

Hauptstudie Umsetzung VA

Niklas Liechti

Inhaltsverzeichnis

1. Architektur	1
2. Technologien	2
2.1. Frontend	2
2.2. Backend	6
2.3. Database	6
2.4. Deployment	6
2.5. Buildsystem	6



1. Architektur



2. Technologien

2.1. Frontend

Aktuell ist das Frontend immer noch massiv von Javascript dominiert. Langsam kommen Konkurrenten wie [Web Assembly](<https://webassembly.org/>) hervor. Diese sind aber noch viel zu Jung und zu wenig verbreitet um momentan für eine Applikation ein zu setzten.

Daher habe ich mich dazu entschieden noch mit einem Javascript Framework weiter zu machen.

Frontend Javascript Frameworks sind seit Jahren im starken Wandel.

2.1.1. Framework

Im Frontend bereich habe ich von den "modernen" Frameworks nur Erfahrung in Angular.js, welches aber schon seit einigen Jahre eigentlich überholt ist.

Daher habe ich mich bei der Entscheidung an einen Blogpost von [dev.to](<https://dev.to/duomly/the-best-front-end-framework-to-learn-in-2019-dn7>) und Erfahrung von Arbeitskollegen angelehnt.



Angular 2+

Angular ist das nachfolge Framework von angular.js, welches weiterhin von Google maintained wird.

Mit Angular habe ich minimale Erfahrung durch einen 1 Monatigen einsatz in einem kleineren Projekt.

Wie unten ersichtlich ist das Lernen und aufsetzen eines Angular Projektes relativ komplex. Grundsätzlich ist es extrem gut einsetzbar für grössere Singlepage apps. Die vorgegebene Struktur ist ein grosser Vorteil für Projekte mit vielen Personen, was in diesem Projekt aber nicht relevant ist.



Figure 1. Angular



Vue.js

Vue.js ist das neuste aller Frameworks. Es ist sehr einfach zu lernen und



Figure 2. Vue.js



React.js

React.js ist eine möglichst schlank gehaltene Frontend Library, welche von Facebook maintained wird. Das Konzept ist wie der Name schon sagt reaktives Verhalten. Das heisst man beschreibt grundsätzlich welche Daten dargestellt werden sollen und das Framework kümmert sich dann um das "reagieren" auf changes des states.



Figure 3. React.js



2.2. Backend

Das Backend ist grundsätzlich seit jeher stabiler, da diese Plattform meist voll unter Kontrolle des betreibenden liegt. Diese Wahl wurde in den letzten Jahren aber auch immer flexibler, da mit Docker fast jede beliebige Betriebssystem Umgebung gebaut werden kann.

Die meiste Erfahrung habe ich selber mit Java EE Applikationen, welche aber hier ein zu grosses Kaliber wären.

Zusätzlich habe ich in letzter Zeit viel Positives über [Kotlin](<https://kotlinlang.org/>) gehört. Kotlin ist eine auf der JVM basierte Sprache, die nahe an Java ist, aber sehr viel Syntactic Sugar anbietet und das Entwickeln von genau solchen kleinen Applikationen extrem einfach macht.

2.2.1. Framework

Als Framework habe ich mir nach kurzem recherchieren und nach Feedback von Arbeitskollegen [Javalin](<https://javalin.io/>) ausgesucht.

Javalin ist ein leichtgewichtiges Microservice Framework, welches mit nativer Kotlin Unterstützung kommt.

2.2.2. Sprache

2.3. Database

2.4. Deployment

Da das Deployment der Applikation auf einem Standard

2.5. Buildsystem