

Ruhr-Universität Bochum

Bachelorarbeit

---

# Entwicklung eines erweiterbaren Kommandozeileninterfaces zur Generierung von Web-Applikationen

---

Schriftliche Prüfungsarbeit  
für die Bachelor-Prüfung des Studiengangs Angewandte Informatik an der  
Ruhr-Universität Bochum

vorgelegt von  
Michael David Kuckuk

am  
Lehrstuhl für Informatik im Bauwesen  
Prof. Dr.-Ing. Markus König

Abgabedatum: 27. Juli 2021  
Matrikelnummer: 108 017 207 503  
1. Prüfer: Prof. Dr.-Ing. Markus König  
2. Prüfer: Stephan Embers, M. Sc.

## **Abstract**

The ABSTRACT is to be a fully-justified text following the title page. The text will be formatted in 12 pt, single-spaced Computer Modern. The title is “Abstract”, set in 12pt Computer Modern Sans Serif, centered, boldface type, and initially capitalized. Writing the abstract in English is mandatory even if the thesis itself is written in German. The length of the abstract should be roughly about 200 words but must not exceed 250 words.

As usual, the abstract should clearly summarize the aim, the background and the results of your thesis so that an interested reading can decide if he or she wants to further read your (interesting) written report. Also, the abstract should be kept simple. That is, do not include tables, figures or cross references in the abstract. You will have plenty of time in your thesis to explain things more clearly.

## **Erklärung**

Ich erkläre, dass das Thema dieser Arbeit nicht identisch ist mit dem Thema einer von mir bereits für eine andere Prüfung eingereichten Arbeit.

Ich erkläre weiterhin, dass ich die Arbeit nicht bereits an einer anderen Hochschule zur Erlangung eines akademischen Grades eingereicht habe.

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, habe ich unter Angabe der Quellen der Entlehnung kenntlich gemacht. Dies gilt sinngemäß auch für gelieferte Zeichnungen, Skizzen, bildliche Darstellungen und dergleichen.

## **Statement**

I hereby declare that except where the specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text.

---

Datum

---

Unterschrift

# Inhaltsverzeichnis

<b>1</b>	<b>Stand der Technik</b>	<b>1</b>
1.1	React . . . . .	1
1.2	Angular . . . . .	1
1.3	Vue . . . . .	2
1.4	Vergleich der Möglichkeiten zwischen Frameworks . . . . .	2

# 1 Stand der Technik

Um dieses Konfigurationsproblem zu automatisieren oder zumindest zu erleichtern, gibt es bereits verschiedene Lösungen. Momentan sind drei Frontend-Frameworks besonders beliebt (React, Angular und Vue) [1] und für jedes dieser Frameworks existiert jeweils ein Programm, was (unter anderem) zur initialen Erstellung von Projekten mit dem jeweiligen Framework empfohlen wird.

## 1.1 React

In der Dokumentation für React wird empfohlen, neue Projekte über das Command Line Interface (CLI) `create-react-app` (CRA) zu erstellen. Dieses kann per Node Package Manager (NPM) installiert werden und ist dann in der Lage, den Inhalt eines angegebenen Templates (oder des Standardtemplates) in ein angegebenes Verzeichnis zu kopieren. Daraufhin werden die benötigten Abhängigkeiten per NPM oder mittels eines anderen installierten Paketmanagers (wie z.B. Yarn) installiert.

Dritten ist es möglich, eigene Templates für CRA zu erzeugen, die dann wie Erstanbieteremplates zur Erzeugung des neuen Projekts genutzt werden können. Wenn man also mittels CRA ein Projekt erzeugen möchte, das bereits mit gewissen Bibliotheken oder Werkzeugen ausgestattet ist, muss dafür ein Template erstellt worden sein. Die Wahrscheinlichkeit, dass das jedoch passiert ist, sinkt mit der Spezifität der Wünsche.

## 1.2 Angular

Das Angular-Team stellt das sog. Angular-CLI zur Verfügung, das Entwickelnden beim Programmieren verschiedene wiederkehrende und repetitive Aufgaben abnehmen soll (wie z.B. die Erstellung und Einbindung neuer Komponenten). Neben diesen Aufgaben wird das Tool auch zur Erstellung neuer Angular-Projekte genutzt.

Der Befehl `ng new <project-name>` führt Nutzende in einen Dialog, bei dem einige Fragen zur gewünschten Konfiguration des Projektes gestellt werden. Es können vier Features konfiguriert werden und daraufhin wird das Projekt eingerichtet.

Im Gegensatz zu CRA sind hier die Möglichkeiten, die Nutzenden geboten werden, sehr eingeschränkt, da es nicht wie bei den Templates für Dritte die Möglichkeit gibt, neue Libraries mitsamt entsprechender Konfiguration in den Erstellungsprozess einzubinden. Dafür können sich Nutzende hier (im Rahmen der beschränkten angebotenen Optionen) eine beliebige Konfiguration aussuchen und sind nicht darauf angewiesen, dass jemand vor ihnen schon denselben Wunsch hatte. Außerdem wird so auch Anfänger:innen die Entdeckung und der Einstieg in neue Libraries erleichtert.

## **1.3 Vue**

Von den drei Frameworks bietet Vue in Bezug auf die Projekterstellung das umfangreichste CLI. Hier trifft man zunächst eine Vorauswahl von zehn Features, die man haben oder nicht haben möchte. Daraufhin werden zu den ausgewählten Features detailliertere Fragen gestellt. Insgesamt stehen einem durch dieses Tool über 20 verschiedene Libraries ohne weiteren Konfigurationsaufwand zur Verfügung.

## **1.4 Vergleich der Möglichkeiten zwischen Frameworks**

Alle drei Tools bieten Entwicklenden die Möglichkeit, eigene Erweiterungen zu erarbeiten und zu veröffentlichen. Im Falle von React muss dies in Form eines Templates geschehen. Da nur ein Template zur Erstellung eines Projektes genutzt werden kann, sind derartige Erweiterungen hier also exklusiv.

Bei Angular und React ist es jedoch möglich, auch Erweiterungen zu entwickeln, die zusätzlich zu anderen Optionen und Erweiterungen nutzbar sind. Für das Angular CLI kann man sogenannte Schematics entwickeln, die das Hinzufügen und Einbinden einer Bibliothek vollautomatisch übernehmen. Diese Schematics müssen aber leider nach der Installation ausgeführt werden und müssen insbesondere von Nutzenden entdeckt werden. Hierfür gibt es keine eigene Plattform o.ä. und die Auswahl der Schematics, die schon bei der Projekterstellung auswählbar sind, beschränkt sich auf Angular-interne Features (z.B. die Einfügung eines Routers). Selbst Libraries, die auch vom Angular-Team betreut werden (z.B. Angular Material) müssen später per Schematic nachgerüstet werden.

Im Rahmen des Vue CLI's ist es immerhin möglich, auch Libraries von Dritten direkt bei der Projekterstellung einzubinden. Allerdings ist auch hier die anfängliche Auswahl nicht erweiterbar. Dafür können, wie auch schon bei Angular, anschließend automatisch über Drittanbieterplugins neue Libraries heruntergeladen, importiert und demonstriert werden.

Tabelle

Außerdem bieten alle drei der genannten Tools keine Unterstützung für andere Frameworks. Daher ist beispielsweise der große Umfang der Features des Vue CLI's leider nicht in einem Angular-Projekt nutzbar.

Tabelle 1: Automatische und initiale Installierbarkeit verschiedener Features in Angular und Vue Projekten

Feature	Angular		Vue	
	Automatisch installierbar	Initial auswählbar	Automatisch installierbar	Initial auswählbar
TypeScript	wird erzwungen		✓	✓
Router	✓	✓	✓	✓
PWA-Support	✓	×	✓	✓
Linten	✓	×	✓	✓
Formatierer	✓	×	✓	✓
CSS Reset	×	×	×	×
CSS Präprozessor	✓	✓	✓	✓
Design Framework	✓	×	✓	×
State Management	wird erzwungen		✓	✓
Unit Testing Framework	✓	×	✓	✓
E2E Testing Framework	✓	×	✓	✓

## Literatur

- [1] Sacha Greif und Raphaël Benitte. Front-end Frameworks — State of JS 2020, 2020. [https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/#front\\_end\\_frameworks\\_experience\\_ranking](https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/#front_end_frameworks_experience_ranking) [Zugriff am 26. Juli 2021].