cb-compose-engine

Release 0.1

Feb 19, 2020

Contents

This microservice is intended to communicate with other services in order to determine intents and entities to retrieve the answer for the channel.

Check also th API description here:

CHAPTER 1

APIS

Here we have the different views of the app.

CHAPTER 2

Services

Here we present all the services used to communicate with the other microservices of the system.

By default, in the settings.ini is described every address of the other microservices

These are the main methods of this class:

`services.`**`discover_intent`**`(`*agent*, *text*`)`

This method connects to the microservice *NLP Engine* in order to retrieve the intents from a raw text.

Parameters

- **`agent`** – The agent name of the chatbot
- **`text`** – The raw input to discover the intent

Returns The uri of a intent

`services.`**`discover_entities`**`(`*agent*, *text*`)`

This method connects to the microservice *NLP Engine* in order to retrieve the entities from a raw text.

Parameters

- **`agent`** – The agent name of the chatbot
- **`text`** – The raw input to discover the intent

Returns A list with the URIS of the entities and type of entities.

`services.`**`get_requirements`**`(`*intent*`)`

This service connects to the microservice *Intents Managment* in order to retrieve requirements of an intent

Parameters **`intent`** – The intent from where requirements will be retrieved

Returns A list of URIS of the different entities needed to complete an intent

`services.`**`get_options`**`(`*entity*`)`

This service connects to the microservice *Intents Management* in order to retrieve options of an entity type

Parameters **`entity`** – The entity from where options will be retrieved

Returns A list of options to complete an entity

Todo: give a list of entities

`services.`**`get_answer`**`(`*intent*, *entities*`)`

This service connects to the microservice *Intents Management* in order to retrieve the answer of an intent.

Parameters

- **`intent`** – The intent from where answer will be retrieved
- **`entities`** – A list of entities used to retrieve answer

Returns A list of options to complete an entity

`services.`**`find_in_context`**`(`*user*, *entities*`)`

This method looks for information in the conversation thread.

param user The id of the user to find information

param entities The entities to be found in context.

`services.`**`get_agent_data`**`(`*agent*`)`

This service connects to the microservice *Intents Management* in order to information about and agent

Parameters **`agent`** – A valid agent name

Returns A dict containing agent, a description and the different intents.

`services.`**`get_intent_rq`**`(`*intent*, *entity*`)`

This service connects to the microservice *Intents Management* in order to get the resolution question of an intent

Parameters

- **`intent`** – A valid intent
- **`entity`** – A valid entity

Returns A dict containing intent, a entity and the resolution question.

CHAPTER 3

Utils

Here we present all the utility methods to help transform the data to present to the user.

These are the main methods of this class:

`compose_utils.clean_uri(uri)`

This method removes the url part of the URI in order to obtain just the property or class

Parameters **`uri`** – An uri to be cleaned

Returns The name of the property or the class

`compose_utils.check_requirements(requirements, entities)`

This method compares the existing entities and the entities required to complete an intent.

Parameters

- **`requirements`** – The list of the entities needed
- **`entities`** – The list of current entities

Returns If entities are missing, a list of this missing entities

`compose_utils.build_answer(raw_answer, answer_type)`

This method builds the answer, depending of the type of answer.

Parameters

- **`raw_answer`** – A dict containing the template of the answer, and the different part of tha answer
- **`answer_type`** – The type of answer to be constructed

Returns the raw text of the final answer

`compose_utils.update_entities(current_entities, new_entities)`

This method updates the current list of entities, by looking for the same type of entity.

Parameters:

param current_entities Current list of entities

param new_entities New list of entities

return Current list of entities updated

Check also the TODO lis for this project:

Todo: give a list of entities

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/cb-compose-engine-ms/checkouts/latest/kbsbot/compose_engine/services.py:docstring of services.get_options, line 7.)

This project is part of the architecture described in: Herrera, Andre & Yaguachi, Lady & Piedra, Nelson. (2019). Building Conversational Interface for Customer Support Applied to Open Campus an Open Online Course Provider. 11-13. 10.1109/ICALT.2019.00011.

Index