GUIDE FOR ADDING NEW INTENTS

**15.1.2018**

There’s three things you have to do if you want to add a new intent to Libby. Firstly, you have to create a new intent for the LibbyDev bot. Secondly, you also have to create methods for the intent which are executed when Libby hears certain keywords. Thirdly, you have to connect your methods to the main handler for all intents.

# **LIBBYDEV BOT**

Creating new intent for a Lex bot is pretty easy. First log in to Amazon Web Service (AWS) - as usual - and then search for Amazon Lex. Now you should see all of our bots. Select LibbyDev. On the left side of the page there is a list of intents. Press the ‘+’-button top of the list and create a new intent. Name you intent and voilá, you have a fresh new intent.

To Sample utterances you can write the questions users can ask to trigger this intent. Use {slotname} to add variables in the input (e.g. “how much does {drink} cost”). To Lambda Initialization and validation check the checkbox and choose LibbyDevelop lambda function. To slots you may want to add your own slot. You can add one in the left side where the slot list is. Press the ‘+’-button and give your slot type a name, description (optional), and possible slots (variables). These variables can be passed to intent later. Most of the time you do not have to care about the rest of the options on the page.

Now you can save your intent (far down on the page) and build the bot (up right). After the build has finished you can test your intent with the “Test chatbot”-function on the right side of the page (it will not do anything yet).

# **ADDING METHODS TO LAMBDA FUNCTION**

**NOTE: DO THIS LOCALLY!**

Do not use the service provided by Amazon where you can write code in the browser because it is very buggy. Create a new python file in your text editor under the ‘lambda\_func’ directory. You probably want to import ‘util’ (“from . import util”). Then you can add your functions which process the input data and returns an output message. You can pass the input data as a parameter for the function (more about this later). The output message is basically a string which your function has to form. Most of the time your function should be fine returning ‘util.elicit\_intent({}, message)’. Sometimes you may want Libby to end the conversation after the answer. Then you can return ‘util.close(({}, ‘Fulfilled’, message)’. There are also other possibilities for the return, but those two should be enough in most cases. You should also write tests for your function. More about these in the test guide.

# **CONNECTING FUNCTIONS TO MAIN HANDLER**

Connecting your functions to the main handler is also pretty straight-forward. First add your method into the intents dictionary located in the ‘Router’ class in the file ‘router.py’ by adding ‘“your\_intent\_name”: your\_function\_name’. If your function needs parameters then add your parameters also in the route method’s if-chain just like there are a few examples already. Now if you have done everything correctly (and this guide is correct) you can try your function by writing tests for it and then by making a commit in GitHub. Then if everything works as expected you should be able to try your bot from the Amazon Lex’s test feature or with a RaspberryPi connected to Lex bot.