

Chatbot für die Unterstützung von Banken: BankAccountBot

Miniprojekt im Modul AI_Foundations an der Fachhochschule Ost, Standort Rapperswil

Dozentin: Mitra Purandare

Author: Samuel Lemmenmeier

Datum Einreichung: 31.10.2021

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1. Einführung	3
1.1.Aufgabenstellung	3
1.2.Idee	3
1.3.Vorgehensweise	3
2. Node.js-Client	3
3. Dialogflow Backend	4
3.1.Ideengestaltung	4
3.2.Intent	4
3.3.Entities	7
4. Demonstration	7
5. Anpassungen	10
5.1.Erweiterungen	10
5.2.Verbesserungen	10
6. Diskussion	10
6.1.Probleme	10
6.1.1.Session	10
6.1.2.Google Dialogflow-Webpage	10

1. Einführung

1.1. Aufgabenstellung

Das Ziel des Miniprojectes ist es eine einfache Chatapplikation zu programmieren, die mit Google Dialogflow über die entsprechende API kommuniziert.

1.2. Idee

Die Idee die versucht wurde umzusetzen, ist ein Chatbot der auf einer Bankenplattform zum Einsatz kommen könnte und somit den Kundendienst der Bank entlasten sollte. Dabei sollten in erster Linie, regelmässige und einfache Anfragen automatisch bearbeitet werden können. Dies sind z.B. die Bankkarte sperren lassen, eine Ersatzkarte anfordern oder ein Beratungsgespräch vereinbaren.

1.3. Vorgehensweise

Es wurde komplett auf vordefinierte Dialogflow-Agents oder den zur Verfügung gestellten Java-Client verzichtet.

2. Node.js-Client

Ueber den Client kann mit dem BankAccountBot ge-chatet werden. Gibt man eine Frage ein und drückt Enter, get dies auf die /answer Route und ein Post Request an den Dialogflow wird abgesetzt.

Der Client ist mit Node.js, Express Framework und Handlebars als View-Engine aufgebaut. Dies ermöglicht einen simplen Aufbau und gleichzeitig auch viele Freiheiten. So ist der Client nicht nur ein rein Consolen basierter Chat, sondern der Output kann direkt mit HTML gestaltet werden und schliesslich gerendert abgebildet werden. Dies ergibt eine wesentlich ansprechendere Oberfläche.

Der Zugriff auf den in Dialogflow definierten Agent über die API musste daher zusätzlich selbst gemacht werden. Dies wurde mit folgender Library gemacht: [Dialogflow API: Node.js Client \(https://github.com/googleapis/nodejs-dialogflow\)](https://github.com/googleapis/nodejs-dialogflow).

Die Antworten und Fragen vom Chatverlauf werden als Objekte in einem Array gehalten.

Der Client hat verschiedene Routes:

- get "/" ruft showChatBot auf
- post "/answer" ruft calculateAnswer auf
- get "/resetContext" ruft newSessionId auf

showChatBot rendert die HTML Ansicht und übergibt ihr den Array. Handlebar iteriert dann über den Array, und stellt schliesslich den Chatverlauf wie im Kapitel Demonstration dargestellt ist, dar.

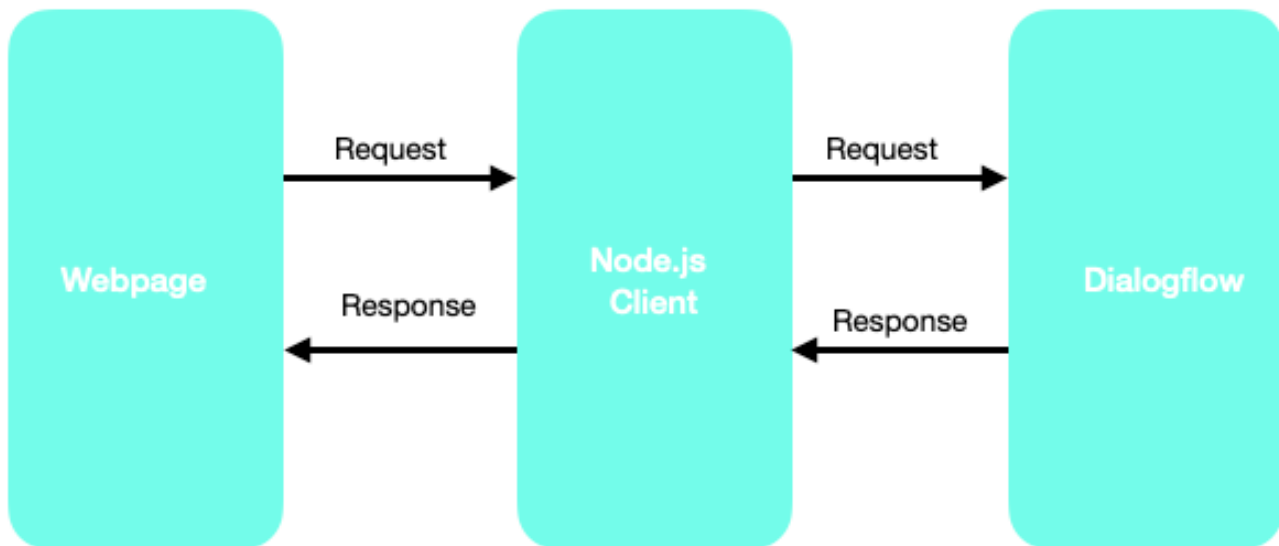
calculateAnswer nimmt aus dem request.body der Anfrage die Frage und schickt sie an den Agent. Aus der response wird schliesslich die Antwort ausgelesen und im Array mit der Frage zusammen gespeichert.

newSessionId setzt eine neue sessionId, dadurch wird eine neue Session gestartet und alle Contexts sind gelöscht. Wie wenn man auf der Google Dialogseite den Context löscht.

Es gibt noch weitere Funktionen, die oben genannten sind aber die wichtigsten!

Am Ende von calculateAnswer und resetContext wird ein redirect auf "/" aufgerufen. So wird vermieden, dass bei einem Page Reload nochmals ein Request mit dem gleichen body als Payload abgesetzt wird.

reset context Button ruft die Funktion newSessionId auf.



Webpage und Client hier entkoppelt dargestellt, laufen aber auf dem gleichen Port

3. Dialogflow Backend

Der Dialogflow wurde als komplett neuer Agent erstellt. Lediglich der Default Welcome und Fallback Intent wurden dabei generiert. Es wurde ein relativer simpler Agent konstruiert, aber dennoch mit ein paar wenigen Besonderheiten um verschiedene Möglichkeiten und Techniken wie Fallback Intent, Followup Intent, Context, Webhook auszuprobieren. Der Informationsfluss bzw. Gesprächsfluss im Dialogflow ist linear und non-linear aufgebaut.

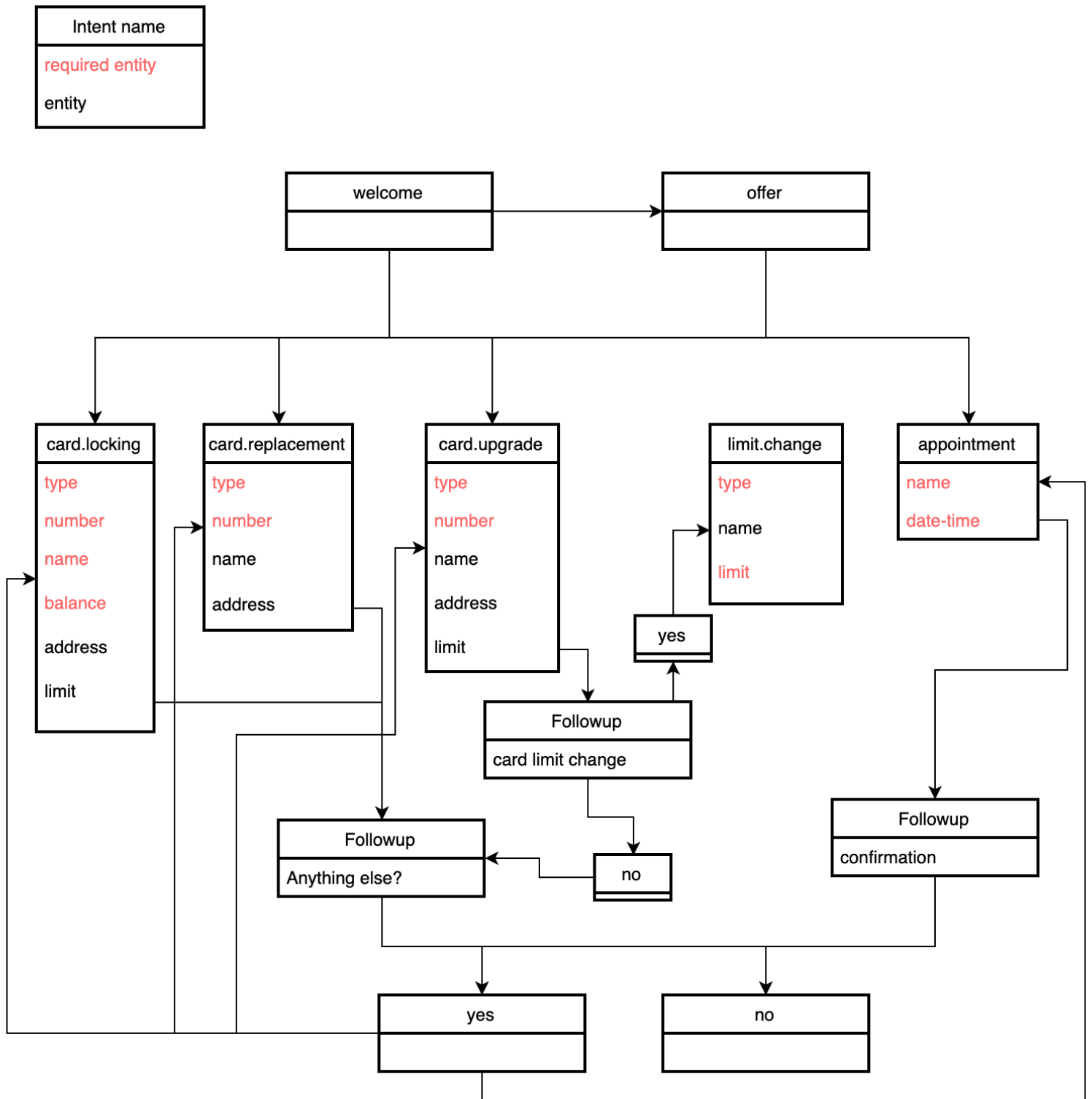
3.1. Ideengestaltung

Der Chatbot soll gewisse Aufgaben eines Bankangestellten übernehmen können. Diese sind in die verschiedenen Intents aufgegliedert. Die Namen der Intents sollten die Absicht des Intents bereits erklären. Die Idee war es noch, verschiedene Bankkarten Typen und Account Limiten zu definieren. Es gibt folgende Bankkarten, bronze, silver, gold, die in dieser Reihenfolge mehr Möglichkeiten für die Kunden bieten. Account Limiten sind 1000, 2000, 5000 und unlimited. In der momentanen BankAccountBot Version, sind theoretisch alle Kombinationsmöglichkeiten von Karten Typen und Account Limiten möglich.

3.2. Intents

Folgende Grafik stellt den grundlegenden Informationsfluss im Dialogflow dar. Als start dient der welcome Intent. Natürlich kann auch nach gewissen NO oder auch YES Entscheidungen wieder jeder Intent (ausser limit.change) aufgenommen werden. Dies wurde aber nicht explizit eingezeichnet.

Intents und wie sie miteinander verbunden sind, Informationsfluss folgt den Pfeilen



Es sind verschiedene Intents definiert:

- **welcome:** Dient als Einstieg in die Konversation.
- **offer:** Erklärt was der Chatbot ist und wie er helfen kann.
- **card.locking:** Wenn man die Karte sperren lassen möchte.
- **card.replacement:** Wenn man eine Ersatzkarte möchte.
- **card.upgrade:** Dieser Intent hat einen Webhook implementiert. Wenn man eine bessere Karte möchte. Es gibt bronze, silver, gold. Wenn alle nötigen Daten gesammelt sind, type + number, berechnet der Client ob ein Upgrade möglich ist. Ein Upgrade ist nur möglich, wenn die Karte die Farbe bronze oder silver hat, gold ist ja schön die "beste" Bankkarte. Per Webhook wird diese Antwort dann dem Agent mitgeteilt.
- **limit.change:** Wenn das Kartenlimit geändert werden möchte. Für Demonstrationszwecke kann dieser Intent nur nach dem card.upgrade Intent erreicht werden. Dies wurde erreicht über ein explizites Setzen von einem Output-Context yescl in card.upgrade und Input-Context yescl in limit.change.

Ein Intent ist abgearbeitet bzw. erfüllt, sobald alle Parameter (Entity Werte) die rot dargestellt sind, gesetzt sind. Dazu wurden Prompts definiert, die die entsprechende Nachfrage machen. Die Parameter in schwarz sind nicht required.

Wie der BankAccountBot genau funktioniert, wird noch veranschaulicht im Kapitel Demonstration.

Um den Datenfluss einfacher zu gestalten und weil nur der card.upgrade Intent einen Webhook hat, wurde beim welcome Intent ein langlebiger Output session-variables Context gesetzt und die Parameter in den verschiedenen Intents haben als Default Value den entsprechenden Parameter aus dem session-variables Context gesetzt bekommen. So wird z.B. bei einem Intent Wechsel von card.locking zu appointment der Parameter name nur einmal abgefragt. Dies kann nachfolgende Intents wesentlich verkürzen und der User bekommt nicht mehrmals die gleiche Aufforderung seinen Namen, Karten Typen etc. mitzuteilen.

Definierte Intents im Dialogflow

The screenshot shows the Dialogflow Essentials interface for the 'BankAccountBot' agent. The 'Intents' tab is selected in the left sidebar. The main panel displays a list of 17 defined intents. Each intent is represented by a blue dot icon and a text label. Some intents have a toggle arrow (up or down) next to them. The intents listed are: appointment, appointment-yes, appointment-no, card.locking, card.locking-yes, card.locking-no, card.replacement, card.replacement-no, card.replacement-yes, card.upgrade, card.upgrade-no, card.upgrade-no-no, card.upgrade-no-yes, card.upgrade-yes, Default Fallback Intent, Default Welcome Intent, limit.change, and offer.

3.3.Entities

Dies sind die Entities mit denen BankAccountBot funktioniert. Für den Typen von Bankkarten und die Limiten für die entsprechenden Accounts wurden eigene Entity-Typen definiert um die Werte Bereiche besser abbilden zu können.

Parameter Name	Entity	Mögliche Werte
type	card-type	bronze, silver, gold + synonyme
limit	card-limit	1000, 2000, 5000, unlimited
name	sys.number-name	Peter, Eva Meyer, ...
number	sys.given-integer	1, 23, 124142, ...
address	sys.address	Rapperswil, First Street New York, ...
balance	sys.number-integer	0, 2, 4, 12312, ...
date-time	sys.date-time	Tomorrow, Saturday at 10am, ...

Natürlich sind für die Entities auch Synonyme definiert, z.B. kann ein User auch grey anstatt silver oder brown anstatt bronze schreiben und es wird dennoch richtig erkannt.

4. Demonstration

Zuerst werden jeweils gewisse User Daten abgefragt. Dies soll eine einfache Art von Verifizierung simulieren. User Daten werden dann in session.variables Context gespeichert und sind bei einem nachfolge Intent bereits vorhanden.

Bild 1: Stellt Startsituation dar

Bild 2: User möchte Karte sperren und eine Besprechung abmachen. Intent offer & card.locking & appointment.

Bild 3: User möchte eine neue Karte und eine Besprechung abmachen. Intent card.replacement & appointment. Synonyme und leicht falsch geschriebene Wörter werden erkannt z.B. brwon wird als brown bzw. bronze (Karten Typ) erkannt.

Bild 4: User möchte seine Karte von gold upgraden. Dies wird aber per Webhook unterbunden, da gold bereits die höchste Stufe der möglichen Karten sind. Intent offer & card.upgrade.

Bild 5: User möchte seine silver Karte upgraden, das wird überprüft per Webhook. Da silver < gold ist,, geht ein Upgrade der Karte. Limit wird aber noch überprüft. Intent card.upgrade & limit.change.

Bild 1

```
samuellenmeier@MacBook-Air:~/Desktop/HSR/HS2021/AI_Foundation
> chatbot@0.0.0 start
> node bin/www.js

nftCreator is running at http://localhost:3000
453552a0-c85a-40a9-a6e6-719866576512
ID453552a0-c85a-40a9-a6e6-719866576512
PATH projects/bankaccountbot-whot/agent/sessions/453552a0-c85a-
Detected intent
Query: hello
Response: Good day! What can I do for you today?
Messages: [object Object]
Intent: Default Welcome Intent
GET / 304 2317.725 ms - -
GET /stylesheets/style.css 304 6.522 ms - -
```

chatBot-Miniproject-by-Samuel-Lemmenmeier

Start chatting with the bot by asking something
Don't forget to hit enter to submit question!

reset context

Good day! What can I do for you today?

Bild 2

Good day! What can I do for you today?

Who are you

I'm a chatbot, I'm here to help you with some specific questions (locking card, new limit, upgrade your card) about your bank card or I can schedule a in person meeting for you!

Ok, yesterday someone has stolen my card 2134, please lock it!

What is the color of your card?

silver

What was the last balance of your account, you can remember?

Not sure, 2000

Tell us your name!

Sorry, my name is Peter Smith

I will lock your card. Do you need anything more?

yes please

Ok, fine tell us whatever you need help with!

Can you arrange an appointment?

When do you prefer the meeting?

Tomorrow in the morning, maybe 10 o'clock

We scheduled your appointment at 2021-11-01T05:00:00/2021-11-01T11:59:59, Do you want a confirmation?

yes

We will send you a confirmation in the next 24h!

Bild 3

reset context

Good day! What can I do for you today?

Hi, this is Peter, my card got broken

What color does your card have?

brwon

What are the last four digits of your card?

2011

We can replace your card. Anything else we can help you with?

I need a meeting

When do you prefer the meeting?

Saturday 4pm

We scheduled your appointment at 2021-11-06T16:00:00, Do you want a confirmation?

no

Bild 4

Good day! What can I do for you today?

Hello, can I have a better card please, right now I have only a silver card.

What are the last four digits of your card?

4777

We can upgrade your card-type. Do you need also new card limit?

yes

Ok, with what limit can we help you?

unlimited

We can try to change your limit, we will contact you in the next 24h. Besides that, do you have another question?

Bild 5

Good day! What can I do for you today?

How can you help me!

I'm a chatbot, I'm here to help you with some specific questions (locking card, new limit, upgrade your card) about your bank card or I can schedule a in person meeting for you!

Ah cool, I have a yellow card right now, what about a gold one

What are the last four digits of your card?

2344

Sorry, you already have a gold card! Do you need also new card limit?

5. Anpassungen

5.1. Erweiterungen

Die Applikation ist so aufgebaut, dass Erweiterungen problemlos gemacht werden könnten. Dies könnten unter anderem sein:

- Mehr Webhooks
- Schöner Oberfläche
- Mehr Intents
- Mehr Entities
- Datenbank anbindung
- Daten auf Korrektheit und in Bezug auf andere Daten überprüfen

5.2. Verbesserungen

Die Applikation ist sehr einfach gehalten, daher wären Verbesserungen nötig würde der Chatbot bei einer richtigen Bank zum Einsatz kommen. Grundsätzlich müsste der User Verifikationsprozess umfangreicher und besser, auf die richtigen Daten abgestimmt, gemacht werden. Momentan werden Werte bezüglich Karten Limite, Karten Nummern nicht überprüft. Auch ist keine Datenbank im Hintergrund, die es ermöglichen würde, User mit entsprechender Karte zu überprüfen.

Der Chatbot ist sehr simpel, d.h. es sind nur wenige einfache Intents definiert. Diese besitzen zwar viele verschiedene Trainings Sätze, trotzdem kann es sein, dass der Default Fallback zum Zuge kommt. Dann braucht es eine korrekte Frage oder zumindest z.B. "How can you help me?" or "What do you offer?" um den offer Intent zu starten, damit der User sich nicht im Kreis dreht.

6. Diskussion

Dieses Projekt war sehr spannend, besonders im Ausblick auf die Zukunft. Ich denke, es ist sehr gut möglich, dass ich in der Zukunft mal einen Chatbot in ein bestehendes oder neues Projekt implementieren muss.

Ich war erstaunt, wie viel Zeit es doch braucht, um sich in diese neue Thematik einzuarbeiten. Für einen gut entwickelten Chatbot, der mit einer komplizierten non-linearen Gesprächsführung umgehen kann, braucht es doch einiges an Wissen.

6.1. Probleme

Probleme die bei mir aufgetreten sind, manche auch selbstverursacht. Um diese Schwierigkeiten zu lösen, brauchte es teilweise auch viel Zeit. Dafür kann man diese Probleme hoffentlich in der Zukunft vermeiden.

6.1.1. Session

Die Session wurde mir mehrmals zum Verhängnis.

Lokal, als Bash-Session, zum Verbinden bzw. Authentifizieren mit der Dialogflow-API. Ich habe mit Webstorm gearbeitet. Der Pfad zum key.json File habe ich auch nur pro Session gesetzt, nur war die Webstorm-Console und Bash-Console nicht die gleiche Session, dann wurde das key.json File nicht gefunden => kein Zugriff auf Dialogflow.

Dialogflow öffnet eine Session pro neuen Chat, diese Session wird beibehalten über mehrere Intents. Musste einen Weg finden um eine neue Session und damit alle Contexts im Dialogflow über Front- bzw. Backend zu starten.

6.1.2. Google Dialogflow-Webpage

- Die Dialogflow-Webpage von Google ist extrem langsam! Teilweise hat die Seite lange gebraucht um zu laden bzw. den bearbeiteten Intent zu speichern. Musste man mehrere Intents nochmals bearbeiten, z.B. um neue Entity-Variabel hinzuzufügen, dauerte das ewig und hat enorm Nerven gekostet. Der Agent wurde dabei aber nicht trainiert, dies wurde zumindest nicht angezeigt!
- Definiere eine Prompt zu einer Entity, drücke close => wurde aber nicht gespeichert.