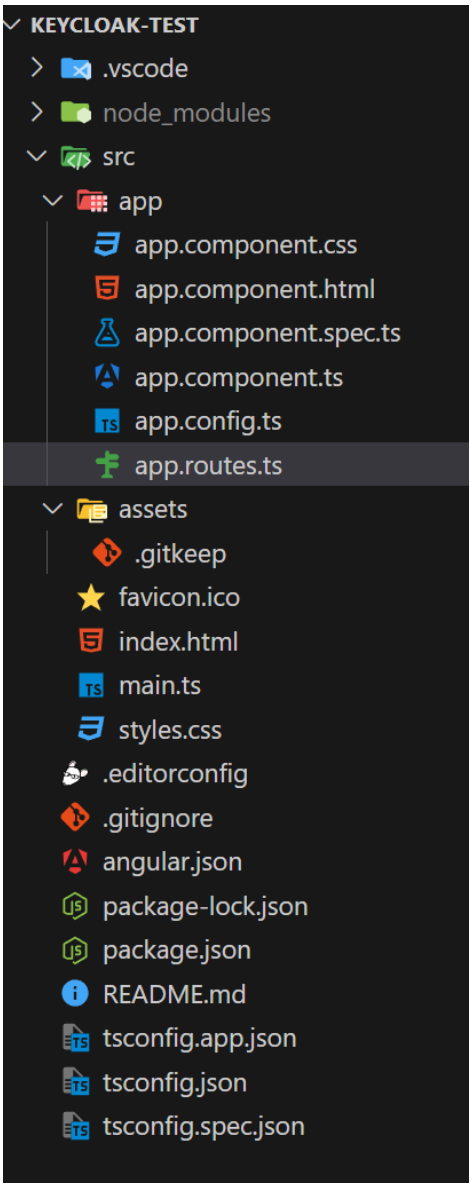
Keycloak mit Angular aufsetzen

Minimale Systemanforderungen:
Angular 17
Node js: v18.13
Typescript: 5.2

Zudem empfehle ich Visual Studio Code zu nutzen!

1. Create Angular 17 Project:
   ng new keycloak-test
   Stylesheet: css
   Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)?: No

   So ungefähr sollte die Ordnerstruktur nach der Erstellung des Projekts aussehen:



Keycloak mit Angular aufsetzen

2. Benötigte Datein einfügen

   Fügen Sie folgenden Source-Code in ihre app.config.ts ein.

```
Source-Code:
import {APP_INITIALIZER, ApplicationConfig, Provider} from
'@angular/core';
import { provideRouter } from '@angular/router';

import { routes } from './app.routes';
import {KeycloakBearerInterceptor, KeycloakService} from "keycloak-
angular";
import {HTTP_INTERCEPTORS, provideHttpClient, withInterceptorsFromDi}
from "@angular/common/http";
import { provideAnimationsAsync } from '@angular/platform-
browser/animations/async';

// Function to initialize Keycloak with the necessary configurations
function initializeKeycloak(keycloak: KeycloakService) {
  return () =>
    keycloak.init({
      config: {
        url: 'https://auth.htl-leonding.ac.at', // URL of the
Keycloak server
        realm: 'htl-leonding', // Realm to be used in Keycloak
        clientId: 'htlleonding-service' // Client ID for the
application in Keycloak,
      },
      initOptions: {
        onLoad: 'check-sso', // Action to take on load
        //enableLogging: true, // Enables logging
        pkceMethod: 'S256', // Proof Key for Code Exchange (PKCE)
method to use
        // IMPORTANT: implicit flow is no longer recommended, but
using standard flow leads to a 401 at the keycloak server
        // when retrieving the token with the access code - we leave
it like this for the moment until a solution is found
        flow: 'implicit',
        silentCheckSsoRedirectUri:
          window.location.origin + '/assets/silent-check-sso.html' //
URI for silent SSO checks
      },
      // Enables Bearer interceptor
      enableBearerInterceptor: true,
      // Prefix for the Bearer token
      bearerPrefix: 'Bearer',
      // URLs excluded from Bearer token addition (empty by default)
      //bearerExcludedUrls: []
    });
}

// Provider for Keycloak Bearer Interceptor
```

Keycloak mit Angular aufsetzen

```
const KeycloakBearerInterceptorProvider: Provider = {
  provide: HTTP_INTERCEPTORS,
  useClass: KeycloakBearerInterceptor,
  multi: true
};

// Provider for Keycloak Initialization
const KeycloakInitializerProvider: Provider = {
  provide: APP_INITIALIZER,
  useFactory: initializeKeycloak,
  multi: true,
  deps: [KeycloakService]
}

export const appConfig: ApplicationConfig = {
  providers: [
    provideHttpClient(withInterceptorsFromDi()), // Provides
HttpClient with interceptors
    KeycloakInitializerProvider, // Initializes Keycloak
    KeycloakBearerInterceptorProvider, // Provides Keycloak Bearer
Interceptor
    KeycloakService, // Service for Keycloak
    provideRouter(routes),
    provideAnimationsAsync()
  ]
};
```

Link des Source-Code:
leo_keycloak_demos/Angular_WebAPI/src/app/app.config.ts at master · markushaslinger/leo_keycloak_demos · GitHub

Führen Sie danach folgende Commands aus:

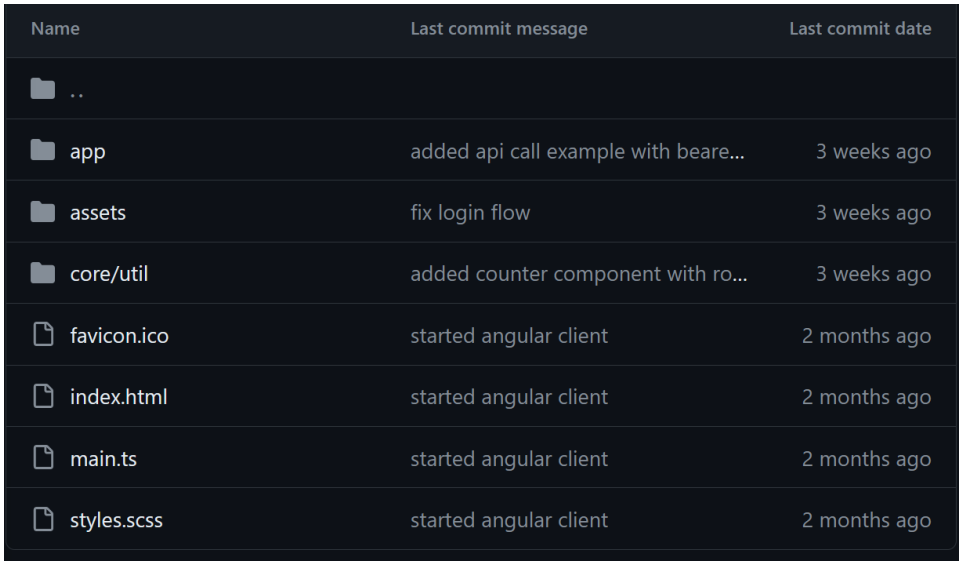
```
npm install
npm install keycloak-angular
```

Danach sollten keine Errors mehr angezeigt werden.

3. Einfügen von Code für das abfangen von Daten der Leocloud

Keycloak mit Angular aufsetzen

Erstellen Sie innerhalb des src-Ordners den Ordner „core“ und innerhalb dem „util“



Name	Last commit message	Last commit date
..		
app	added api call example with beare...	3 weeks ago
assets	fix login flow	3 weeks ago
core/util	added counter component with ro...	3 weeks ago
favicon.ico	started angular client	2 months ago
index.html	started angular client	2 months ago
main.ts	started angular client	2 months ago
styles.scss	started angular client	2 months ago

Erstellen Sie dann im Util-Ordner auth-guard.ts und leo-token.ts

Gehen Sie dann auf
https://github.com/markushaslinger/leo_keycloak_demos/tree/master/Angular_WebAPI/src/core/util

und fügen Sie den Code in die jeweillige Datei ein

4. Demo-Projekt
   Erstellen Sie ein Login-Component und fügen Sie folgendes Code in die login-component.ts Datei ein.

```
import {Component} from '@angular/core';
import {KeycloakService} from "keycloak-angular";
import {MatButton} from "@angular/material/button";

@Component({
  selector: 'app-home',
  standalone: true,
  imports: [
    MatButton
  ],
  templateUrl: './home.component.html',
  styleUrl: './home.component.scss'
})
export class HomeComponent {
  isLoggedIn = false;

  constructor(private keycloakService: KeycloakService) {
    this.isLoggedIn = this.keycloakService.isLoggedIn();
    this.keycloakService.getToken().then(token => {
      console.log(token);
    });
  }

  async login(): Promise<void> {
```

Keycloak mit Angular aufsetzen

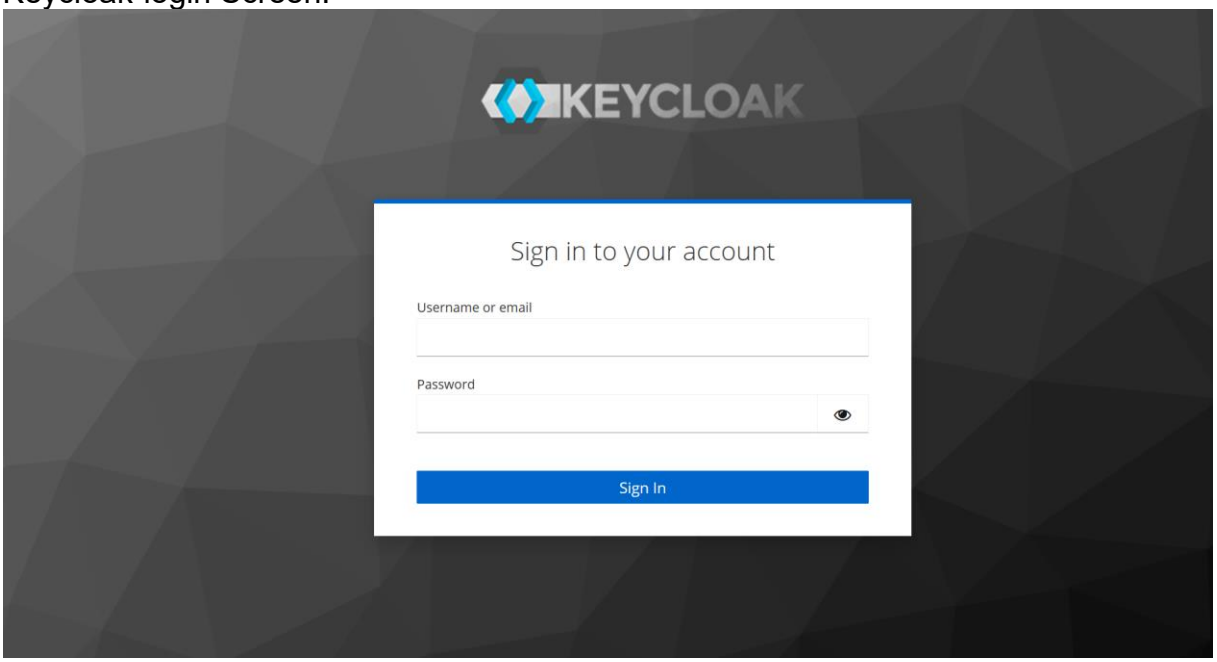
```
    if (this.isLoggedIn) {
      return
    }
    await this.keycloakService.login();

  }

  async logout(): Promise<void> {
    if (!this.isLoggedIn) {
      return;
    }
    await this.keycloakService.logout();
  }
}
```

HTML-Datei:

```
@if (isLoggedIn){
  <span>You are logged in</span>
  <button mat-button (click)="logout()">Logout</button>
} @else {
  <span>You are not logged in</span>
  <button mat-button (click)="login()">Login</button>
}
```

Beim einloggen werden Sie dann direkt an die Leo-Cloud-Login Website weitergeleitet.

Keycloak-login Screen:



Für mehere Information:
https://github.com/markushaslinger/leo_keycloak_demos/tree/master/Angular_WebAPI

Keycloak mit Angular aufsetzen

Github-Repo (erstellt von Prof. Haslinger):
https://github.com/markushaslinger/leo_keycloak_demos