Mensa-Auskunft Documentation

Release 1.0

Bogdan Kostić, Maria Lomaeva, Monique Noss, Olha Zolotaren

30.10.2019

Contents:

KAPITEL 1

Lambda Function

lambda_function.**details_intent_handler**(*handler_input*)

Der Intent listet die Details zu einem bestimmten Gericht auf.

(Alle verfügbaren Informationen der OpenMensa API.)

Dazu gehören:

- Der vollständige Titel des Gerichts
- Die Preise für Studenten, Angestellte und Andere
- Die Kategorie des Gerichts
- Die zusätzlichen Notes

Der Benutzeranfrage und den bereitgestellten Slot Values werden folgende Daten entnommen:

- Gerichtnummer

Über die Session-Attributes müssen folgende Daten abgerufen werden:

- **Daten zu den gespeicherten Gerichten:**
  - ‚name'
  - ‚price'
  - ‚category'
  - ‚notes'

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**list_dishes_intent_handler**(*handler_input*)

Der Intent listet alle verfügbaren Gerichte auf, die nach verschiedenen Mustern gesucht werden:

1. Zunächst wird eine Mensa und ein Datum festgelegt. Die Slots werden elizitiert, wenn die Nutzeranfrage diese noch nicht enthält. 2. Die Suchanfrage kann spezifiziert werden, indem der User zusätzlich bis zu zwei Zutaten mit angeben kann, die entweder im Gericht enthalten sind oder nicht im Gericht enthalten sein sollen.

Der Benutzeranfrage und den bereitgestellten Slot Values werden folgende Daten entlockt:

- Tag bzw. Datum der Suchanfrage (erforderlich)
- Mensaname (erforderlich)
- (nicht) erwünschte Zutaten (optional)

Anschließend wird eine Suchanfrage mithilfe der Mensa-ID und dem Datum gestartet und die Daten von der OpenMensa API erfragt. Diese werden ebenfalls in den Session-Attributes gespeichert.

Gibt es zusätzlich Zutaten in der Nutzeranfrage, werden die API-Daten gefiltert.

Zuletzt werden die Ergebnisse in einen String überführt, für den Chunking benutzt wird, um die Antwort von Alexa kurz zu halten (siehe lambda_utility.py).

Außerdem werden bei jedem Turn nur vier Gerichte ausgegeben. Will der Nutzer mehr Gerichte erfahren, muss er nach „weiteren Gerichten“ fragen.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**price_intent_handler**(*handler_input*)

Der Intent gibt den Preis für ein bestimmtes Gericht zurück. Der Benutzer muss dabei die Nummer des Gerichts angeben und kann optional eine Zielgruppe (Studenten, Angestellte, Andere) definieren.

Der Benutzeranfrage und den bereitgestellten Slot Values werden folgende Daten entlockt:

- Nummer des Gerichts (erforderlich)
- Zielgruppe (optional)

Über die Session-Attributes müssen folgende Daten abgerufen werden:

- **Daten zu den gespeicherten Gerichten:**
  - ‚name‘
  - ‚prices‘

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**address_intent_handler**(*handler_input*)

Der Intent gibt die Adresse einer Mensa zurück. Benötigt wird der Name der Mensa. Ist dieser im Katalog, wird die Adresse zurückgegeben. Ist dieser nicht vorhanden, wird eine Fehlermeldung zurückgegeben.

Der Benutzeranfrage und den bereitgestellten Slot Values werden folgende Daten entlockt:

- Name und ID der Mensa (erforderlich)

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**list_mensas_intent_handler**(*handler_input*)

Der Intent gibt eine Liste mit Mensen in einer Stadt zurück. Benötigt wird der Name der Stadt. Sind Mensen vorhanden, werden diese zurückgegeben; gibt es keine oder ist die Stadt nicht in der Datenbank, wird eine Fehldermeldung zurückgegeben.

Bei jedem Turn werden nur vier Mensen ausgegeben. Will der Nutzer mehr Mensen erfahren, muss er nach „weiteren Mensen“ fragen.

Der Benutzeranfrage und den bereitgestellten Slot Values werden folgende Daten entlockt:

- Stadt (erforderlich)

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**get_nearest_mensa_intent_handler**(*handler_input*)

Der Intent gibt die vom Standort des Nutzers aus nächste Mensa und ihre Adresse zurück. Dafür muss der Benutzer seinen Standort für den Skill freigegeben haben.

Um die nächste Mensa berechnen zu können, werden die Koordinaten des Nutzers extrahiert. Anschließend wird mithilfe der Haversine-Formel die zum Benutzer nächstgelegene Mensa berechnet. Es wird also die Mensa mit der kleinsten Luftlinie zum Nutzer zurückgegeben.

Die Koordinaten des Nutzer werden folgendermaßen extrahiert:

a) Falls der Benutzer ein mobiles Gerät verwendet, werden die aktuellen Koordinaten aus dem GPS-Sensor des Geräts des Nutzers aus dem von Alexa an das Backend gesendete Request-JSON-Objekt extrahiert.

b) Falls der Benutzer ein stationäres Gerät verwendet, wird die vom Benutzer in der Alexa-App bzw. angegebene Adresse aus der Alexa-API extrahiert. Anschließend werden die Koordinaten der extrahierten Adresses mithilfe der Nominatim-API ermittelt.

Um dies zu ermöglichen, muss der Nutzer Zugriff auf seinen aktuellen Standort bzw. auf seine Adressdaten in der Alexa-App erlauben.

Wenn der Benutzer seinen Standort nicht freigegeben hat oder dieser gerade nicht extrahierbar ist, weil z.B. kein GPS-Signal verfügbar ist, werden je nach Errortype unterschiedliche Fehlermeldungen ausgegeben.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**log_response**(*handler_input*, *response*)

Response logger.

lambda_function.**log_request**(*handler_input*)

Request logger.

lambda_function.**launch_request_handler**(*handler_input*)

Der Intent wird beim Öffnen des Skills durch Modal-Launchphrasing getriggert. Er gibt eine Willkommensnachricht zurück.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**no_intent_handler**(*handler_input*)

Der Intent wird ausgelöst, wenn der Benutzer eine Rückfrage des Skills verneint. Er gibt eine Verabschiedung zurück und beendet den Skill.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**next_intent_handler**(*handler_input*)

Der Intent führt den ListDishesIntent weiter und listet weitere Gerichte auf. Die benötigten Daten werden aus den Session-Attributes entnommen.

Jeder Turn gibt nur vier Gerichte aus. Will der Nutzer mehr Gerichte erfahren, muss er nach „weiteren Gerichten“ fragen.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**help_intent_handler**(*handler_input*)

Der Intent unterstützt den Nutzer bei der Bedienung des Skills. Er informiert den Nutzer über den Umfang des Skills und gibt Beispielanfragen zurück.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**exit_intent_handler**(*handler_input*)

Der Intent stoppt den Skill mit einer Verabschiedung.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**session_ended_request_handler**(*handler_input*)

Der Intent stoppt den Skill ohne Verabschiedung.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die response ohne Antwort zurück

Rückgabetyp Response

lambda_function.**fallback_intent_handler**(*handler_input*)

Der Intent informiert den Benutzer darüber, dass der Skill die gewünschte Funktionalität nicht besitzt und beendet ihn.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**all_exception_handler**(*handler_input*, *exception*)

Der Intent ist ein Fallback für alle Exceptions. Er informiert darüber, dass der Skill die gewünschte Funktionalität nicht besitzt und beendet ihn.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**unhandled_intent_handler**(*handler_input*)

Ein Intent für alle „Unhandled requests“. Er informiert darüber, dass der Skill die gewünschte Funktionalität nicht besitzt und beendet ihn.

Parameter **handler_input** (*(HandlerInput) -> Response*) – HandlerInput

Rückgabe Gibt die vollständige Skill-Antwort zurück

Rückgabetyp Response

lambda_function.**invoke_skill**()

POST Methode, die den Skill als FLASK app startet.

Rückgabe Startet den Skill

Rückgabetyp dispatch_request

KAPITEL 2

Lambda Utility

lambda_utility.**create_mensa_url**(*mensa_id*, *date*)
Baut den String für die OpenMensa API Anfrage.

Parameter

- **mensa_id** (*str*) – die Mensa-ID als String
- **date** (*str*) – das Datum als String

Rückgabe Gibt die Webseite der API als String zurück

Rückgabetyp str

lambda_utility.**random_phrase**(*str_list*)
Gibt einen zufälligen String aus einer String-Liste zurück.

Parameter **str_list** (*List[str]*) – Liste mit Strings

Rückgabe Gibt einen String zurück

Rückgabetyp str

lambda_utility.**http_get**(*url*, ***kwargs*)
Erfragt Daten mithilfe des requests-Moduls und fängt Error ab.

Parameter **url** (*str*) – die API url, die angefragt wird

Verursacht

- **ValueError** – ValueError, wenn die Response leer ist
- **[ErrorType]** – Je nach Statuscode, den die Response enthält

Rückgabe Gibt eine Antwort des requests-Moduls als json zurück.

Rückgabetyp json

lambda_utility.**http_get_iterate**(*url*)
Erfragt Daten mithilfe des requests-Moduls und fängt Error ab. Iteriert über alle Seiten einer API

Parameter **url** (*str*) – die API url, die angefragt wird

Verursacht **[ErrorType]** – Je nach Statuscode, den die Response enthält

Rückgabe Gibt die Antworten des requests-Moduls als json zurück.

Rückgabetyp json

lambda_utility.**chunking**(*meal*)
Hilfsfunktion, die naives Chunking auf die Gerichte anwendet, für die es keine ähnlichen Gerichte in der globalen Liste gibt. Gesplittet wird an Präpositionen.

Parameter **meal** (*str*) – Name des Gerichts, auf den Chunking angewendet wird.

Rückgabe Gibt den verkürzten String zurück.

Rückgabetyp str

lambda_utility.**find_difference**(*duplicates*, *all_dishes*)
Hilfsfunktion für Chunking, die ähnliche Gerichte in der Liste aller Gerichte sucht und diese so verkürzt, dass nach der Präposition „mit“ der Teil dieser ähnlichen Gerichte steht, der sie unterscheidet.

Parameter **duplicates** – Liste ähnlicher Gerichte

Rückgabe Gibt die Liste der ähnlichen Gerichte zurück, wenn diese vorkommen

Rückgabetyp List

lambda_utility.**make_chunking**(*all_dishes*)
Hauptfunktion des Chunkings der Gerichte. Die Länge der Namen der Gerichte in der endgültigen Liste wird durch den parameter cutoff von der overlap-Funktion kontrolliert.

Parameter **all_dishes** (*List*) – Liste aller Gerichte.

Rückgabe Gibt eine Liste mit verkürzten Namen der Gerichte zurück

Rückgabetyp List

lambda_utility.**build_mensa_speech**(*mensalist*, *start_idx*)
Baut den String, der anschließend für den Prompt des ListMensasIntents benutzt wird. Die Liste ist möglicherweise sehr lang. Deshalb wird nur das Element am Startindex und drei weitere Elemente in den Prompt gebaut, um zu lange Antworten zu vermeiden.

Der letzte Index wird gemerkt und zusammen mit dem String zurückgegeben. Dieser fungiert beim nächsten Durchlauf als Startindec, damit die Liste an derselben Stelle fortgesetzt wird.

Parameter

- **mensalist** (*List[str]*) – Eine Liste mit Mensen als Strings
- **start_idx** (*int*) – Der Index, bei dem angefangen werden soll zu suchen.

Rückgabe Gibt den fertigen String un den letzten Index für den nächsten Durchlauf zurück.

Rückgabetyp str, int

lambda_utility.**build_dish_speech**(*dishlist*, *start_idx*)
Baut den String, der anschließend für den Prompt des ListDishesIntents benutzt wird. Die Liste ist möglicherweise sehr lang. Daher wird nur das Element am Startindex und drei weitere Strings in den Prompt gebaut, um zu lange Antworten zu vermeiden.

Der letzte Index wird gemerkt und zusammen mit dem String zurückgegeben. Dieser fungiert beim nächsten Durchlauf als Startindex, damit die Liste an derselben Stelle fortgesetzt wird.

Parameter

- **dishlist** (*List[str]*) – Eine Liste mit Gerichten als Strings
- **start_idx** (*int*) – Der Index, bei dem angefangen werden soll zu suchen.

Rückgabe Gibt den fertigen String und den letzten Index für den nächsten Durchlauf zurück

Rückgabetyp str, int

lambda_utility.**build_preposition_speech**(*ingredients*)
Baut einen zusätlichen String, der Zutaten enthält, um dem Nutzer mitzuteilen, wonach er gesucht hat.

Parameter **ingredients** (*dict*) – Ein Dictionary mit Zutaten-Strings

Rückgabe Gibt den fertigen String zurück

Rückgabetyp str

lambda_utility.**build_price_speech**(*price*, *user*)
Baut die Prompt für den Preis.

Parameter

- **price** (*str*) – Der Preis als String
- **user** (*str*) – Die Zielgruppe als String

Rückgabe Gibt den fertigen String zurück

Rückgabetyp str

lambda_utility.**get_resolved_value**(*request*, *slot_name*)
Die Funktion löst einen slot name mithilfe von slot resolutions auf, um einen „dahinterliegenden“ Wert zu finden.

„Resolve the slot name from the request using resolutions.“

Parameter

- **request** (*IntentRequest*) – Die Anfrage des Intents
- **slot_name** (*str*) – Name des Slots

Verursacht

- **AttributeError** –
- **KeyError** –
- **ValueError** –
- **IndexError** –
- **TypeError** –

Rückgabe Gibt entweder den resolved slot value zurück oder None

Rückgabetyp str, None

lambda_utility.**get_slot_values**(*filled_slots*)
Extrahiert zusätzliche Informationen zum Slot filling, wie z.B. die ID eines slots, ob der Slot in den Slot values des Interaction Models vorhanden ist und seinen resolved value.

„Return slot values with additional info.“

Parameter **filled_slots** (*Dict[str, Slot]*) – Ein Dictionary mit allen Slots

Verursacht

- **AttributeError** –
- **KeyError** –
- **ValueError** –
- **IndexError** –
- **TypeError** –

Rückgabe Gibt die extrahierten Values mit zusätzlichen Infos zurück

Rückgabetyp Dict[str, Any]

lambda_utility.**find_matching_dishes**(*api_response*, *ingredients={'first': None, 'second': None}*)
Filtert die aus der API-Antwort erhaltenen Gerichte nach bestimmten Zutaten. Hat der User keine Zutaten angebeben, sind diese „None“ und es wird die komplette API-Antwort zurückgegeben.

Parameter

- **api_response** (*json*) – Die json response der OpenMensa API

- **ingredients** (*Dict[str, [None, str]] (Default: {'first' : None, 'second' : None})*) – Dictionary mit Zutaten-String

Rückgabe Liste mit den erwünschten Gerichten

Rückgabetyp List[str]

lambda_utility.**ingredient_in_dish**(*ingredient*, *dish*)

Checkt, ob eine bestimmte Zutat in einem Gerichte-String oder in den Notes enthalten ist.

Parameter

- **ingredients** (*Dict[str, [None, str]]*) – Dictionary mit Zutaten-String
- **dish** (*Dict[str, Any]*) – Dictionary mit allen Daten zu einem Gericht

Rückgabe Gibt True zurück, wenn die Zutat gefunden wurde, sonst False

Rückgabetyp Boolean

lambda_utility.**ingredient_fleisch**(*dishlist*, *first_prep*, *second_prep*, *is_first_ingredient=True*)

Filtert die erhaltenen Gerichte nach veganen und vegetarischen Gerichten, da dies nicht aus den Daten der API ersichtlich ist.

Parameter

- **dishlist** (*List[str]*) – Liste mit Gerichten
- **first_prep** (*str*) – Die erste Präposition
- **second_prep** (*str*) – Die zweite Präposition
- **is_first_ingredient** (*Boolean (Default: True)*) – TODO Docu

Rückgabe Liste mit Gerichten mit/ohne Fleisch

Rückgabetyp List[str]

lambda_utility.**calculate_nearest_mensa**(*user_coordinates*, *mensa_list*)

Berechnet die zum User nächstgelegene Mensa (Luftlinie) mithilfe der Haversine-Formel.

Parameter

- **user_coordinates** (*Tuple[float]*) – Paar mit Koordinaten des Users
- **mensa_list** (*List[dict]*) – Liste aller Mensen

Rückgabe Paar mit Name und Adresse der nächsten Mensa

Rückgabetyp Tuple[str]

lambda_utility.**convert_acronyms**(*speech*)

Konvertiert Akronyme in einem Speech-Output in eine Form, die Alexa zeigt, dass das Akronym nicht als Wort sondern Buchstabe für Buchstabe ausgesprochen werden soll. Diese Funktion ist wichtig, da Alexa ansonsten Universitäten wie „FU Berlin“ als [fuː bɛʊˈliːn] aussprechen würde.

Parameter **speech** (*str*) – Speech-Output als String

Rückgabe Gibt einen umgewandelten Speech-String zurück

Rückgabetyp str

KAPITEL 3

Indices and tables

- genindex
- modindex
- search

Python-Modulindex

l

Stichwortverzeichnis

S

U