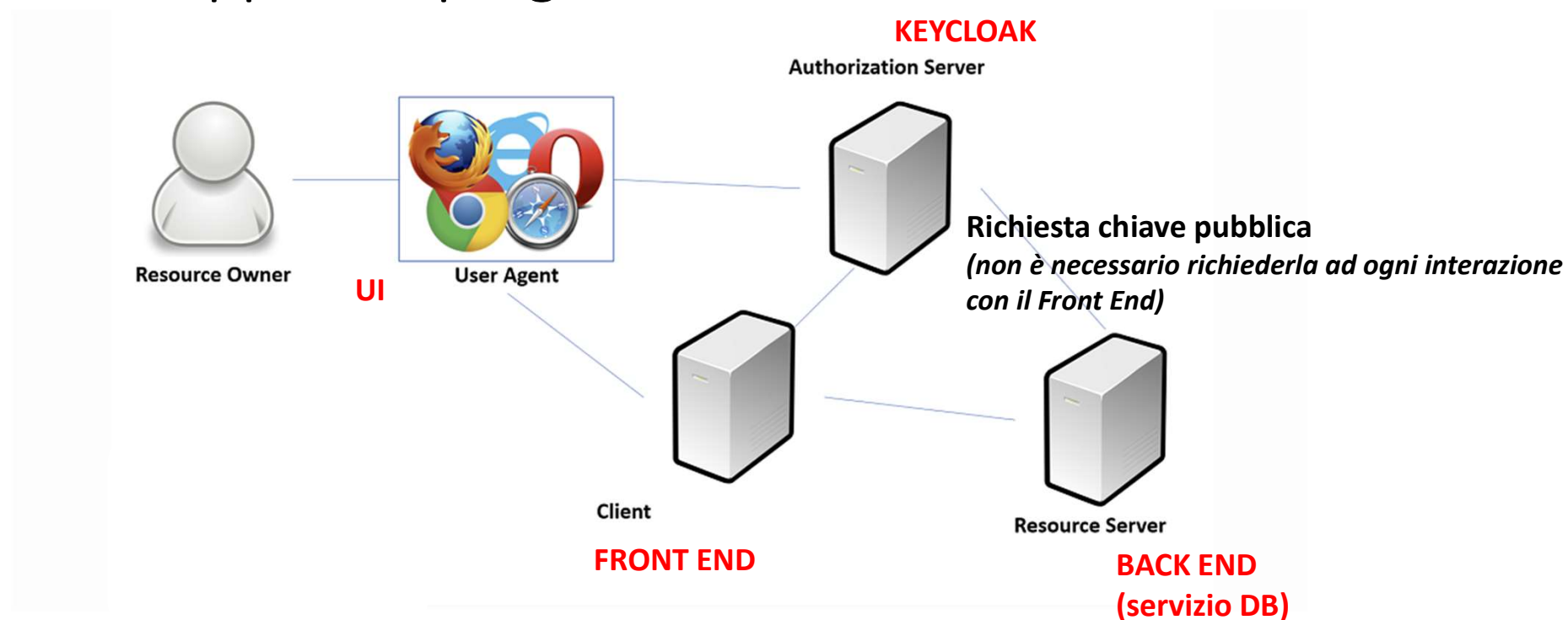


Autenticazione e autorizzazione con OAuth2, Open ID Connect tramite server Keycloak

Sommario

- Il protocollo OAUTH2
- JSON Web Token
- Il server Keycloak
- Applicazione al progetto
- Alcuni esperimenti

Come si applica al progetto



Esempio 1: applicazione NodeJS che usa Keycloak

Il primo esempio di applicazione NodeJS è descritta nel cap. 2 del libro – comprende un frontend ed un backend e mostra come si possano proteggere (alcune del)le funzionalità del backend limitando l’accesso ad utenti autenticati (tramite Keycloak)

Le due applicazioni si trovano nella cartella ch2 del repository scaricabile da github:

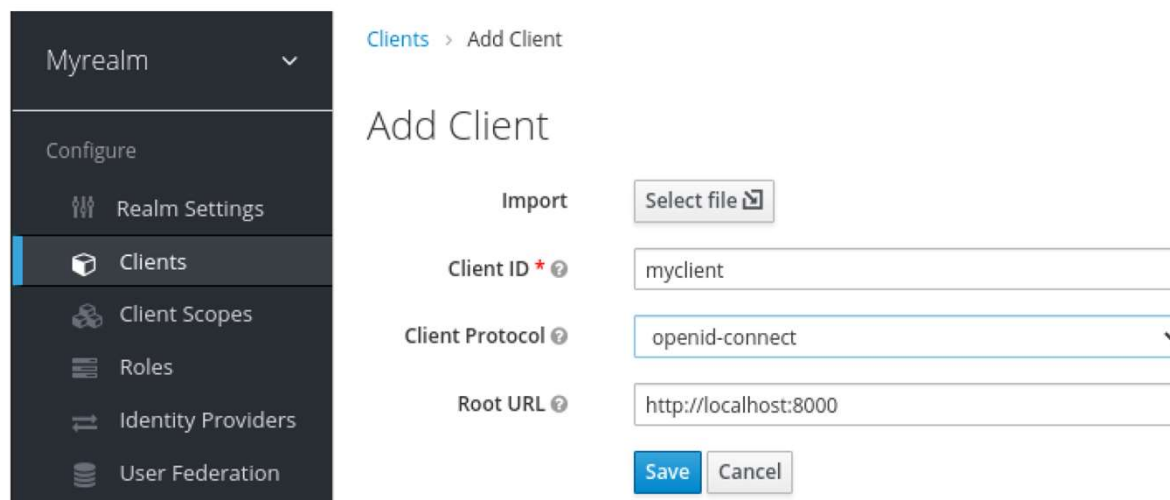
<https://github.com/PacktPublishing/Keycloak-Identity-and-Access-Management-for-Modern-Applications>

Su youtube trovate una playlist che mostra come avviarle :

<https://www.youtube.com/playlist?list=PLcLcvrwLe187DykEKXg-9Urd1Z6MQT61d>

Una prima applicazione di esempio

Prima di iniziare: preparare un realm «myrealm» con un utente (scegliete voi un nome, nel mio esempio lo username è «keycloak») al quale è stato assegnato un ruolo «myrole» (creato per l'esempio all'interno di myrealm). Inoltre occorre registrare l'app «myclient» nel realm



The screenshot displays the Keycloak Admin Console interface. On the left, a dark sidebar menu shows the navigation options: 'Myrealm' (selected), 'Configure', 'Realm Settings', 'Clients' (highlighted with a blue bar), 'Client Scopes', 'Roles', 'Identity Providers', and 'User Federation'. The main content area is titled 'Clients > Add Client'. It features an 'Import' section with a 'Select file' button. Below this, there are three form fields: 'Client ID' with the value 'myclient', 'Client Protocol' with a dropdown menu set to 'openid-connect', and 'Root URL' with the value 'http://localhost:8000'. At the bottom of the form are 'Save' and 'Cancel' buttons.

Figure 2.4 – Creating the client in the admin console

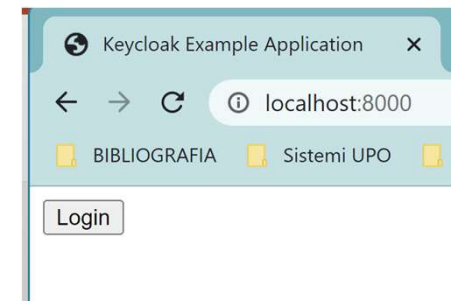
Una prima applicazione di esempio (avviare l'app)

- Avviare keycloak (sulla porta di default, la 8080)
- Eseguire all'interno delle due cartelle frontend e backend (occorre avere già installato nodeJS)
 - npm install
 - npm start
- Il frontend (la client app) si raggiunge tramite browser alla URL localhost:8000 mentre il backend è attivo sulla porta 3000

Sperimentare il primo esempio

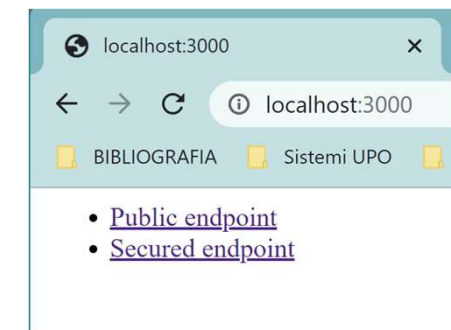
FRONTEND

Su localhost:8000 si vede solo un bottone «login»



BACKEND

Su localhost:3000 ci sono 2 link:



- Se si tenta di fare login senza aver registrato l'app su keycloak, si otterrà un errore perché l'applicazione non viene riconosciuta.
- Cliccando su «Secured endpoint» lato backend si ottiene un errore «Access denied» - accedendo dal frontend previa autenticazione si riuscirà ad accedere alla parte protetta del backend

The screenshot shows a web browser window with the URL `localhost:8080/auth/realms/myrealm/protocol/openid-connect/auth?client_id=myclient&redirect_uri=http%3A%2F%2Flocalhost:8000/`. The page displays a "Sign in to your account" form with fields for "Username or email" and "Password", and a "Sign In" button. A callout box points to the "Sign In" button with the text: "Cliccando su Login si viene rediretti su Keycloak per l'autenticazione".

The Chrome DevTools network panel shows a list of requests:

Metodo	Metodo	Dominio	File	Iniziatore	Tipo	Trasferito	Dime...
GET	localhost:8080	auth?client_id=myclient&redi	keycloak.js:1345 (docum...	html	5,28 kB	3,81 kB	
GET	localhost:8080	all.css	stylesheet	x-unk...	NS_ERR...	0 B	
GET	localhost:8080	base.css	stylesheet	css	6,12 kB	38,84 ...	
GET	localhost:8080	app.css	stylesheet	css	52,31 kB	508,7...	
GET	localhost:8080	patternfly.min.css	stylesheet	css	32,04 kB	182,7...	
GET	localhost:8080	patternfly-additions.min.css	stylesheet	css	31,23 kB	225,0...	

The console panel shows the following messages:

- 16 richieste
- 1,31 MB di 460,49 kB trasferiti
- Completato: 326 ms
- DOMContentLoaded: 58 ms
- load: 387 ms

The network panel details for the first request (GET http://localhost:8080/auth/realms/myrealm/protocol/openid-connect/auth?client_id=myclient&redirect_uri=http://localhost:8000/&state=c6505b65-919e-439b-87e5-e4d6616f4d7f&response_mode=fragment&response_type=code&scope=openid&nonce=7572510e-6a6e-49e6-8c32-b5b5dfd0a5a2) show:

- Stato: 200 OK
- Versione: HTTP/1.1
- Trasferito: 5,28 kB (dim. 3,81 kB)
- Referrer Policy: strict-origin-when-cross-origin

Qui l'utente è autenticato, notiamo la richiesta del token

The screenshot shows a web browser window with a dark theme. The address bar displays 'localhost:8000'. The page content includes a navigation bar with buttons: 'Logout', 'Show ID Token', 'Show Access Token', 'Refresh', and 'Invoke Service'. Below this, the text 'Hello Giuliana Franceschinis' is displayed next to a small square placeholder. The bottom of the browser window shows the 'Rete' (Network) tab of the developer tools. The network log lists several requests, with the 'token' request highlighted in blue. The details for this request show a JSON response with fields: 'code', 'grant_type', 'client_id', and 'redirect_uri'. The status bar at the bottom indicates 9 requests, 99.74 kB transferred, and a load time of 206 ms.

Logout Show ID Token Show Access Token Refresh Invoke Service

Hello Giuliana Franceschinis

Analisi pagina Console Debugger Rete Editor stili Prestazioni Memoria Archiviazione Accessibilità Applicazione

Stato	Metodo	Dominio	File	Iniziatore	Tipo	Trasferito	Dime...
404	GET	localhost:8000	favicon.ico	FaviconLoader.jsm:180 (...)	html	In cache	150 B
200	GET	localhost:8080	step1.html	subdocument	html	1,44 kB	955 B
200	GET	localhost:8080	step2.html	step1.html:24 (subdocu...	html	1,63 kB	1,14 kB
200	GET	localhost:8080	login-status-iframe.html	subdocument	html	4,20 kB	3,71 kB
200	POST	localhost:8080	token	keycloak.js:790 (xhr)	json	4,14 kB	3,62 kB
204	GET	localhost:8080	init?client_id=myclient&origi...	login-status-iframe.html...	xml	224 B	0 B

9 richieste | 99,74 kB di 34,64 kB trasferiti | Completato: 5,50 s | DOMContentLoaded: 204 ms | load: 206 ms

Filtra parametri di richiesta

Dati dei moduli

code: "93827d1d-51ac-40b1-be12-c276a20dc92f.0202bcbf-fc14-4978-b146-d0efec8697cc.80bb0d2b-aa7a-4827-a947-0e7bfd82829c"
grant_type: "authorization_code"
client_id: "myclient"
redirect_uri: "http://localhost:8000/"

Filtra messaggi

Errori Avvisi Log Info Debug CSS XHR Richieste

E l'access token in risposta da Keycloak

LogoutShow ID TokenShow Access TokenRefreshInvoke Service

Hello Giuliana Franceschinis

Crea PDF

Analisi paginaConsoleDebuggerReteEditor stiliPrestazioniMemoriaArchiviazioneAccessibilitàApplicazione

Filtra URL

TuttiHTMLCSSJSXHRCaratteriImmaginiMediaWSAltroDisattiva cacheNessun limite

Stato	Meto...	Dominio	File	Iniziatore	Tipo	Trasferito	Dime...	Header	Cookie	Richiesta	Risposta	Tempi	Analisi dello stack
404	GET	localhost:8000	favicon.ico	FaviconLoader.jsm:180 (...)	html	In cache	150 B	Filtro proprietà					
200	GET	localhost:8080	step1.html	subdocument	html	1,44 kB	955 B	JSON					
200	GET	localhost:8080	step2.html	step1.html:24 (subdocu...	html	1,63 kB	1,14 kB	Non elaborata (raw)					
200	GET	localhost:8080	login-status-iframe.html	subdocument	html	4,20 kB	3,71 kB	access_token: "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXLTUzZmNjZGE2MWEiLCJpc3MiOiJodHRwOi8vbG9jYXRob3N0OjgweODAvYXV0aC9yZWZsbXVxXyZWZsbSIsImF1ZC16ImFjY291bnQlLCJzdWiiOiI5Mjc3NzlyMC1kNzE5LTQzMWQyTE2MS1mYVY3Mjg2MmMyYmEiLCJ0eXAiOiJCZWZyZXliLCJhenAiOiJteWNsaWVudCIsIm5vbmNIjoiNzU3MjUxMGUtNmE2ZS00OWU2LTJhMzItYjVlNWRmZDBhNWYyIiwic2Vzcl21bGllbmFuZnJlbnNlc2NoaW5pcr0RnbWFEnbC5ib20ifO.i87zK7AmHc3WM2r4alop4.fDWLn3zUTN					
200	POST	localhost:8080	token	keycloak.js:790 (xhr)	json	4,14 kB	3,62 kB						
204	GET	localhost:8080	init?client_id=myclient&origi	login-status-iframe.html...	xml	224 B	0 B						
9 richieste99,74 kB di 34,64 kB trasferitiCompletato: 5,50 sDOMContentLoaded: 204 msload: 206 ms													

UNIVERSITÀ DEL PIEMONTE ORIENTALE

Access token:

Nel payload del token
vediamo il nome utente
che viene visualizzato
anche sul browser

HAi0jE2ODM40TAzMDASImIhdcI6MTY4Mzg5MDAw
MCwiYXV0aF90aW1lIjoXNjgzODkwMDAwLcJqdGk
i0iJlY2IyNjM3NC0xMWZhLTQ2OTUtOGQ1NC1hZT
U3ZmNjZGE2MWEiLcJpc3MiOiJodHRwOi8vbg9jY
Wxob3N00jgwODAvYXV0aC9yZWZfbXVxbXlyZWZs
bSIsImF1ZCI6ImFjY291bnQilcJzdWII0iI5Mjc
3NzIyMC1kNzESLTQzMWQwYTE2MS1mYWY3Mjg2Mm
MyYmEiLcJ0eXAiOiJCZWZlYXZiLcJhenAiOiJte
WNsaWVudCIsIm5vbmlIiJoiNzU3MjUxMGUtNmE2
ZS00WU2LTGJzIyZlYiViNWrmZDBhNWeyIiwiC2V
zc2lhb19zdGh0ZSI6ImFjYmJiY2JmLWJzMTQTND
k3OC1mTQ2LWQwZWZlYzg2OTdjYyIsImFjcjciOiI
jEiLcJhbGxvd2VklW9yaWdpbnMiOl5iaHR0cDov
L2xvY2FsaG9zdDo4MDAwIl0sInJlYWxtX2FjY2V
zcyI6eyJyb2xlcYi6WyJvZmZsaW5lX2FjY2Vzcy
IsInVtYV9hdXRob3JpemF0aW9uIiwibXlyb2x1I
l19LcJyZXNvdXJjZV9hY2Nlc3MiOnsiYWNjb3Vu
dCI6eyJyb2xlcYi6WyJtYW5hZ2UtYWNjb3VudCI
sIm1hbmFnZS1hY2NvdW50LWxpbnmtzIiwiZmlldy
1wcm9maWx1Ii19fSwic2NvcGUiOiJvcGVuaWQgc
HJvZm1sZSB1bWFPbCIsImVtYW1sX3Zlcm1maWVk
IjpmYXZzZWibmFkdzSI6IkdpdWxpYW5hIEZyYW5
j2XNjGlaUaMiLcJwcmVmZXJyZWZrfdXNlcm5hbW
U0iOiJrZXljY29hYyIsImdpdmVudX25hbmWU0iOiJHa
XVsaWZlY29hYyIsImZhbmW1seV9uYW1lIjoiRnJhbmNl
c2NoaW5pcyIsImVtYW1sIjoiZ2l1bG1hbmEuZnJ
hbmNlc2NoaW5pc0BnbWFPbC5jb20ifQ. j8ZkZA
mHs3WM2r4algp4_fDWHp3zUTN4Jbdx7r424Xz9Y
rFmpn4FEkFXqGf-
JUaUxamai9Exgcr9TDZAeA_ubsVwLjGp9mEWgX0
tF_-R8Tk17CYe-
82HVcDrFTEBNZbnjD97xjeMsgbdlIH4RS82w1aAB
1p_4eOrn66sXqvXU0vLejUdThR0cX4YEkGzJSX8
vOn0kIDjUyx4vHqFFXE80fwkEgHqtmNsFIiUlkq
v6t-rIjGp0K_vdTP0.

```

    "kid": "1-D0xQRzH16nL7pwXKZWfN9u8n2J5n1nL6fFoQVzKc"
  }
}

PAYLOAD: DATA

{
  "exp": 1683890300,
  "iat": 1683890000,
  "auth_time": 1683890000,
  "jti": "ecb26374-11fa-4695-8d54-ae57fccda61a",
  "iss": "http://localhost:8080/auth/realms/myrealm",
  "aud": "account",
  "sub": "92777220-d719-431d-a161-faf72862c2ba",
  "typ": "Bearer",
  "azp": "myclient",
  "nonce": "7572510e-6a6e-49e6-8c32-b5b5dfd0a5a2",
  "session_state": "0202bcbf-fc14-4978-b146-d0efec8697cc",
  "acr": "1",
  "allowed-origins": [
    "http://localhost:8080"
  ],
  "realm_access": {
    "roles": [
      "offline_access",
      "uma_authorization",
      "myrole"
    ]
  },
  "resource_access": {
    "account": {
      "roles": [
        "manage-account",
        "manage-account-links",
        "view-profile"
      ]
    }
  },
  "scope": "openid profile email",
  "email_verified": false,
  "name": "Giuliana Franceschinis",
  "preferred_username": "keycloak",
  "given_name": "Giuliana",
  "family_name": "Franceschinis",
  "email": "giuliana.franceschinis@gmail.com"
}

```

Show Access Token

Il pannello mostrato dall'app permette di visualizzare il contenuto del token (che naturalmente è lo stesso ottenuto decodificando l'access token in jwt.io)

Logout Show ID Token Show Access Token Refresh Invoke Service

Hello Giuliana Franceschinis




```
{
  "exp": 1683890300,
  "iat": 1683890000,
  "auth_time": 1683890000,
  "jti": "ecb26374-11fa-4695-8d54-ae57fccda61a",
  "iss": "http://localhost:8080/auth/realms/myrealm",
  "aud": "account",
  "sub": "92777220-d719-431d-a161-faf72862c2ba",
  "typ": "Bearer",
  "azp": "myclient",
  "nonce": "7572510e-6a6e-49e6-8c32-b5b5dfd0a5a2",
  "session_state": "0202bcbf-fc14-4978-b146-d0efec8697cc",
  "acr": "1",
  "allowed-origins": [
    "http://localhost:8000"
  ],
  "realm_access": {
    "roles": [
      "offline_access",
      "uma_authorization",
      "myrole"
    ]
  },
  "resource_access": {
    "account": {
      "roles": [
        "manage-account",
        "manage-account-links",
        "view-profile"
      ]
    }
  }
}
```

Invoke service: richiama la funzione protetta del backend

[Logout](#) [Show ID Token](#) [Show Access Token](#) [Refresh](#) [Invoke Service](#)

Hello Giuliana Franceschinis



200

Secret message!

Il token viene passato al backend nell'header della richiesta