

M151 Projekt

Noah Ziltener

Was macht die Applikation?

- Auktionshaus für Autos
- User ansehen, erstellen, abändern und löschen
- Auktionen ansehen (Vergangen, Offene, Gewonnen, von mir erstellte)
- Auktionen erstellen (mit Auto, Startpreis & Direktkaufpreis)
- Auf eine Auktion bieten oder Direktkaufen
- Alle meine Gebote anzeigen
- Auktionen laufen nach 5 min oder nach einem Direktkauf ab

Gewählte Technologien

- Java 11
- Java Spring
- DB Migrationen: Flyway
- Redis Cache
- Docker (DB & Cache)

Schnittstellenbeschreibung

- REST Schnittstelle
- Spring
- Basic Auth
- Alle Beispiel in Git Repo (Postman)

Schnittstellenbeschreibung User

- Endpunkte um Informationen zu aktuellen User, eine Liste von allen Usern bekommt. Auch können neue User angelegt, aktualisiert und gelöscht werden.
- Get /user/info
- Get /user/
- POST /user/ (user json)
- PUT /user/{id}
- DELETE /user/{id}

Schnittstellenbeschreibung Auktionen

- Schnittstellen um alle offenen, geschlossenen, von mir erstellten und gewonnen Auktionen zu sehen. Auch noch Detail zu einer einzelnen Auktion und das Erstellen von einer neuen sind vorhanden.
 - Auktionen bearbeiten wurde nicht umgesetzt, dafür viel andere Funktionen
 - Auktionen werden automatisch nach 5 min geschlossen. (Manchmal wenig länger, da mit Scheduled Task gelöst)
-
- Get /auction/
 - Get /auction/{id}
 - GET /auction/history
 - GET /auction/myAuctions
 - GET /auction/wonAuctions
 - POST /auction/ (auction json)

Schnittstellenbeschreibung Bid & Direkt

- Schnittstellen für Gebote und Direktkäufe abzuschliessen. Auch kann ich meine Gebote Anschauen.
- POST /auction/bid/{id} (bid json)
- POST /auction/directbuy/{id}
- GET /auction/bid/

Herausforderungen

- Nicht viel Erfahrung mit Java
- Keine Erfahrung mit Spring, Flyway oder Redis
- Wenig Erfahrung mit Docker
- Zeit

Lessons learned & Erfahrungen

- Java, Spring & Docker sind Cool
- Weiter Projekte mit Java

Demo

- Postman File in Repo