loanbot Documentation

Release

Anne LoVerso, Mimi Kome

May 08, 2017

Contents:

CHAPTER 1

mission

Product Definition

Loan Wrangler is an easy and fun chatbot conversational interface. It serves as a simple way to check out tools and interface with the tool loaning system at Olin. The tool help includes: resources, reminders to return due tools, and more

Project Risks

- Complications of natural language processing and/or anticipating all use cases (incl. trolls/misuse)
- Synchronizing with TIND for optimal use/adoption
- There're a lot of great bonus features that will take some time to go through

Project Roadmap

Between now and April 20:

- Start setting up TIND interface to get tools list
- Being able to ask about tool availability
- Getting full list of all tools when asked
- More testing

Between April 20 and April 24:

- Fully fledged testing
- Better documentation
- Digital signage

CHAPTER 2

server.py

CHAPTER 3

messenger_client.py

class messenger_client.**MessengerClient**

A class that interfaces with facebook Messenger and receives and sends message data.

handle_received_data(*data*)

Handle data received from the webhook.

Extracts the message text and sender_id and returns.

send_message(*recipient_id*, *message_text*, *quickreply_options_list*)

Send a custom FB message with message_text as the body.

This allows for quickreply options.

CHAPTER 4

database_client.py

class database_client.**DatabaseClient**
A client which connects to Mongo and deals with Mongo database operations.

find_or_create_user(*sender_id*, *name*)
Find or crate a user in the database.

Either finds a user by sender_id, or creates that user in the database. Returns the user.

find_tool_by_id(*tool_id*)
Find a tool by searching on the id field.

find_tool_by_name(*name*)
Find a tool by searching on the name field.

find_user(*field_name*, *field_value*)
Find a user from a given field, allowing to search by any field.

find_user_by_sender_id(*sender_id*)
Find one user by using the sender_id as the search field.

get_all_available_tools()
Return all non checked out tools in the Tools database.

get_all_tools()
Return all tools in the Tools database.

get_all_users()
Return all users in the Users database.

update_tool(*updated_tool*)
Update a tool in the database.

Given an updated tool dictionary with the same _id, replace the old database entry with the new one

update_user(*updated_user*)
Update a user in the database.

Given an updated user dictionary with the same sender_id, replaces the old database entry with the new one

class database_client.**User**(*sender_id*, *name*)
The class sructure used for the User in the Mongo database.

class database_client.**DatabaseClient**
A client which connects to Mongo and deals with Mongo database operations.

find_or_create_user(*sender_id*, *name*)
Find or crate a user in the database.

Either finds a user by sender_id, or creates that user in the database. Returns the user.

find_tool_by_id(*tool_id*)
Find a tool by searching on the id field.

find_tool_by_name(*name*)
Find a tool by searching on the name field.

find_user(*field_name*, *field_value*)
Find a user from a given field, allowing to search by any field.

find_user_by_sender_id(*sender_id*)
Find one user by using the sender_id as the search field.

get_all_available_tools()
Return all non checked out tools in the Tools database.

get_all_tools()
Return all tools in the Tools database.

get_all_users()
Return all users in the Users database.

update_tool(*updated_tool*)
Update a tool in the database.

Given an updated tool dictionary with the same _id, replace the old database entry with the new one

update_user(*updated_user*)
Update a user in the database.

Given an updated user dictionary with the same sender_id, replaces the old database entry with the new one

CHAPTER 5

conversation_handler.py

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

m

Index