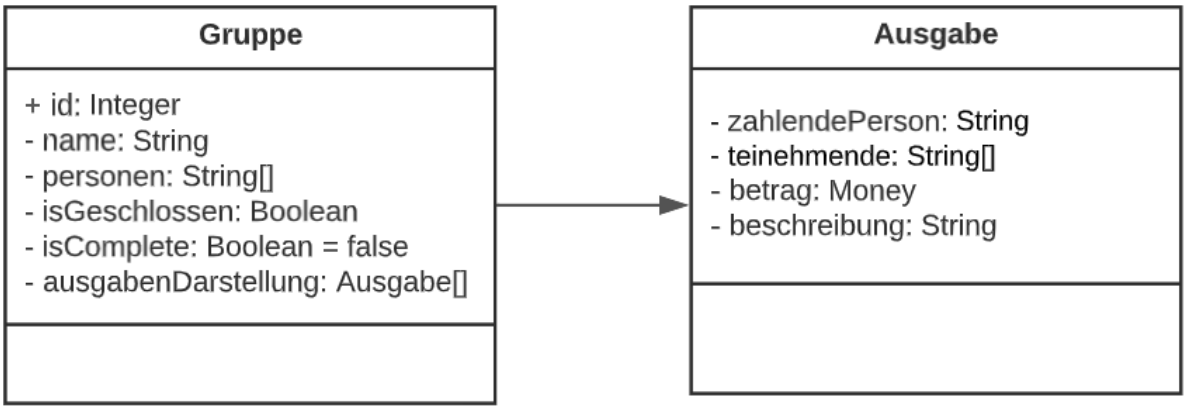
Splitter Dokumentation

Kontextabgrenzung und Komponentenstruktur:

Domain:

Die Gruppen-Klasse ist das Aggregate Root.

Ausgabe:
Das Ausgabe-Record ist ein zur Gruppen-Klasse dazugehöriges Value Objekt.



Der User wird anhand des eingeloggten GitHub Handles identifiziert und wird intern als String abgespeichert.

Im Gruppenobjekt werden die Transaktionen ausgeglichen und Ausgaben gespeichert.

Zur Domain gehören einige Helper Klassen:

- AusgabeAlsString:
  Parsen einer Ausgabe als String für die HTML-Ausgabe.
- AusgabeAlsStringMitPers:
  Hinzufügen der Information ob die Person an der Ausgabe beteiligt ist. Wird für die Hervorhebung der vom User getätigten Ausgaben im HTML verwendet.
- AusgleichHelper:
  Methoden die in der berechneAusgleich Methode der Gruppe verwendet werden.
- VonZuMitBetragAlsString:
  Parsen des VonZuMitBetrag-Records zu einem String für die HTML-Ausgabe.
- EUR:
  Shortcut für Money.of(xx, “EUR”).

- MoneyHelper:
  Zuständig für die Rundung eines Money-Objekts mit Hilfe des default roundings der Moneta Bibliothek.
- VonZuMitBetrag:
  Das VonZuMitBetrag-Record dient um den Zusammenhang zwischen einem zahlenden und einem kassierenden Nutzer zu modellieren.
- PersonMitBetrag:
  Das PersonMitBetrag-Record stellt den Zusammenhang zwischen einer Person und einem Geldbetrag dar.

Web (Interface-Schicht):

Im web-package befindet sich der WebController mit einigen Helper-Methoden. Darin verschachtelt befindet sich das api-package mit dem JSONController und weiteren Helper-Methoden für die API.

Service (Geschäftslogik-Schicht):

Im Service package liegen die Klasse SplitterService und das Interface SplitterRepository.

DB (Persistenz-Schicht):

Enthält die Implementierung des SplitterRepository Interface.

Datenbank-Modell:

Das Datenbank-Modell besteht aus zwei Tabellen, in denen Gruppen und Ausgaben gespeichert werden können. Als Primary Key wird in beiden Tabellen für die ID der serial Pseudotyp der PostgreSQL Datenbank verwendet, um eine AUTO INCREMENT Funktion zu gewährleisten.

Die Datenbank wird in einem Docker-Container betrieben, der aus dem Image postgres:14-alpine erzeugt wird.

Im Service sind einige Methoden mit @Transactional annotiert, um die Transaktionssicherheit der Datenbankzugriffe zu gewährleisten.

Die Methode findById im SpringDataSplitterRepo wird mit pessimistischem Write gelockt.