**EAI: Semester Assignement**

**E-Shop Backend with Rest**

Ein Studentenprojekt der Fachhochschule Nordwestschweiz



**Github:**

|  |
| --- |
| <https://github.com/dubidamdam/EAI_Shop> |

Projekthandbuch

|  |  |
| --- | --- |
| Dokumentinformationen | |
| Projektmitglieder | Herr Fabian Dubach |
| Dozierende | Herr Andreas Martin |
| Version | final |
| Studiengang | Fachhochschule Nordwestschweiz  Wirtschaftsinformatik, 5. Semester |
| Datum | 14.12.2018 |

**Inhalt**

[Projekthandbuch 1](#_Toc532653751)

[1. Kurzbeschreibung des Projekts 3](#_Toc532653752)

[2. Funktionales 4](#_Toc532653753)

[2.1 Eshop 4](#_Toc532653754)

[2.1.1 Methode whatIntent 4](#_Toc532653755)

[2.1.2 Methode AddProdukt 4](#_Toc532653756)

[2.1.3 Methode handleCheckout 5](#_Toc532653757)

[2.2 Inventory 5](#_Toc532653758)

[2.3 Shipment 5](#_Toc532653759)

[2.4 Payment 5](#_Toc532653760)

[3. Tools 6](#_Toc532653761)

[3.1 IntelliJ 6](#_Toc532653762)

[3.2 H2 DB 6](#_Toc532653763)

[3.3 Dialogflow 6](#_Toc532653764)

[3.4 Ngrok 6](#_Toc532653765)

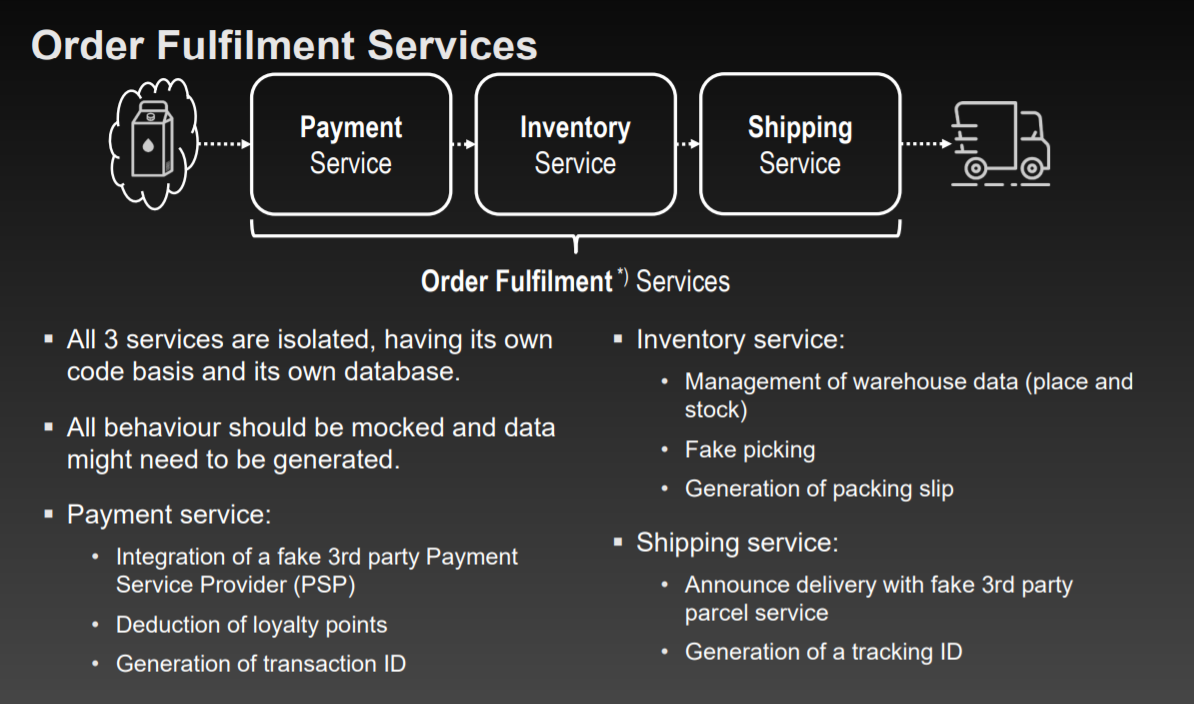
[3.5 Jsonschema2pojo 6](#_Toc532653766)

[4. Schwierigkeiten 7](#_Toc532653767)

[5. Fazit 8](#_Toc532653768)

# Kurzbeschreibung des Projekts

End-to-end E-Shop welcher aus unabhängigen Services besteht. Alles Services haben eine eigene h2-DB. Sie kommunizieren via Rest. Der E-Shop fungiert als Kontroller für das ganze Geschehen.

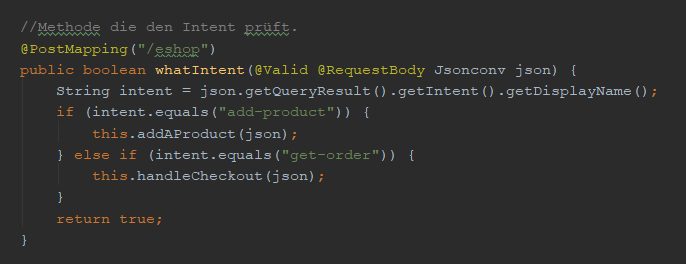


# Funktionales

## Eshop

### Methode whatIntent

Die Methode whatInten nimmt ein json von Dialogflow entgegen und schaut was der übergebene „Intent“ (Absicht) ist. Je nach dem wird die entsprechende interne Methode aufgerufen.



### Methode AddProdukt

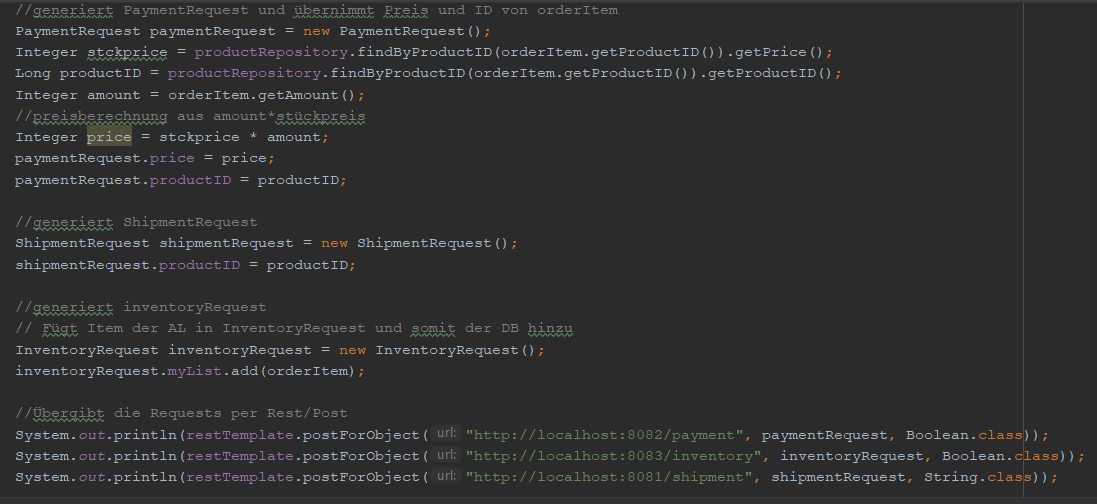
Die AddProdukt Methode stellt sicher das der Shopbetreiber neue Produkte erstellen kann. Die Funktion nimmt ein Produkt entgegen. Das Produkt wird dann in der Eshop Datenbank mit der EshopRepository Klasse eingetragen. Ausserdem wird ein RestPost an Inventory gesendet und dort das selbe Produkt eingetragen. Die Produkte stehen mit dem Attribut „ProduktID“ (nicht zu verwechseln mit ID) im Bezug zu einander. Sozusagen ein Foreignkey, welcher jeweils per Rest übergeben wird.

Die Methode nimmt ein json entgegen welches mit Hilfe von Java Klassen welche mit jsonschema2pojo erstellt wurden, die benötigten Parameter ausliest und ins Produkt schreibt.



### Methode handleCheckout

Diese Methode bearbeitet eine Bestellung und sendet alle nötigen Informationen an die anderen Services. Sie nimmt das json und liest wiederum von dort die benötigten Parameter ein.



## Inventory

Im Inventar wird von der Checkout Methode ein InventoryRequest entgegengenommen. Dieses beinhaltet eine Liste mit Items, die Bestellt wurden. Diese werden dann mit der Methode takeFromInventory in der Datenbank angepasst..

Die addNewProduct Methode fügt der Datenbank alfällige neue Produkte hinzu.

## Shipment

Das Shipment ist relativ simpel gestaltet. Es generiert mit dieser Line eine neue UUID:



Diese wird dann auch zurückgegeben und zusammen mit der productID in der Shipment DB abgelegt.

## Payment

Das Payment erhält über einen PaymentRequest einen Gesamtbetrag, welcher bereits im Eshop, in der Checkout Methode errechnet wurde. Dieser trägt es dann in die Payment DB ein. Ausserdem wäre eine Methode getPaymentStatus bereit welche noch mit der Bezahl-Logik abgefüllt werden müsste. Für unser Projekt gibt diese einfach immer true zurück.

# Tools

## IntelliJ

Wieso: Weil Lieblingsumgebung.

Wofür: Ganze Java/Spring Implementation

## H2 DB

Wieso: Weil so empfohlen.

Wofür: Jeweils unabhängige Datenbanken für die vier Services.

## Dialogflow

Wofür: Post Inputs via Sprache/Texteingabe

Wieso: Weil optionale Anforderung und weil cool.

## Ngrok

<https://ngrok.com/>

Wieso: Die Verbindung zu Dialogflow ist nur mühsam konfigurierbar, sofern man lokal arbeitet.

Wofür: HTTP/S Verbindung auf localhost von Dialogflow

## Jsonschema2pojo

<http://www.jsonschema2pojo.org/>

Wieso: Ich hatte mit den übergebenen JSONs zu kämpfen.

Wofür: Um JSON in Java umzuwandeln.

Special thanks: Felix Lehner, welcher mir dieses Tool empfohlen hat.

# Schwierigkeiten

Abgesehen davon, dass mein komplettes Projektteam (namentlich Team Rocket) zwei Wochen vor Abgabe abgesprungen ist, habe ich eigentlich vor allem kleinere Hindernisse gehabt.

Vor allem der Start ins Projekt verlief eher harzig, da es ohne Erfahrung im Bereich EAI, doch sehr schwierig ist, irgendwo mit dem Projekt anzufangen.

Danach hatte ich eigentlich nur noch Probleme mit der „Skalierung“ des Projekts. Da ich jeweils die Services immer funktional ein bisschen verbessert habe (Step-by-Step), hatte ich zwischendurch viel Aufwand in Form von Clean-Up des Codes. In einem weiteren Projekt würde ich zuerst die Services genauer „designen“ und danach programmieren. Hier habe ich halt mehr oder weniger einfach mal „drauflos programmiert“.

# Fazit

Schade, dass das Projekt von der Studiengangleitung aus nicht mehr gewichtet wird. Das Projekt war zwar als EAI-Noob extrem aufwändig, aber auch extrem lehrreich. Teilweise hätte ich gerne noch weitere Funktionen eingebaut (gerade im Bereich Assistant/Dialogflow), aber musste Aufwand und Ertrag auch etwas in Relation setzen, weil wir momentan ja noch weitere Abgaben (ToBIT, Web-Eng), haben oder hatten.

Trotzdem finde ich das Module wie EAI unbedingt solche Projekte beinhalten müssen, da diese viel mehr Inhalt bieten als trockene Vorlesungen und Prüfungen wo auf Papier programmiert wird. Die Projekte müssen jedoch auch entsprechend gewichtet werden.