



Shopstantly EAI Projekt

Mario Allemann, Michael Tu, Philip Käppeli, Pascal Moll

Inhaltsverzeichnis

I.	Einleitung	2
II.	Vorgehen	2
III.	Tools	3
IV.	Datenbanken.....	3
V.	Dialog Flow.....	4
VI.	Fazit.....	5

Einleitung

Das Modul Enterprise Application Integration (EAI) im 5. Semester des Wirtschaftsinformatikstudium beinhaltet eine studentische Arbeit bei der in 4er Gruppen eine Applikation mit Microservices erstellt werden soll. Diese Projektarbeit beinhaltet die Programmierung von 3 Microservices (Inventory, Payment, Shipping), das Aufsetzen des Dialogflow-Bots, und das Deployment auf einem Webserver. Schlussendlich sollte es möglich sein einen Bestellvorgang via Dialogflow zu simulieren.

Vorgehen

1. Phase: Vorbereitung

Zuerst mussten alle Projektmitglieder sich in die Materie einlesen. Dann wurden die Rollen der Teammitglieder verteilt:

- Mario Allemann: Dialogflow-Bot aufsetzen + Datenbanken
- Michale Tu: Datenbanken planen und aufsetzen
- Pascal Moll: Spring Projekte erstellen + erstellen und führen der Dokumentation
- Philipp Käppeli: Verbindung aller Komponenten + Deployment auf Heroku

2. Phase: Entwicklung

Zuerst wurde gemeinsam die DAO's der Datenbanken festgelegt, um dann mittels der H2 Datenbank alle zu erstellen.

Die ersten Versuche das Projekt dann auf Heroku zu deployen liefen schief, erst nach einer längeren Nachforschung fiel auf, dass es anscheinend nicht möglich ist eine H2 Datenbank auf Heroku zu integrieren. Dies führte zum Beschluss des Projektteams mit MySQL Datenbanken zu arbeiten. Doch kurze Zeit später fiel auf, dass pro Heroku Projekt nur eine Datenbank initialisiert werden kann. Das Projektteam entschied sich deswegen 4 GIT, Spring-Boot und Heroku Projekte für die verschiedenen Services zu starten und diese dann im Anschluss zu verbinden.

Anschliessend konnten die Microservices programmiert werden und damit auch die Zugriffe auf die Datenbanken geschrieben werden. In einem weiteren Schritt wurde die Kommunikation zwischen den Microservices ermöglicht. Schlussendlich konnte man mit dem Dialogflow-Bot Bestellungen aufgeben.

3. Phase Testing

Die Applikation wurde getestet und es konnten erfolgreich die 3 Produkte (Reis, Pasta und Pizza) bestellt werden.

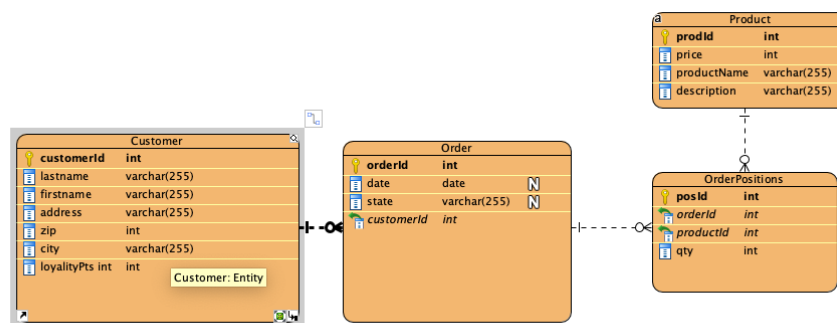
Tools

- Spring
- Postman
- Heroku
- Dialogflow
- Eclipse
- Github
- MySQL

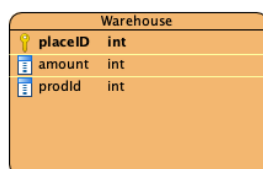
Datenbanken

Die Datenspeicherung wurde auf 4 verschiedene Datenbanken aufgeteilt, Order-Management und die 3 Microservices:

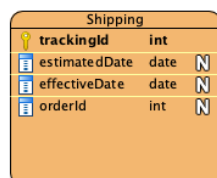
Order-Management:



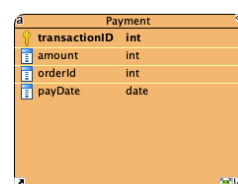
Warehouse



Shipping:



Payment



Dialog Flow

Events

Training phrases

Search training phrases

^

99

Add user expression

99

Ich habe Lust auf **Pasta**

99

Ich will **Pizza**

99

Ich möchte **Milch** bestellen

99

Pasta

99

Ich will **Milch** bestellen

Action and parameters

Enter action name

^

REQUIRED

PARAMETER NAME

ENTITY

VALUE

IS LIST

☐

product

@product

\$product

☐

☐

Enter name

Enter entity

Enter value

☐

+ New parameter

Abbildung 1 Dialogflow Angabe des Produkts

Training phrases

Search training phrases

^

99

Add user expression

99

3 Mal

99

1

99

3 Stück

99

Ich brauche **3** Kilogramm

99

2 Liter

99

2 Kilo

Action and parameters

OrderProduct OrderProduct-custom

^

REQUIRED

PARAMETER NAME

ENTITY

VALUE

IS LIST

☐

qty

@quantity

\$qty

☐

☐

product

@product

#OrderProduct-followup product

☐

☐

Enter name

Enter entity

Enter value

☐

+ New parameter

Abbildung 2 Dialogflow Angabe der Anzahl

Fazit

Da wir anfangs grosse Schwierigkeiten hatten, überhaupt zu verstehen wie wir das Projekt angehen sollten und wir die Materie noch nicht genau begriffen hatten, sind wir froh mit einem Ergebnis abzuschliessen welches uns zufriedenstellt. Die Funktionalität konnte voll implementiert werden und man kann nun über den Dialogflow Produkte bestellen.

Die Projektarbeit gab uns allen einen grossen Lernzuwachs und jeder konnte seine Stärken miteinbringen.