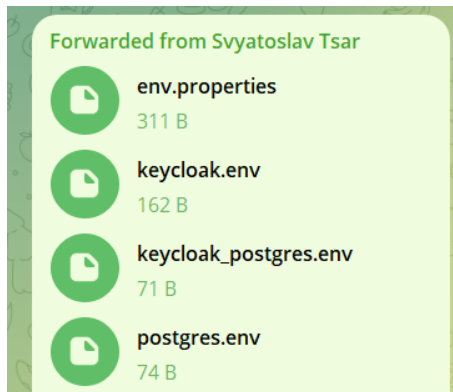
Search

☰

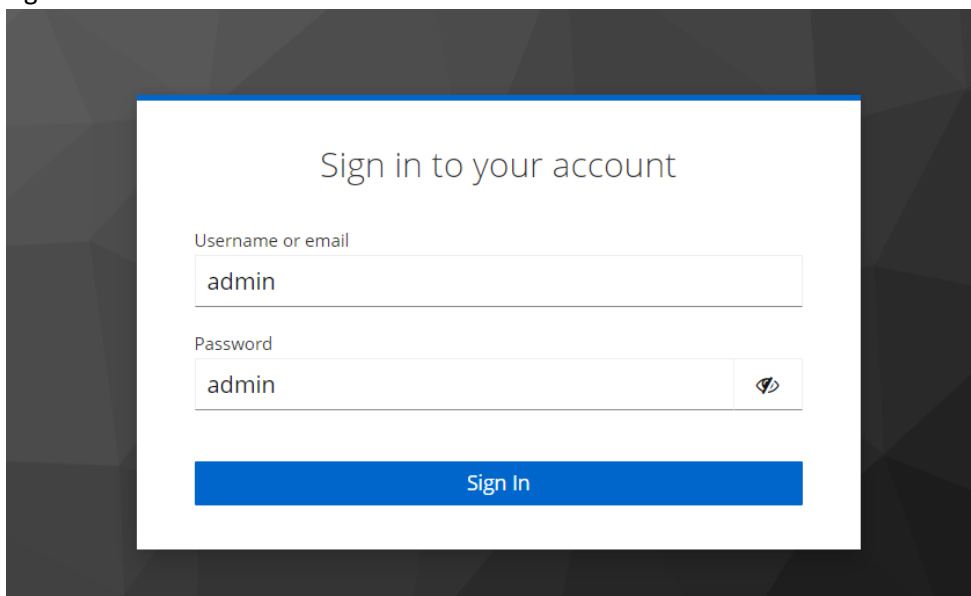
Only show running containers

<div><input type="checkbox"/></div>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<div><div><input type="checkbox"/></div><div>▼</div></div>	<div><div></div><div>libraryapp</div></div>		Running (3/3)	1.32%		3 minutes ago	<div><div><div></div></div><div><div></div></div><div><div></div></div></div>
<div><input type="checkbox"/></div>	<div><div></div><div><div>postgres_container</div><div>c0420411f955 </div></div></div>	<div><a href="#">postgres:16</a></div>	Running	0.59%	<div><a href="#">5433:5432</a> </div>	3 minutes ago	<div><div><div></div></div><div><div></div></div><div><div></div></div></div>
<div><input type="checkbox"/></div>	<div><div></div><div><div>keycloak-db-1</div><div>1e1472d5437d </div></div></div>	<div><a href="#">postgres:16</a></div>	Running	0%		3 minutes ago	<div><div><div></div></div><div><div></div></div><div><div></div></div></div>
<div><input type="checkbox"/></div>	<div><div></div><div><div>keycloak_container</div><div>9800167a4ec8 </div></div></div>	<div><a href="#">keycloak/keycloak</a></div>	Running	0.73%	<div><a href="#">8090:8080</a> </div>	3 minutes ago	<div><div><div></div></div><div><div></div></div><div><div></div></div></div>

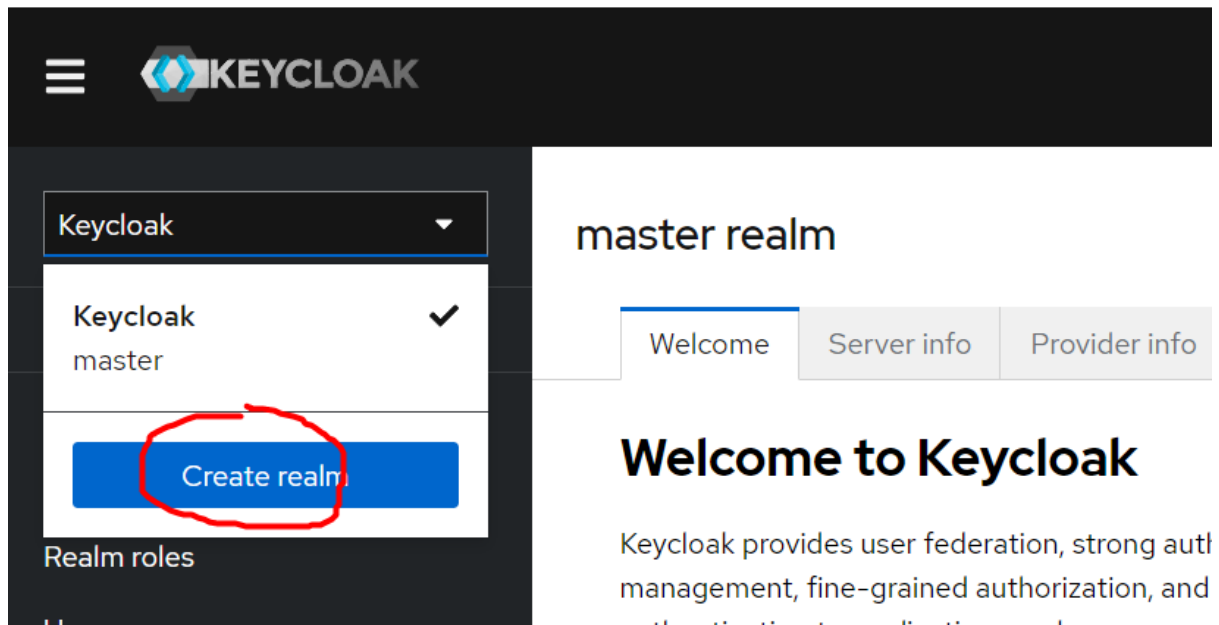
- 3) Move these files to backend root (there are files with security configs and they are not store on any public sources including GitHub. To get them, please contact any team member)



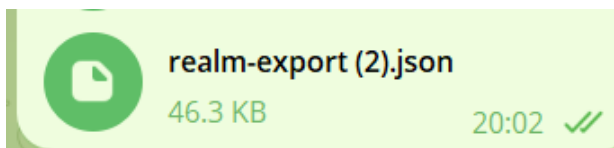
- 4) Start backend  
Download frontend dependencies using the command “npm install”  
Start frontend using the command “npm start”
- 5) Navigate to <http://localhost:8090> (authorization server)  
sign in as “admin”: “admin”



Create new realm



Click "browse" and import real-export (2).json and press "Create" (That is also file not for sharing it publicly, so, please contact any team member to get it)



## Create realm

A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and can only manage the users that they control.

**Resource file**

Drag a file here or browse to upload Browse... Clear

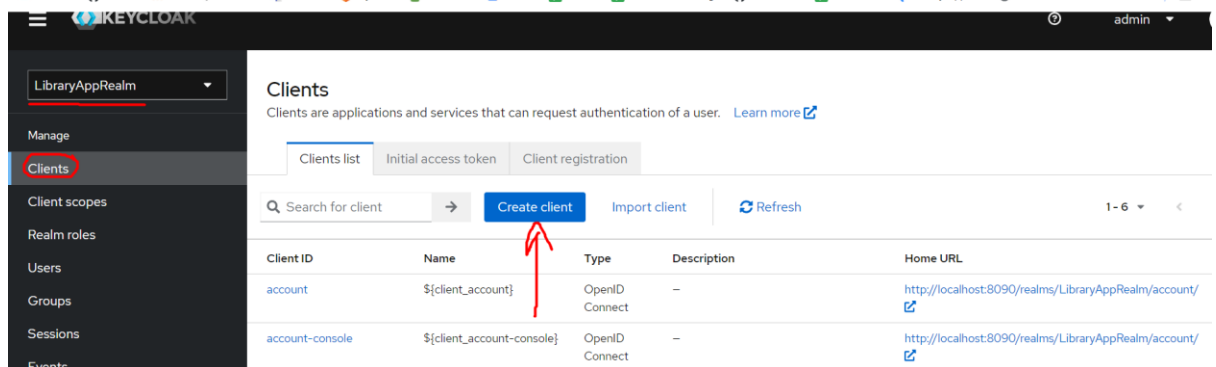
1

Upload a JSON file

**Realm name \***

**Enabled** ☒ On

## Navigate to this one



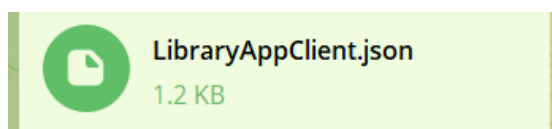
**Clients**  
Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list Initial access token Client registration

→ Create client Import client Refresh 1-6

Client ID	Name	Type	Description	Home URL
account	\$_client_account	OpenID Connect	—	<a href="http://localhost:8090/realms/LibraryAppRealm/account/">http://localhost:8090/realms/LibraryAppRealm/account/</a>
account-console	\$_client_account-console	OpenID Connect	—	<a href="http://localhost:8090/realms/LibraryAppRealm/account/">http://localhost:8090/realms/LibraryAppRealm/account/</a>

Press “Import client” and import LibraryAppClient.json and press “Save” (This file can also be received from any team member)



**Clients**  
Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list | Initial access token | Client registration

Search for client → [Create client](#) [Import client](#) [Refresh](#)

Client ID	Name	Type	Description	Ho
<a href="#">account</a>	\${client_account}	OpenID Connect	–	<a href="#">http</a>
<a href="#">account-console</a>	\${client_account-console}	OpenID Connect	–	<a href="#">http</a>
<a href="#">admin-cli</a>	\${client_admin-cli}	OpenID	–	–

Enter “\*” in “Web origins” below and save the configuration

**LibraryAppClient** (OpenID Connect)  
Clients are applications and services that can request authentication of a user.

Settings | Keys | Credentials | Roles | Client scopes | Sessions | Advanced

**General settings**

Client ID \* [?](#) LibraryAppClient

Name [?](#)

Description [?](#)

Always display in UI [?](#) ☐ Off

**Access settings**

Root URL [?](#)

Home URL [?](#)

Valid redirect URIs [?](#) \*

[Add valid redirect URIs](#)

Valid post logout redirect URIs [?](#)

[Add valid post logout redirect URIs](#)

Web origins [?](#) \*

[Add web origins](#)

### **START USING THE APP**

LOGIN – navigate to <http://localhost:8080> (it redirects to keycloak server localhost:8090), where you can login (or register a new) user.

LOGOUT – navigate to “/logout” route on frontend (<http://localhost:3000/logout>) (it redirects to keycloak server localhost:8090 default logout route), and submit that you want to logout.

**P.S.:** after logout we are not automatically redirected anywhere, because this feature is difficult to implement as for now (work in progress).