INSTALLATIONSANLEITUNG

* Git-Repository klonen (bevorzugt in Visual Studio Code): <https://github.com/deutho/DKE-PR>
* Download node.js: <https://nodejs.org/en/download/>
* Installiere Angular
  + öffne node.js command prompt
  + navigiere zu project root
  + tippe „npm install -g @angular/cli“ ein und führe es mit der Return-Taste aus

## Login/Registration Service Konfiguration (Port: 3000)

* installiere den MySQL Community Server (inkl. Workbench): <https://dev.mysql.com/downloads/mysql/>
* bei Installation, Passwort auf „**rootroot**“ setzen
* neue Verbindung erstellen (Hostname und Port: Standard; user: **root**; password: **rootroot**)
* neues Schema erstellen: in SCHEMAS -> rechte Maustaste -> create Schema
* als Name „LoginRegistrationDb“ eingeben
* neuen SQL-Tab erstellen, um eine Query ausführen zu können
* users.sql importieren/kopieren in Schema in MySQL und ausführen
  + users.sql findet man unter ..\Backend\LoginRegistrationService

Datenbank sollte nun lauffähig sein -> nächster Schritt: API starten

* öffne neues Terminal in VS Code
* navigiere zu ..\ Backend\LoginRegistrationService
* tippe „npm install“ ein, um Abhängigkeiten zu installieren
* tippe „npm start“ ein, um Service zu starten

## Networking Service Konfiguration (Port: 8083)

* installiere die Neo4j Desktop Version: <https://neo4j.com/download/>
* starte Neo4j Desktop
* erstelle eine lokale Neo4j Datenbank
* deaktiviere die Authentifizierung für erstellte Datenbank
  + öffne den lokalen Root von neo4j **..\Neo4jDesktop\relate-data\dbmss\dbms-*name*\conf**
  + öffne die Datei neo4j (conf-default-file)
  + setze dbms.security.auth\_enabled=**false** (das ist standardmäßig auf true gesetzt)
* öffne das Projekt in IntelliJ, da dieser Service in Java geschrieben wurde
* pom.xml als Maven-Projekt bestimmen und pom.xml installieren
* starte die Spring Boot Application: Application\_neo4j.java

## Notification Service Konfiguration (Port: 5050)

* Installiere Redis: <https://redis.io/download>
* Windows Installer: <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.msi>
* Öffne ein neues Terminal in VS Code
* Navigiere zu **..\Backend\NotificationService**
* Tippe „**npm install**“ ein um die Abhängigkeiten zu installieren
* Tippe „**npm start**“ ein um den Service zu starten

## Posting Service Konfiguration (Port: 27017)

* installiere MongoDB Community Server: <https://www.mongodb.com/try/download/community>
* erstelle eine neue Datenbank -> database-name: **postingDb** ; collection- name: **postings**
* öffne neues Terminal in VS Code
* navigiere zu **..\Backend\PostingService**
* tippe „**npm install**“ ein um die Abhängigkeiten zu installieren
* tippe „**npm start**“ ein um den Service zu starten

## Message Broker Konfiguration (Port: 15672)

* installiere Erlang als Admin: <https://www.erlang.org/downloads>
* wichtig: Zuerst Erlang installieren!!
* installiere RabbitMQ als Admin: <https://www.rabbitmq.com/download.html>
* öffne RabbitMQ Command Prompt
* tippe **rabbitmq-plugins enable rabbitmq\_management** ein
* öffne im Browser **localhost:15672**
* Verwende folgende Anmeldedaten: username: **guest** password: **guest**

## Starte Frontend (Port: 4200)

* öffne ein neues Terminal in VS Code und navigiere zum Ordner **..\Frontend**
* tippe „**npm install**“ ein um die Abhängigkeiten zu installieren
* tippe „**ng serve**“ ein um das Frontend zu starten
* öffne Google Chrome mit deaktivierter Web-Security, um mit den verschiedenen Services im Frontend kommunizieren zu können:
  + öffne cmd als Administrator
  + tippe „**cd C:\Program Files\Google\Chrome\Application**“ ein
  + tippe „**chrome.exe --user-data-dir="C:/Chrome dev session" --disable-web-security**“ ein
  + öffne im Browser **localhost:4200**, registriere einen Account und lege los 😊