ASSIGNMENT 2: BUILDING A CHATBOT REPORT

David Poyitt_103953398_Date: 4/6/2022

Lecturer: Golam Zilani

Contents

1. Approach

There are various approaches that can effectively build the framework and the chatbot for the assignment. The method of development must be well structured. The creation of the framework and the order this was made had to be meticulous to fulfill all assignment requirements and outcomes and to create a functioning chatbot and Django framework that would communicate as an API, while meeting the needs of customers.

The Django framework was completed first to build the foundations of the application. This was essential in setting up the webpage so it could host a functioning chatbot. It turned out that the app didn't need to be connected for the assignment to be completed, but it was still critical for the foundations and framework of the website that the chatbot was of a high standard and satisfactory for all customers.

Amazon Lex and Lambda were the next two sections of the assignment that needed to be completed to create, test and distribute a working chatbot. There were certain criteria required to have a chatbot that would fulfill all customer requirements. At least four relevant utterances and at least five slots per intent needed to be included. These needed some planning so that the right utterances could be specified to aid the process of placing a booking for flights, hotel accommodation and car hire. Precise utterances were something that was important to get right as it involved making predictions of what the customer would ask the chatbot. To test these utterances, the chatbot was given to numerous real life test customers and a short survey of the utterances they used helped to shape some additional utterances. This gave a good indication as to what utterances would be made by customers. Lambda was also used to make improvements in the chatbot by adding essential code to make it more interactive and unique.

During the creation of the Lambda functions and the connection with the chatbot in Amazon Lex there were some issues that caused a need to restart this part of the assignment. A new Lambda file was created, and a new Amazon Lex Bot was created to re-start and improve the connectivity. This worked well and ensured working connection between the two web services. Although this was tedious and took additional time to complete, it was worth it as it created an improved, working chatbot. (See Figure 4)

2. Test Driven Development

Test-driven development (TDD) is such an important step in the development process. It is fundamental to producing a fail-safe software product that is user friendly. Before TTD was a part of the programming process, programmers used to write code and forget about checking for errors (Adzic, 2009). This would result in programs developed and released to the public that were full of bugs. The principle "test early, test often" encouraged programmers to test as they coded and removed most bugs that once plagued new release software products (Adzic, 2009). TDD is now a common practice, it was developed in the late 1990's by Kent Beck and is such an important process for programmers to master in this contemporary programming age (Qiu, 2021).

Many programmers state that TTD is not a natural process, it is a discipline that needs to be learnt and maintained by software developers (Percival, 2014). It has also reduced the production time to market as tests can occur during development instead of waiting for the end of the development to locate the errors (Acharya, 2013).

Implementing TDD into the assignment required the consistent checking of various projects to enable all of them to function together. It was particularly important to have working code for each of the specific areas of this assignment before moving onto the subsequent steps. For example, when using Lambda and adding the code using Python, it was important to consistently check that the code passed the tests and anytime that it failed, it had to be changed and re-published to enable an error free connection to the Amazon Lex Bot. As shown in figure.1, the code in this instance has no error and a new version of the Lex file can be published.

Continuously testing during the development stage enabled all errors to be located and solved. The importance of an error free Lambda code was vital to ensure the connection to the chatbot was working properly and secure.

One frustrating error that I spent two days trying to solve involved the final fulfillment of making the confirmation booking. Figure 2 shows the error message that would show every time a booking tried to be confirmed. To solve this issue the Lambda code needed to be re-assessed and checked for the error that was causing this. A similar error box also occurred numerous times throughout the bot-building process. This other error stated that there was an invalid chat-bot configuration error. This meant that the chatbot had a communication problem with Amazon Lex. This error was resolved by ticking some of the slot boxes that had for some reason, by default, been left unticked. (See figure 3)

The other important TDD process that was often applied was the actual testing of the chatbot, which required user input. The chatbot was active during the entire assignment process to test functionality, code and outcomes that needed to be fulfilled. This was the ultimate way to find out if all code was communicating and working according to the given criteria.

If there were improvements that could be adapted onto the assignment for next time it would be to add a greeting intent that would welcome the customer to the chatbot. This would allow for a more customer friendly experience. Another addition would be to customise a confirmation email to be sent out to the user once the booking is successfully, with all the necessary details included.

TTD was also used for the Django framework, although setting this part of the assignment up was relatively simple due to the familiarity of working with PyCharm and its simpleness when compared to the more recent systems Lex and Lambda that I had to learn.

3. Links

The link below contains all assignment material.

Find GitHub link here…https://github.com/103953398/Swinburne_Assignment_2

4. References

Acharya, S., 2013. *Test-driven development with Mockito.* 1st Edition ed. Birmingham: Packt Publishing.

Adzic, G., 2009. *Test Driven .NET Development with FitNesse.* 2nd Edition ed. London: Neuri Limited.

Amrit, C., 2018. Effectiveness of Test- Driven Development and Continuous Integration. *TEST-DRIVEN DEVELOPMENT AND CONTINUOUS INTEGRATION,* XX(1), pp. 27-36.

Beck, K., 2033. *Test-driven development by example.* 1st Edition ed. Boston: Addison-Wesley.

Madeyski, L., 2010. *Test-Driven Development: An Empirical Evaluation of Agile Practice.* 1st Edition ed. Berlin: Springer-Verlag.

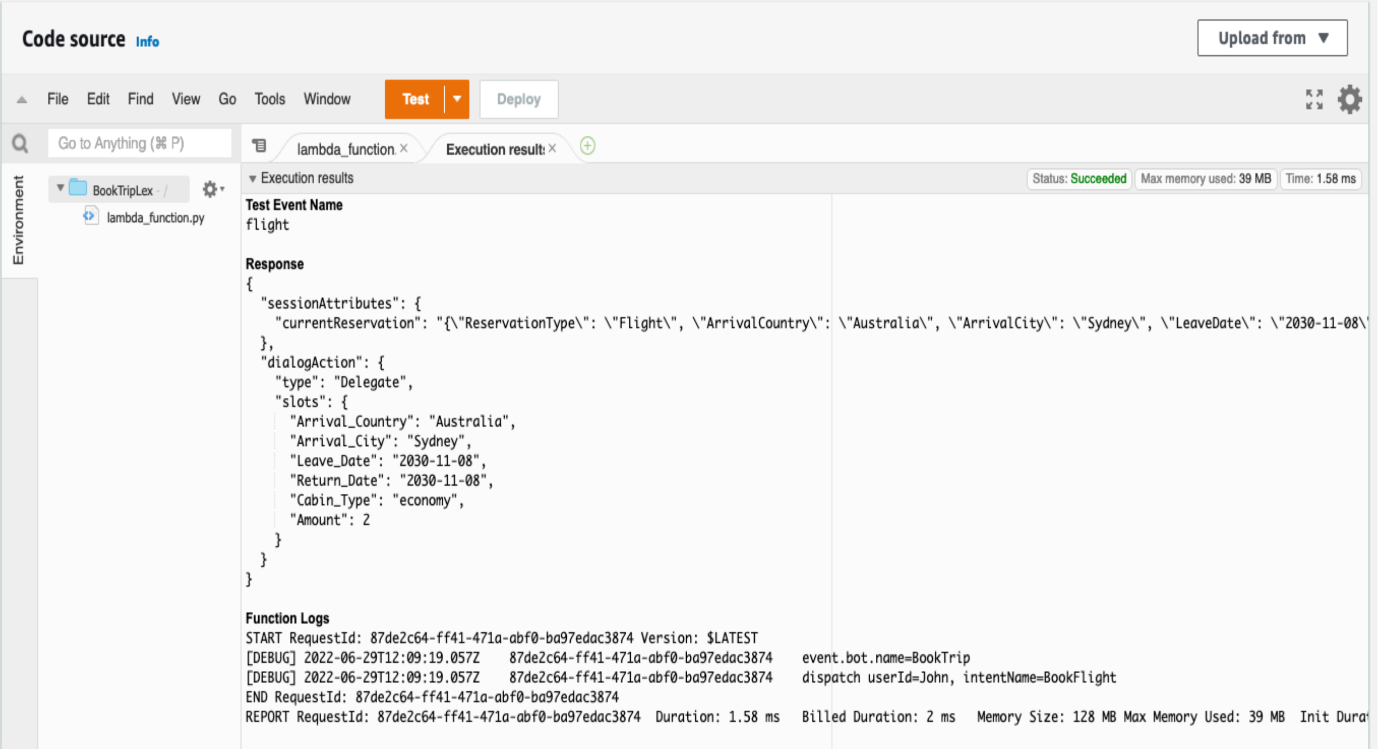
Percival, H., 2014. *Test-driven development with Python.* 1st Edition ed. California: O'Reilly.

Qiu, J., 2021. *Test-Driven Development with React : Apply Test-Driven Development in Your Applications.* 4th Edition ed. Place of publication not identified: Apress.

Singman, P., 2020. How I write meanigngful tests for AWS Lambda functions. *Towards Data Science*, 19 August.

Swinburne, 2022. *Module 6.7.1: Test-driven development.* s.l.:Swinburne Online.

5. Appendix



Status: Succeeded | Max memory used: 39 MB | Time: 1.58 ms

Test Event Name

```
flight
```

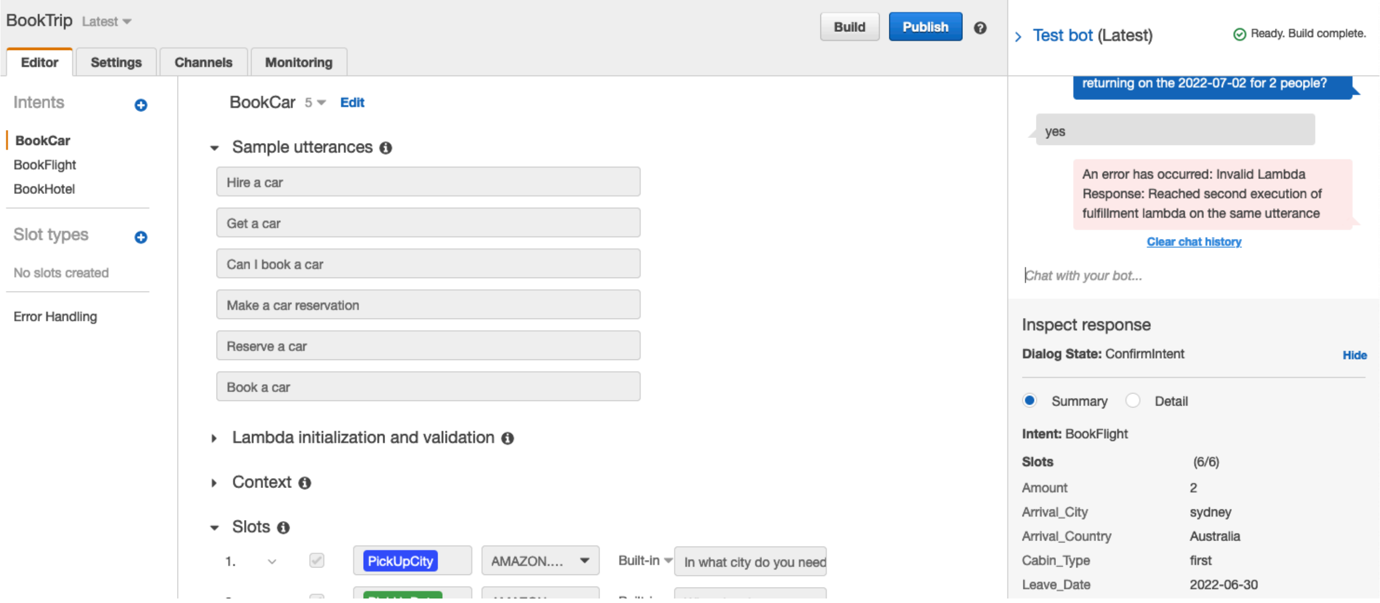
Response

```
{
  "sessionAttributes": {
    "currentReservation": "{\"ReservationType\": \"Flight\", \"ArrivalCountry\": \"Australia\", \"ArrivalCity\": \"Sydney\", \"LeaveDate\": \"2030-11-08\'
  },
  "dialogAction": {
    "type": "Delegate",
    "slots": {
      "Arrival_Country": "Australia",
      "Arrival_City": "Sydney",
      "Leave_Date": "2030-11-08",
      "Return_Date": "2030-11-08",
      "Cabin_Type": "economy",
      "Amount": 2
    }
  }
}
```

Function Logs

```
START RequestId: 87de2c64-ff41-471a-abf0-ba97edac3874 Version: $LATEST
[DEBUG] 2022-06-29T12:09:19.057Z    87de2c64-ff41-471a-abf0-ba97edac3874    event.bot.name=BookTrip
[DEBUG] 2022-06-29T12:09:19.057Z    87de2c64-ff41-471a-abf0-ba97edac3874    dispatch userId=John, intentName=BookFlight
END RequestId: 87de2c64-ff41-471a-abf0-ba97edac3874
REPORT RequestId: 87de2c64-ff41-471a-abf0-ba97edac3874  Duration: 1.58 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max Memory Used: 39 MB  Init Dura
```

Figure 1: See status in top right in green that says ‘succeeded’. This code can then be published and pushed to the Lex Bot.



Sample utterances

- Hire a car
- Get a car
- Can I book a car
- Make a car reservation
- Reserve a car
- Book a car

1. ☑ PickUpCity AMAZON.... Built-in In what city do you need

returning on the 2022-07-02 for 2 people?

yes

An error has occurred: Invalid Lambda Response: Reached second execution of fulfillment lambda on the same utterance

Inspect response

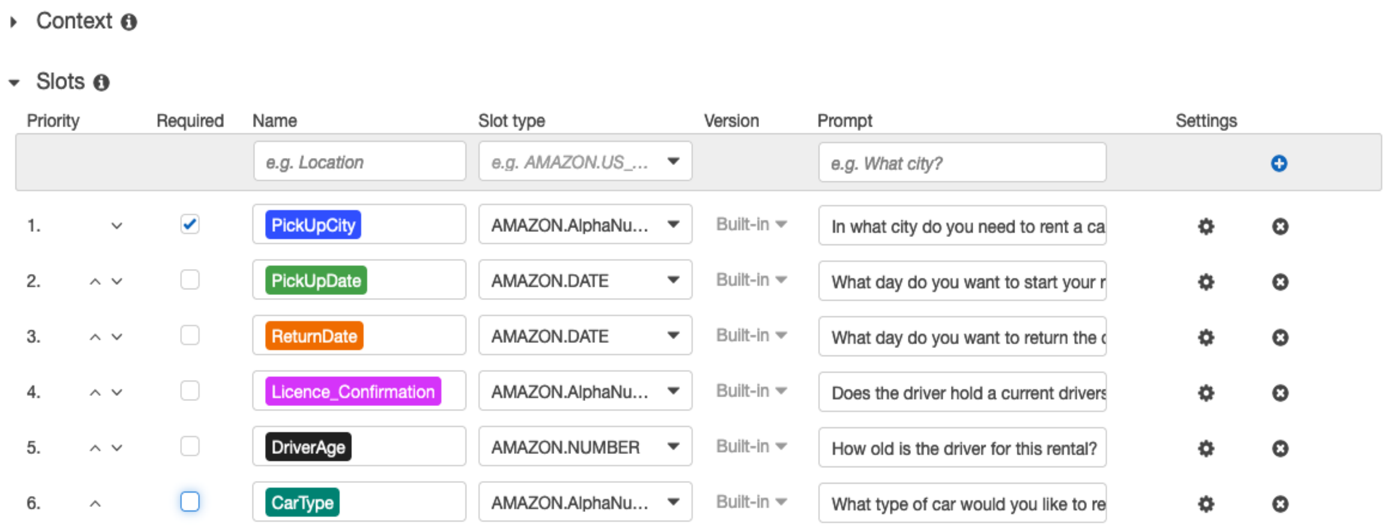
Dialog State: ConfirmIntent

● Summary ○ Detail

Intent: BookFlight

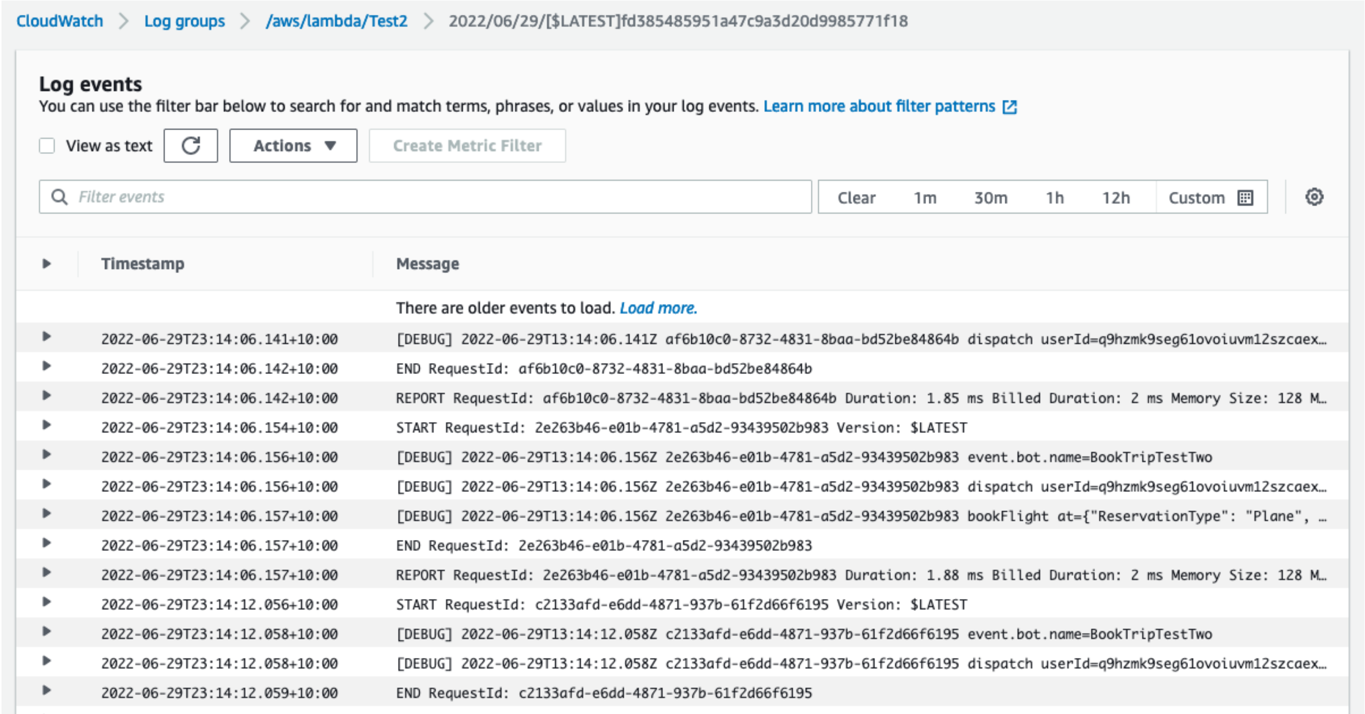
Slots	(6/6)
Amount	2
Arrival_City	sydney
Arrival_Country	Australia
Cabin_Type	first
Leave_Date	2022-06-30

Figure 2: This image shows an error with the final fulfillment of booking the flight.



Priority	Required	Name	Slot type	Version	Prompt	Settings
		e.g. Location	e.g. AMAZON.US_...		e.g. What city?	
1.	☑	PickUpCity	AMAZON.AlphaNu...	Built-in	In what city do you need to rent a ca	
2.	□	PickUpDate	AMAZON.DATE	Built-in	What day do you want to start your r	
3.	□	ReturnDate	AMAZON.DATE	Built-in	What day do you want to return the c	
4.	□	Licence_Confirmation	AMAZON.AlphaNu...	Built-in	Does the driver hold a current drivers	
5.	□	DriverAge	AMAZON.NUMBER	Built-in	How old is the driver for this rental?	
6.	□	CarType	AMAZON.AlphaNu...	Built-in	What type of car would you like to re	

Figure 3: Notice the boxes next to the slots are unticked. This caused the error and they needed to be all ticked.



CloudWatch > Log groups > /aws/lambda/Test2 > 2022/06/29/[$LATEST]fd385485951a47c9a3d20d9985771f18

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

□ View as text
Actions
Create Metric Filter
Filter events
Clear
1m
30m
1h
12h
Custom

```
   Timestamp                        Message
                                    There are older events to load. Load more.
▶  2022-06-29T23:14:06.141+10:00   [DEBUG] 2022-06-29T13:14:06.141Z af6b10c0-8732-4831-8baa-bd52be84864b dispatch userId=q9hzmk9seg61ovoiuvm12szcaex…
▶  2022-06-29T23:14:06.142+10:00   END RequestId: af6b10c0-8732-4831-8baa-bd52be84864b
▶  2022-06-29T23:14:06.142+10:00   REPORT RequestId: af6b10c0-8732-4831-8baa-bd52be84864b Duration: 1.85 ms Billed Duration: 2 ms Memory Size: 128 M…
▶  2022-06-29T23:14:06.154+10:00   START RequestId: 2e263b46-e01b-4781-a5d2-93439502b983 Version: $LATEST
▶  2022-06-29T23:14:06.156+10:00   [DEBUG] 2022-06-29T13:14:06.156Z 2e263b46-e01b-4781-a5d2-93439502b983 event.bot.name=BookTripTestTwo
▶  2022-06-29T23:14:06.156+10:00   [DEBUG] 2022-06-29T13:14:06.156Z 2e263b46-e01b-4781-a5d2-93439502b983 dispatch userId=q9hzmk9seg61ovoiuvm12szcaex…
▶  2022-06-29T23:14:06.157+10:00   [DEBUG] 2022-06-29T13:14:06.156Z 2e263b46-e01b-4781-a5d2-93439502b983 bookFlight at={"ReservationType": "Plane", …
▶  2022-06-29T23:14:06.157+10:00   END RequestId: 2e263b46-e01b-4781-a5d2-93439502b983
▶  2022-06-29T23:14:06.157+10:00   REPORT RequestId: 2e263b46-e01b-4781-a5d2-93439502b983 Duration: 1.88 ms Billed Duration: 2 ms Memory Size: 128 M…
▶  2022-06-29T23:14:12.056+10:00   START RequestId: c2133afd-e6dd-4871-937b-61f2d66f6195 Version: $LATEST
▶  2022-06-29T23:14:12.058+10:00   [DEBUG] 2022-06-29T13:14:12.058Z c2133afd-e6dd-4871-937b-61f2d66f6195 event.bot.name=BookTripTestTwo
▶  2022-06-29T23:14:12.058+10:00   [DEBUG] 2022-06-29T13:14:12.058Z c2133afd-e6dd-4871-937b-61f2d66f6195 dispatch userId=q9hzmk9seg61ovoiuvm12szcaex…
▶  2022-06-29T23:14:12.059+10:00   END RequestId: c2133afd-e6dd-4871-937b-61f2d66f6195
```

Figure 4: This shows the log events for the Amazon services providing details of errors made throughout the assignment. Keeping a log of all this also helped detect and solve any recurring coding issues.