# Vorbereitung

* NodeJS installieren
  + <https://nodejs.org/en/>
* CMD öffnen
  + Den Befehl „npm install -g @angular/cli“ ausführen
* IDE installieren (z.B. IntelliJ, Visual Studio Code)
  + Ich werde am Workshop IntelliJ benutzen
* PersonRestServer
  + Datei PersonRestServer.jar herunterladen
    - <https://github.com/markusborer/PersonRestServer/tree/master/release>
  + PersonRestServer.jar ausführen
    - CMD öffnen
    - In das Verzeichnis mit dem JAR wechseln
    - Den Befehl „java -jar PersonRestServer.jar“ ausführen
  + test (<http://localhost:8080/person?name=borer>)

# Applikation erstellen

* CMD als **Administrator** öffnen
* In den Ordner für das Projekt wechseln
* ng new PersonApp --routing true
* In den Projektordner wechseln
* ng serve
  + <http://localhost:4200/>
* CLI Commands zeigen
* ng test
* ng e2e
* Projekt in IDE öffnen
* TSLint ausschalten
* Repository in github.com erstellen
* TortoiseGit -> Settings -> Git -> Remote -> URL -> <https://github.com/markusborer/PersonApp.git>
* git push

# Home hinzufügen

* CMD öffnen
* In den Projektordner wechseln
* ng generate component Home

## app-routing.module.ts

**import** { NgModule } **from '@angular/core'**;  
**import** { Routes, RouterModule } **from '@angular/router'**;  
**import** {HomeComponent} **from "./home/home.component"**;

**const** routes: Routes = [  
 {  
 **path**: **''**,  
 **redirectTo**: **'home'**,  
 **pathMatch**: **'full'** },  
 {  
 **path**: **'home'**,  
 **component**: HomeComponent  
 },  
 {  
 **path**: **'\*\*'**,  
 **redirectTo**: **'home'** }  
]

@NgModule({  
 **imports**: [RouterModule.forRoot(routes)],  
 **exports**: [RouterModule]  
})  
**export class** AppRoutingModule { }

## app.component.html

<**div style="text-align**:**center"**>  
 <**h1**>  
 Welcome to {{ title }}!  
 </**h1**>  
</**div**>  
<**ul**>  
 <**li**>  
 <**h2**><**a routerLink="home"**>Home</**a**></**h2**>  
 </**li**>  
</**ul**>  
<**router-outlet**></**router-outlet**>

# Personensuche hinzufügen

* CMD öffnen
* In den Projektordner wechseln
* ng generate class shared/Person
* ng generate service shared/Person
* ng generate component PersonSearch

## person.ts

**export class** Person {  
 **name**: **string**;  
 **vorname**: **string**;  
}

## person-service.ts

**import** {Injectable} **from '@angular/core'**;  
**import** {HttpClient, HttpParams} **from '@angular/common/http'**;  
**import** {Observable} **from 'rxjs/Observable'**;  
**import** {Person} **from './person'**;  
  
@Injectable()  
**export class** PersonService {  
  
 **private url** = **'http://localhost:8080/person'**;  
  
 **constructor**(**private** httpClient: HttpClient) {  
 }  
  
 search(searchTerm: **string**): Observable<Person[]> {  
 **let** options = {};  
 **if** (searchTerm !== undefined) {  
 options = { **params**: **new** HttpParams().set(**'name'**, searchTerm) };  
 }  
 **return this**.httpClient.get<Person[]>(**this**.**url**, options);  
 }  
  
}

## person-search.component.ts

**import** { Component, OnInit } **from '@angular/core'**;  
**import** {Person} **from '../shared/person'**;  
**import** {PersonService} **from '../shared/person.service'**;  
  
@Component({  
 **selector**: **'app-person-search'**,  
 **templateUrl**: **'./person-search.component.html'**,  
 **styleUrls**: [**'./person-search.component.css'**]  
})  
**export class** PersonSearchComponent **implements** OnInit {  
  
 **searchTerm** = **''**;  
 **persons**: Person[] = [];  
  
 **constructor**(**private** personService: PersonService) {  
 }  
  
 ngOnInit() {  
 }  
  
 search() {  
 **this**.personService.search(**this**.**searchTerm**).subscribe(  
 (data) => {  
 ***console***.log(data);  
 **this**.**persons** = data;  
 }  
 );  
 }  
  
}

## person-search.component.html

<**h1**>Personen</**h1**>  
<**input type="text" [(ngModel)]="searchTerm" name="searchTerm" (ngModelChange)="search()"**>  
<**table**>  
 <**thead**>  
 <**tr**>  
 <**th**>Name</**th**>  
 <**th**>Vorname</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 <**tr \*ngFor="let *person* of persons"**>  
 <**td**>{{***person***.**name**}}</**td**>  
 <**td**>{{***person***.**vorname**}}</**td**>  
 </**tr**>  
 </**tbody**>  
</**table**>

## app.component.html

<**div style="text-align**:**center"**>  
 <**h1**>  
 Welcome to {{ title }}!  
 </**h1**>  
</**div**>  
<**ul**>  
 <**li**>  
 <**h2**><**a routerLink="home"**>Home</**a**></**h2**>  
 </**li**>  
 <**li**>  
 <**h2**><**a routerLink="person-search"**>Personen-Suche</**a**></**h2**>  
 </**li**>  
</**ul**>  
<**router-outlet**></**router-outlet**>

## app-routing.module.ts

**import** { NgModule } **from '@angular/core'**;  
**import** { Routes, RouterModule } **from '@angular/router'**;  
**import** {HomeComponent} **from "./home/home.component"**;

**import** {PersonSearchComponent} **from "./person-search/person-search.component"**;

**const** routes: Routes = [  
 {  
 **path**: **''**,  
 **redirectTo**: **'home'**,  
 **pathMatch**: **'full'** },  
 {  
 **path**: **'home'**,  
 **component**: HomeComponent  
 },  
 {  
 **path**: **'person-search'**,  
 **component**: PersonSearchComponent  
 },  
 {  
 **path**: **'\*\*'**,  
 **redirectTo**: **'home'** }  
]

@NgModule({  
 **imports**: [RouterModule.forRoot(routes)],  
 **exports**: [RouterModule]  
})  
**export class** AppRoutingModule { }

## app.module.ts anpassen

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;  
  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 PersonSearchComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

# Was ist jetzt noch zu tun?

* Überholende Responses
* ErrorHandler
* Objekte
* Debouncing
* Loader
* Berechtigung
* Library-Versionen fixieren

# Überholende Responses (Variante 1)

## person-search.component.ts

**import** { Component, OnInit } **from '@angular/core'**;  
**import** {Person} **from '../shared/person'**;  
**import** {PersonService} **from '../shared/person.service'**;  
**import** {Subscription} **from 'rxjs/Subscription'**;  
  
@Component({  
 **selector**: **'app-person-search'**,  
 **templateUrl**: **'./person-search.component.html'**,  
 **styleUrls**: [**'./person-search.component.css'**]  
})  
**export class** PersonSearchComponent **implements** OnInit {  
  
 **searchTerm** = **''**;  
 **persons**: Person[] = [];  
 **subscription**: Subscription;  
  
 **constructor**(**private** personService: PersonService) {  
 }  
  
 ngOnInit() {  
 }  
  
 search() {  
 **if** (**this**.**subscription**) {  
 **this**.**subscription**.unsubscribe();  
 }  
 **this**.**subscription** = **this**.personService.search(**this**.**searchTerm**).subscribe(  
 (data) => {  
 ***console***.log(data);  
 **this**.**persons** = data;  
 }  
 );  
 }  
  
}

# Überholende Responses (Variante 2)

## person-search.component.html

<**h1**>Personen</**h1**>  
<**form #searchForm="ngForm"**>  
 <**input type="text" [(ngModel)]="searchTerm" name="searchTerm"**>  
</**form**>  
<**table**>  
 <**thead**>  
 <**tr**>  
 <**th**>Name</**th**>  
 <**th**>Vorname</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 <**tr \*ngFor="let *person* of persons"**>  
 <**td**>{{***person***.**name**}}</**td**>  
 <**td**>{{***person***.**vorname**}}</**td**>  
 </**tr**>  
 </**tbody**>  
</**table**>

## person-search.component.ts

**import** {Component, OnInit, ViewChild} **from '@angular/core'**;  
**import** {Person} **from '../shared/person'**;  
**import** {PersonService} **from '../shared/person.service'**;  
  
@Component({  
 **selector**: **'app-person-search'**,  
 **templateUrl**: **'./person-search.component.html'**,  
 **styleUrls**: [**'./person-search.component.css'**]  
})  
**export class** PersonSearchComponent **implements** OnInit {  
  
 **searchTerm** = **''**;  
 **persons**: Person[] = [];  
 @ViewChild(**'searchForm'**) **form**;  
  
 **constructor**(**private** personService: PersonService) {  
 }  
  
 ngOnInit() {  
 **this**.**form**.**control**.**valueChanges** .**switchMap**(s => **this**.personService.search(**this**.**searchTerm**))  
 .subscribe(  
 data => {  
 ***console***.log(data);  
 **this**.**persons** = data;  
 }  
 )  
 ;  
 }  
  
}

## app.module.ts

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;

**import 'rxjs/add/operator/switchMap'**;  
  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 PersonSearchComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

# ErrorHandler

## person-search.component.ts

**import** {Component, OnInit, ViewChild} **from '@angular/core'**;  
**import** {Person} **from '../shared/person'**;  
**import** {PersonService} **from '../shared/person.service'**;  
**import** {Observable} **from 'rxjs/Observable'**;  
  
@Component({  
 **selector**: **'app-person-search'**,  
 **templateUrl**: **'./person-search.component.html'**,  
 **styleUrls**: [**'./person-search.component.css'**]  
})  
**export class** PersonSearchComponent **implements** OnInit {  
  
 **searchTerm** = **''**;  
 **persons**: Person[] = [];  
 @ViewChild(**'searchForm'**) **form**;  
  
 **constructor**(**private** personService: PersonService) {  
 }  
  
 ngOnInit() {  
 **this**.**form**.**control**.**valueChanges** .**switchMap**(s => **this**.personService.search(**this**.**searchTerm**)  
 .**catch**(error => {  
 ***console***.log(error);  
 **return** Observable.*of*(undefined);  
 })  
 )  
 .subscribe(  
 data => {  
 ***console***.log(data);  
 **this**.**persons** = data;  
 }  
 )  
 ;  
 }  
  
}

## app.module.ts

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;  
  
**import 'rxjs/add/observable/of'**;  
**import 'rxjs/add/operator/catch'**;  
**import 'rxjs/add/operator/switchMap'**;  
  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 PersonSearchComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

# Objekte

## person.ts

**export class** Person {  
 **name**: **string**;  
 **vorname**: **string**;  
  
 getEmail(): **string** {  
 **return this**.**vorname** + **'.'** + **this**.**name** + **'@adesso.ch'**;  
 }  
}

## person-search.component.html

<**h1**>Personen</**h1**>  
<**form #searchForm="ngForm"**>  
 <**input type="text" [(ngModel)]="searchTerm" name="searchTerm"**>  
</**form**>  
<**table**>  
 <**thead**>  
 <**tr**>  
 <**th**>Name</**th**>  
 <**th**>Vorname</**th**>  
 <**th**>EMail</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 <**tr \*ngFor="let *person* of persons"**>  
 <**td**>{{***person***.**name**}}</**td**>  
 <**td**>{{***person***.**vorname**}}</**td**>  
 <**td**>{{***person***.getEmail()}}</**td**>  
 </**tr**>  
 </**tbody**>  
</**table**>

## Warum funktioniert das nicht?

## person-service.ts

**import** {Injectable} **from '@angular/core'**;  
**import** {HttpClient, HttpParams} **from '@angular/common/http'**;  
**import** {Observable} **from 'rxjs/Observable'**;  
**import** {Person} **from './person'**;  
  
@Injectable()  
**export class** PersonService {  
  
 **private url** = **'http://localhost:8080/person'**;  
  
 **constructor**(**private** httpClient: HttpClient) {  
 }  
  
 search(searchTerm: **string**): Observable<Person[]> {  
 **let** options = {};  
 **if** (searchTerm !== undefined) {  
 options = { **params**: **new** HttpParams().set(**'name'**, searchTerm) };  
 }  
 **return this**.httpClient.get<Person[]>(**this**.**url**, options)  
 .**map**(persons => persons.map(person => **new** Person(person)));  
 }  
  
}

## person.ts

**export class** Person {  
 **name**: **string**;  
 **vorname**: **string**;  
  
 **constructor**(person: Person) {  
 **this**.**name** = person.**name**;  
 **this**.**vorname** = person.**vorname**;  
 }  
  
 getEmail(): **string** {  
 **return this**.**vorname** + **'.'** + **this**.**name** + **'@adesso.ch'**;  
 }  
}

## app.module.ts

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;  
  
**import 'rxjs/add/observable/of'**;  
**import 'rxjs/add/operator/catch'**;  
**import 'rxjs/add/operator/map'**;  
**import 'rxjs/add/operator/switchMap'**;  
  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 PersonSearchComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

# Debouncing

Requests im Browser anzeigen.

Wie könnte man das beheben?

## person-search.component.ts

**import** {Component, OnInit, ViewChild} **from '@angular/core'**;  
**import** {Person} **from '../shared/person'**;  
**import** {PersonService} **from '../shared/person.service'**;  
**import** {Observable} **from 'rxjs/Observable'**;  
  
@Component({  
 **selector**: **'app-person-search'**,  
 **templateUrl**: **'./person-search.component.html'**,  
 **styleUrls**: [**'./person-search.component.css'**]  
})  
**export class** PersonSearchComponent **implements** OnInit {  
  
 **searchTerm** = **''**;  
 **persons**: Person[] = [];  
 @ViewChild(**'searchForm'**) **form**;  
  
 **constructor**(**private** personService: PersonService) {  
 }  
  
 ngOnInit() {  
 **this**.**form**.**control**.**valueChanges** .**debounceTime**(200)  
 .**switchMap**(s => **this**.personService.search(**this**.**searchTerm**)  
 .**catch**(error => {  
 ***console***.log(error);  
 **return** Observable.*of*(undefined);  
 })  
 )  
 .subscribe(  
 data => {  
 ***console***.log(data);  
 **this**.**persons** = data;  
 }  
 )  
 ;  
 }  
  
}

## app.module.ts

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;  
  
**import 'rxjs/add/observable/of'**;  
**import 'rxjs/add/operator/catch'**;  
**import 'rxjs/add/operator/debounceTime'**;  
**import 'rxjs/add/operator/map'**;  
**import 'rxjs/add/operator/switchMap'**;  
  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 PersonSearchComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

# Loader

ng generate enum shared/LoadingStatus

## loading-status.enum.ts

**export enum** LoadingStatus {  
 *LOADING* = 0,  
 LOADED= 1,  
 *ERROR* = 2  
}

## person-search.component.html

<**h1**>Personen</**h1**>  
<**form #searchForm="ngForm"**>  
 <**input type="text" [(ngModel)]="searchTerm" name="searchTerm"**>  
</**form**>  
<**label \*ngIf="status===LoadingStatus.***LOADING***"**>Loading...</**label**>  
<**label \*ngIf="status===LoadingStatus.***ERROR***"**>Error getting persons</**label**>  
<**table \*ngIf="status===LoadingStatus.***LOADED***"**>  
 <**thead**>  
 <**tr**>  
 <**th**>Name</**th**>  
 <**th**>Vorname</**th**>  
 <**th**>EMail</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 <**tr \*ngFor="let *person* of persons"**>  
 <**td**>{{***person***.**name**}}</**td**>  
 <**td**>{{***person***.**vorname**}}</**td**>  
 <**td**>{{***person***.getEmail()}}</**td**>  
 </**tr**>  
 </**tbody**>  
</**table**>

## person-search.component.ts

**import** {Component, OnInit, ViewChild} **from '@angular/core'**;  
**import** {Person} **from '../shared/person'**;  
**import** {PersonService} **from '../shared/person.service'**;  
**import** {Observable} **from 'rxjs/Observable'**;  
**import** {LoadingStatus} **from '../shared/loading-status.enum'**;  
  
@Component({  
 **selector**: **'app-person-search'**,  
 **templateUrl**: **'./person-search.component.html'**,  
 **styleUrls**: [**'./person-search.component.css'**]  
})  
**export class** PersonSearchComponent **implements** OnInit {  
  
 **LoadingStatus** = LoadingStatus;  
  
 **searchTerm** = **''**;  
 **persons**: Person[] = [];  
 @ViewChild(**'searchForm'**) **form**;  
 **status**: LoadingStatus;  
  
 **constructor**(**private** personService: PersonService) {  
 }  
  
 ngOnInit() {  
 **this**.**form**.**control**.**valueChanges** .**debounceTime**(200)  
 .**do**(() => {  
 **this**.**persons** = [];  
 **this**.**status** = LoadingStatus.*LOADING*;  
 })  
 .**switchMap**(s => **this**.personService.search(**this**.**searchTerm**)  
 .**catch**(error => {  
 ***console***.log(error);  
 **this**.**status** = LoadingStatus.*ERROR*;  
 **return** Observable.*of*(undefined);  
 })  
 )  
 .subscribe(  
 data => {  
 **if** (data !== undefined) {  
 ***console***.log(data);  
 **this**.**persons** = data;  
 **this**.**status** = LoadingStatus.*LOADED*;  
 }  
 }  
 )  
 ;  
 }  
  
}

## app.module.ts

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;  
  
**import 'rxjs/add/observable/of'**;  
**import 'rxjs/add/operator/catch'**;  
**import 'rxjs/add/operator/debounceTime'**;  
**import 'rxjs/add/operator/do'**;  
**import 'rxjs/add/operator/map'**;  
**import 'rxjs/add/operator/switchMap'**;  
  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 PersonSearchComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

# Login

* ng generate component Login
* ng generate service shared/Login

## login.service.ts

**import** { Injectable } **from '@angular/core'**;  
  
@Injectable()  
**export class** LoginService {  
  
 **private loggedIn**: **boolean** = **false**;  
  
 **constructor**() {  
 }  
  
 login() {  
 **this**.**loggedIn** = **true**;  
 }  
  
 logout() {  
 **this**.**loggedIn** = **false**;  
 }  
  
 isLoggedIn() {  
 **return this**.**loggedIn**;  
 }  
  
}

## login.component.ts

**import** { Component, OnInit } **from '@angular/core'**;  
**import** {LoginService} **from "../shared/login.service"**;  
**import** {ActivatedRoute, Router} **from "@angular/router"**;  
  
@Component({  
 **selector**: **'app-login'**,  
 **templateUrl**: **'./login.component.html'**,  
 **styleUrls**: [**'./login.component.css'**]  
})  
**export class** LoginComponent **implements** OnInit {  
  
 **needsLogin**: **boolean**;  
  
 **constructor**(**private** loginService: LoginService, **private** route: ActivatedRoute, **private** router: Router) {  
 route.**params**.subscribe(params => {  
 **this**.**needsLogin** = params[**'needsLogin'**] === **'true'**;  
 });  
 }  
  
 ngOnInit() {  
 }  
  
 login() {  
 **this**.loginService.login();  
 **this**.router.navigate([**'home'**]);  
 }  
  
 logout() {  
 **this**.loginService.logout();  
 **this**.router.navigate([**'home'**]);  
 }  
  
 isLoggedIn() {  
 **return this**.loginService.isLoggedIn();  
 }  
  
}

## login.component.html

<**p**>  
 login works!  
</**p**>  
<**p \*ngIf="needsLogin"**>  
 Sie müssen sich anmelden  
</**p**>  
<**p**>  
 <**button (click)="login()" \*ngIf="!isLoggedIn()"**>Login</**button**>  
 <**button (click)="logout()" \*ngIf="isLoggedIn()"**>Logout</**button**>  
</**p**>

## app.module.ts

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppRoutingModule } **from './app-routing.module'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { HomeComponent } **from './home/home.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;  
  
**import 'rxjs/add/observable/of'**;  
**import 'rxjs/add/operator/catch'**;  
**import 'rxjs/add/operator/debounceTime'**;  
**import 'rxjs/add/operator/do'**;  
**import 'rxjs/add/operator/map'**;  
**import 'rxjs/add/operator/switchMap'**;  
**import** {LoginService} **from "./shared/login.service"**;  
**import** { LoginComponent } **from './login/login.component'**;  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 HomeComponent,  
 PersonSearchComponent,  
 LoginComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 AppRoutingModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService,  
 LoginService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

## app.component.html

<**div style="text-align**:**center"**>  
 <**h1**>  
 Welcome to {{ **title** }}!  
 </**h1**>  
</**div**>  
<**ul**>  
 <**li**>  
 <**h2**><**a routerLink="home"**>Home</**a**></**h2**>  
 </**li**>  
 <**li**>  
 <**h2**><**a routerLink="person-search"**>Personen-Suche</**a**></**h2**>  
 </**li**>  
 <**li**>  
 <**h2**><**a routerLink="login"**>Login</**a**></**h2**>  
 </**li**>  
</**ul**>  
  
<**router-outlet**></**router-outlet**>

## app-routing.module.ts

**import** { NgModule } **from '@angular/core'**;  
**import** { Routes, RouterModule } **from '@angular/router'**;  
**import** {HomeComponent} **from "./home/home.component"**;  
**import** {PersonSearchComponent} **from "./person-search/person-search.component"**;  
**import** {LoginComponent} **from "./login/login.component"**;  
  
**const** routes: Routes = [  
 {  
 **path**: **''**,  
 **redirectTo**: **'home'**,  
 **pathMatch**: **'full'** },  
 {  
 **path**: **'home'**,  
 **component**: HomeComponent  
 },  
 {  
 **path**: **'login'**,  
 **component**: LoginComponent  
 },  
 {  
 **path**: **'person-search'**,  
 **component**: PersonSearchComponent  
 },  
 {  
 **path**: **'\*\*'**,  
 **redirectTo**: **'home'** }  
];  
  
@NgModule({  
 **imports**: [RouterModule.*forRoot*(routes)],  
 **exports**: [RouterModule]  
})  
**export class** AppRoutingModule { }

# Berechtigung

Wie kann man das machen?

* ng generate service shared/AuthGard

## auth-gard.service.ts

**import** { Injectable } **from '@angular/core'**;  
**import** {ActivatedRouteSnapshot, CanActivate, Router, RouterStateSnapshot} **from "@angular/router"**;  
**import** {LoginService} **from "./login.service"**;  
  
@Injectable()  
**export class** AuthGardService **implements** CanActivate {  
  
 **constructor**(**private** router: Router, **private** loginService: LoginService) {  
 }  
  
 canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot): **boolean** {  
 **if** (!**this**.loginService.isLoggedIn()) {  
 **this**.router.navigate([**'login'**, { **needsLogin**: **true** }]);  
 }  
 **return true**;  
 }  
  
}

## app.module.ts

**import** { BrowserModule } **from '@angular/platform-browser'**;  
**import** { NgModule } **from '@angular/core'**;  
**import** { FormsModule } **from '@angular/forms'**;  
**import** { HttpClientModule } **from '@angular/common/http'**;  
  
**import** { AppRoutingModule } **from './app-routing.module'**;  
  
**import** { AppComponent } **from './app.component'**;  
**import** { HomeComponent } **from './home/home.component'**;  
**import** { PersonSearchComponent } **from './person-search/person-search.component'**;  
**import** { PersonService } **from './shared/person.service'**;  
  
**import 'rxjs/add/observable/of'**;  
**import 'rxjs/add/operator/catch'**;  
**import 'rxjs/add/operator/debounceTime'**;  
**import 'rxjs/add/operator/do'**;  
**import 'rxjs/add/operator/map'**;  
**import 'rxjs/add/operator/switchMap'**;  
**import** {LoginService} **from "./shared/login.service"**;  
**import** { LoginComponent } **from './login/login.component'**;  
**import** {AuthGardService} **from "./shared/auth-gard.service"**;  
  
@NgModule({  
 **declarations**: [  
 AppComponent,  
 HomeComponent,  
 PersonSearchComponent,  
 LoginComponent  
 ],  
 **imports**: [  
 BrowserModule,  
 AppRoutingModule,  
 FormsModule,  
 HttpClientModule  
 ],  
 **providers**: [  
 PersonService,  
 LoginService,  
 AuthGardService  
 ],  
 **bootstrap**: [AppComponent]  
})  
**export class** AppModule { }

## app-routing.module.ts

**import** { NgModule } **from '@angular/core'**;  
**import** { Routes, RouterModule } **from '@angular/router'**;  
**import** {HomeComponent} **from "./home/home.component"**;  
**import** {PersonSearchComponent} **from "./person-search/person-search.component"**;  
**import** {LoginComponent} **from "./login/login.component"**;  
**import** {AuthGardService} **from "./shared/auth-gard.service"**;  
  
**const** routes: Routes = [  
 {  
 **path**: **''**,  
 **redirectTo**: **'home'**,  
 **pathMatch**: **'full'** },  
 {  
 **path**: **'home'**,  
 **component**: HomeComponent  
 },  
 {  
 **path**: **'login'**,  
 **component**: LoginComponent  
 },  
 {  
 **path**: **'person-search'**,  
 **component**: PersonSearchComponent,  
 **canActivate**: [AuthGardService]  
 },  
 {  
 **path**: **'\*\*'**,  
 **redirectTo**: **'home'** }  
];  
  
@NgModule({  
 **imports**: [RouterModule.*forRoot*(routes)],  
 **exports**: [RouterModule]  
})  
**export class** AppRoutingModule { }

# Library-Versionen fixieren

Welche Angular-Version verwendet unsere Applikation jetzt und in einem Jahr?

Warum soll man sicherstellen, dass sich die Version nicht ändert?

Wie kann man sicherstellen, dass sich die Version nicht ändert?

npm shrinkwrap