Technische Features

Dienstag, 21. Mai 2019 22:35

Camunda

- Modelle
  - 6 BPMN
  - 1 DMN
- No Backend (ERP/CRM) directly exposed to app - everything goes through a process
- Digitalisierung primär über Java Klassen
  - Auch eingesetzt: Skripte, Expressions
- Genutzte BPMN Features:
  - parallel multi instance to handle lists
  - subtasks to bundle tasks
- Genutzte Camunda Features:
  - org.camunda.spin: um eigene komplexe Objekte als Data Transfer Objects brauchen zu können
  - http-connector für alle Integromat aufrufe
  - Zwei Pattern, um Variablen zur Verfügung zu stellen
    - Einmal warten über einen Receive Task bis der Client die Variablen abholt und eine Message schickt
    - Als Bestandteil einer Antwort auf den REST-Call, der den Prozess startet
  - processes.xml, um die verwendeten Prozesse selektiv zu hosten (ansonsten scannt Camunda alle Ordner und versucht alle gefundenen Modell zu hosten)
  - Eigentlich Java-Feature: UUID, um die IDs zu generieren
  - Rollen für die User Tasks (TODO: Das muss noch kommen!)

Watson Services

- Android Integration über Android SDK: https://github.com/watson-developer-cloud/android-sdk
- Zusätzliche Integration über Java SDK: https://github.com/watson-developer-cloud/java-sdk
- Text to Speech
- Speech to Text
- Visual Recognition
  - Default classifier wurde verwendet
- Watson Assistant (Chatbot)
  - Ein paar generell verfügbare Dialoge wurden importiert (Begrüssung, Standard Reaktionen und so en Kram)
  - Dialog, um Formular zu erstellen für Präferenzen
  - Dialog in zweifacher Ausführung (Male / Female), um die Masse zu nehmen, falls das Item mass geschneidert werden muss

Android App

- simplen chat entwickelt
- icon eingebunden
- retrofit zwecks kommunikation mit Camunda
- Camunda REST API verwendet zwecks Integration mit Prozessen und deren Variablen
- SDK für Watson Services
- ObjectMapper verwendet, um die serialisierten Objekte, die in Camunda Prozessen gespeichert wurden wieder in Java Objekte zu deserialisieren
- Data Binding für Formulare
- Integration visual recognition ist speziell:
  - Bild wird an visual recognition service von ibm geschickt
  - Zurück kommt eine Reihe an Tags, die erkannt worden mit verschiedenen confidence leveln
  - Diese werden zu einem whitespace-separierten String zusammen gepappt und an den chatbot service von ibm geschickt

    - Der chatbot assistant nimmt jeweils die tokens heraus, die es erkennen konnte
    - Notiz: Viele Bilder funktionieren nicht, obwohl der Assistent korrekt das Intent und die gesuchten Attribute erkennt. Der Grund ist nicht klar. Für diese Bilder kann dann aber explizit der Bot nochmals trainiert werden

Integromat

- Backend implementierung über Google Sheets
- Webhook, um APIs zur Verfügung zu stellen