



**Hochschule
Kaiserslautern**
University of
Applied Sciences

**Informatik und
Mikrosystemtechnik**
Zweibrücken

Fachhochschule Kaiserslautern
- Campus Zweibrücken -

Fachbereich für IMST

Projektbericht

**Erstellung einer progressiven Web-App zum Austausch von Textnachrichten in einem
Kontakt Netzwerk - Keep in Touch**

Autor: Alea I. Sauer - 870723
Eric Werner - 870019

Version vom: 22.05.2020

Betreuer: Prof. Dr. Michael Bender

Abstract

Keep in Touch

Im Zuge der Projektarbeit soll eine progressive Web-Applikation für Android und IOS erstellt werden, die es den Nutzern ermöglicht textbasierte Inhalte mit einer Kontaktliste zu teilen. Die App soll eine Registrierung sowie einen Login besitzen, bei der jedoch keine überflüssigen sensiblen Daten benötigt werden. Alle hinterlegten Daten werden in einem Backend verwaltet. Als registrierter Nutzer kann ich Textnachrichten schreiben, teilen und löschen. Die maximale Anzahl der Zeichen einer Textnachricht ist begrenzt. Nutzer können über den Nicknamen gefunden werden und zu einer Freundesliste hinzugefügt und entfernt werden. Alle Personen, die sich in der eigenen Kontaktliste der App befinden, können Inhalte von mir anzeigen lassen und lesen.

Für die Applikation werden wir ein benutzerfreundliches Ionic-Frontend, das vor allem auf mobile Nutzer abgestimmt ist, entwickeln. Erweitert wird es durch ein Backend zur Datenverwaltung. Das Backend wird über das Framework Spring Boot realisiert, welches in einer Docker-Container Umgebung läuft.

Inhaltsverzeichnis

Abstract	II
1 Einführung	1
1.1 Motivation	1
1.2 Aufgabenstellung	1
2 Frameworks	3
2.1 Docker	3
2.1.1 Verwendung	3
2.2 Spring Boot	4
2.2.1 Verwendung	4
2.2.2 Dependencies	4
2.2.3 Controller	5
2.3 Ionic	6
2.3.1 Verwendung	7
2.3.2 Ablaufdiagramm	7
3 Pages	8
3.1 Login	8
3.2 Registration	8
3.3 Messages	9
3.4 Follower	10
3.5 Profile	11
4 Installationsanleitung	12
4.1 Projektstruktur	12
4.2 Docker	12
4.3 Spring Boot	14
4.4 Ionic	16
4.4.1 Fehlerbehandlung	18
5 Projekt Ausführen	19
5.1 Start Backend	19
5.2 Start Frontend	19
5.3 Fehlerbehandlung	19
6 Ausblick	21
Literaturverzeichnis	IV

1 Einführung

Im Zuge der Projektarbeit soll eine progressive Web-Applikation für Android und IOS erstellt werden, die es den Nutzern ermöglicht textbasierte Inhalte mit einer Kontaktliste zu teilen. Die App soll eine Registrierung sowie einen Login besitzen, bei der jedoch keine überflüssigen sensiblen Daten benötigt werden. Alle hinterlegten Daten werden in einem Backend verwaltet. Als registrierter Nutzer kann ich Textnachrichten schreiben, teilen und löschen. Die maximale Anzahl der Zeichen einer Textnachricht sind begrenzt. Nutzer können über den Nicknamen gefunden werden und zu einer Freundesliste hinzugefügt und entfernt werden. Alle Personen, die sich in der eigenen Kontaktliste der App befinden, können Inhalte von der Person anzeigen lassen und lesen.

In einem weiteren Schritt könnten die im folgenden beschriebenen Funktionen die Applikation erweitern. Nutzern ist es möglich, Beiträge über ein "Like" positiv oder negativ zu bewerten. Außerdem können Nutzer Kommentare unter Inhalten schreiben und so Feedback zu Inhalten liefern.

Für die Applikation wird ein benutzerfreundliches Ionic-Frontend, das vor allem auf mobile Nutzer abgestimmt ist, entwickelt. Erweitert wird es durch ein Backend zur Datenverwaltung.

1.1 Motivation

Der App-Markt ist ein stetig wachsender und die Auswahl der App zur Kommunikation mit anderen Menschen ist schier unendlich. Telegram und Whats App sind nur einige sehr erfolgreiche Beispiele dafür. Jedoch sind die Anwendungen zur Kommunikation überladen mit zusätzlichen Funktionen und Einstellungen. Hinzu kommt, dass niemand so wirklich weiß, was mit den dort gesammelten Informationen überhaupt passiert. Die sensiblen Daten des Nutzers werden nicht selten für Werbung genutzt und die Apps fordern meist Zugriffsrechte auf (fast) alle Bereiche des Mobiltelefons, doch wozu wird nur sehr oberflächlich erklärt.

Unsere Motivation war es also eine Anwendung zu erstellen, die auf den ursprünglichen Zweck der textuellen Kommunikation reduziert ist und nicht durch mehr Funktionalität vom eigentlichen Sinn der Tätigkeit ablenkt. Außerdem war es uns wichtig, die App "datensparsam" zu gestalten, das heißt, der Nutzer muss nur die notwendigsten Daten hinterlegen und auf das Benutzertracking wird vollständig verzichtet.

In der Projektarbeit soll also eine benutzerfreundliche und datensparsame Kommunikations-Applikation erstellt werden. Sie soll nur wenige sensiblen Daten des Benutzers benötigen und verwalten.

1.2 Aufgabenstellung

In der Projektarbeit soll eine progressiv Web-Applikation für Android und IOS erstellt werden, die die folgenden Funktionen bieten soll:

- Nutzer können sich registrieren und zu einem späteren Zeitpunkt mit ihren Daten wieder einloggen.
- Nutzer der App können nach Freunden suchen und Freunde hinzufügen.
- Nutzer können Inhalte publizieren (Textstatus max. 150 Zeichen) und so mit ihren Freunden teilen.
- Nutzer können sich Inhalte von Freunden anzeigen lassen.

Erweiterungen der genannten Basisfunktionen könnten sein:

- Eine Feedback-Funktion (Like/Dislike) für Beiträge von Freunden. Nutzer können so auf Beiträge reagieren.
- Nutzer können Kommentare zu Beiträgen von Freunden verfassen, um ihre Reaktion (Like/Dislike) zu ergänzen oder nur textuell zu reagieren.

Die Anwendung soll grob wie folgt aufgebaut sein:

- Ein Ionic-Frontend. Das User Interface (UI) und die Logik sind auf mobile Endgeräte abgestimmt.
- Ein Backend für die Datenverwaltung. Auf sensible Daten des Nutzers wird weitestgehend verzichtet.

2 Frameworks

In diesem Kapitel werden die verwendeten Frameworks genannt sowie deren Verwendungszwecke kurz vorgestellt.

2.1 Docker

Docker ist eine Software zur Erstellung sowie Bereitstellen von Anwendungen mittels virtualisierten Containern. Container sind dabei eine Art "Behälter" für Anwendungen die in diesen laufen können. Die einzelnen Container enthalten wichtige Daten, welche für die Ausführung der Applikation benötigt werden. Dazu gehören auch Betriebssystem-Komponenten wie beispielsweise Libraries oder Binaries, die eine leichtgewichtige Form der Virtualisierung ermöglichen. Die Container können aber unabhängig voneinander ausgeführt werden, da sie voneinander abgekapselte Einheiten darstellen. Durch ihre Portabilität sind die Container auch auf unterschiedlichen Systemen ausführbar, solange das System einen Container virtualisieren kann. Gemeinsam mit dem installierten Betriebssystem sorgt die Container Engine dann für die Virtualisierung. Gestartet wird ein Container aus einem Abbild, dem Container-Image. Ein Container-Image enthält dabei alle Abhängigkeiten der Anwendung. Der Container sorgt für eine Trennung und Verwaltung der auf dem Host-System genutzten Ressourcen beziehungsweise einzelnen Containern. Dabei ist zu beachten, dass immer nur ein Dienst oder eine Anwendung pro Container existieren sollte. Zum Beispiel: Eine Datenbank und ein Webserver laufen in zwei unterschiedlichen Containern, können aber über die Container hinaus miteinander kommunizieren.

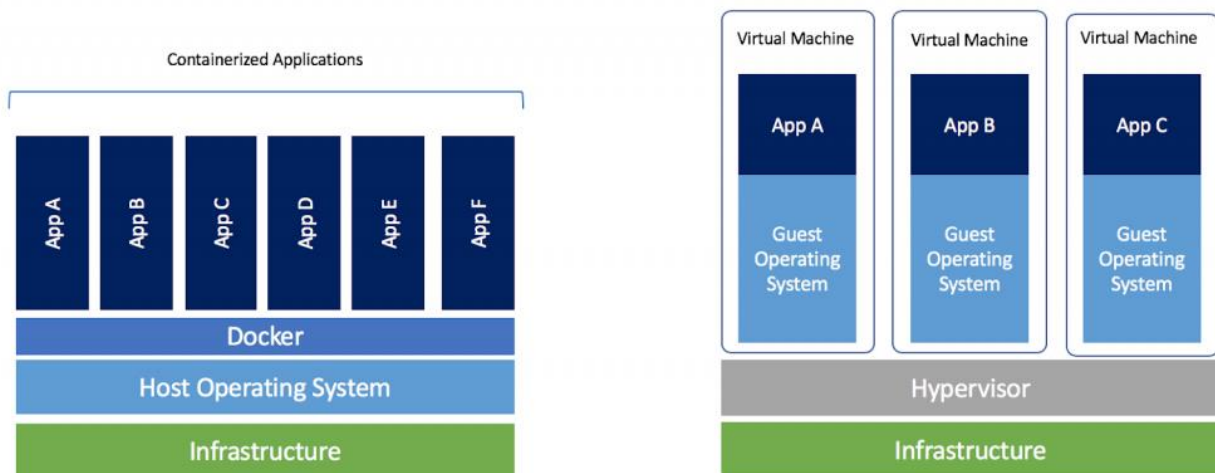


Abbildung 2.1: Unterschied zwischen Container und virtuellen Maschinen [Gae19]

2.1.1 Verwendung

Docker wurde in diesem Projekt dazu genutzt, um auf unterschiedlichen Host-Systemen die für das Projekt entwickelte REST-API-Schnittstelle, sowie eine MYSQL-Datenbank zur Verfügung zu stellen, auf welche die Ionic-Anwendung zugreifen soll.

Docker selbst ist eine der bekanntesten Container-Technologien, weshalb wir uns bei diesem Projekt für diese bewährte Technologie entscheiden haben. Im Umgang mit Docker hatten wir vor diesem Projekt nur wenig bis keine Erfahrungen gesammelt.

2.2 Spring Boot

Spring oder auch Spring-Applikation genannt, ist ein Framework zum Erstellen von Java-Anwendungen. Spring Boot ist eine einfache, schon vorkonfigurierte Spring-Applikation beziehungsweise Erweiterung zu Spring und wird häufig dazu genutzt um beispielsweise Web-Anwendungen oder Web-Schnittstellen (REST-API) zu erstellen. Das Ziel von Spring Boot ist es, die Entwicklung von Anwendungen zu vereinfachen und schnell komplexe Anwendungen zu erzeugen. Spring Boot greift hierzu auf eine Menge von Funktionen wie dem Spring Initializers zurück. Darüber können benötigte Abhängigkeiten wie zum Beispiel Datenbanktreiber, Sicherheitskomponenten oder andere Services zu Beginn ausgewählt werden. Diese werden dann per Dependency Injection beim Anlegen eines Projektes automatisch einbezogen und konfiguriert.

2.2.1 Verwendung

Spring Boot wurde in diesem Projekt dazu genutzt, um eine REST-API-Schnittstelle zwischen der Frontend-Anwendung und der Datenbankanwendungen zu realisieren. Dabei fungiert die Rest-API nicht nur als einfacher Vermittler zwischen Frontend-Anwendungen und Datenbank, sondern auch als Authentifizierungs- und Autorisierungsdienst. So muss sich der Nutzer zunächst registrieren um Zugriff auf weitere Funktionalitäten der Rest-API Schnittstelle zu erlangen. Hierzu wird das JWT (kurz für: JSON Web Token) Standard genutzt. Dieser erzeugt bei jedem erfolgreichen Login einen einzigartigen JWT, über den ein Nutzer sich gegenüber dem Server authentifiziert. Der JWT enthält neben einigen nicht kritischen Nutzerinformationen, eine Signatur. Die Signatur dient dazu die Echtheit und Unverfälschtheit des JWT zu garantieren.

Es folgt eine kurze Erklärung des Ablaufs:

Ein Client sendet einen HTTP-Request an den Server beziehungsweise an den Controller. Der Controller empfängt den HTTP-Request und leidet diesen an die entsprechende Methode oder Schnittstelle im Controller weiter. Die Schnittstelle liest nun alle benötigten Daten aus und bearbeitet die Anfragen. Dabei greift die Methode auf andere Komponenten des Servers wie zum Beispiel eine Datenbank zu. Könnte die Anfrage erfolgreich bearbeitet werden, sendet der Controller einen HTTP-Response, gegebenenfalls mit den ermittelten Daten zurück an den Client.

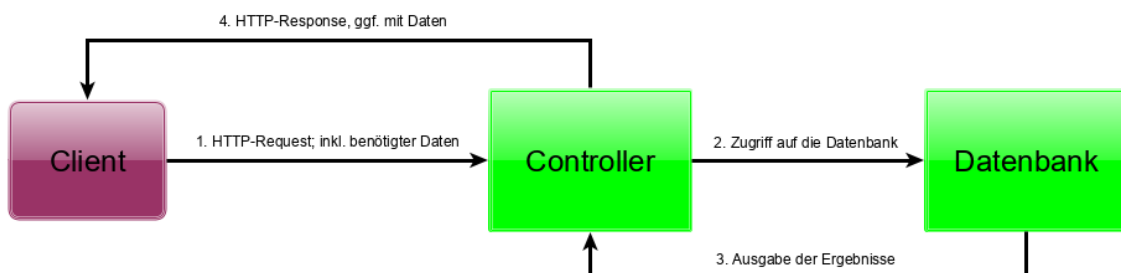


Abbildung 2.2: Ablaufdiagramm einer HTTP-Anfrage an den Server

2.2.2 Dependencies

Folgende wichtigen Dependencies wurden in dem Projekt verwendet:

- Spring Boot Web Starter

Starter für die Erstellung von Web-Anwendungen, einschließlich RESTful, mit Spring MVC. Verwendet Tomcat als den standardmäßig eingebetteten Container.

- Spring-Boot Dev Tools

Spring-Boot Dev Tools erweitert die schon vorhandenen Spring-Funktionalitäten um weitere, vor allem für die Entwicklung nützliche Features. Wie zum Beispiel das automatische Compilieren und Starten der Anwendung nach größeren Änderungen am Quellcode.

- Spring Data JPA

Spring Data JPA ermöglicht die Vereinheitlichung sowie den leichten Zugriff auf die verschiedenen Arten von Persistenzspeichern, sowohl auf relationale Datenbanksysteme als auch auf NoSQL-Datenspeicher.

- MySQL Connectors

MySQL bietet einen auf dem Standard basierenden Treiber für JDBC, ODBC und .net, mit denen Entwicklern Datenbankanwendungen in der Sprache ihrer Wahl erstellen können. Darüber hinaus ermöglicht es einem native C-Bibliothek Entwickler die direkte Einbettung von MySQL in eine Anwendung.

- Spring Security

Spring Security ist ein leistungsstarkes und hochgradig anpassbares Framework für Authentifizierung und Zugriffskontrolle.

- JSON Web Tokens

JSON Web Token (JWT) ist ein offener Standard (RFC 7519), der eine kompakte und in sich geschlossene Methode zur sicheren Übertragung von Informationen zwischen Parteien als JSON-Objekt definiert. Diese Informationen können verifiziert werden und sind vertrauenswürdig, da sie digital signiert sind. JWTs können mit einem geheimen (mit dem HMAC-Algorithmus) oder einem öffentlich oder privaten Schlüsselpaar mit RSA oder ECDSA signiert werden.

2.2.3 Controller

Eine Spring Boot Anwendung kann so aufgebaut werden, dass diese eine Schnittstelle - auch Controller genannt - nach außen hin öffnet. Diese können über gewisse Pfade (URLs) angesprochen werden. Ein Controller besitzt dabei mehrere vorher festgelegte Einstiegspunkte. Per Injection können diesen Einstiegspunkte Pfade zugewiesen werden. Es ist zwar möglich lediglich einen einzigen Controller zu nutzen, es bietet sich jedoch an, für verschiedenen Szenarien unterschiedliche Controller zu definieren. Je nach Schnittstelle werden unterschiedliche Eingabeparameter benötigt. Innerhalb der Schnittstellen werden nun unterschiedliche Operationen ausgeführt, zum Beispiel die Überprüfung der Logging-Daten oder das Anlegen eines Datenbankeintrags. Am Ende wird ein Request erzeugt und an das Gerät, welches eine Anfragen an die REST-API gestellt hat, zurück gesendet.

Für dieses Projekt wurden folgende Controller und Schnittstellen definiert:

- FriendController

FriendController dient dazu Verknüpfungen zwischen zwei Nutzern abzuspeichern. Dabei wird lediglich die bei der Registrierung erzeugte ID eines Nutzers abgespeichert. Über diese ID können anschließend weitere Nutzerdaten heruntergeladen werden.

- getFollowerData: Über einen Username, können Nutzerdaten abgefragt werden.
- addFollower: Fügt einen Nutzer der eigenen Follower-Liste hinzu.
- findAllFollower: Stellt **alle** Follower-Informationen zur Verfügung.
- findAllFollowerDataById: Sucht alle Follower-Daten zu einen Nutzer, sowie alle weiteren Daten zu den Followern.

- findAllFollowerById: Übersicht alle Follower-Daten.
- deleteFollowerById: Löscht einen Follower aus der Liste.

■ LoginController

Der LoginController wird genutzt um einen Nutzer zu Authentifizieren.

- login: Benötigt den Username und das Passwort, um darüber einen Benutzer zu authentifizieren. Anschließend wird für diese Session ein einmaliger Token generiert, über den sich ein Benutzer für weitere Aktionen authentifizieren kann, ohne das jedes Mal sein Passwort mit gesendet werden muss. Über diesen Token erfolgt auch die Autorisierung.

■ MessageController

Der MessageController verwaltet alle Anfragen zu den erstellten Messages. Eine Message ist eine einfache Textnachricht, welcher ein Benutzer der Anwendung erzeugen kann. Um eine Message von anderen Benutzern zu empfangen, muss der andere Nutzer als Follower registriert werden.

- findAll: Stellt **alle** erzeugten Messages bereit.
- add: Fügt eine neue Message hinzu.
- loadMessages: Lädt alle Messages, die ein Nutzer empfangen kann und sortiert diese nach deren Erstellung, sodass die neusten Messages oben in der Liste angezeigt werden.

■ MyAccountController

Der MyAccountController verwaltet alle Anfragen zu einem Account. Ein Account definiert dabei alle Informationen zu dem gerade eingeloggten Nutzer.

- findAll: Sucht alle existierende Accounts.
- getUserData: Stellt die Nutzerdaten zu einem Nutzer bereit (außer das Passwort).
- updateUser: Updatet die Nutzerdaten eines bestimmten Nutzers.

■ RegistrationController RegistrationController wird genutzt, um die Anfragen bei der Registrierung zu verwalten.

- isUsernameAvailable: Überprüft ob der übergebenen Nutzernamen schon vergeben ist.
- isEmailAvailable: Überprüft ob die übergebenen E-Mail-Adresse schon vergeben ist.
- registration: Fügt einen neuen Nutzer der Datenbank hinzu.

2.3 Ionic

Das Open-Source-Webframework Ionic eignet sich zur Erstellung von Hybrid-Apps und Progressive-Web-Apps auf Basis von HTML5, CSS und Javascript oder Typescript Sprachen, wie z.B. Angular, Apache Cordova, Vue oder React. Ionic stellt hierbei eine aus verschiedenen Modulen bestehende Programmbibliothek zur Verfügung. Es besteht aus einem npm-Modul, wodurch es sehr einfach ist weitere Module nachzuladen. Ein Beispiel hierzu wäre das Entwickeln von Mobilien Anwendungen durch Ionic, hierzu muss das bestehende npm-Modul um das npm-Modul zur Entwicklung unter eine anderen Plattform (z.B. IOS oder Android) einfach nachgeladen werden und kann anschließend genutzt werden.

2.3.1 Verwendung

ionic wurde in diesem Projekt dazu genutzt um das Frontend der Anwendung zu entwickeln. Ziel dabei war es eine Anwendung zu erstellen, welche ohne großen Aufwand auf drei verschiedenen Plattform lauffähig ist (Web, IOS und Android).

Die Anwendung soll dabei eine Registrierungsseite bieten über welche sich neue Nutzer registrieren können und eine Logging-Page über welche die Nutzer sich nach der Registrierung oder bei einer späteren Nutzung der App jederzeit einloggen können. Zum Einloggen in die App müssen die Nutzer ihren Benutzernamen und ihr Passwort eingeben und gelangen bei einem erfolgreichen Logging auf dem Dashboard der App. Auf dieser erhalten die Benutzer eine Übersicht über alle getätigten Nachrichten der Personen denen sie folgen und können jeder Zeit eigenen Nachrichten verfassen. Auch können die Nutzer über die Tabbar im unteren Bildschirmbereich einerseits auf die Follower-Liste andererseits auf die Informationen ihres Accounts zugreifen. In der Follower-Liste erhalten die Nutzer eine Übersicht über alle Benutzer denen Sie selbst folgen. Auch kann ein Nutzer dort über den "Freund hinzufügenButton eine Person Suchen und dieser Person anschließend folgen. Auf der Account-Page erhalten die Nutzer die wichtigsten Informationen über Ihren Account und können gegebenenfalls gewisse Inhalte wie beispielsweise den Nutzernamen oder das Passwort ändern.

2.3.2 Ablaufdiagramm

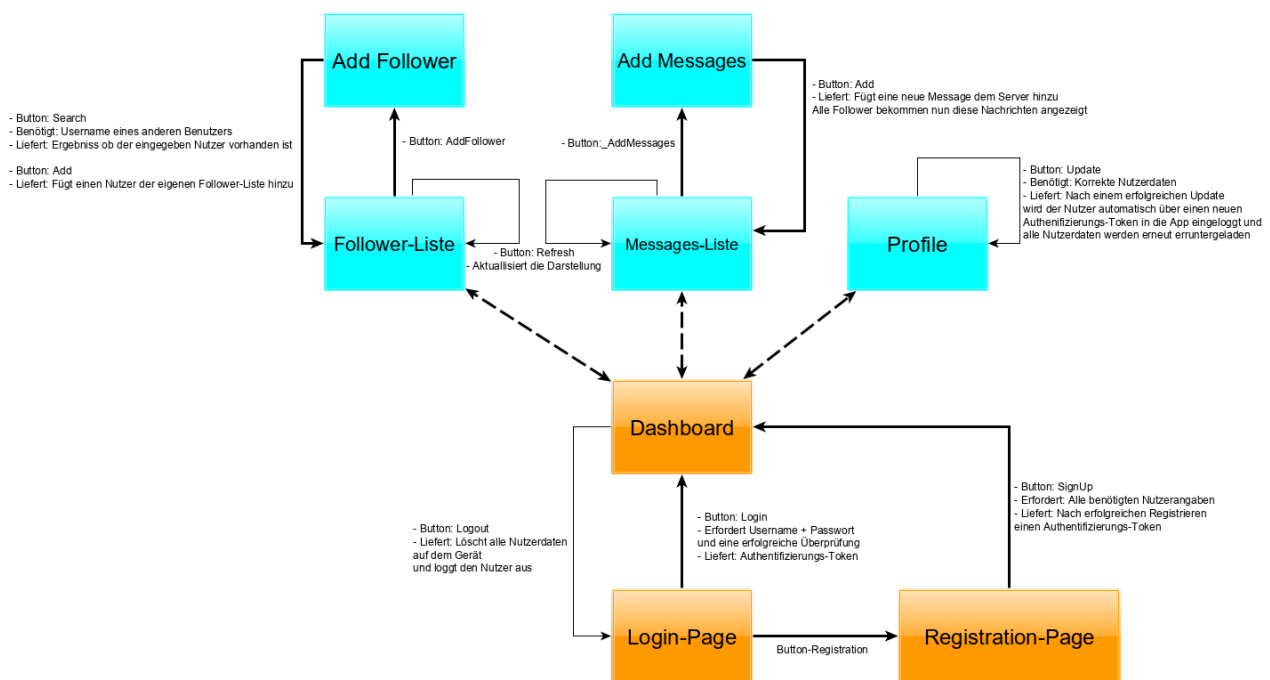


Abbildung 2.3: Ablaufdiagramm der Ionic Anwendung "Keep in Touch"

3 Pages

Folgend werden die einzelnen Pages des Ionic Frontends aufgezeigt und deren Funktionalität kurz erklärt. Alle Pages des Frontend werden als Android- und IOS-Ansichten präsentiert.

3.1 Login

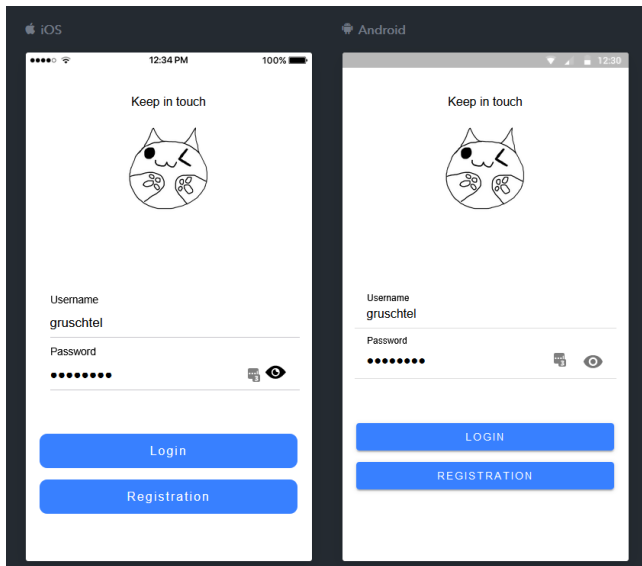


Abbildung 3.1: Login-Page

Auf der Login-Seite kann sich der Nutzer mit seinen Daten (Nutzername und Passwort) anmelden oder sich über den "RegistrationButton als neuer Nutzer registrieren.

3.2 Registration

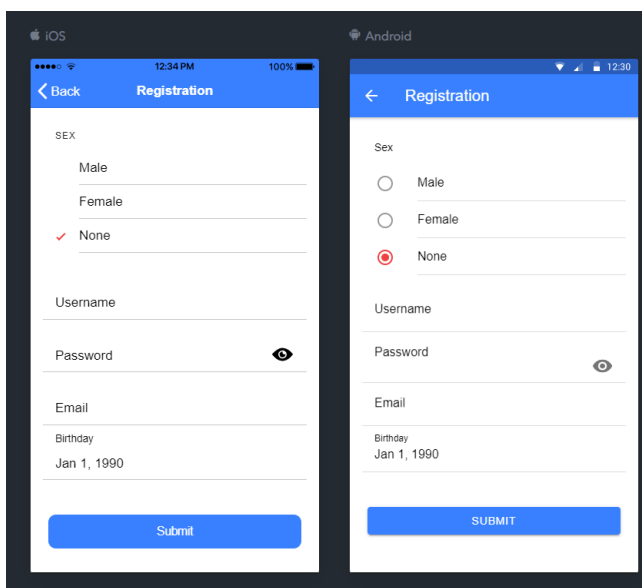


Abbildung 3.2: Registrierungs-Page

Will sich ein Nutzer neu registrieren muss er das Formular ausfüllen. Dazu sollte er folgende Angaben machen: Geschlecht (Sex), einmaliger Nutzernamen (Username), Passwort (Password), E-Mail-Adresse (Email) und Geburtsdatum (Birthday). Ist das Formular befüllt kann sich der Nutzer über den SSubmitButton für die App "Keep in touch"registrieren.

3.3 Messages

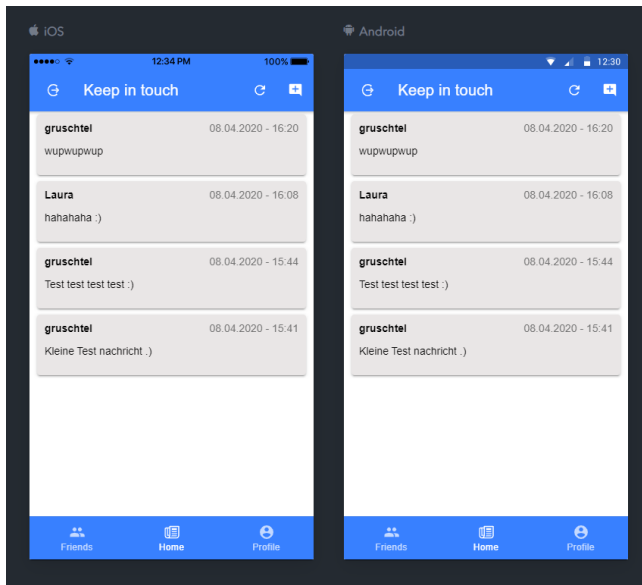


Abbildung 3.3: MessageListe-Page

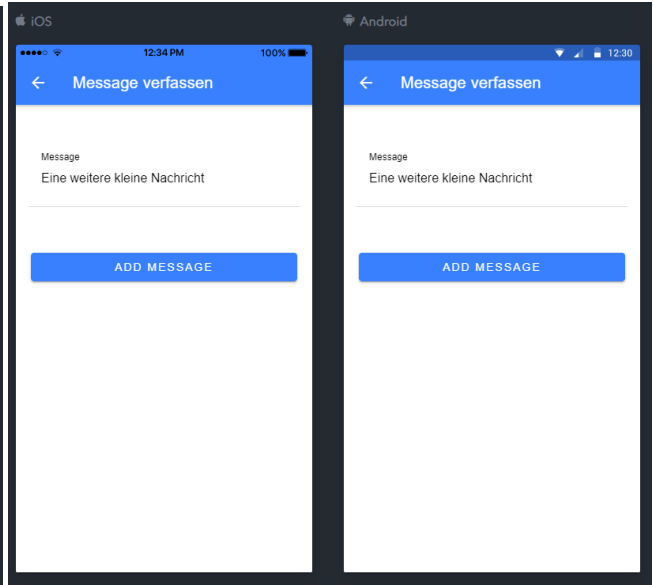


Abbildung 3.4: AddMessage-Page

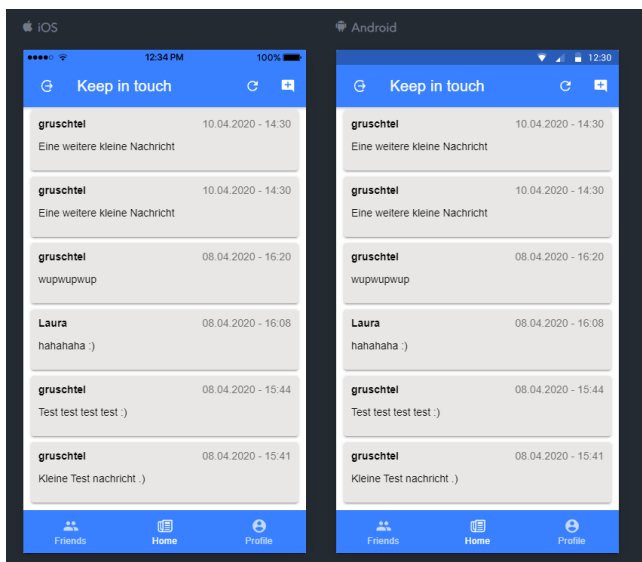


Abbildung 3.5: MessageListe-Page: Neue Nachrichten werden dargestellt.

In der Übersicht der Nachrichten (Home) kann der Nutzer alle Nachrichten sehen. Fügt er eine neue Nachricht über den "Nachricht verfassenButton hinzu erscheint diese auf der aktualisierten Übersicht der Nachrichten. Jede Nachricht wird mit einem Zeitstempel bestehend aus Datum und Uhrzeit versehen. Verfasst der Nutzer eine neue Nachricht gelangt er auf eine neue Ansicht. Über das eintippen von Text und

dem bestätigen der Nachricht durch den Add MessageButton wird die Nachricht gesendet und erscheint in der aktualisierten Übersicht. Durch die Nutzung des SZurückButtons kann man diesen Vorgang abbrechen, die Nachricht wird nicht gesendet. Gesendete Nachrichten können nicht gelöscht werden. Auch wenn diese Funktion im Vorfeld geplant wurde haben wir uns im Entwicklungsprozess aus Designgründen dagegen entschieden.

3.4 Follower

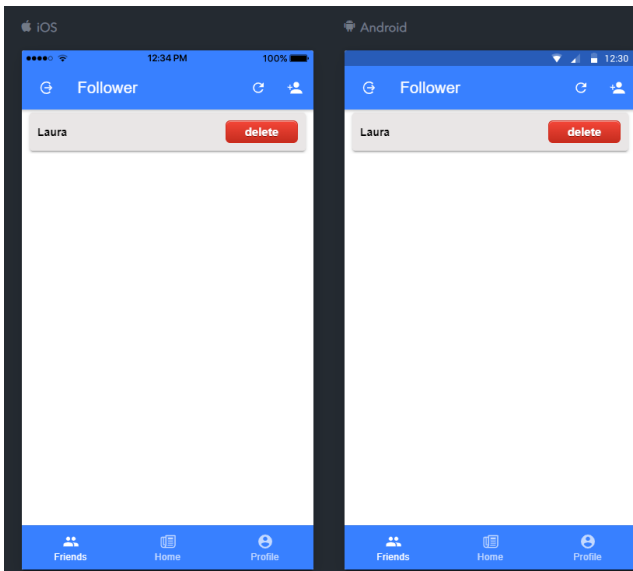


Abbildung 3.6: Follower-Liste-Page

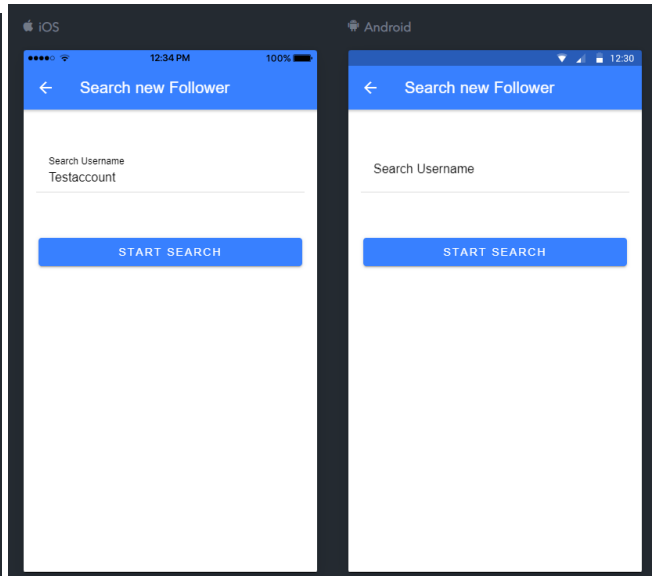


Abbildung 3.7: Add-Follower-Page: Ein Benutzer wird gesucht.

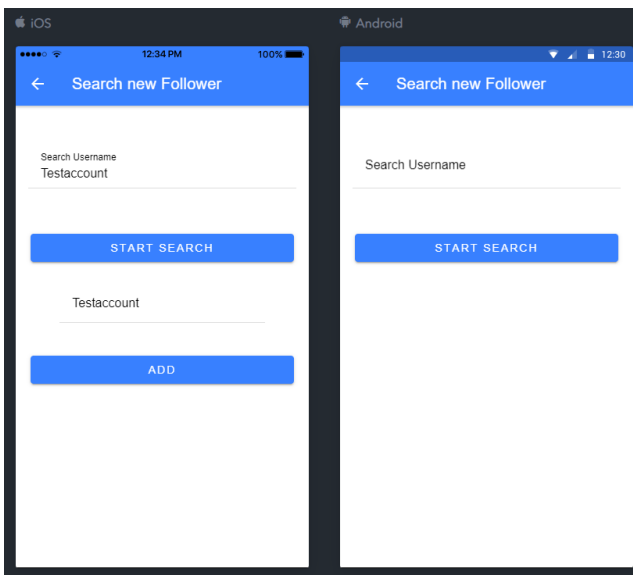


Abbildung 3.8: Add-Follower-Page: Ein Benutzer wird als Follower hinzugefügt.

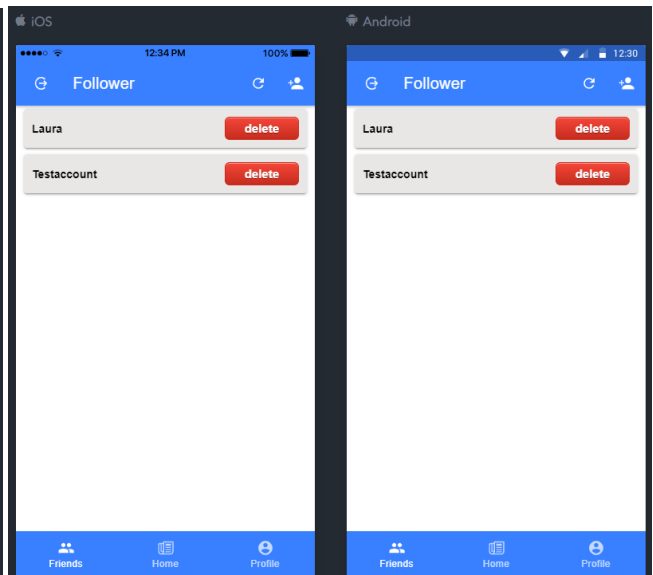


Abbildung 3.9: Follower-Liste-Page: Neuer Follower wird angezeigt.

Auf der Friends-Page hat man einen Übersicht aller Follower. Man kann über den "DeleteButton" auch Follower von der Liste löschen. Über den "Freunde hinzufügenButton" kann man andere Nutzer suchen und zur Liste hinzufügen, sodass man auch ihre Inhalte künftig sehen kann.

3.5 Profile

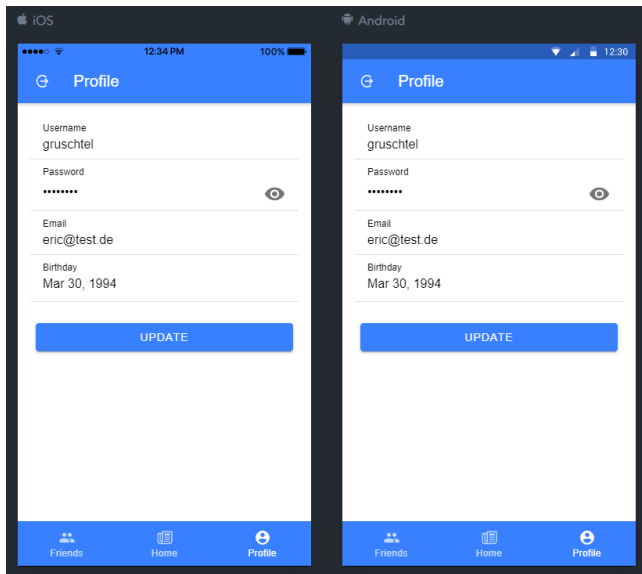


Abbildung 3.10: Profil-Page

Im Profil kann der Nutzer seine Daten sehen (Nutzername, Passwort, E-Mail-Adresse und Geburtsdatum). Außerdem kann er hier seine Angaben bearbeiten und diese Änderungen über den UpdateButton in seinem Profil speichern.

4 Installationsanleitung

Da verschiedene Technologien zum Umsetzen des Projektes beziehungsweise der Anwendungen genutzt wurden, folgt nun für jedes eingesetzte Werkzeug eine kompakte Installationsanleitung. Eine Installationsanleitung ist notwendig, da je nach Werkzeug einige Besonderheiten bei der Installation zu beachten sind. Außerdem werden je nach Werkzeug zusätzliche Plugins oder Erweiterungen benötigt, um die Anwendung auf einem anderen Host-System erfolgreich ausführen zu können.

Wichtig an dieser Stelle ist, dass diese Installationsanleitung nur für Nutzer mit dem Betriebssystem "Windows 10 Pro" verfasst wurde. Da auf allen verwendeten Host-Systemen dieses Betriebssystem installiert war. Nach aktuellen Erkenntnissen sollte es jedoch kein Problem sein, die in diesem Projekt genutzten Programme und Tools auch auf einem anderen Betriebssystem zu installieren. In diesem Fall verweisen wir jedoch auf die von den Herstellern bereitgestellten Installationsanleitungen.

Zunächst werden wir alle benötigten Programme und Abhängigkeiten installieren und anschließend wird erklärt, wie der Server und die Ionic-Anwendung gestartet werden kann.

Neben den hier vorgestellten Tools und Werkzeugen wurde Visual Studio Code (VS Code) als Entwicklungsumgebung für das Projekt genutzt. VS Code hat den Vorteil, ein schlankes und leichtgewichtiges Anwendungsumfeld zu sein, welches mittels Plugins (Extension) um benötigte Funktionen erweiterbar ist. Es kann sowohl zum Entwickeln und Starten des Backends als auch für das Frontend genutzt werden.

4.1 Projektstruktur

Das Projekt besteht aus drei Hauptordnern: Backend, Frontend und Dokumentation. In dem Ordner Backend befinden sich alle für den Server benötigten Daten. Im Ordner Frontend alle Daten für die Ionic-Anwendung und im Ordner Dokumentation alle Materialien zur Dokumentation.

4.2 Docker

Docker vereinfacht die Anwendungsbereitstellung, indem es den Transport und die Installation von Containern erleichtert, die alle erforderlichen Pakete als Dateien enthalten. Container gewährleisten die Trennung und Verwaltung der auf einem Computer verwendeten Ressourcen. Dazu gehören nach Angaben der Entwickler: Code, Laufzeitmodul, Systemwerkzeuge, Systembibliotheken - eben alles, was auf einem Computer installiert werden kann.

1. Zunächst muss der Installations-Client für Docker heruntergeladen werden. Den Link dazu finden Sie hier: <https://docs.docker.com/docker-for-windows/install/>

Wichtig: Für Windows wird entweder die Pro, Enterprise oder Education Version benötigt.

2. Während der Installation von Docker werden Sie gefragt, ob Sie Windows Container oder Linux Container verwenden möchten. Stellen Sie sicher, dass Sie das Kästchen in der angezeigten Option nicht ankreuzen (vgl. Abbildung). Der Docker verwendet also Linux Container. Docker-Container verwenden oft Linux-Komponenten, womit Windows-Container oft Probleme haben.

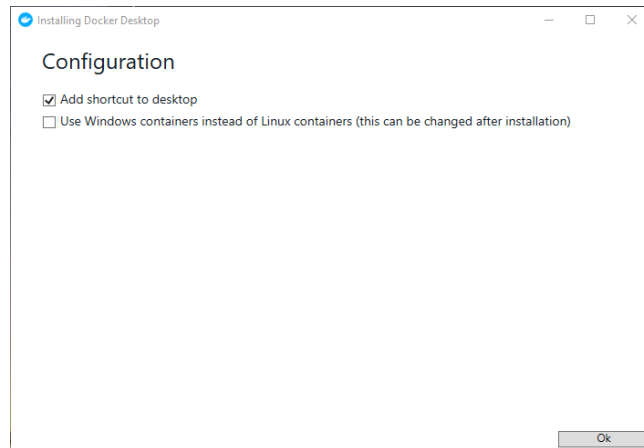


Abbildung 4.1: Das Feld "Windows-Containers" sollte frei bleiben.

3. Unter Windows müssen Sie Hyper-V und Container aktivieren. Hyper-V und Container können unter den Windows-Funktionen aktiviert werden. Dazu drücken Sie zunächst die Windows-Taste + Q und geben dort die Windows-Features ein. Aktivieren Sie dann Hyper-V und Container und klicken Sie auf "ÖK". Folgen Sie dann der Windows-Installation und installieren Sie die Features. Nach der Installation der Windows-Features müssen Sie den PC neu starten.

Wichtig: Um Virtualisierungsdienste nutzen zu können, müssen diese gegebenenfalls im Bios-System aktiviert werden.

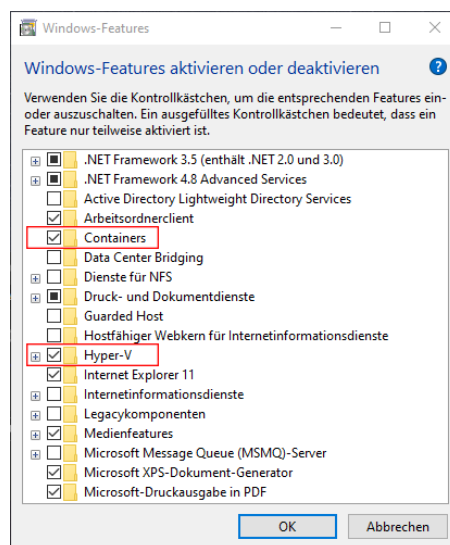


Abbildung 4.2: Benötigte Windows Features - Containers und Hyper-V.

4. Es bietet sich an unter Windows die "Windows PowerShell" zu nutzen, da diese für Docker nützliche Funktionalitäten im Gegensatz zur einfachen Eingabeaufforderung mitbringt. Wenn die Windows PowerShell noch nicht installiert wurde, können Sie es mit den Windows-Funktionen installieren. Drücken Sie dazu zunächst die Windows-Taste + Q und geben Sie "Windows-Features" ein. Schalten Sie dann Windows PowerShell ein, und drücken Sie dann "ÖK". Installieren Sie PowerShell und starten Sie dann den Computer neu.

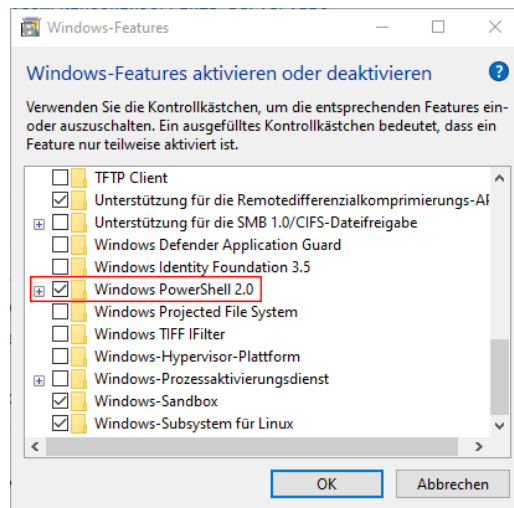


Abbildung 4.3: Benötigte Windows Features - PowerShell.

- Um nun einen Docker-Container unter Windows nutzen zu können, benötigt Docker noch Zugriff auf den Speicherort, auf welchem sich der Docker-Container befindet. Hierzu muss Docker Zugriff auf die Festplattenpartition gewährt werden, auf der die Docker-Container gespeichert sind. Um dies zu ermöglichen muss mit der linken Maustaste auf das Docker-Symbol in der Taskleiste das Einstellungsmenü geöffnet werden. Anschließend navigieren Sie zu dem Reiter "Resources / File Sharing". Dort können Sie Docker Zugriff auf die Partitionen gewähren oder Sie wieder aufrufen. An diesem Punkt werden Sie auch nach dem aktuellen Windows-Benutzerkennwort gefragt. Sollten sie nach einem Passwort gefragt werden, geben Sie Ihr aktuelles Windows-Passwort ein, um Docker Zugriff auf die Partition zu gewähren.

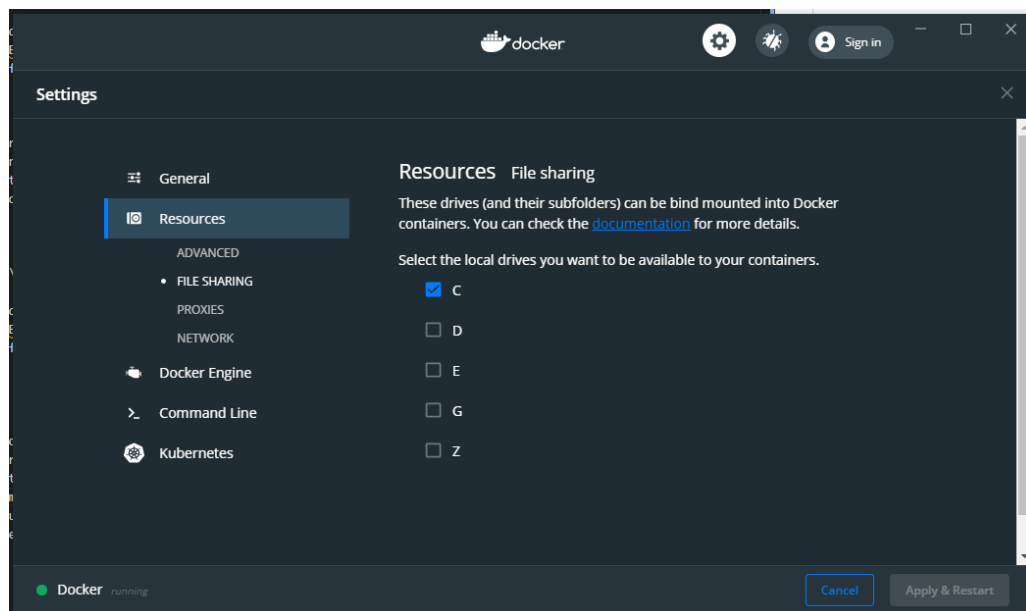


Abbildung 4.4: Freigabe einzelner Partitionen für Docker.

4.3 Spring Boot

Wurde Docker erfolgreich installiert erfolgt nun die Anleitung zur Installation und Inbetriebnahme der Spring Boot Anwendung. Die Installation zu Spring Boot ist recht einfach dazu benötigen wir lediglich das

4 Installationsanleitung

Java Developer Kit 1.8 und Build-Management-Tool Maven.

1. Die Installationsanwendung zu Java Developer Kit 1.8 kann unter folgendem Link aufgerufen werden:
<https://www.oracle.com/java/technologies/javase-jdk8-downloads.html>
2. Im nächsten Schritt wird zunächst Maven heruntergeladen. Mit Maven kann man insbesondere Java-Programme standardisiert erstellen und verwalten. Den Link dazu finden Sie hier: <https://mirror.dkd.de/apache/maven/3/3.6.3/binaries/apache-maven-3.6.3-bin.zip>.

Wichtig: Es sollte darauf geachtet werden, dass die Binaries heruntergeladen werden. Diese enthalten einen Ordner mit der Bezeichnung "bin", welcher für spätere Zwecke benötigt wird.

3. Nachdem die Binaries heruntergeladen wurden, entpacken wir den Ordner. Da Maven jedes Mal wenn das Programm kompiliert werden soll benötigt wird, bietet es sich an den Ordner dauerhaft an einen "sicheren" Ort abzuspeichern. Es bietet sich an Maven in den Umgebungsvariable bekannt zu machen. Dadurch muss nicht bei jedem Maven-Befehl der korrekte Pfad zu den entpackten Ordner angegeben werden. Ist Maven dem System bekannt, wird wenn ein Maven-Befehl aufgerufen wird, der Pfad zum entpackten Ordner automatisch durch das System hinzugefügt.
4. Nun öffnen Sie per Windows-Taste + Q das Suchfeld und suchen dort nach "Systemumgebungsvariablen bearbeiten". Es sollte sich nun ein Fenster öffnen. Dort drücken Sie auf "Umgebungsvariablen bearbeiten".

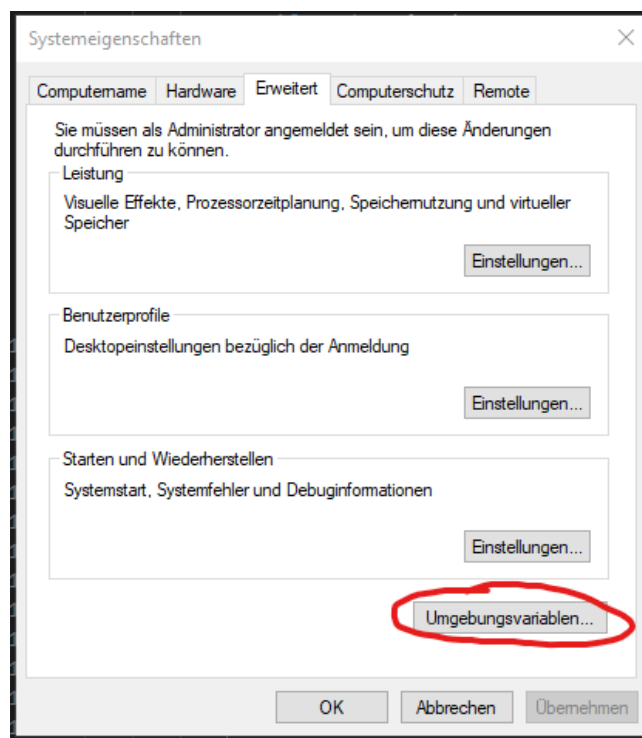


Abbildung 4.5: Umgebungsvariable bearbeiten.

5. Im nächsten Fenster wählen Sie zunächst unter Systemvariablen den Wert Path aus und drücken dann auf bearbeiten.

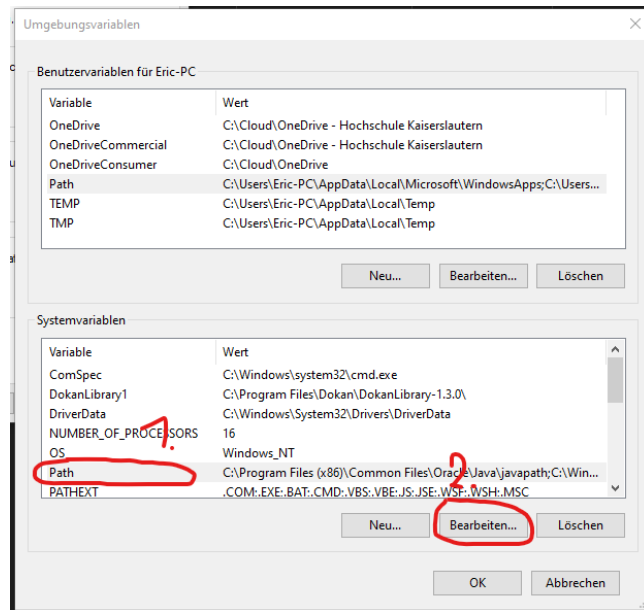


Abbildung 4.6: Umgebungsvariable bearbeiten.

6. Dort fügen Sie mit "Neu" einen neuen Eintrag in der Tabelle hinzu und können nach auswählen der neuen Spalte über "Durchsuchen" den Pfad zu dem Maven-Ordner angeben.

Wichtig: Zu beachten ist, dass der Ordner "bin" auf jeden Fall angegeben werden muss. Ansonsten kann Windows die benötigten Dateien nicht finden.

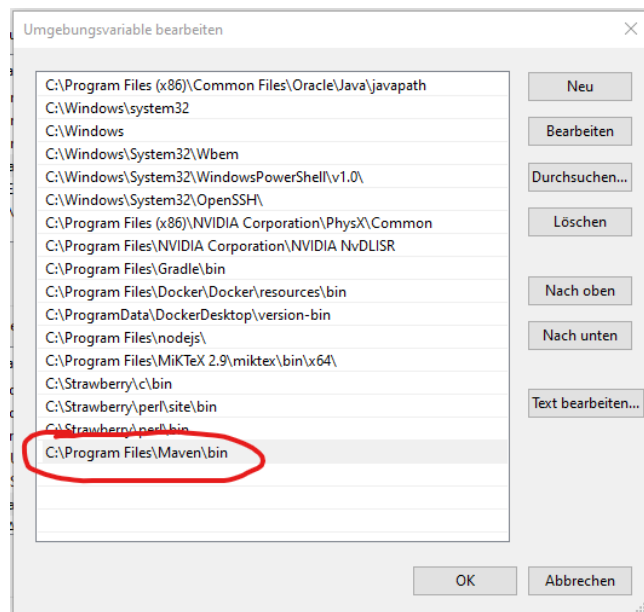


Abbildung 4.7: Umgebungsvariable hinzufügen.

4.4 Ionic

Um eine Ionic-Anwendung zu programmieren wird neben einer Entwicklungsumgebung lediglich NodeJS und einige Erweiterungen benötigt.

4 Installationsanleitung

1. Der erste Schritt ist das Herunterladen und Installieren von Node.JS. Sie finden Node.JS unter folgendem Link: <https://nodejs.org/en/download/current/>
2. Starten Sie nach der Installation von Node.JS die Eingabeaufforderung oder Windows PowerShell und geben Sie den Befehl `npm -version` ein. Wenn Sie die aktuelle Versionsnummer sehen, wurde Node.JS korrekt installiert.
3. Nun müssen noch Ionic und Cordova installiert werden. Geben Sie zunächst an der Eingabeaufforderung `npm install -g cordova` ein. Dadurch wird das Cordova-Framework installiert, das später für die Bereitstellung der Anwendung auf verschiedenen Plattformen benötigt wird. Danach benötigen Sie das Ionic-Framework, um Anwendungen in Ionic zu erstellen. Sie können Ionic mit dem Befehl `npm install -g ionic` installieren.
4. Ionic unterstützt verschiedene Programmiersprachen. Standardmäßig werden Ionic-Anwendungen in JavaScript programmiert. Es ist jedoch auch möglich Ionic-Anwendungen in Angular, React oder Vue zu programmieren. Um Ionic-Anwendungen in anderen Sprachen programmieren zu können, müssen Sie auch die entsprechende CLI auf Ihrem System installieren. Für dieses Projekt wurde die TypeScript-Sprache Angular genutzt. Um Angular zu installieren muss folgender Befehl eingegeben werden: `npm install -g @angular/cli`
5. Nach der Installation von Angular starten Sie die Eingabeaufforderung oder Windows PowerShell und navigieren sich in den "FrontendOrdner des Projektes. Führen Sie dort die folgenden Befehle aus:
 - `npm install`
 - `npm cache verify`Mit diesen Befehlen installieren sie alle benötigten Abhängigkeiten für das kompilieren und ausführen von Ionic- beziehungsweise Angular-Anwendungen.

Im zweiten Schritt werden nun alle Plugins installieren, die Ionic zur Unterstützung von anderen Plattformen benötigt.

1. Damit Ihre Ionic-Anwendung eine Plattform unterstützt, öffnen Sie zunächst die Eingabeaufforderung und wechseln Sie in das aktuelle Verzeichnis Ihres Ionic-Projekts.
2. Geben Sie folgenden Befehle ein um IOS als Ziel-Plattform hinzuzufügen: `ionic cordova platform add ios`
3. Geben Sie folgenden Befehle ein um Android als Ziel-Plattform hinzuzufügen: `ionic cordova platform add android`
4. Anschließend müssen noch die Ressourcen der Plattform hinzugefügt werden. Hierzu muss der Befehl `npm install -g cordova-res` ausgeführt werden.

Zur Unterstützung der Anwendung direkt auf einem Android-Gerät werden noch einige Programme benötigt.

1. Zunächst benötigen Sie Android Studio. Eine Einführung in die Installation und Einrichtung finden Sie unter <https://ionicframework.com/docs/developing/android>
2. Wurde Android Studie und alle benötigten Feature installiert, kann die Anwendung für Android Geräte gebuildet und anschließend verteilt werden. Dazu geben Sie den Befehl `ionic cordova build android` und anschließend `ionic cordova emulate android` ein.

Alternativ zu einem Emulator können Sie auch Ihr eigenes Android-Smartphone über USB an Ihren PC anschließen und Ihre Ionic-App auf Ihrem eigenen Telefon installieren. Siehe: <https://javatutorial.net/connect-android-device-android-studio>

Zur Unterstützung der IOS-Plattform benötigen Sie einen Computer mit MAC-OS und installiertem XCode. Eine Einführung in die Installation und Einrichtung finden Sie unter: <https://ionicframework.com/docs/developing/ios>

4.4.1 Fehlerbehandlung

Ein häufiger Fehler der beim Verteilen beziehungsweise neu Einrichten von Ionic-Projekten passiert ist das Abhängigkeiten nicht korrekt aufgelöst werden können. Oft genügt es lediglich im Ordner "Frontend" die Datei package-lock.json und den Ordner node_modules zu löschen und anschließend neu zu laden. Hierzu müssen folgende zwei Befehle über die Eingabeaufforderung oder PowerShell (im Projektordner) ausgeführt werden:

- npm install
- npm cache verify

Ein weiterer Fehler der auftreten kann ist, dass beim Builden von Android das Programm mit dem Fehler "Gradle could not be found." fehlschlägt. In diesem Fall müssen Sie Gradle Herunterladen laden, entpacken und wie in Kapitel 4.3 (ab Punkt 4) dem Build-Management-Tool Maven dem System über die Umgebungsvariablen bekannt machen.

5 Projekt Ausführen

Im folgenden Abschnitt wird kurz erklärt, wie man zunächst den Server startet und anschließend die Ionic-Anwendung über einen Browser darstellen kann. Sollte VS Code als Entwicklungsumgebung genutzt werden, bietet es sich an eine Instanz von VS Code für das Backend und eine Instanz von VS Code für das Frontend zu starten.

5.1 Start Backend

1. VS Code bietet im Reiter den Punkt "Terminal" an. Darüber kann direkt in VS Code ein Terminal geöffnet werden. Das Terminal befindet sich schon im richtigen Ordner. Alternativ kann auch die Eingabeaufforderung oder die PowerShell gestartet werden und in den Projektordner "Backend" navigiert werden.
2. Im ersten Schritt wechseln Sie in den Ordner vorhanden "rest_controller".
Befehl: `cd rest_controller`
3. Zunächst muss das Projekt kompiliert werden. Dies kann per Befehl `mvn clean | mvn package` ausgeführt werden.
Wichtig: Maven muss auf dem System vorhanden und über die Umgebungsvariablen dem System bekannt sein.
4. Wurde das Projekt erfolgreich kompiliert, kann der Server per folgendem Befehl gestartet werden:
`docker-compose up --build`
Auch hier sei zu beachten, dass sich zum einen das Terminal im Ordner "rest_controller" befinden sollte und zum anderen der Docker-Deamon (Docker.exe) gestartet ist. Das Starten des Servers kann je nach Hardware einige Minuten benötigen.

5.2 Start Frontend

1. Auch beim Starten des Frontends wird zunächst ein neues Terminal geöffnet. Wahlweise durch VS Code oder durch Starten eines eigenen Terminal Clients.
2. Zum Starten der Ionic-Anwendung muss jedoch nur der Befehl `ionic serve --lab` ausgeführt werden. Ionic übernimmt alle weiteren Schritte zum Starten der Ionic-Anwendung. Diese wird auch automatisch im Webbrowser geöffnet, sobald alle Daten kompiliert wurden. Wird der Befehl `ionic serve --lab` zum ersten Mal ausgeführt, kann es sein, dass Sie noch einige Abhängigkeiten installieren müssen.

5.3 Fehlerbehandlung

Ein möglicher Fehler der Auftreten kann, wäre dass Spring Boot den Pfad zur JDK nicht findet und beim Compilieren der Spring Boot Anwendung mit der Fehlermeldung "No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?" abstürzt.

In diesem Fall muss in der pom-Datei des Servers die Pfad-Angabe des JDK angepasst werden. Die pom-Datei befindet sich im Ordner "Backend\rest_controller\".

```
98
99     <!-- -->
100     <!-- Plugins -->
101     <!-- -->
102     <build>
103         <plugins>
104             <plugin>
105                 <groupId>org.apache.maven.plugins</groupId>
106                 <artifactId>maven-compiler-plugin</artifactId>
107                 <version>3.1</version>
108                 <configuration>
109                     <verbose>true</verbose>
110                     <fork>true</fork>
111                     <!-- !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! -->
112                     <!-- GEGEBENFALLS PFAD ÄNDERN -->
113                     <!-- !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! -->
114                     <executable>C:\Program Files\Java\jdk1.8.0_241\bin\javac</executable>
115                     <compilerVersion>1.8</compilerVersion>
116                 </configuration>
117             </plugin>
118         </plugins>
119     </build>
120 </project>
```

Abbildung 5.1: Path-Angabe zur JDK muss angepasst werden.

Ein weiterer Fehler könnte der sogenannte CORS-Fehler der Ionic beim Ausführen der Ionic-Anwendung im Browser auftreten kann. Die gemeinsame Nutzung von Ressourcen aus verschiedenen Quellen (Cross-Origin Resource Sharing, CORS) ist ein Standard, der es einem Server erlaubt, die Politik der gleichen Herkunft zu lockern. Dies wird verwendet, um einige Anfragen aus verschiedenen Quellen explizit zuzulassen und andere abzulehnen. Wenn eine Website beispielsweise einen integrierbaren Dienst anbietet, kann es notwendig sein, bestimmte Einschränkungen zu freizugeben. [Moz04]

Eine mögliche Fehler-Quelle kann sein, dass der Browser CORS-Anfrage ablehnt. Je nach Browser zum Beispiel Chrome werden einige Plugins angeboten, welche ermöglichen CORS-Anfrage zu versenden und zu empfangen. Für Google Chrome wäre dies Allow CORS: Access-Control-Allow-Origin". Aber auch für Firefox sollte ein ähnliches Plugin zur Verfügung stehen.

6 Ausblick

In diesem Kapitel wird auf Erweiterungen und Ideen eingegangen, welche während der Entwicklung des Projektes entstanden sind und um welche die App in zukünftigen Projekten erweitern werden könnte.

- Die Möglichkeit Bilder zu veröffentlichen und zu bearbeiten
- Direkt zu Bildern oder Textnachrichten ein Kommentar zu hinterlassen
- Beiträge über ein "Like" positiv oder negativ zu bewerten
- Zugriff auf das Profil von anderen Personen
- Die Möglichkeit ein Profilbild hinzuzufügen
- Das Sortieren beziehungsweise Filtern von Nachrichten
- Das Sortieren beziehungsweise Filtern von Freunden
- Den eigenen Account löschen

Literaturverzeichnis

- [Gae19] Gael Thomas. A beginner's guide to Docker — how to create your first Docker application [online]. Hrsg. von freeCodeCamp. 2019.
Verfügbar unter: <https://www.freecodecamp.org/news/a-beginners-guide-to-docker-how-to-create-your-first-docker-application-cc03de9b639f/>.
Zuletzt geprüft am: 09. 04. 2020.
- [Moz04] Mozilla. CORS errors [online]. 10.04.2020.
Verfügbar unter: <https://developer.mozilla.org/de/docs/Web/HTTP/CORS/Errors>.
Zuletzt geprüft am: 10. 04. 2020.