

ALCUNE PRECISAZIONI SUL PRIMO ESEMPIO: frontend-backend nella sottocartella ch2 del repository

<https://github.com/PacktPublishing/Keycloak-Identity-and-Access-Management-for-Modern-Applications>

Per questo esempio è necessario utilizzare il nome **myrealm** per il realm: infatti nella cartella ch2/backend si trova un file di configurazione keycloak.json dove è indicato "realm": "myrealm".

Inoltre per poter accedere da frontend all'esempio di servizio protetto del backend è necessario aggiungere al realm "myrealm" un ruolo "myrole" ed aggiungerlo all'utente registrato che si autenticcherà. Se create due user, uno con e uno senza questo ruolo nel primo caso dovrete ricevere una risposta positiva dal backend (cioè quando si preme sul bottone Invoke Service si ottiene la risposta "200 Secret message!"), nel secondo caso invece non sarà concessa l'autorizzazione ad accedere alla pagina protetta (cioè quando si preme sul bottone Invoke Service si ottiene la risposta "403 Access denied"). Per verificare la presenza del ruolo myrole, premendo su "Show Access Token" si può vedere il token e verificare che nella sezione "realm_access" – "roles" sia presente "myrole".

In questo stesso file è indicata la URL di keycloak: <http://localhost:8080/auth> pertanto l'applicazione si aspetta che keycloak sia fatto partire sulla porta di default, la 8080.

L'applicazione utilizza l'adapter Node.js messo a disposizione di keycloak la cui documentazione (per chi volesse approfondire) si trova qui: https://www.keycloak.org/docs/latest/securing_apps/#_nodejs_adapter

Per eseguire frontend e backend contenuti nella cartella ch2 (avendo già installato node js e il package manager npm):

- aprire un terminale nella cartella frontend
- eseguire npm install (questa operazione legge il contenuto di package.json e scarica in una sottocartella node_modules tutte le librerie necessarie)
- eseguire npm start (start è definito in package.json ed è semplicemente uno script che esegue "node app.js")
- ripetere i tre passaggi sopra per la cartella backend