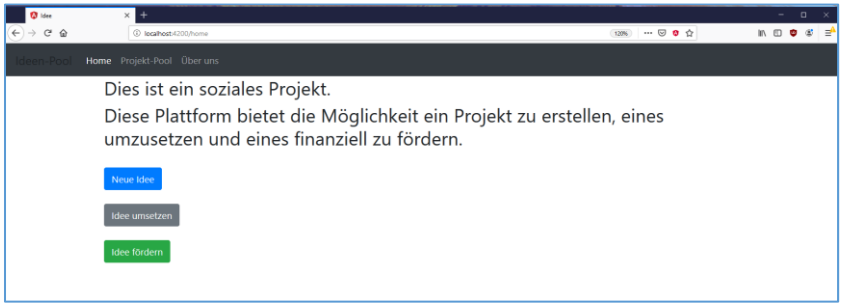
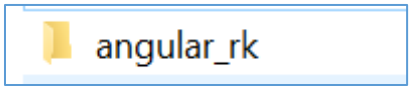
Projekt Rotes Kreuz



1)Neues Projekt erstellen inkl. routing

- Dazu erstelle am Beginn einen neuen Ordner, hier auf C: namens „angular_rk“



Mittels CMD arbeiten wir direkt in der Angular CLI, um die Grundstruktur des Projektes anzulegen.

- Navigiere dann in CMD (Benutzereingabe) mittels „cd ..“ in den neuen Ordner, zuerst einmal aus dem sich öffnenden bis hin zu C:

```
C:\Users\edi>cd ..

C:\Users>cd ..

C:\>
```

- Dann in den neuen Ordner hinein mit „cd angular_rk“

```
C:\>cd angular_rk

C:\angular_rk>
```

- Erstelle nun hier das neue Projekt namens „idee“ mittels dem Code:

  ng new idee –p rk –routing

  Man wird trotzdem nach „routing“ gefragt. Hier „Y“ eingeben.

```
C:\angular_rk>ng new idee -p rk -routing
? Would you like to add Angular routing? (y/N) y
```

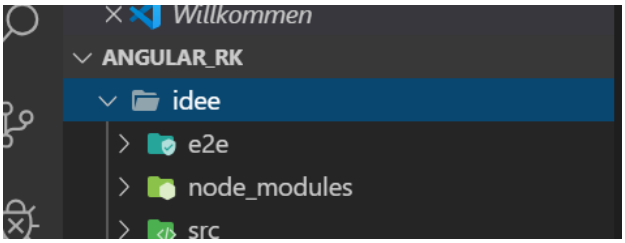
  - Der Parameter „–p rk“ sorgt dafür, dass das Präfix „rk“ (frei erfunden) eingetragen wird. Ansonsten wäre es wieder das Standardwort „app“. Auf die Dauer sieht das für Insider nicht professionell aus. ☺
  - Dann „CSS“ wählen

```
C:\angular_rk>ng new idee -p rk -routing
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
  SCSS   [ http://sass-lang.com/documentation/file.SASS_REFERENCE.html#syntax ]
  Sass   [ http://sass-lang.com/documentation/file.INDENTED_SYNTAX.html       ]
```

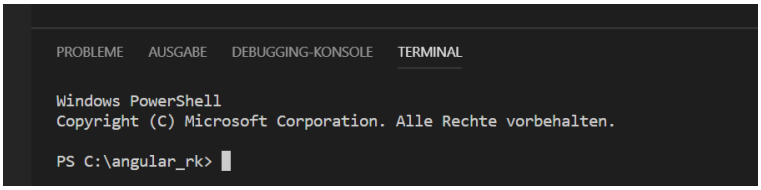
Beim Anlegen eines neuen Projekts inkl. „routing“ mit Hilfe der Angular CLI kann man die Routing-Konfiguration generieren lassen.

So wird ein eigenes Feature-Module mit dem Namen „AppRoutingModule“ angelegt. Die Definition der Routen sowie die Deklaration der verwendeten Komponenten erfolgt komplett in dieser Datei.

- Wenn alles fertig ist, schließe die CMD.
- Starte VS-Code und öffne den Ordner.



- Öffne „Neues Terminal“.



2)Erste Komponenten (Seiten) erstellen

Ziel:

- home, pool, about in der Navigation oben
- ansonsten folgende Seiten: ideeneu, ideeumsetzen, ideefoerdern, ideedanke

Daher muss für jede eine neue Komponente angelegt werden.

Für die Anlage einer neuen Komponente verwende wieder die Angular CLI (Terminal). Als Name wähle zuerst einmal „pool“.

Info:

- Dateinamen werden in „**Kebab-Case**“ geschrieben. Das bedeutet, dass sie aus Kleinbuchstaben bestehen und dass einzelne Wörter durch ein Minus getrennt sind.
- Komponenten werden automatisch im Ordner „src/app“ abgelegt.

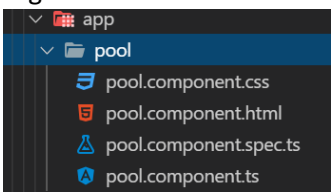
Man sollte sich im Hauptverzeichnis befinden, um folgenden Code im Terminal zu schreiben:

ng g c pool

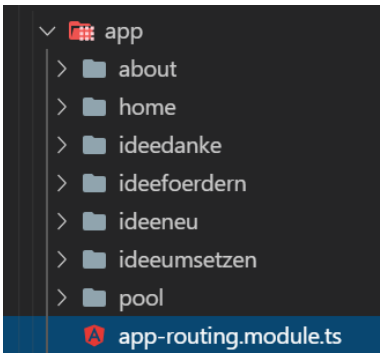
Natürlich muss man zuerst in das Projekt ☺ „idee“ wechseln:

```
PS C:\angular_rk> ng g component pool
The generate command requires to be run in an An
PS C:\angular_rk> cd idee
PS C:\angular_rk\idee> ng g component pool
```

Ergebnis:



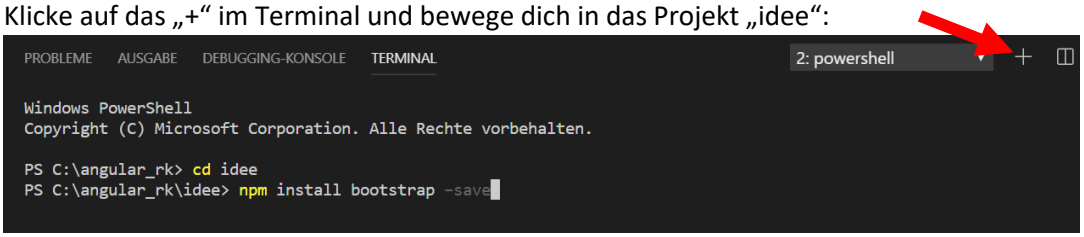
Erstelle danach alle andern Komponenten ebenso:



3)Bootstrap einbinden

Bootstrap wird, wie andere Bibliotheken auch, mit dem Paketmanager npm installiert.

Klicke auf das „+“ im Terminal und bewege dich in das Projekt „idee“:



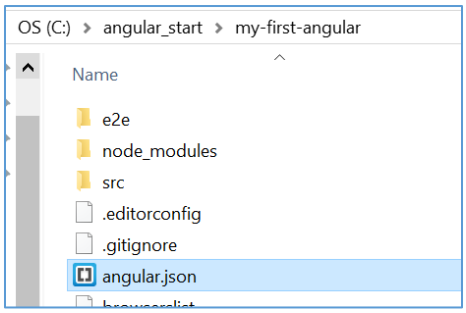
npm install bootstrap (ohne –save)

Dieser Befehl durchsucht standardmäßig das online zur Verfügung stehende npm-Repository und lädt Bootstrap in den Ordner „node_modules“.

```
C:\angular_start\my-first-angular>npm install bootstrap
```

Damit das CLI das Stylesheet von Bootstrap auch in die Anwendung einbindet, muss man sie in der Datei „.angular.json“ im Array „styles“ eintragen.

Öffne dafür den Ordner in der Windows Explorer Ansicht:

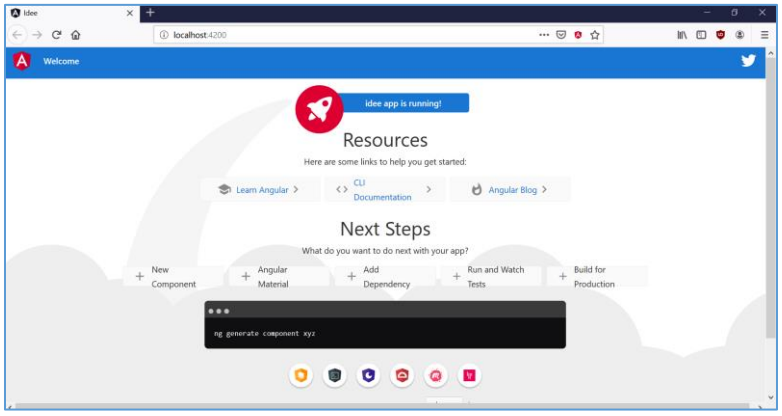


"node_modules/bootstrap/dist/css/bootstrap.min.css",

```
26      "styles": [
27        "node_modules/bootstrap/dist/css/bootstrap.min.css",
28        "src/styles.css"
29      ],
```

Nach einem erneuten Start der Anwendung mit „ng serve –open“ sollte das Browserfenster die Überschrift ohne Serifen präsentieren.

Ergebnis:



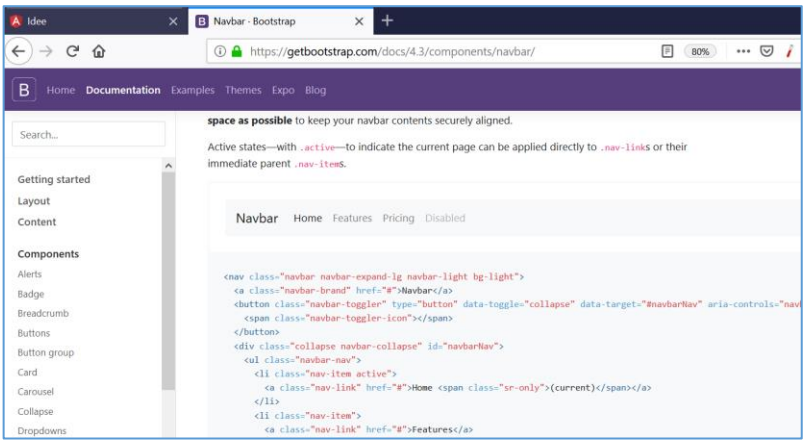
Welche Version sieht man in „package.json“:

```
21      "@angular/router": "~8.1.0",
22      "bootstrap": "^4.3.1",
23      "rxjs": "~6.4.0",
```

4)Startseite

Öffne die Datei „app.component.html“ und lösche den gesamten Inhalt.

Dann öffne die Website „www.getbootstrap.com“ und wähle „navbar“ um folgenden Code zu kopieren und in die Datei einzufügen.



Nur der Bereich mit den List-Einträgen <li> muss als **Angular-Code** geschrieben werden.

- Hier ist nämlich statt „href“ das „routerLink“ nötig.
- Verwende auch das „routerLinkActive“.
- Dafür muss man das „active“ bei ersten „nav-item“ entfernen.
- Die kann man auch entfernen:

```
 8
 9    outerLinkActive="active" >Home <span class="sr-only">(current)</span></a>
10
```

```
<li class="nav-item">
  <a routerLink='/home' routerLinkActive="active" class="nav-link">Home</a> </li>
<li class="nav-item">
  <a routerLink='/pool' routerLinkActive="active" class="nav-link">Projekt-Pool</a> </li>
<li class="nav-item">
  <a routerLink='/about' routerLinkActive="active" class="nav-link">Über uns</a> </li>
```

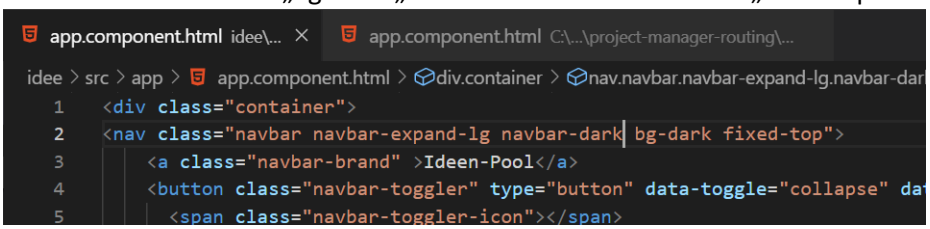
angular.json | app.component.html | pool.component.html

src > app > app.component.html > div.container > div.content

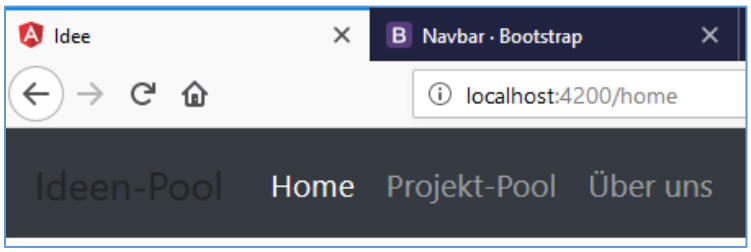
```
1   <div class="container">
2   <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top" >
3     <a class="navbar-brand" href="#">Bild</a>
4     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls=
5       <span class="navbar-toggler-icon"></span>
6     </button>
7     <div class="collapse navbar-collapse" id="navbarNav">
8       <ul class="navbar-nav">
9         <li class="nav-item active">
10          <a class="nav-link" router-link="/home" routerLinkActive="active" >Home </a>
11        </li>
12        <li class="nav-item">
13          <a class="nav-link" router-link="/pool" routerLinkActive="active">Projekt-Pool</a>
14        </li>
15        <li class="nav-item">
16          <a class="nav-link" router-link="/about" routerLinkActive="active">Über uns </a>
17        </li>
18      </ul>
19    </div>
20  </nav>
```

In der Zeile 3 für „brand“ kann das „href“ gelöscht werden.

In Zeile 2 ändere beide „light“ in „dark“ und daneben schreibe „fixed-top“:

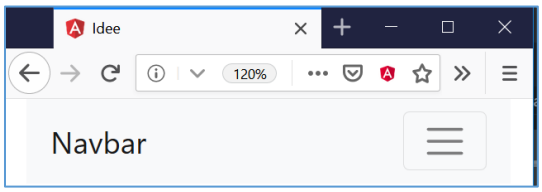


Ergebnis:

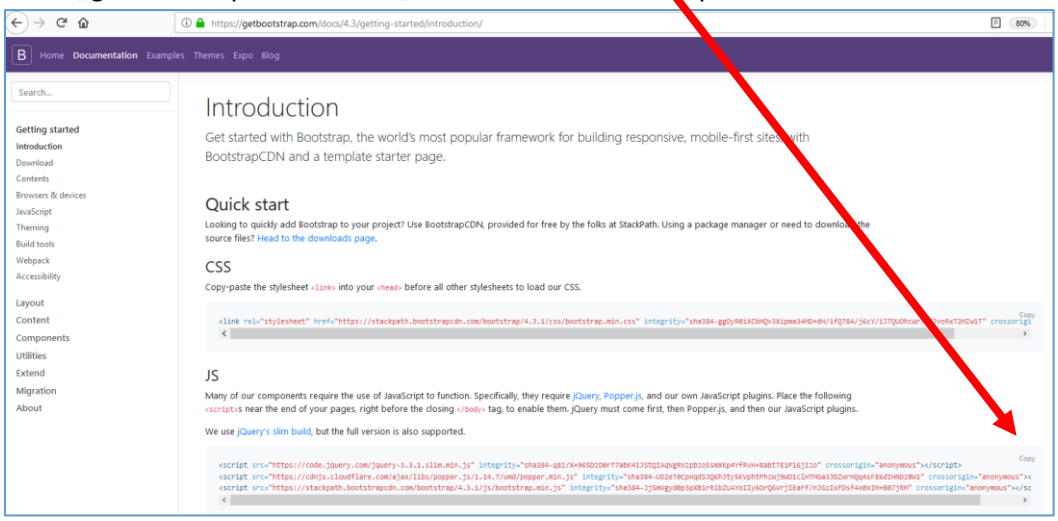


<u>Navigation verbessern:</u>

Auch das Design für Smartphones funktioniert:



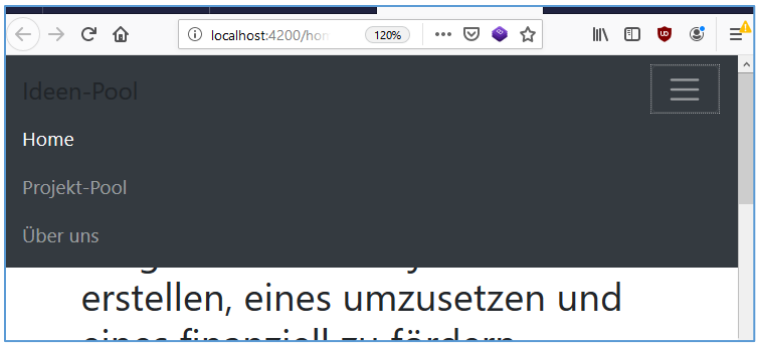
Aber der Button funktioniert noch nicht. Dazu benötigt man das „jQuery“ von „Bootstrap“. Öffne daher „getbootstrap.com“ in der „Documentation“ und kopiere die 3 JS-Zeilen:



Füge diese in (und nur hier) die „index.html“ im <head> Bereich ein:

```
about.component.html   index.html
idee > src > index.html > html > head > script
1   <!doctype html>
2   <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Idee</title>
6     <base href="/">
7
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <link rel="icon" type="image/x-icon" href="favicon.ico">
10    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7
11  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="
12  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha
13  </head>
14  <body>
```

Ergebnis:



Nun soll unterhalb dieser Navigation der **dynamische Teil** vorbereitet werden:

- Die Navigation bleibt auf allen Seiten bestehen und wird immer angezeigt
- Der Teil darunter wird immer gewechselt, je nachdem, welcher Link angeklickt wird.

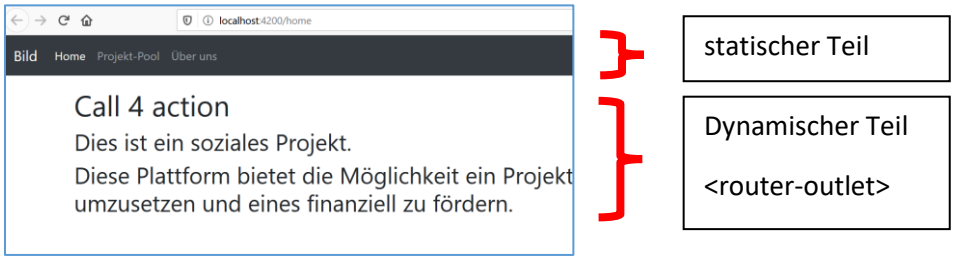
Der dynamische Teil besteht hier nur aus einem Platzhalter:

<router-outlet>

```
20  </nav>
21  <div class="content">
22
23    <router-outlet></router-outlet>
24
25  </div>
26  </div>
```

Dieser Platzhalter wird vom Router dynamisch durch die geladene Komponente ersetzt.

Das mit der Klasse „content“ dient hier nur der Information und ist typisch „Bootstrap“.

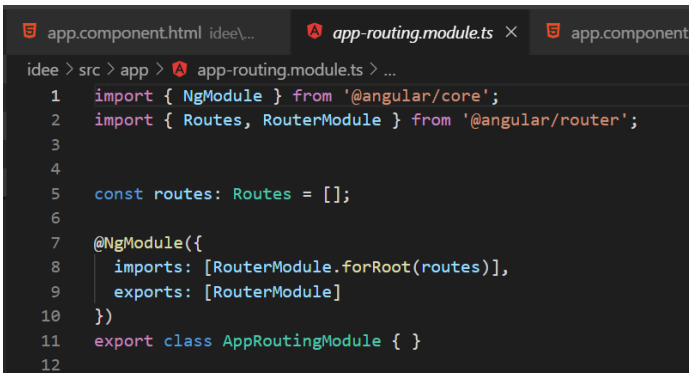


5)Routing erstellen

Die automatische Anlage des „routing“ bei der Erstellung des Projektes hat schon vieles automatisch angelegt. Ein gutes Beispiel dafür ist die Datei „app-routing.module.ts“

Öffne die Datei „app-routing.module.ts“.

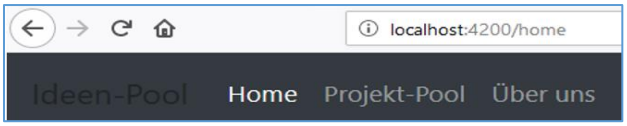
Grundlegende Zeilen sind schon vorhanden:



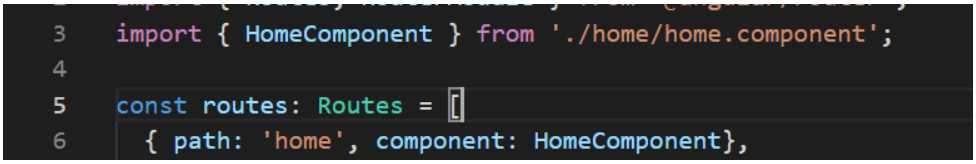
Es ist bereits im Modul (RouterModule) die Methode „forRoot“ enthalten, die eine Liste von Routenkonfigurationen als Parameter entgegennimmt.

5.1.)Routes erstellen

Es müssen drei Routen erstellt werden, nämlich für die neue Home, die Pool-Liste und die About-Site.

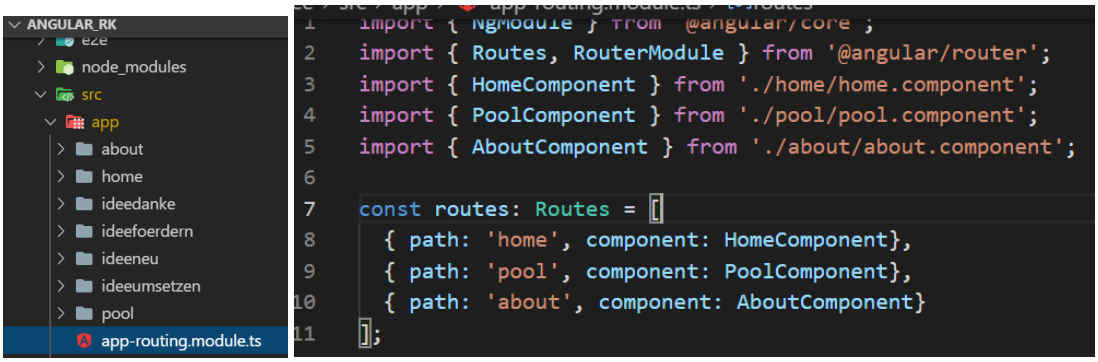


- Erstelle in den eckigen Klammern von der Zeile 5 den ersten Pfad:
  {path: 'home', component: HomeComponent},
  Es wird automatisch der passende Import erstellt. Siehe Zeile 3:
  TIPP:
  Wenn nach der „component:“ das „HomeC“ begonnen wird zu schreiben poppt ein Fenster auf, dass das Wort anbietet, übernommen zu werden. Benutze das, damit der „import“ auch wirklich automatisch folgt.



  Kopiert man dann diese „home“ Zeile, kann man nur das „home“ ausbessern und bei dem Namen der „component:“ wieder die Automatik verwenden, zum Erstellen der anderen Pfade:

- Erstelle auch die beiden anderen Pfade. Man kann links in der Übersicht sich nochmals die Namen als Erinnerung ablesen:



Noch zu berücksichtigen:

Beim Start fehlt die Route, weil ja nicht z.B. „home“ eingegeben ist, sondern nur die Start-URL, wie z.B. „localhost:4200“. Aber ein „RouterOutlet“ darf nie leer sein.
Daher muss man eine neue Route anlegen, in der der „path“ leer ist. In der legt man dann fest, dass der Root-Pfad auf die URL /home weiterleiten soll, damit der Nutzer sofort zur Startseite gelangt.
Daher benötigt man eine Routenweiterleitung.

- Man verwendet die Eigenschaft „redirectTo“ und gibt einen Pfad zur Weiterleitung an.
- pathMatch: ‘full‘ – damit greift die Route sicher

Diese Route füge vor der ersten ein.

{path: '', redirectTo: 'home', pathMatch: 'full'},

```
6
7   const routes: Routes = [
8     { path: '', redirectTo: 'home', pathMatch: 'full'},
9     { path: 'home', component: HomeComponent},
```

- Speichern.

6)Home-Seite befüllen

Nun können die erstellten Seiten einmal vorsorglich mit etwas Text befüllt werden.

6.1.) Home Site

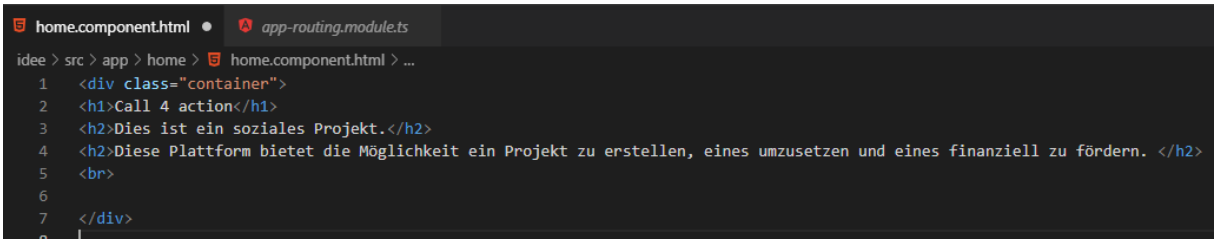
Öffne home.components.html .
Ziel:

- Willkommenstext
- 3 Buttons

Verwende das bekannte Bootstrap-Framework.

Code:

```
<div class="container">
<h1>Call 4 action</h1>
<h2>Dies ist ein soziales Projekt.</h2>
<h2>Diese Plattform bietet die Möglichkeit ein Projekt zu erstellen, eines umzusetzen und eines finanziell zu fördern. </h2>
<br>
</div>
```



Darunter sollen 3 Buttons folgen, die auf bestimmte Seiten verzweigen:

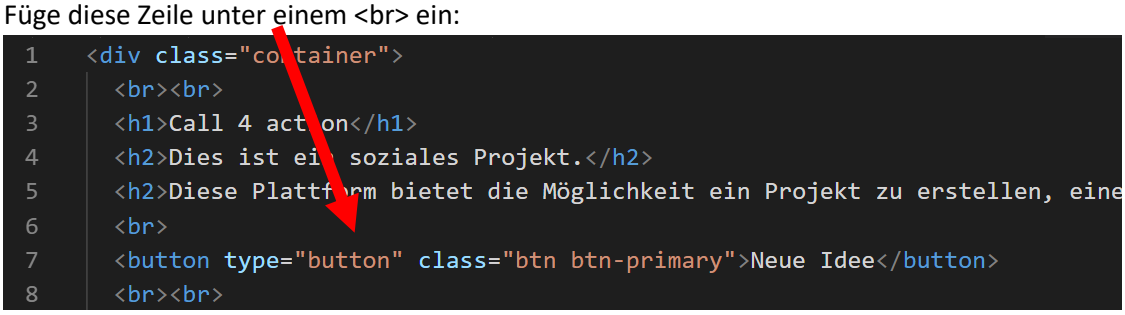
Bootstrap verwenden:

Da das Projekt mit Bootstrap läuft, kann man hier auf der Website www.getbootstrap.com unter „button“ gerne den Code organisieren:

Code:

```
<button type="button" class="btn btn-primary">Neue Idee</button>
```

Füge diese Zeile unter einem
 ein:



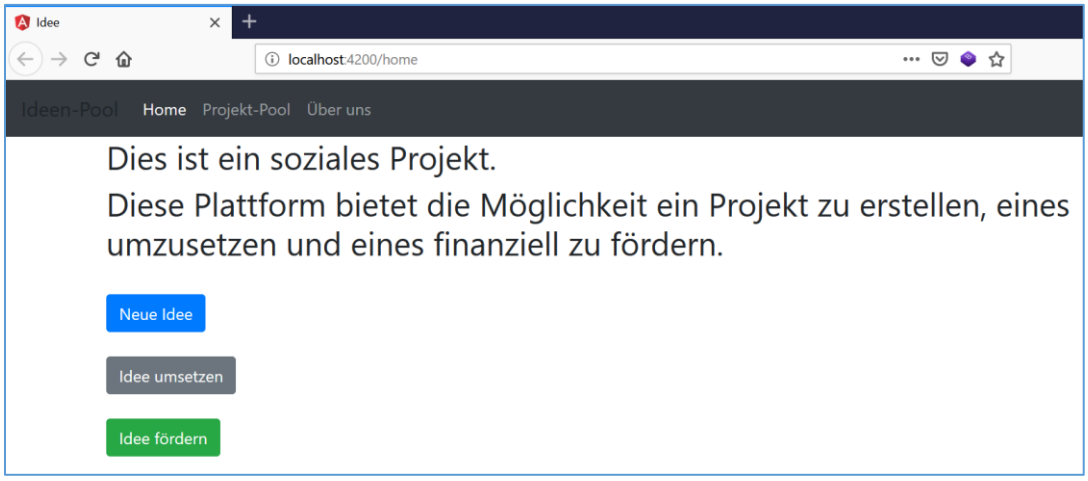
Ändere folgendes:

- Der Link (also die Route) fehlt noch, daher füge ein: die typische Angular Router-Direktive „routerLink“, und zwar auf die Komponente, die dazu passt „ideeneu“ mit Slash davor.
- Die routerLinkActive=“active“ einfügen
- Als Text „Neue Idee “

Code:
routerLink="/ideeneu" routerLinkActive="active"

Nun erstelle auch die beide anderen Buttons, mit einem
 dazwischen und anderer Farbe.

```
<br>
<button type="button" class="btn btn-primary" routerLink="/ideeneu" routerLinkActive="active">Neue Idee</button>
<br><br>
<button type="button" class="btn btn-secondary" routerLink="/ideeumsetzen" routerLinkActive="active">Idee umsetzen</button>
<br><br>
<button type="button" class="btn btn-success" routerLink="/ideefoerdern" routerLinkActive="active">Idee fördern</button>
</div>
```



Info: RouterLinkActive:

Styling des aktiven Links in der Navigationsleiste. Angular stellt hier eine bequeme Lösung zur Verfügung. So kann man die Klasse „active“ zu einem Listenelement hinzufügen, wenn dieser Link aktiv ist. Man verwendet folgenden HTML-Code

```
<li routerLinkActive="active">
    <a routerLink='/home' class="link">Home</a
</li>
```

Somit ist der aktive Menüeintrag farblich hervorgehoben.

https://www.youtube.com/watch?v=Nehk4tBxD4o ab ca. 9:22 Minuten

6.2.) Button Navigation

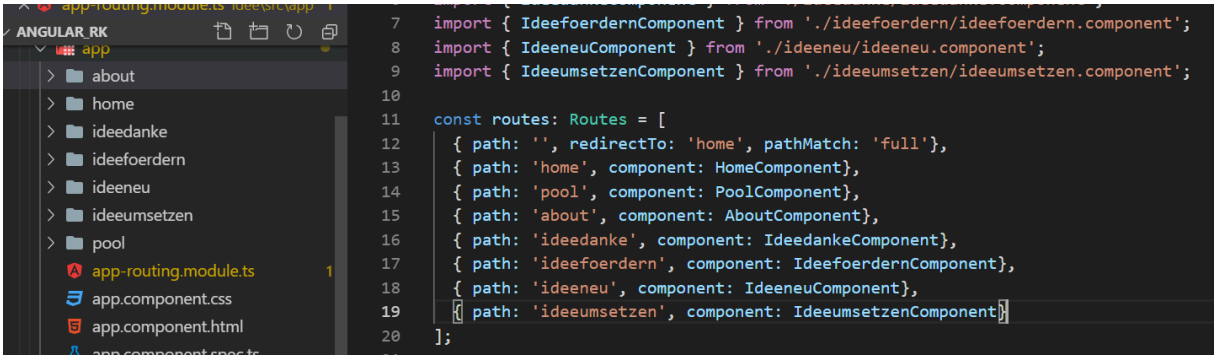
Die Navigation zu diesen Buttonlinks muss aber erst erstellt werden.

In der „app-routing.module.ts“ muss man nun die fehlenden Routes für die Unterseiten der Buttons anlegen. Zusätzlich erstelle hier auch die noch nicht aktuell benötigte „ideedanke“ – Route für später.

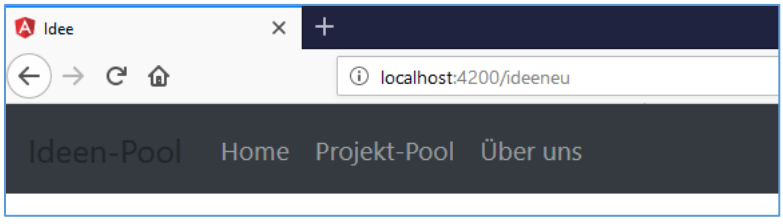
Nach der „about“ ist ein Beistrich zu erstellen, und dann die erste Route anzulegen (Zeile 13). Der Import (Zeile 6) erstellt sich dann von selbst:

```
6   import { IdeedankeComponent } from './ideedanke/ideedanke.component';
7
8   const routes: Routes = [
9     { path: '', redirectTo: 'home', pathMatch: 'full'},
10    { path: 'home', component: HomeComponent},
11    { path: 'pool', component: PoolComponent},
12    { path: 'about', component: AboutComponent},
13    { path: 'ideedanke', component: IdeedankeComponent}
14  ];
```

Erstelle alle drei anderen ebenfalls. Welche sieht man ja in der Darstellung der Komponenten links:



Die Verlinkung funktioniert dann sofort:



7)Seiten befüllen (HTML)

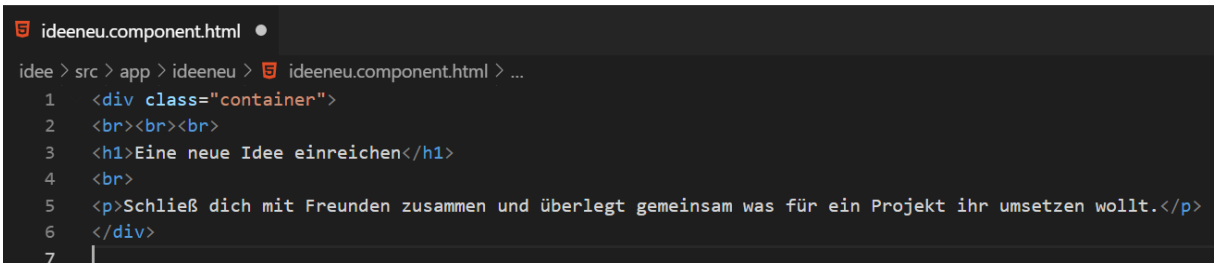
Dazu muss man sich in die betreffenden HTML Seiten der Komponenten stellen.

7.1.) Idee Neu

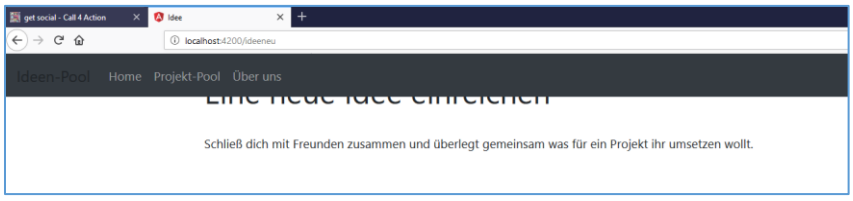
Öffne daher „ideeneu.component.html“.

Gib folgenden Code ein:

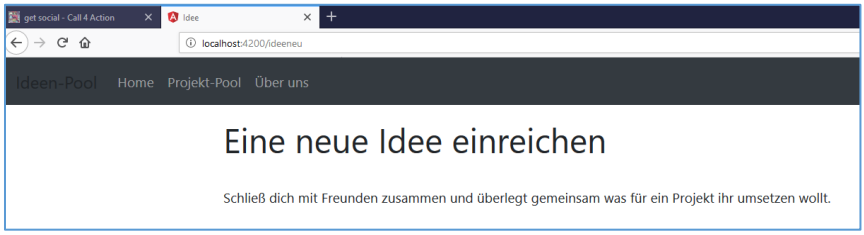
```
<div class="container">
<br>
<h1>Eine neue Idee einreichen</h1>
<br>
<p>Schließ dich mit Freunden zusammen und überlegt gemeinsam was für ein Projekt ihr umsetzen wollt.</p>
</div>
```



Ergebnis im Browser: nicht ideal, da die Navigation fix angezeigt wird und der Text verdeckt wird. Daher muss man einige <Breaks> erstellen.



Besser: mit 3

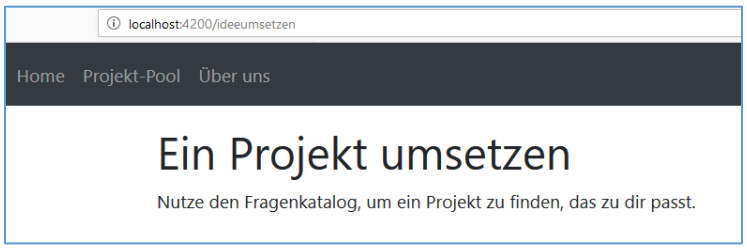


7.2.) Idee umsetzen

Öffne „ideeumsetzen.component.html“ und schreibe:

```
<div class="container">
  <br><br><br>
  <h1>Ein Projekt umsetzen</h1>
  <p>Nutze den Fragenkatalog, um ein Projekt zu finden, das zu dir passt.</p>
</div>
```

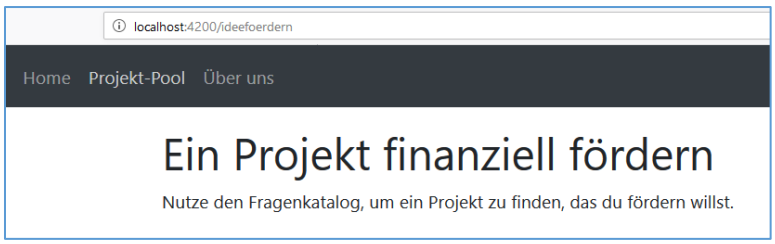
Ergebnis:



7.3.) Idee fördern

Öffne „ideefoerdern.component.html“ und schreibe:

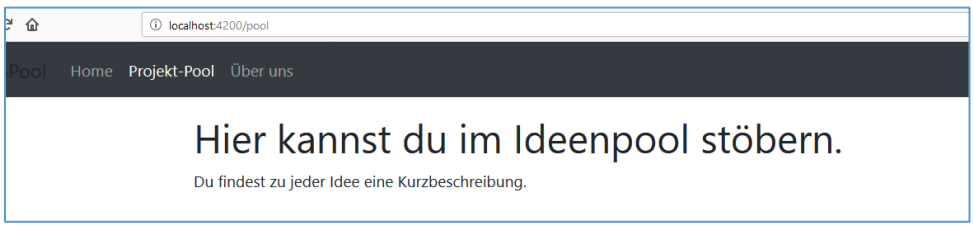
```
<div class="container">
  <br><br><br>
  <h1>Ein Projekt finanziell fördern</h1>
  <p>Nutze den Fragenkatalog, um ein Projekt zu finden, das du fördern willst.</p>
</div>
```



7.4.) Projekt Pool

Öffne die Datei „pool.component.html“.

```
<div class="container">
  <br><br><br>
  <h1>Hier kannst du im Ideenpool stöbern.</h1>
  <p>Du findest zu jeder Idee eine Kurzbeschreibung.</p>
</div>
```



7.5.) Über uns

Öffne „about.component.html“.

about.component.html

idee > src > app > about > about.component.html > ...

```
1  <div class="container">
2    <br><br><br>
3    <h1>Über uns</h1>
4    <p>Wir sind das Team aus der 5CK DigBiz-Mistelbach</p>
5  </div>
```