

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
 - o öffne node.js command prompt
 - o navigiere zu project root
 - o tippe „npm install -g @angular/cli“ ein und führe es mit der Return-Taste aus

1. 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
 - o 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

2. 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
 - o öffne den lokalen Root von neo4j ..\Neo4jDesktop\relate-data\dbmss\dbms-name\conf
 - o öffne die Datei neo4j (conf-default-file)
 - o 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

3. 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

4. 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

5. 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**

6. 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:
 - o öffne cmd als Administrator
 - o tippe „**cd C:\Program Files\Google\Chrome\Application**“ ein
 - o tippe „**chrome.exe --user-data-dir="C:/Chrome dev session" --disable-web-security**“ ein
 - o öffne im Browser **localhost:4200**, registriere einen Account und lege los 😊