

Überblick

"Stock-Portfolio" ist ein Studienprojekt zum Erlernen der Fähigkeiten, die ein moderner "Fullstack-Entwickler" benötigt, um Enterprise-Applikationen als Mitglied in einem agilen Scrum-Team erstellen zu können. Das Projekt soll eine Web- und Mobil-Anwendung zur Verwaltung von Aktien-Depos realisieren und setzt dabei auf eine Mehrschichten-Architektur mit Frontend, Middleware (Backend-for-Frontend) und Datenbank-Backend.

Als Programmiersprache kommt in erster Linie Javascript in Verbindung mit NodeJS zum Einsatz. Die verwendeten Javascript-Frameworks sind u. a. VueJS für das Frontend und ExpressJS für die Middleware. Die Middleware stellt in erster Linie eine Rest-API für Service-Requeste des Frontends zur Verfügung. Darüber werden auch sämtliche Datenbank-Zugriffe vorgenommen. Für die Beschaffung von Aktien-Daten sollen externe Rest-APIs konsumiert werden, z. B. Yahoo Finance. Das Datenbank-Backend soll mit Postgresql realisiert werden. Zum Authentifizieren und Autorisieren des Benutzers soll ein JWT-Server (Keycloak) zum Einsatz kommen. Als Test-Framework haben wir uns für Mocha entschieden.

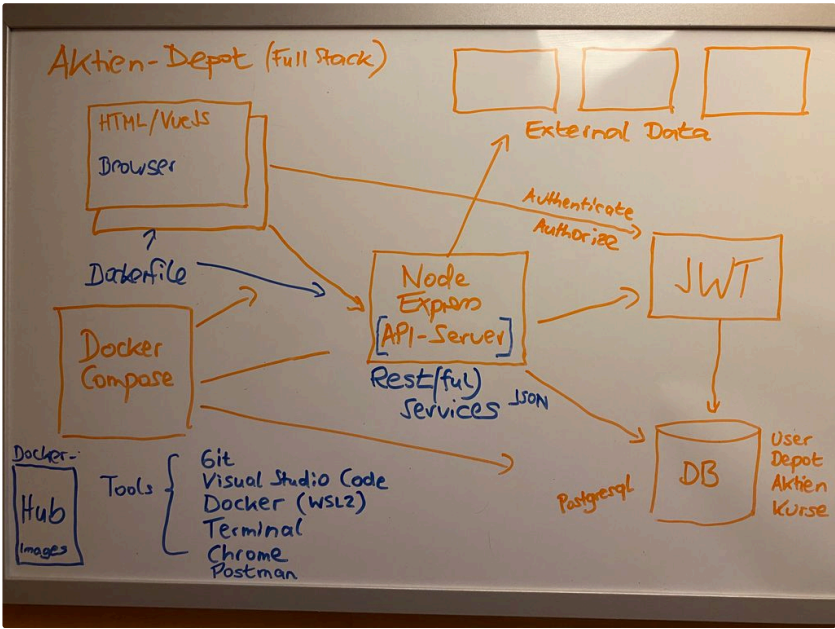
Der Toolstack umfasst folgende Programme: Visual Studio Code, Git (Gitlab Community Edition), cURL, Postman, Google Chrome, Windows Terminal und HeidiSQL.

Die gesamte Anwendung soll von Anfang an in mehreren Containern laufen. Dazu wird Docker Compose eingesetzt und dies mit Hilfe von Visual Studio Devcontainer schon in der Entwicklungsphase genutzt.

Zur Dokumentation des Projektes werden folgende Tools eingesetzt: Jira für das Issue-Tracking, Confluence für die Gesamt-Dokumentation, Miro für Brainstorming, Visual Paradigm Community Edition als UML Tool und SqldbM für die Modellierung der Datenbank. Für Präsentationen wird Microsoft Powerpoint genutzt.

Zum Abschluss - als Add-On - wird noch eine Auslieferung der fertigen Software in einen Kubernetes-Cluster angestrebt. Dazu soll im Rahmen einer lokalen Kubernetes-Installation mittels Minikube oder k3s auch eine kleine Devops-Pipe in Gitlab realisiert werden.

Das folgende Bild veranschaulicht die grundlegende Architektur des Studienprojektes.



Core team

Leon Hessel Team Lead	Thomas Hessel Tech Lead	
--------------------------	----------------------------	--

Schnell-Navigation

- Glossar
- Anleitungsartikel
- Workshops
- Datenmodell

Nützliche Links

Link	Description
Confluence 101: organize your work in spaces	Chances are, the information you need to do your job lives in multiple places. Word docs, Evernote files, email, PDFs, even Post-it notes. It's scattered among different systems. And to make matters worse, the stuff your teammates need is equally siloed. If information had feelings, it would be lonely. But with Confluence, you can bring all that information into one place.
Confluence 101: discuss work with your team	Getting a project outlined and adding the right content are just the first steps. Now it's time for your team to weigh in. Confluence makes it easy to discuss your work - with your team, your boss, or your entire company - in the same place where you organized and created it.
Confluence 101: create content with pages	Think of pages as a New Age "document." If Word docs were rotary phones, Confluence pages would be smart phones. A smart phone still makes calls (like their rotary counterparts), but it can do so much more than that