



**Hochschule  
Kaiserslautern**  
University of  
Applied Sciences

**Informatik und  
Mikrosystemtechnik**  
Zweibrücken

Fachhochschule Kaiserslautern  
- Campus Zweibrücken -

Fachbereich für IMST

## **Installationsanleitung**

**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:** 10.05.2020

**Betreuer:** Prof. Dr. Michael Bender

# Inhaltsverzeichnis

<b>1</b>	<b>Installationsanleitung</b>	<b>1</b>
1.0.1	Projektstruktur . . . . .	1
1.1	Docker . . . . .	1
1.2	Spring Boot . . . . .	3
1.3	Ionic . . . . .	5
1.3.1	Fehlerbehandlung . . . . .	7
1.4	Ausführen . . . . .	7
1.4.1	Start Backend . . . . .	7
1.4.2	Start Frontend . . . . .	7
1.4.3	Fehlerbehandlung . . . . .	8
	<b>Literaturverzeichnis</b>	<b>III</b>

# 1 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 leichtgewichtige Anwendungen zu sein, welche mittels Plugins (Extension) um benötigte Funktionen erweiterbar ist. Es kann sowohl zum Entwickeln und Starten des Backends als auch für das Frontend genutzt werden.

## 1.0.1 Projektstruktur

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

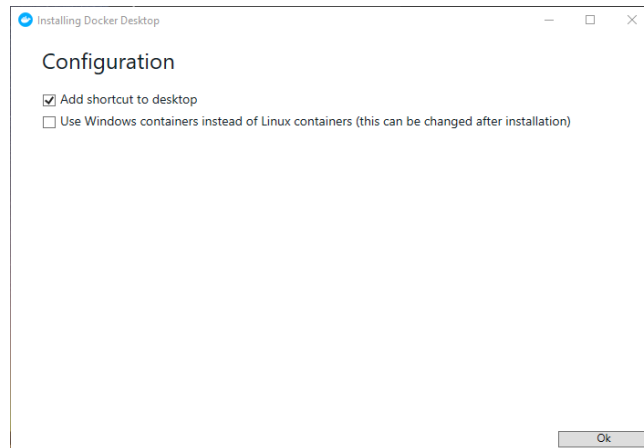
## 1.1 Docker

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.

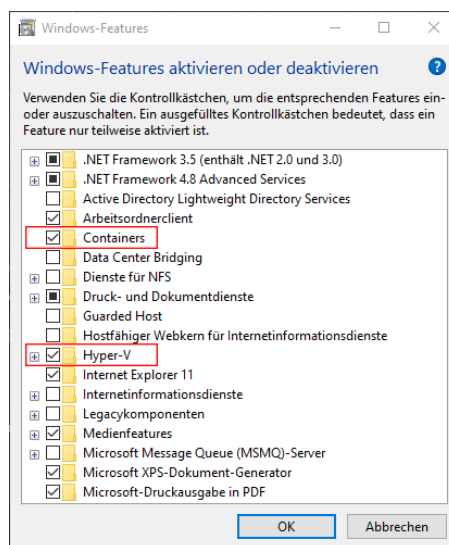
2. Während der Installation von Docker werden Sie gefragt, ob Sie Windows Container oder Linux Container verwenden möchten. Stellen Sie sicher, dass Sie das Kästchen in der angezeigten Option nicht ankreuzen (vgl. Abbildung). Der Docker verwendet also Linux Container. Docker-Container verwenden oft Linux-Komponenten,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



**Abbildung 1.1:** Das Feld "Windows-Containers" sollte frei bleiben.

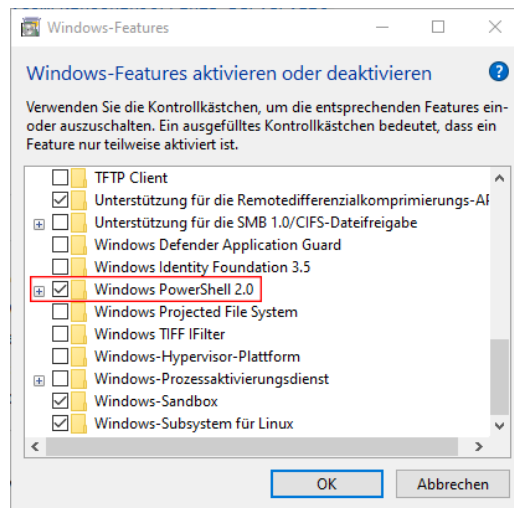
3. Unter Windows müssen Sie Hyper-V und Container aktivieren. Hyper-V und Container können unter den Windows-Funktionen aktiviert werden. Dazu drücken Sie zunächst die Windows-Taste + Q und geben dort die Windows-Features ein. Aktivieren Sie dann Hyper-V und Container und klicken Sie auf "ÖK". Folgen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



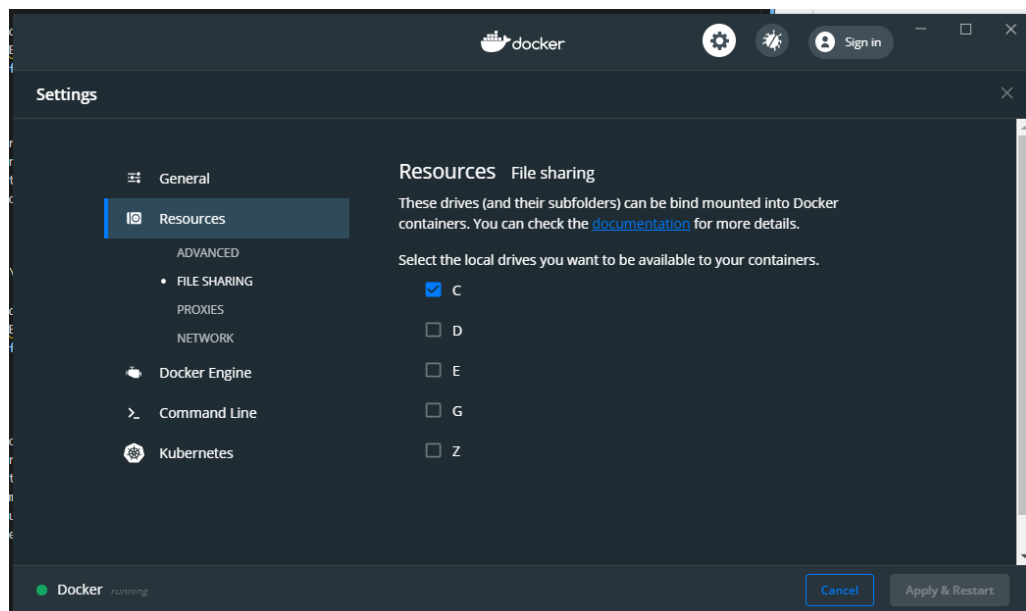
**Abbildung 1.2:** Benötigte Windows Features - Containers und Hyper-V.

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 1.3:** 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 1.4:** Freigabe einzelner Partitionen für Docker.

## 1.2 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

## 1 Installationsanleitung

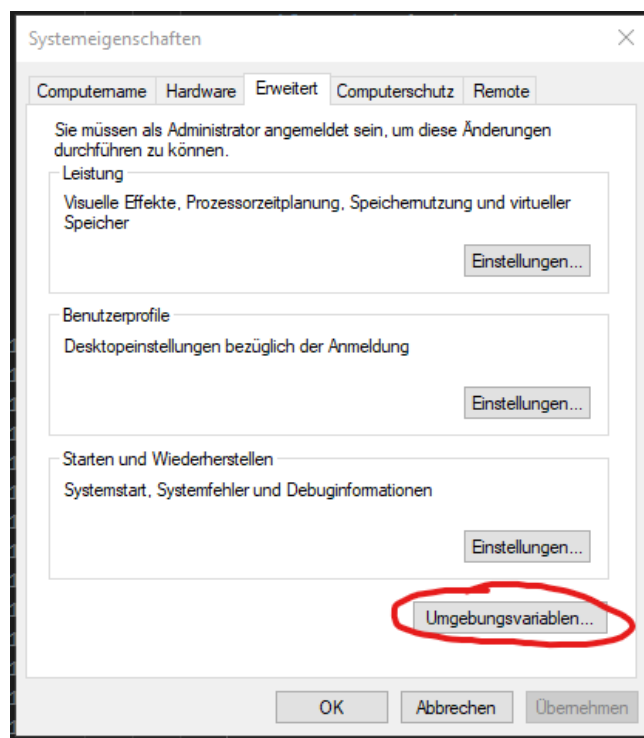
---

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 1.5:** Umgebungsvariable bearbeiten.

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

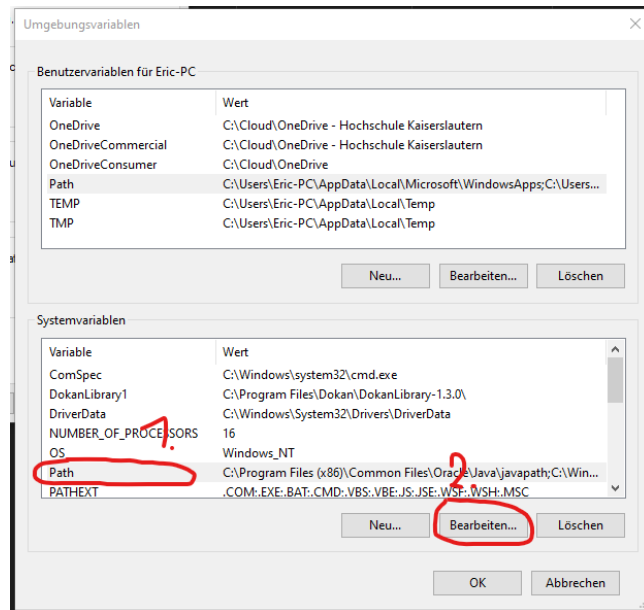


Abbildung 1.6: Umgebungsvariable bearbeiten.

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.

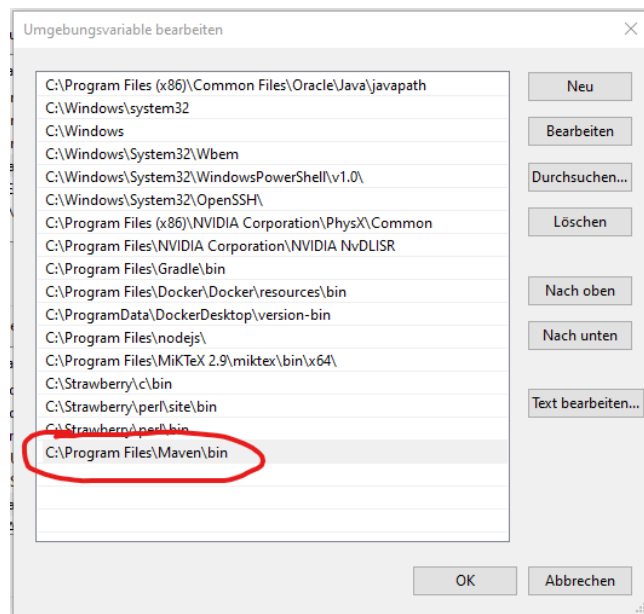


Abbildung 1.7: Umgebungsvariable hinzufügen.

### 1.3 Ionic

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

## 1 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>



### 1.3.1 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 weitere 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 1.2 (ab Punkt 4) dem Build-Management-Tool Maven dem System über die Umgebungsvariablen bekannt machen.

## 1.4 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.

### 1.4.1 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.

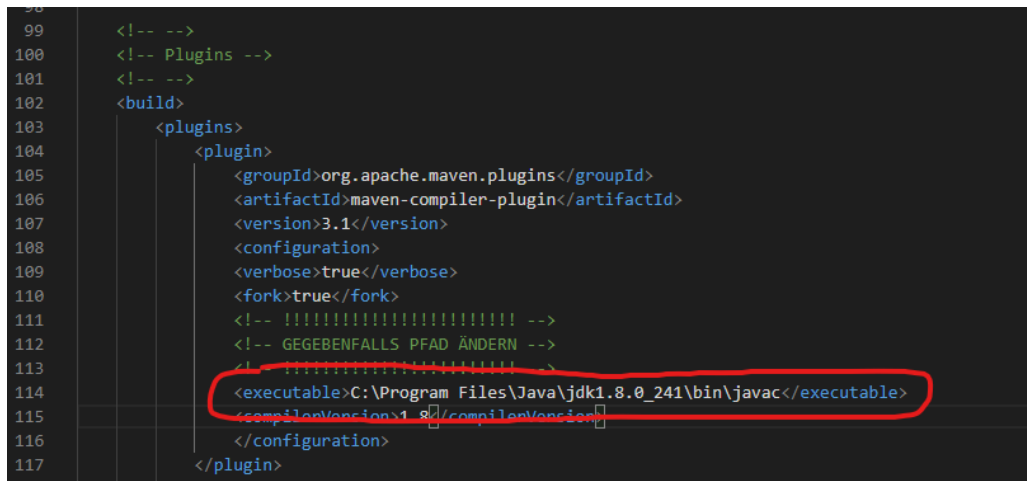
### 1.4.2 Start Frontend

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.

### 1.4.3 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                     <!-- GEBENFALLS 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>
```

**Abbildung 1.8:** 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.

# Literaturverzeichnis

- [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.