

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 die Windows Command Prompt.
 - o Tippe „**npm install -g @angular/cli**“ ein und führe es mit der Enter-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 („...\\Backend\\LoginRegistrationService“) in das neue Schema importieren oder den Inhalt der users.sql Datei in die MySQL-Workbench kopieren und ausführen .

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.

API sollte nun gestartet sein.

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 die neu 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).

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

- Öffne das Projekt in IntelliJ, da dieser Service in Java geschrieben wurde (Auch in VS-Code möglich mithilfe von Plugins).
- Pom.xml als Maven-Projekt bestimmen und pom.xml installieren.
- Starte die Spring Boot Application: "Application_neo4j.java" .

API sollte nun gestartet sein.

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>)
- Redis sollte automatisch gestartet worden sein und bereits laufen. Ist dies nicht der Fall, gehe zu „C:\Program Files\Redis“ (bei Default Installation) und führe „redis-server.exe“ aus.

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

- 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.

API sollte nun gestartet sein.

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** .
- Klicke auf „Connect“, um die Datenbank einzuschalten.

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

- 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.

API sollte nun gestartet sein.

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> .
- Gehe zu „..\Programs\RabbitMQ Server“ und führe „**RabbitMQ Service – start**“ aus - dies startet den Server.

Message Broker sollte nun gestartet sein. Um das Webterminal zu sehen, führe folgendes aus:

- Öffne RabbitMQ Command Prompt (..\Programs\RabbitMQ Server\RabbitMQ Command Prompt (sbin dir)).
- 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.
- Sollte Nutzung und Hosting der Mikroservices am selben System stattfinden, öffne Google Chrome mit deaktivierter Web-Security. Führe dazu folgende Schritte auf Windows aus:
 - o Öffne die Windows Command Prompt 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 😊.