

Fachrichtung Informatik

Schuljahr 2020/2021

DBI-Projekt

Angular

Ausgeführt von:

Sebastian Lindinger, Johannes
Huemer und Maximilian Lehner

Betreuer/Betreuerin:

DI Christoph Gerstberger

Inhaltsverzeichnis

4710 Grieskirchen, Parzer Schulstraße 1	1
.....	1
1 Erstes Angular-Projekt erstellen	3
1.1 Angular installieren	3
1.2 Projekt erstellen	3
2 Geschichte und Gründer	5
3 Vor- und Nachteile von Angular	5
3.1 Vorteile	5
3.2 Nachteile	7
4 Rest-Calls.....	8
5 Besondere Features.....	9
5.1 Angular Material	9
6 Community.....	9
7 Websites die mit Angular erstellt wurden.....	10
7.1 YouTube für PlayStation.....	10
7.2 Netflix.....	10
7.3 IBM	10
8 Angular Lernkurve	11
9 Persönliche Meinung zu Angular.....	11
10 Quellen.....	12

1 Erstes Angular-Projekt erstellen

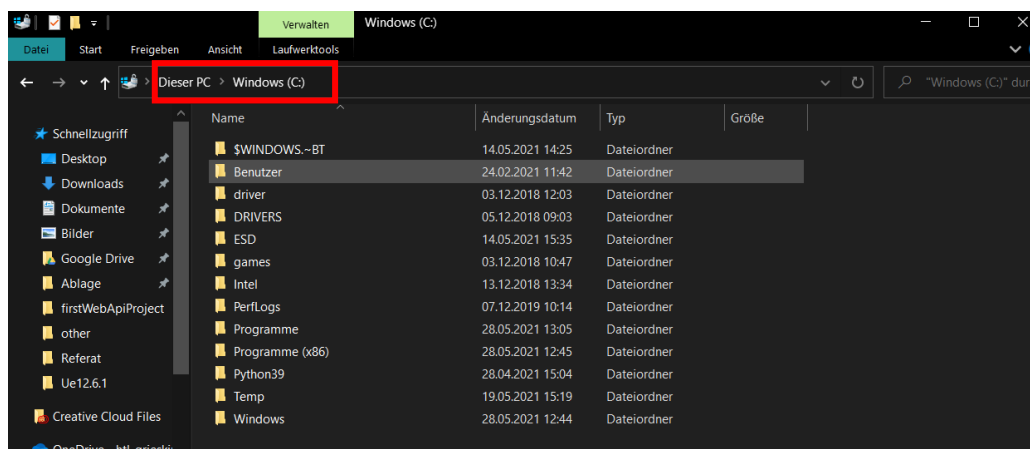
1.1 Angular installieren

Um Angular zu installieren, geben Sie den Befehl „npm install -g @angular/cli“ im Terminal ein. Beachten Sie dabei, dass Sie NodeJS installiert haben müssen, damit dieser Befehl funktioniert. Eine Anleitung zur Installation davon finden Sie hier: (<https://docs.microsoft.com/de-de/windows/dev-environment/javascript/nodejs-on-windows>).

```
Eingabeaufforderung
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\Johannes Huemer>npm install -g @angular/cli
```

1.2 Projekt erstellen

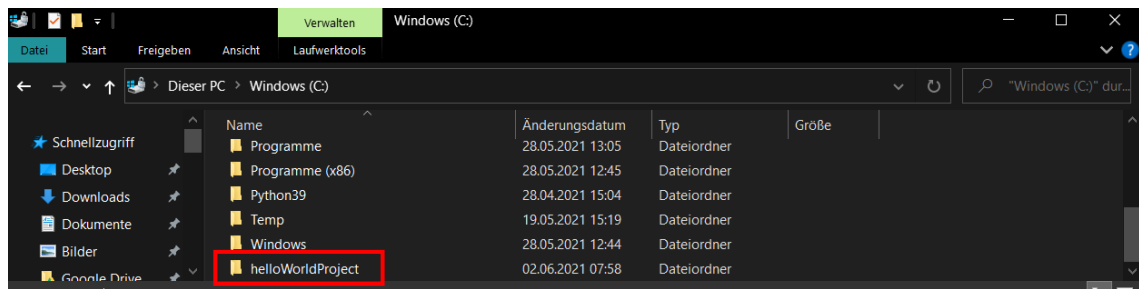
Navigieren Sie in das Verzeichnis, in dem Sie das neue Angular-Projekt erstellen wollen und klicken Sie in Explorer auf den Pfad.



Geben Sie dort „cmd“ ein und drücken Sie die Enter-Taste. Es sollte sich nun das Terminal öffnen. Geben Sie dort den Befehl „ng new helloWorldProject“ ein um ein Projekt mit dem Namen helloWorldProject zu erstellen.

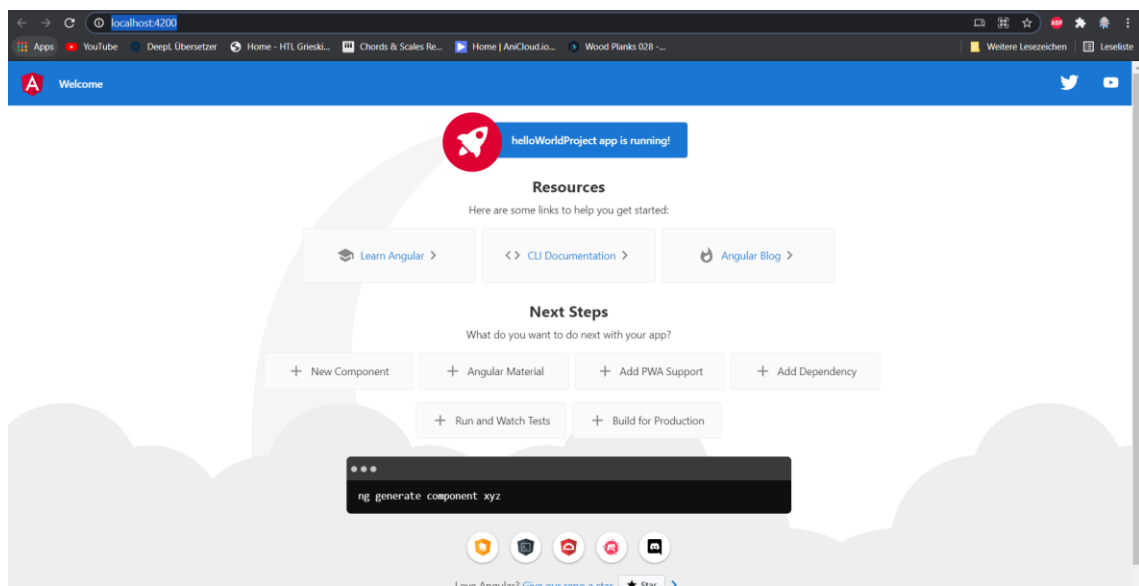
```
C:\>ng new helloWorldProject
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE helloWorldProject/angular.json (3711 bytes)
CREATE helloWorldProject/package.json (1220 bytes)
```

Wie Sie im nachfolgenden Bild sehen können, wurde das Projekt erfolgreich erstellt.



Navigieren Sie nun in das Projekt-Verzeichnis und geben Sie den Befehl „ng serve –open“ ein, um das Projekt zu starten und zugleich im Browser zu öffnen.

```
C:\>cd helloWorldProject  
C:\helloWorldProject>ng serve --open
```



Sie können das Projekt nun mit einer Entwicklungsumgebung Ihrer Wahl bearbeiten. Wir haben zum Beispiel Visual Studio Code verwendet.

2 Geschichte und Gründer

Das Framework wurde von Misko Hevery im Jahr 2009 begründet. Die anfängliche Idee war es, Webdesigner dabei zu unterstützen, ein wenig mehr HTML in deren Code unterzubringen, so dass auch kleine statische Seiten, mehr Funktionalitäten bedienen können. Zum Beispiel eine kleine Pizzaladen-Website, welches ein Pizza-Bestellsystem über einfache HTML Tags einfügen kann.

Misko wechselte dann zu Google als Arbeitgeber, wo er in seiner freien Zeit weiter an AngularJS arbeitete. Zu diesem Zeitpunkt war jedoch der Gedanke, dieses Framework auch für grössere Seiten zu nutzen. Auch Google überzeugte die Idee mit der Zeit und baute dann ein Angular Team auf.

3 Vor- und Nachteile von Angular

3.1 Vorteile

- **Teil des MEAN Stack:** Angular ist Teil des MEAN Technologie Stacks. MEAN steht für (M) MongoDB, (E) EmberJS, (A) AngularJS an noSQL. Mit dem MEAN Stack lassen sich Web Applikationen schreiben, unter der Nutzung einer Sprache, nämlich JavaScript. Der Vorteil hierbei ist, dass alles was entwickelt wird, auf jeder Stufe (Frontend, Backend, Datenbank) auf dieser einen Sprache basiert. Zudem ist die komplette Technologie Open Source und damit Lizenzfrei.
- **MVC:** MVC steht für Model Viewer Controller und ist ein Ansatz, um die Programmierlogik (Backend) und die Designlogik (Frontend) voneinander zu trennen. Es ist hiermit einfacher, Programmierungen im Backend vorzunehmen, ohne unbedingt das Design der Applikation zu ändern. Angular folgt diesem MVC Ansatz. Üblicherweise muss man im MVC Ansatz, die Anwendung in die verschiedenen Komponenten (Model/ Viewer/ Controller) aufteilen und danach Code schreiben, um diese wieder zusammenzufügen. Das macht eine Menge Arbeit. In Angular kann man sich zumindest den Teil des Zusammenfügens sparen, da das Framework dies bereits automatisch ausführt.

- **Single Page Applikation:** Mit dem Framework lassen sich Single Page Applikationen schreiben. Hierbei wird die Seite nur einmal geladen und weitere Komponenten werden in die gleiche Seite geladen, sobald dies vom Nutzer aufgerufen werden. Hierdurch bekommt die Webanwendung das "Look and Feel" einer Desktopanwendung.
- **Grosse Teams können daran arbeiten:** Im herkömmlichen JavaScript Framework ist es fast unmöglich in grossen Teams zusammenzuarbeiten. Angular hat dieses Problem behoben und macht es möglich, dass viele Entwickler gleichzeitig daran Coden und Testen können, ohne sich dabei gegenseitig auf die Füsse zu treten.
- **Das Framework ist im Trend:** Google Trends hilft, unterschiedliche Begriffe in ihrer Popularität zu vergleichen. Stellt man Angular, KnockoutJS, EmberJS und reactJS in den direkten Vergleich auf Google Trends, dann sieht man den enormen Unterschied in der Popularität. Besonders ab 2013 geht die Kurve steil aufwärts.
- **Einfacheres Testing möglich:** Man kann das Framework von Anfang nutzen, um ein einfaches Testing durchführen. Man kann das sogenannte Unit Testing durchführen oder auch ein End-to-end Testing. Dabei können wenige Entwickler, als auch ein grosses Team involviert sein.
- **Stark im Zusammenhang mit Ruby on Rails oder ASP.NET:** Ruby on Rails oder ASP.NET Framework mit der C# Programmiersprache im Backend und Angular im Frontend ergänzen sich gut.

3.2 Nachteile

- **Wird nicht in alten Browsern unterstützt:** Besonders die neue Version AngularJS 2.0 wird ältere Versionen von Browsern nicht mehr unterstützen. Das ist eine Herausforderungen, besonders auch weil in vielen Grossunternehmen noch alte Versionen von Browsern genutzt werden.
- **Für kleinere Projekte gibt es alternative Frameworks:** Nicht alle Websites benötigen robuste Frameworks wie Angular. In vielen Fällen reichen einfachere leichtgewichtigere Frameworks wie Backbone.js.
- **Es ist groß und kompliziert:** Das Framework sollte eher von Profis, als von Anfängern bedient werden (oder zumindest braucht man einiges an Zeit sich darin einzuarbeiten). Zudem haben unterschiedliche Programmierer unterschiedliche Ansätze eine bestimmte Komponente umzusetzen. Dies macht es dann schlussendlich schwieriger unterschiedliche Komponenten von unterschiedlichen Entwicklern zu einer kompletten Lösung zusammenzufügen.
- **Schlechte Skalierbarkeit:** Wenn das Projekt mit der Zeit größer wird, dann wird es in vielen Notwendig, die Programmierung auf Grundlage der neuen Anforderung zu erstellen, da der frühere Ansatz für die neuen Anforderungen höchstwahrscheinlich nicht genügen werden.
- **Viele Probleme werden erst mit Angular 2.0 gelöst:** Viele der Nachteile werden erst mit der neuen Version dieses interessanten Frameworks behoben.

4 Rest-Calls

Für die jeweiligen Calls haben wir uns eine Hilfsklasse erstellt die folgendermaßen aussieht.

```
readonly ROOT_URL;  
  
constructor(private http:HttpClient) {  
    this.ROOT_URL = 'http://localhost:8080/ameisenbert.shop/'  
}  
  
get(url: string){  
    return this.http.get(`${this.ROOT_URL}/${url}`);  
}
```

Das dient dazu, dass wir nicht jedes Mal den ganzen Call ausschreiben müssen.

Ein beispielhafter Class könnte dann so aussehen:

```
products() {  
    return this.apiService.get('products');  
}
```

Der Aufruf der Methode sieht dann folgendermaßen aus:

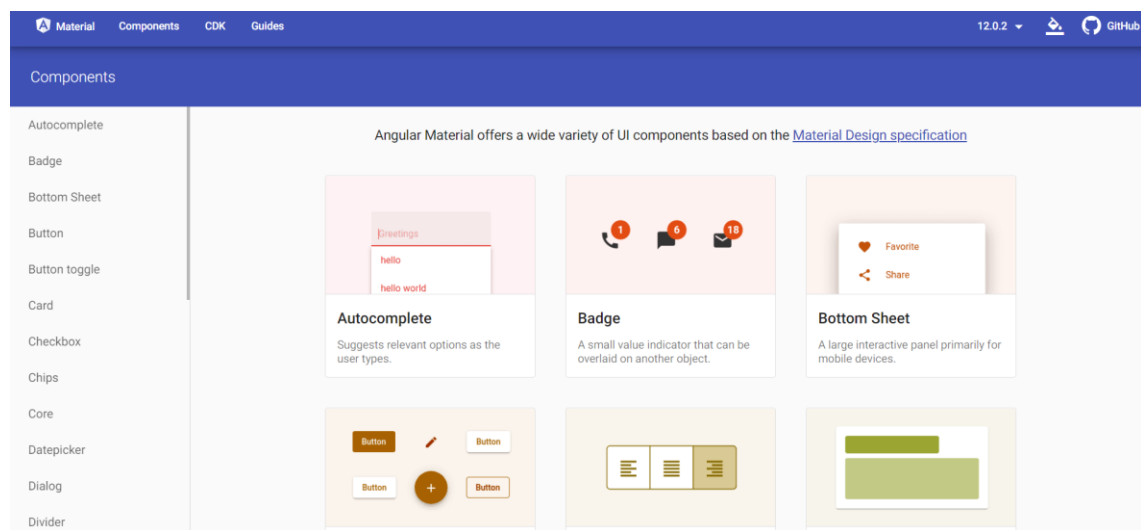
```
getAllProducts() {  
    this.httpservice  
        .products()  
        .subscribe((res) => {  
            console.log(res);  
            this.products = <Product[]>(<unknown>res);  
        });  
}
```


5 Besondere Features

5.1 Angular Material

Bei Angular Material gibt es viele verschiedene Vorlagen, die man Verwenden kann. Diese machen das Design der Website um einiges einfacher, da man dadurch nicht alles von Null aufbauen muss. Eine Anleitung zur Verwendung davon finden Sie unter: <https://material.angular.io/>

Die Website ist außerdem sehr übersichtlich und intuitiv gestaltet

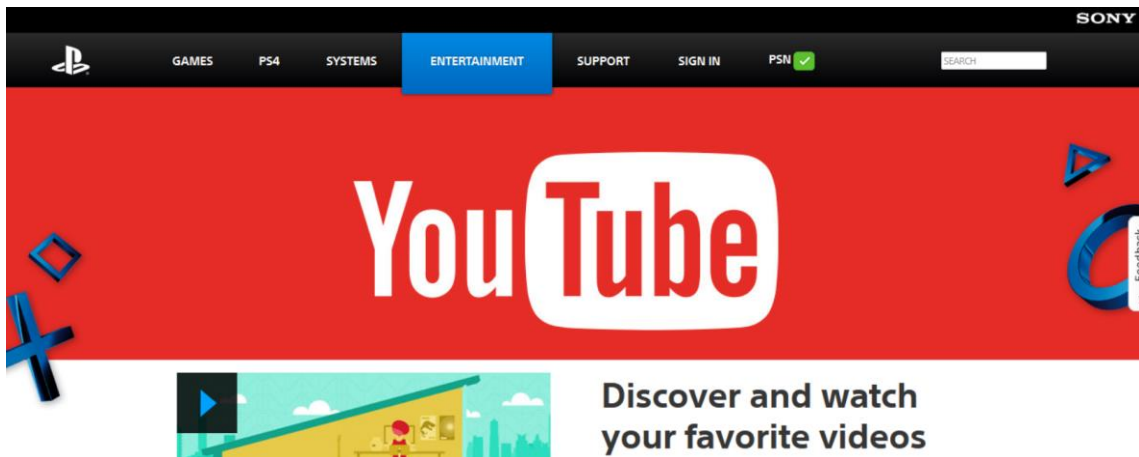


6 Community

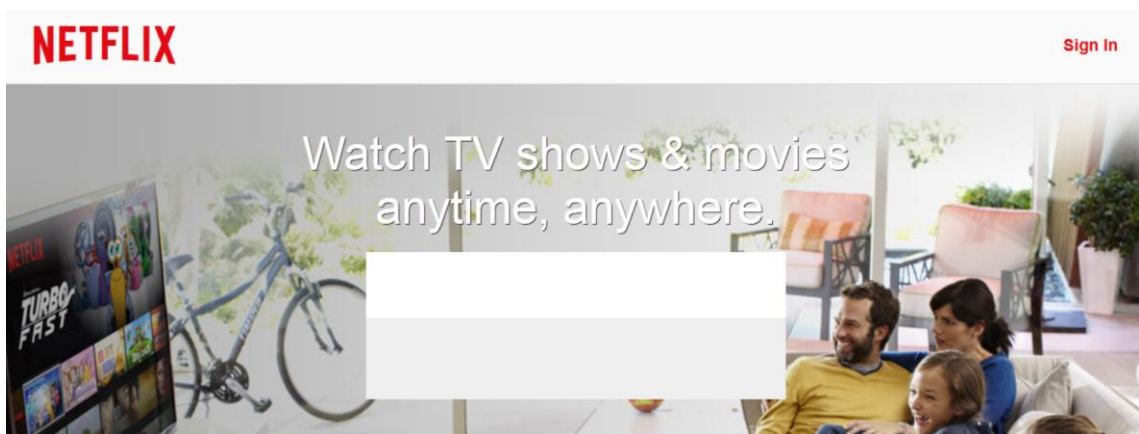
Da es Angular schon relativ lange gibt ist die Community dahinter entsprechend groß. Es ist zu jedem erdenklichen Problem eine Lösung im Netz zu finden und es gibt viele Anleitungen in Video- und Textform.

7 Websites die mit Angular erstellt wurden

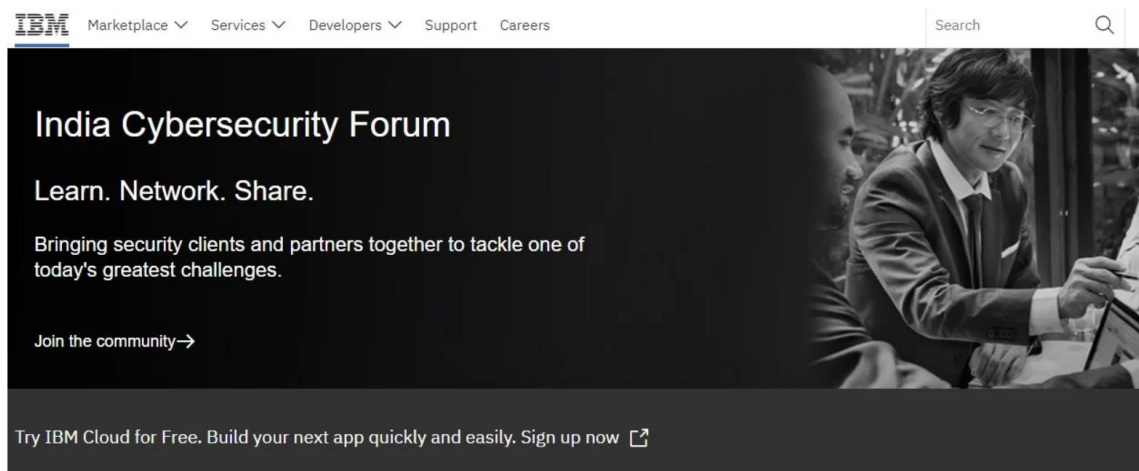
7.1 YouTube für PlayStation



7.2 Netflix



7.3 IBM



8 Angular Lernkurve

Die Lernkurve von Angular ist anfänglich relativ hoch. Es gibt jedoch viele Tutorials und nach kurzer Zeit ist die steile Lernkurve kein Problem mehr. Außerdem hatten wir einen Udemy-Kurs, der uns den Einstieg etwas erleichterte.

9 Persönliche Meinung zu Angular



Den Einsatz von Angular kann man nur empfehlen. Das Framework ist logisch aufgebaut, die Rest-Calls sind leicht zu implementieren und die Verwendung von Angular Materials macht den Website-Bau um vieles schneller und einfacher. Im Großen und Ganzen sind wir mit dem Framework **sehr zufrieden**.

10 Quellen

- Microsoft. (o. D.). Installieren von NodeJS unter Windows. Microsoft Documentation. Abgerufen am 2. Juni 2021, von <https://docs.microsoft.com/de-de/windows/dev-environment/javascript/nodejs-on-windows>
- Google. (o. D.). Angular Material. Angular Material. Abgerufen am 2. Juni 2021, von <https://material.angular.io/>
- Eduonix. (2019, 10. April). Top 20 websites and apps built with AngularJS. <https://blog.eduonix.com/web-programming-tutorials/top-15-websites-and-apps-built-with-angularjs/>
- Udemy. (o. D.). Online Courses - Learn Anything, On Your Schedule. Abgerufen am 2. Juni 2021, von <https://www.udemy.com/>