Rahel H

r.hueppi2008@gmail.com

LB3 m254

Rezept-Suche

Inhaltsverzeichnis

[Einführung 2](#_Toc194911232)

[Entscheidungen 2](#_Toc194911233)

[API 2](#_Toc194911234)

[Vorgehen 2](#_Toc194911235)

[Basic Camunda-Modell erstellen 2](#_Toc194911236)

[Form erstellen 3](#_Toc194911237)

[Skript zur Erstellung der URL als API-Anfrage 4](#_Toc194911238)

[Externes Skript einbinden 4](#_Toc194911239)

Rezept-Suche

Datum in Fusszeile ändern

# Einführung

In meinem Projekt geht es darum, eine Rezept-Suche mit Camunda umzusetzen. Dabei kann der User Zutaten, eine bestimmte Küche oder anderes eingeben und der Prozess sucht Rezepte über eine API. Die Resultate werden per Mail an den User versendet.

# Entscheidungen

## API

Ich habe mich für die Spoonacular-API entschieden, da es bei der API eine Gratisversion hat und einige verschiedene Rezepte anbietet. Ebenfalls ist der Gebrauch sehr einfach, da es zum Beispiel auch eine ausführliche Dokumentation hat.

# Vorgehen

## Basic Camunda-Modell erstellen

Ein Bild, das Screenshot, Rechteck, Reihe, Diagramm enthält.

KI-generierte Inhalte können fehlerhaft sein.

Im ersten Schritt habe ich ein Basic Camunda-Modell erstellt. Das war noch ohne Fehlerbehandlungen. Es war also ein Modell, welches nur dafür war, wenn alles nach Plan lief.

Ebenfalls wurde hier noch nichts genaueres, wie zum Beispiel das Form oder …. (Skript? Oder wie heisst das, bei was es danach weiss, was es tun soll?) bei den Service Tasks definiert.

## Form erstellen

Ich habe in der Dokumentation der Spoonacular API (<https://spoonacular.com/food-api/docs>) nachgeschaut, welche Parameter mitgegeben werden können und ich somit auch für mein Input-Formular verwenden kann.

Besonders ins Auge gefallen sind mir dabei diese Parameter:

(Informationen der Tabelle wurden der offiziellen Dokumentation entnommen)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Example** | **Description** |
| Query | String | Pasta | The (natural language) recipe search query. |
| Cuisine | String | Italien | The cuisine(s) of the recipes. One or more, comma separated (will be interpreted as 'OR'). |
| Intolerances | String | Gluten | A comma-separated list of intolerances. All recipes returned must not contain ingredients that are not suitable for people with the intolerances entered. |
| includeIngredients | String | Tomato, cheese | A comma-separated list of ingredients that should/must be used in the recipes. |
| excludeIngredients | String | Eggs | A comma-separated list of ingredients or ingredient types that the recipes must not contain. |
| Type | String | Main course | The type of the recipe |
| Number | Number | 10 | The number of expected results. |

Aufgrund dieser Parameter habe ich ein Formular gemacht (wirklich genau diese? Nochmals überprüfen)

j

Ich war mir am Anfang unsicher, wie ich das genau machen sollte mit Parametern wie zum Beispiel Cuisine. Bei diesen kann man mehrere angeben, welche mit einem Komma getrennt sind. Man kann aber nicht alles angeben, da es eine Liste hat, welche Eingaben unterstützt werden. Wegen dem wäre es sinnvoll, würde man ein Select machen, da man so nur die Vorgegebenen eingeben kann. Jedoch könnte man so nicht mehrere angeben, weshalb ich es fürs erste eine Texteingabe gelassen habe.

Ein Bild, das Screenshot, Text, Software, Computersymbol enthält.

KI-generierte Inhalte können fehlerhaft sein.

Bild: erste Version meines Formulars

## Skript zur Erstellung der URL als API-Anfrage

In meinem Camunda Diagramm habe ich einen separaten Task eingebaut, welcher die URL mit allen Query Parameter zusammenstellt. Dieses Skript habe ich der einfachheitshalber fürs erste direkt in Camunda in ein internes Skript getan und nicht ein externes File eingebunden. Das habe ich dann im nächsten Schritt gemacht.

## Externes Skript einbinden

Ich habe anschliessend das vorher erstellte Skript als ein externes Skript hinzugefügt. Dies habe ich mit dieser Anleitung gemacht: <https://gitlab.com/ch-tbz-it/Stud/m254/-/tree/main/Unterlagen/Camunda7/Beispiele/Aufg_ExternalSkript?ref_type=heads>  
  
Meine Probleme hatte ich zuerst damit, dass ich nicht verstand, wo ‘deployment://test.js’ hinmuss. Das habe ich dann aber relativ schnell herausgefunden. Anschliessend hat es immer noch nicht funktioniert. Nach einer Zeit habe ich dann herausgefunden, dass bei ‘Format’ «javascript» hineinmuss. Zuvor hatte ich «JavaScript» und mit dem inline script hat es auch funktioniert.

## Skript zum API-Call

Anschliessend habe ich ein Skript dazu geschrieben, welches den API-Call ausführt. Dieses Skript habe ich ebenfalls extern gemacht und es anschliessend wie das erste Skript eingebunden. Beim Schreiben des Skripts war ein bisschen ungewöhnlich, dass man Java dafür einbinden musste. Das war so, da Camunda kein normales Browser-JavaScript benutzt, sondern läuft Java Script auf der Nashorn Engine. Das ist eine Java basierte JavaScript-Umgebung. (Stimmt das? Ev. Anpassen oder skript anpassen) Deshalb war es für mich ungewohnt, so den API Call zu machen.