Shop: Datenbank und Service-Provider

Inhalt:

1. http-Modul für Datenbank-Verbindung
2. Service-Provider ist vorhanden, aber daher umschreiben für API

1)http-Modul für Datenbank-Verbindung

Theorie:
Das HttpClient Modul benötigt man, um http **POST, GET, PUT und DELETE requests** machen zu können. Dadurch wird es möglich Daten zum API (Verbindung zur Datenbank) zu senden bzw. zu erhalten.

API calls, die das http-Client-Modul nutzen sind asychron. Dabei wird mit modernen JavaScript API-Elementen gearbeitet, nämlich

- Promises
- Observables

Promises
Ein „promise“ kann 3 Zustände haben, nämlich pending (warten), fulfilled und rejected.

Observables
ist der neuere Standard ab Ionic5 und kann mehr Features als promises.

Registriere HttpClientModule in AppModule

Öffne den Ordner in VisualStudioCode.

Öffne bzw. schreibe im geöffneten internen Terminal:

Es sollen 2 Features nachinstalliert werden.

- Um http requests machen zu können:

  npm install rxjs

  npm install -save rxjs-compat

Öffne die „app.module.ts“

Um mit diesem http-Modul arbeiten zu können muss man es importieren.

app.module.ts - erstelle den Import von HttpClientModule UND vom Storage:

```
import { HttpClientModule } from '@angular/common/http';
```

```
9   import { HttpClientModule } from '@angular/common/http';
```

Füge dazu auch in den „imports“ dieser nach einem Beistrich ein: In Ionic8 wird es als „depreciated“ bezeichnet, also als veraltet. Es geht aber trotzdem.

```
13    imports: [BrowserModule, IonicModule.forRoot(),
14      AppRoutingModule, HttpClientModule],
```

2)Service Provider erstellen

Um an einer zentralen guten Stelle die Ausgabe zu organisieren, sollte dies immer in einem vorhandenen Service durchgeführt werden.

Erstelle im integrierten Terminal einen Provider ein Service.
Es gibt dafür **2 Varianten:** entweder

1. Gib den Code ein:

   ionic g service shopapi

   ODER:

2. lässt man IONIC für sich arbeiten – es werden die Auswahlmöglichkeiten angeboten:

   ionic g

Auswahl mit Pfeiltasten auf „service“ stellen

```
PROBLEME   AUSGABE   DEBUGGING-KONSOLE   TERMINAL

? What would you like to generate?
  page
  component
> service
  module
  class
  directive
  guard
(Move up and down to reveal more choices)
```

oder selbst eingeben:

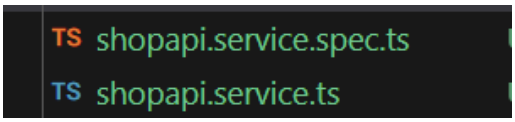
1. ionic generate service

```
PS D:\ionic_start\erstesApp> ionic generate service
```

Dann wird nach dem Namen gefragt. Gib ein

shopapi

Dieser Name wird automatisch folgendermaßen umgesetzt und in die 2 neue Dateien übernommen:



- app.module.ts

Dieses Service muss man NICHT in app.modules.ts importieren!

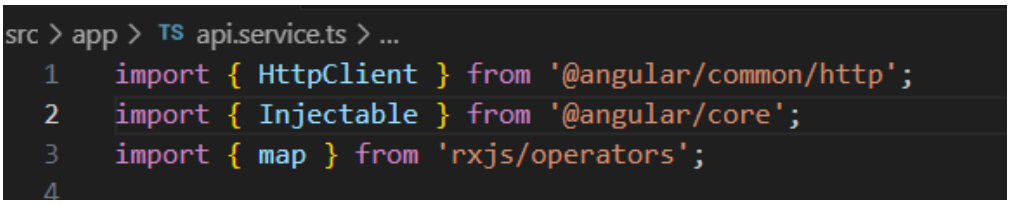
Erstelle http Service mit RxJS Observables

Öffne „shopapi.service.ts“.

Schreibe in die Zeile 2 und 3 folgenden IMPORT:

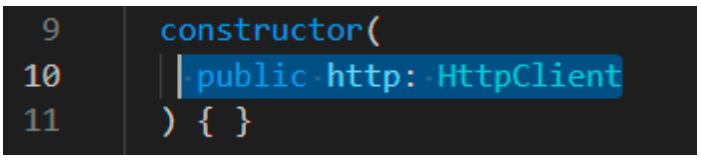
import {HttpClient, HttpHeaders} from '@angular/common/http';

import { map } from 'rxjs/operators';



Neu 2024:

Dann in den Constructor:



Man könnte auch hier wieder zuerst die Methode im constructor schreiben und sich automatisiert den Import vorschlagen lassen und dann annehmen, um den Import in Zeile 2 nicht selbst schreiben zu müssen.