Implementation of all REST API endpoints (CRUD) for the new entity "DailyMessage"

PROJEKTARBEIT

für die Prüfung zum

Bachelor of Engineering/Bachelor of Science

des Studiengangs Informatik

Studienrichtung Angewandte Informatik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Marija Cvetkovic

Abgabedatum

Matrikelnummer

Kurs

Ausbildungsfirma

Betreuer

Erklärung

(gemäß §5(3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. 9. 2015)

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: „ *Aufsetzen einer DevOps-Pipeline von Code bis zum Cloud Deployment*“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum Unterschrift

**Sperrvermerk**

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Ausbildungsstätte vorliegt.

# Ziel dieses Projektes

Mein Betrieb bietet seit einiger Zeit ein Produkt unter dem Name seiriQ auf dem Markt an. In dem Produkt “seiriQ” will mein Team in Zukunft ein Widget anbieten, dass einen täglich wechselnden Text anzeigen soll. Dazu ist es erforderlich die benötigten API Endpoints im Backend (Java) zu implementieren.

# Allgemeines

## Tools

Um das Projekt erfolgreich durchzuführen, muss ich allererst die folgende Programme herunterladen und damit die notwendige Voraussetzungen erfüllen:

* Java >11

Abbildung : Java 11

* Maven (mwn)

Abbildung : Maven (mwn)

* ActiveMQ Classic

Abbildung : Active MQ Apache

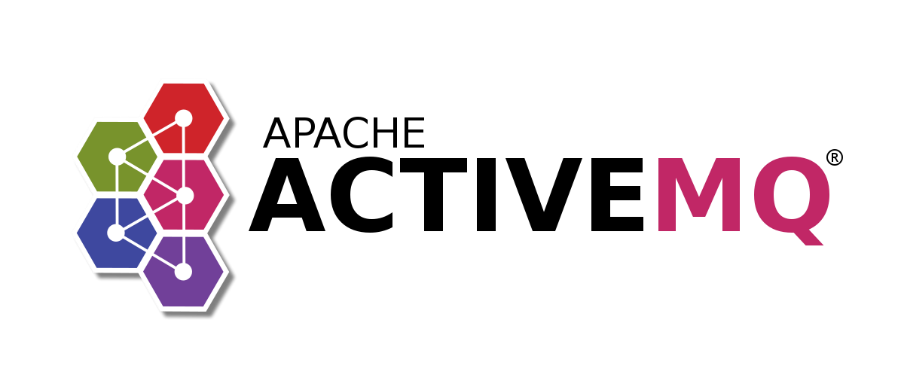
* MongoDB Community Server >= 4.4

Abbildung : MongoDB Community Server

* Node js (LTS)

Abbildung :Node js (LTS)

* Git

Abbildung : git

Amazon Corretto ist eine kostenlose, plattformübergreifende und produktionsbereite Distribution des Open Java Development Kit (OpenJDK).

**Maven** ist ein auf [Java](https://de.wikipedia.org/wiki/Java_(Programmiersprache)) basierendes [Build-Management](https://de.wikipedia.org/wiki/Erstellungsprozess" \o "Erstellungsprozess)-Tool der [Apache Software Foundation](https://de.wikipedia.org/wiki/Apache_Software_Foundation), mit dem insbesondere die Erstellung von Java-Programmen standardisiert verwaltet und durchgeführt werden kann.

**Apache ActiveMQ** ist ein [freier](https://de.wikipedia.org/wiki/Open_Source) Message Broker, der vollständig das [Java Message Service 1.1](https://de.wikipedia.org/wiki/Java_Message_Service) (JMS) implementiert. Apache ActiveMQ verändert die Verbindungen eines Netzwerks zwischen bestehenden Anwendungen, indem die [synchrone Kommunikation](https://de.wikipedia.org/wiki/Synchrone_Kommunikation) zwischen zu integrierenden Applikationen in eine [asynchrone Kommunikation](https://de.wikipedia.org/wiki/Asynchrone_Kommunikation) umgewandelt wird.

**MongoDB** Community Edition ist eine kostenlose Open-Source-Datenbank, die eine beliebte Option für die Bereitstellung moderner Anwendungen ist. Es handelt sich um eine dokumentenbasierte und verteilte Datenbank, die für die Nutzung von Cloud-Technologien entwickelt wurde.

**Node.js** wird aufgrund seiner Single-Thread-Natur in erster Linie für nicht-blockierende, ereignisgesteuerte Server verwendet. Es wird für herkömmliche Websites und Back-End-API-Dienste verwendet, wurde aber mit Blick auf Push-basierte Echtzeit-Architekturen entwickelt.

**Git** ist ein DevOps-Tool, das für die Quellcodeverwaltung verwendet wird. Es handelt sich um ein kostenloses und quelloffenes Versionskontrollsystem, mit dem sich kleine bis sehr große Projekte effizient verwalten lassen. Git dient der Verfolgung von Änderungen am Quellcode und ermöglicht mehreren Entwicklern die Zusammenarbeit bei der nicht-linearen Entwicklung.

## Start des Projektes

Sobald ich alle diese Tools heruntergeladen habe, sowie npm (npm ist in Node.js enthalten) führe ich folgendes aus:

- yarn (npm install --global yarn)

- angular-cli (npm install -g angular-cli)

Was auch nützlich ist herunterzuladen ist **Advanced Rest Client.** Das ist eine einfache und übersichtliche Benutzeroberfläche mit der man APIs testen kann.

NBEnvironment Skripts, nämlich die ZIP-Datei NBEnvironmentSkripts.zip dient zu dem einfachen Start der Entwicklungsumgebung. // Wenn Sie es richtig entzippt ist, können Sie jetzt einfach auf die Batchdateien doppelklicken, um den gewünschten Teil zu starten. //

Um mit meinem Projekt anfangen zu können, klone ich die folgenden Repositorien mit Git:

* main repo

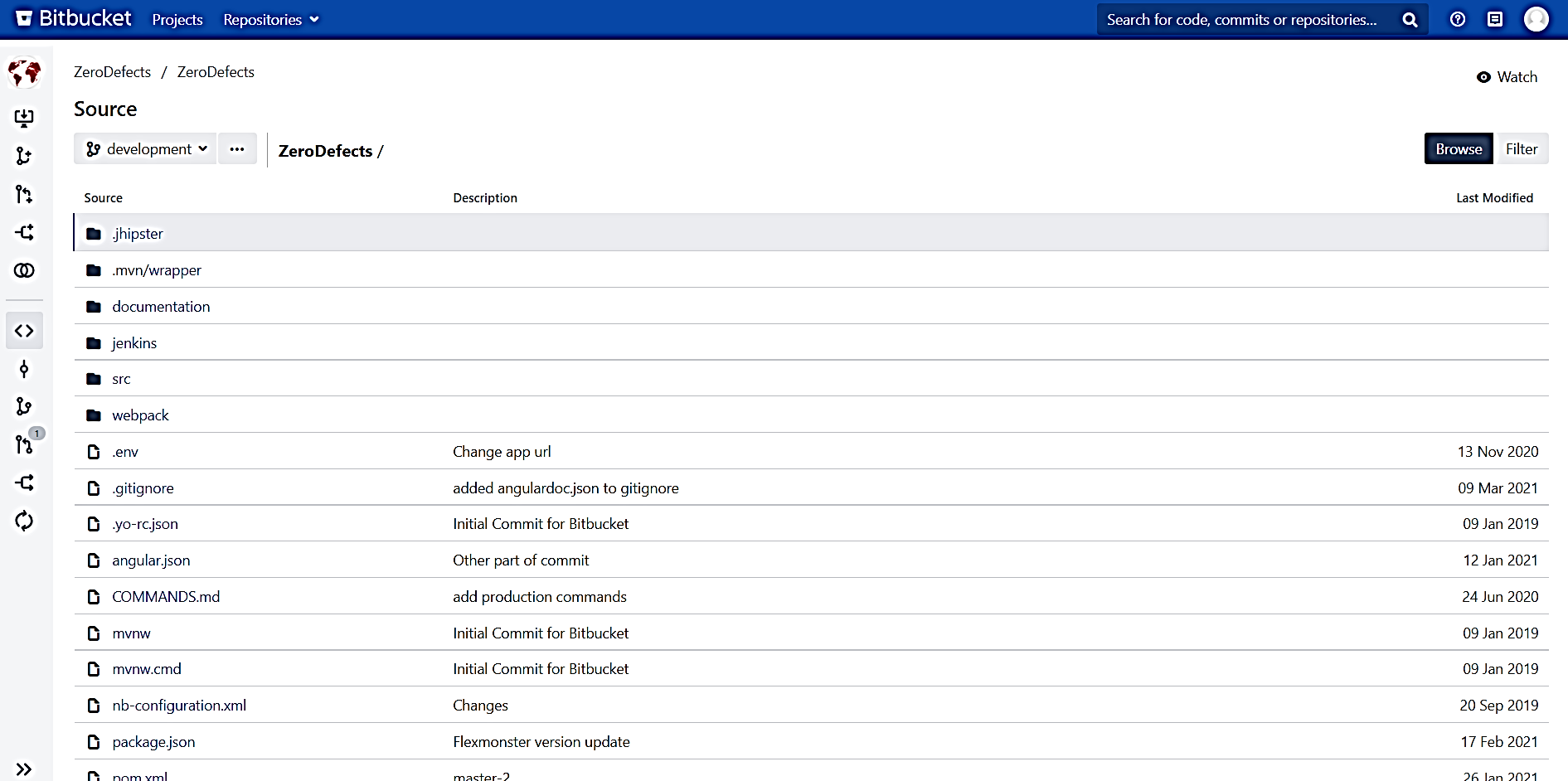


Abbildung : Main Repo

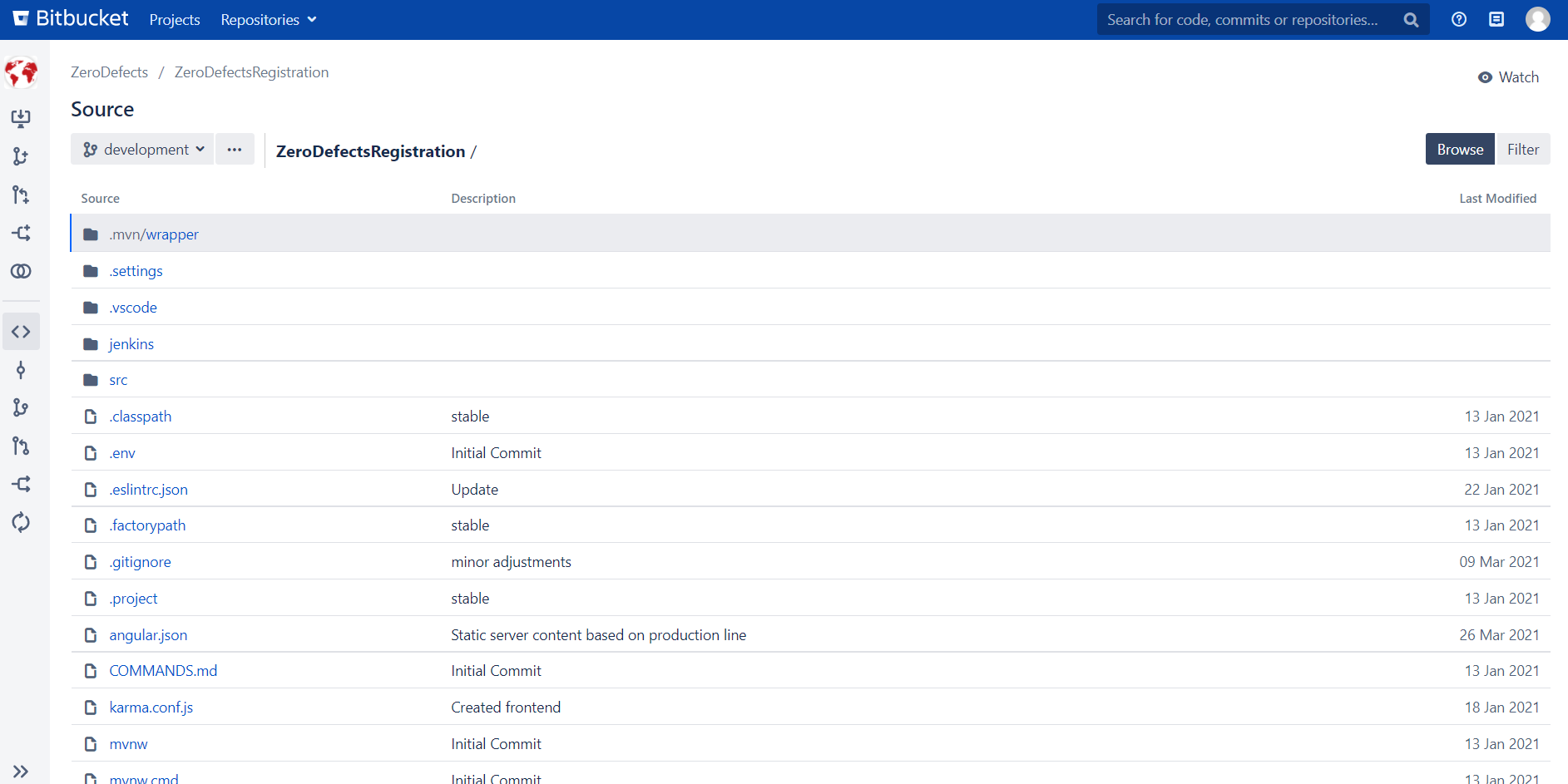
* Registration Service (für neue Tenant Registration)

Abbildung : Registration Service

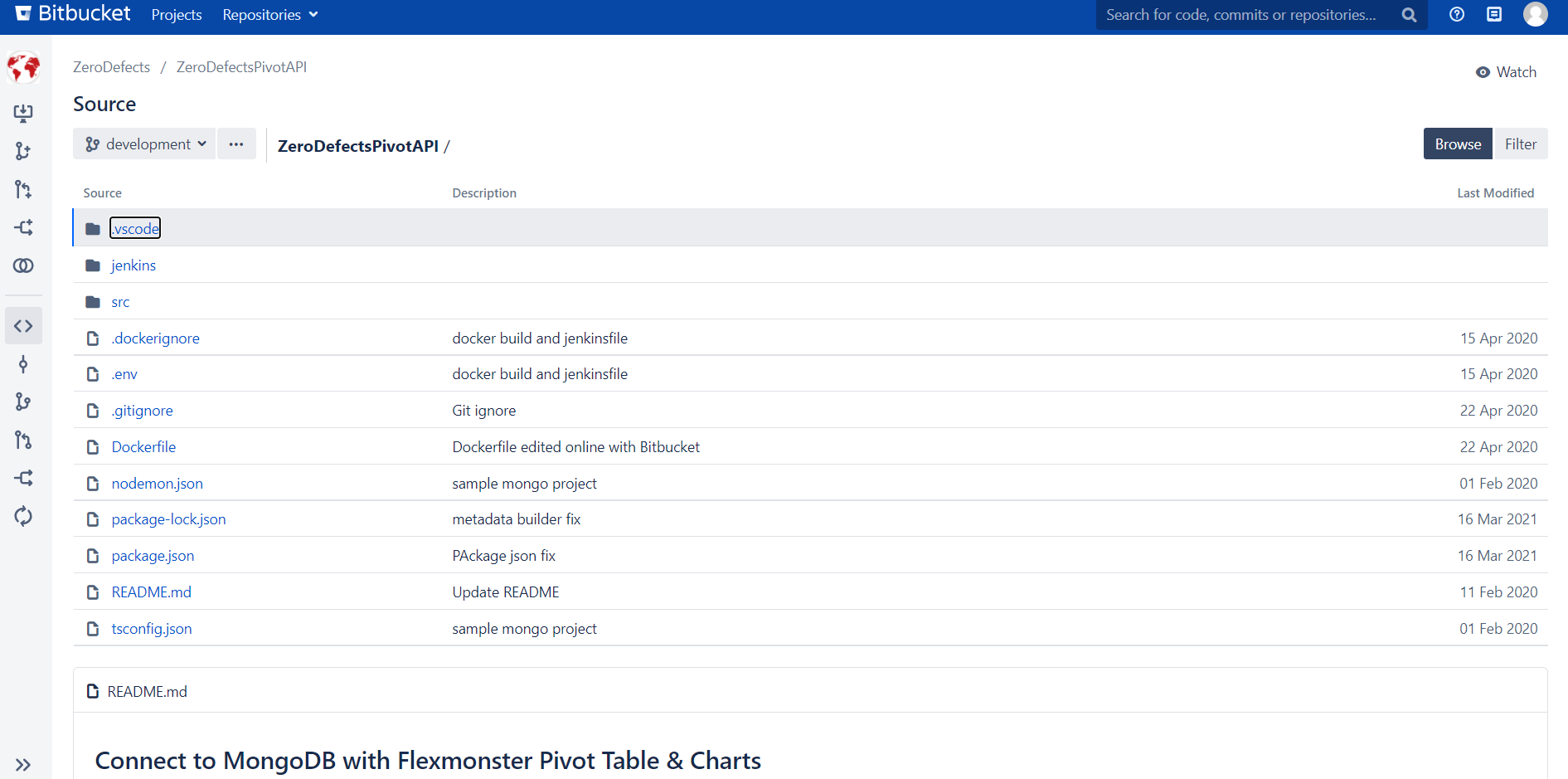
* PivotAPI Microservice

Abbildung : PivotAPI Microservice

* Active MQ JWT Plugin

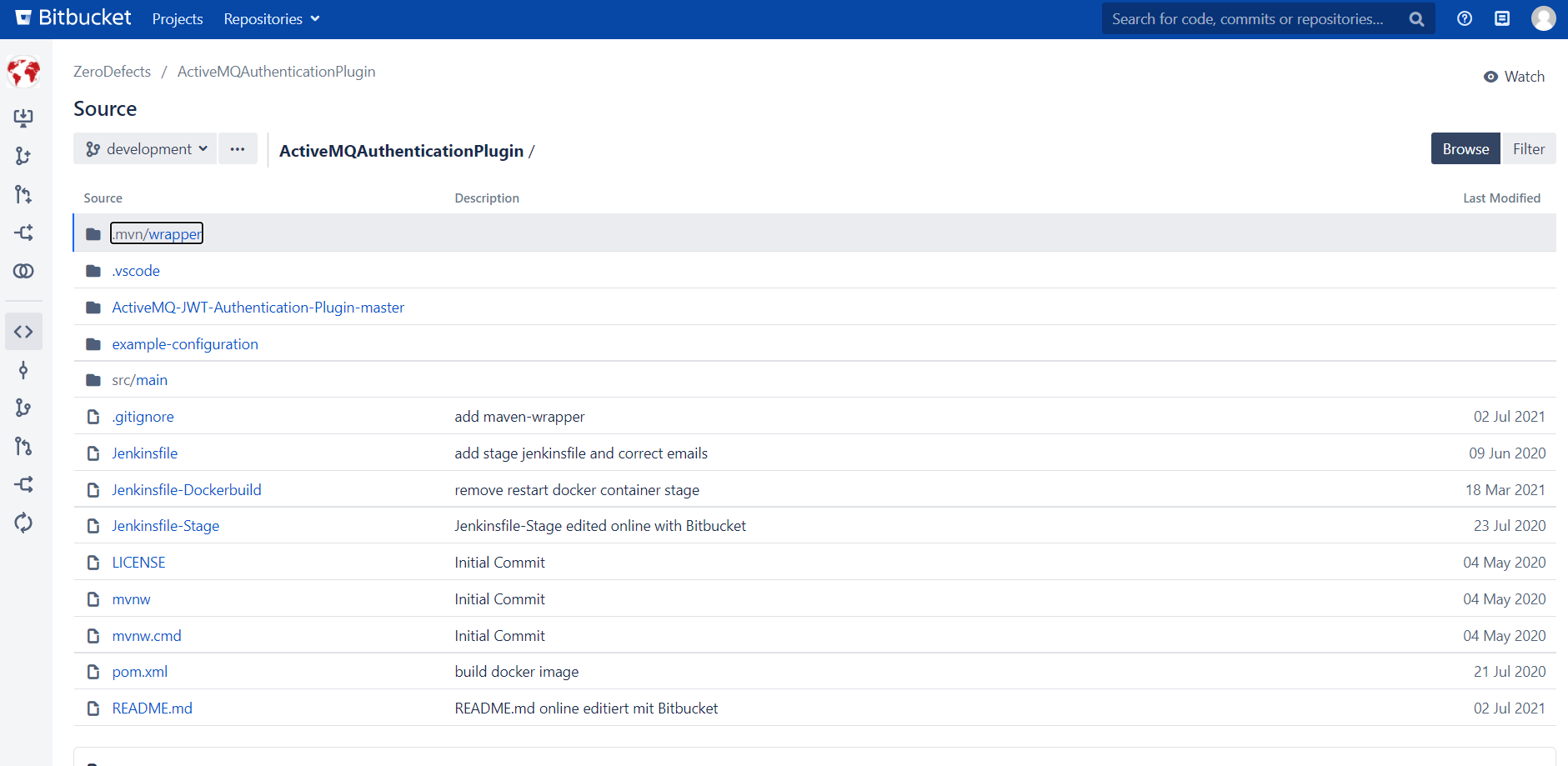


Abbildung : Active MQ JWT Plugin

Nun bevor ich mit meinem Projekt starte, muss ich die folgenden Abhängigkeiten auf meinem Rechner installieren und konfigurieren:

1. Node.js: Ich werde Node verwenden, um einen Entwicklungs-Webserver zu betreiben und das Projekt zu erstellen. // Ich kann Node entweder aus dem Quellcode oder als vorgefertigtes Paket installieren.
2. Yarn: Ich verwende Yarn, um Node-Abhängigkeiten(Dependencies) zu verwalten. Ich kann Yarn entweder aus dem Quellcode oder als vorgefertigtes Paket installieren.

Nach der Installation von Node bon ich in der Lage, den folgenden Befehl auszuführen, um Entwicklungswerkzeuge zu installieren.

*yarn install*

Es wird yarn-Skripte und Webpack als unser Build-System verwendet.

Nun führe ich die folgenden Befehle in zwei separaten Terminals aus, um eine angenehme Entwicklungsumgebung zu schaffen, in der Ihr Browser automatisch aktualisiert wird, wenn sich Dateien auf Ihrer Festplatte ändern.

*./mvnw*

*yarn start*

Yarn wird auch zur Verwaltung der in dieser Anwendung verwendeten CSS- und JavaScript-Abhängigkeiten verwendet. Die Dependencies könne aktualisieret werden, indem man eine neuere Version in package.json angeben. Sie können auch *yarn update* und *yarn install* ausführen, um Abhängigkeiten zu verwalten. // Fügen Sie das Hilfe-Flag zu jedem Befehl hinzu, um zu sehen, wie Sie ihn verwenden können. Zum Beispiel: *yarn help* update.

Der Befehl *yarn run* listet alle Skripte auf, die für dieses Projekt zur Verfügung stehen.

Einrichtung des Repository - Pivot API

Ich sollte die vorhandenen Synsoft-Git-Anmeldeinformationen in package-lock.json und package.json ersetzen:

Strings werden wie folgt aufgebaut:

"git+http://<username>:<password>@git....."

Auch sollten die definierten Abhängigkeiten in package.json installiert werden:

*npm install*

Am Ende starte ich das Beispielprojekt:

*npm run build*

*npm run start*

Aktives MQ JWT-Plugin einrichten

Dieses Plugin wird nur für die Nutzung der Websocket-Funktionen von seiriQ benötigt. Es hat die Aufgabe, jwt Token als gültige Passwörter für die Benutzer mit dem vorkonfigurierten Benutzernamen zu akzeptieren. Zum Beispiel Benutzername *jwtUser*, Passwort: *eyJhbGciOi*....

Der Standardbenutzer u: *admin* p: *admin* und andere Benutzer sind weiterhin verfügbar. Für Nachrichten von Nicht-JWT-Benutzern muss der jms\_user\_id-Header mit der korrekten Benutzerkennung aus der Datenbank gesetzt werden, damit sie funktionieren.

Schritte zum Einrichten des JWT-Plugins:

1. Verschieben in dem ActiveMQ-Installationsverzeichnis den lib-Ordner mit der ActiveMQ jar-Datei, die man auf der rechten Seite findet.

1b. Wenn es Probleme mit der jar-Datei gibt, zum Beispiel wenn sie veraltet ist, klone ich das Projekt und eine gültige jar-Datei erstellen, indem ich *./mvnw clean install* ausführe.

2. Dann einfügen in conf/activemq.xml innerhalb des <broker>-Tags den <plugins>-Tag, falls er noch nicht vorhanden ist, und auch einfügen die Konfiguration aus der Mitte.

3. Diesen Schritt kann ich überspringen. Die Eigenschaft tokenHeader ist der Benutzername aller jwt-Benutzer, er kann in diesem Teil der Konfiguration geändert werden. Property masterSecretKey ist der jwt-Schlüssel aus der application-dev.yml des seiriq-Projekts, der standardmäßig erwartete Schlüssel ist bereits in der mitgelieferten Plugin-Konfiguration vorkonfiguriert.

MIETE !!!!!!!!!!!!!