amazon alexa

Introduction to Build AI Chatbots using AWS LEX

Mariusz Momotko
Senior SDE
Amazon Alexa NLU

amazon alexa

Amazon Development Center in Gdańsk

- Text to Speech (TTS)
- Natural Language Understanding (NLU)
- Alexa Hybrid

amazon alexa

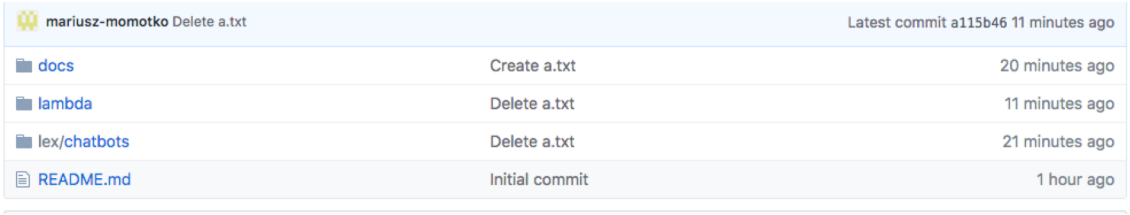
Agenda

- 17:30 – 19:30 Workshop
  - Part 1 – Build a chat bot
  - Part 2 – Build a useful chat bot
  - Part 3 – Build even more useful bot
- 19:30 - … Networking pizza(s)

amazon alexa

MeetUp materials

- All materials are available on [aws-lex-chatbot GitHub project]
  - Lambda functions (+ tests)
  - Bot (definition + deployment)
  - This presentation (PDF)



On slides – this symbol indicates that you can use materials if you do exercises on your own.

amazon alexa

Part 1 – Build a AWS Lex chat bot

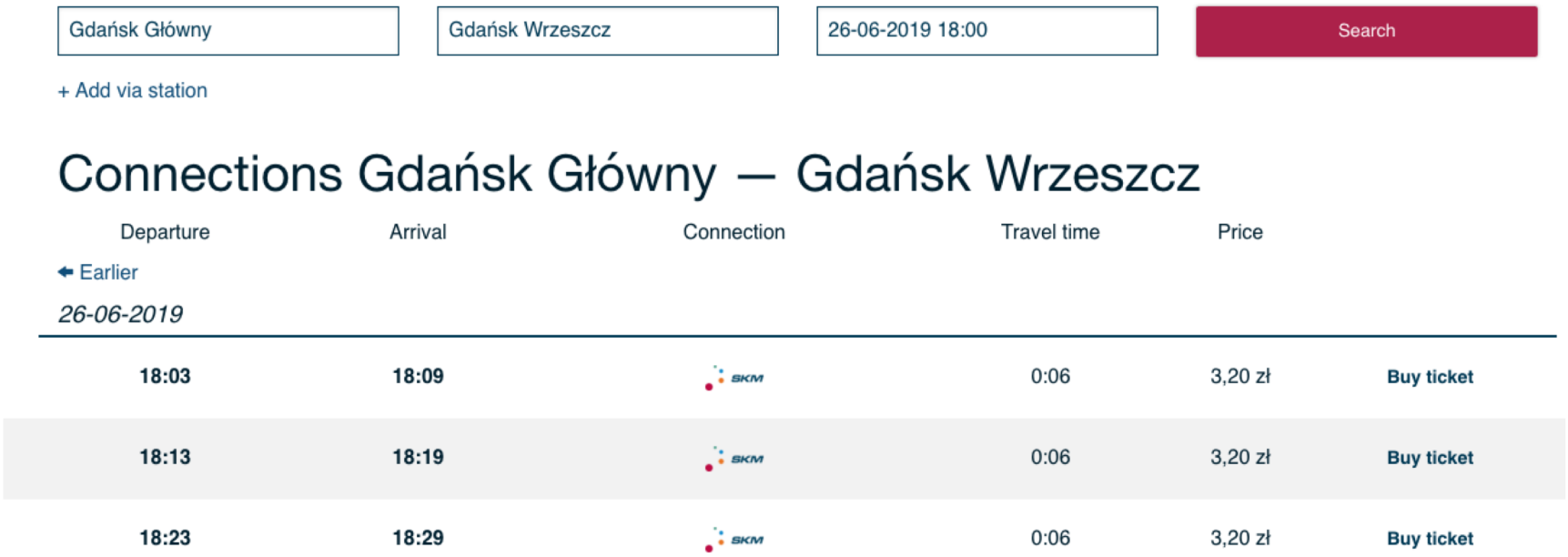
Specify requirements for a bot to support buying a train ticket for SKM

Create a bot

Define intent and slots

amazon alexa

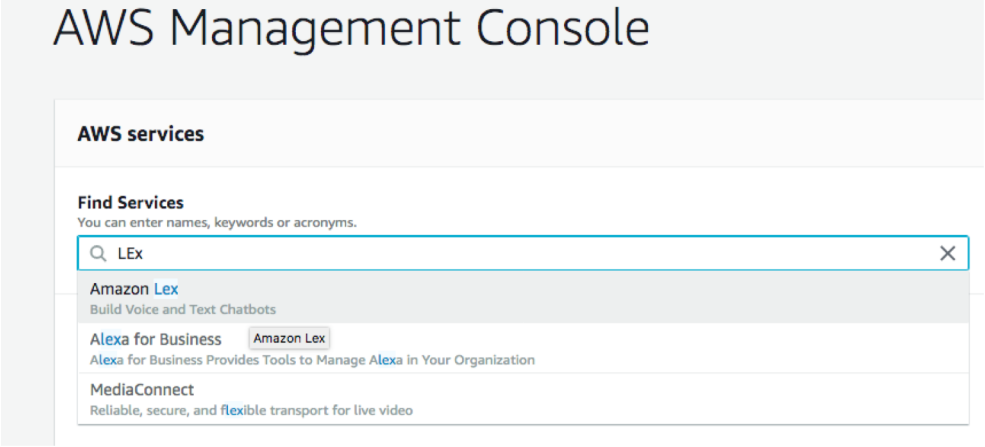
A need – I want to buy a train ticket for SKM

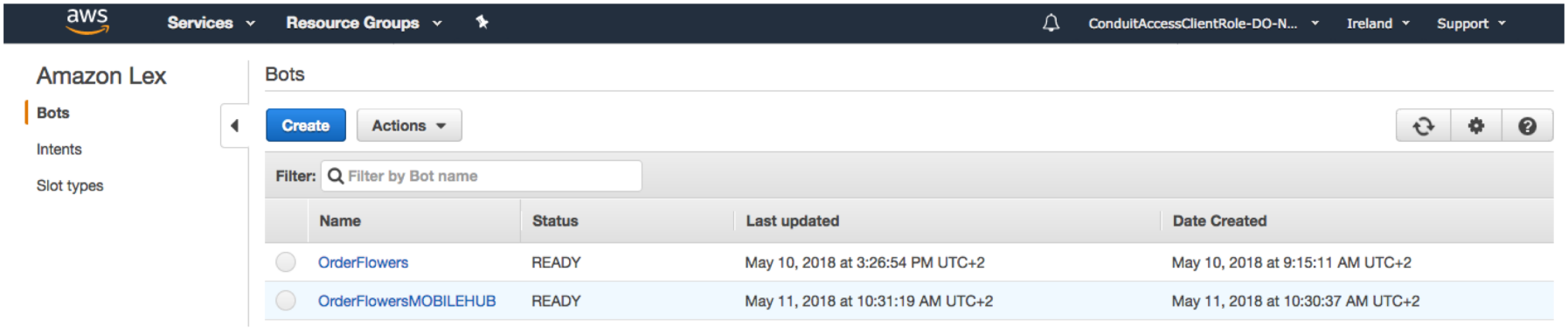


https://bilety.skm.pkp.pl/en/search/gdansk-glowny/gdansk-wrzeszcz/26-06-2019_18:00

amazon alexa

AWS Lex



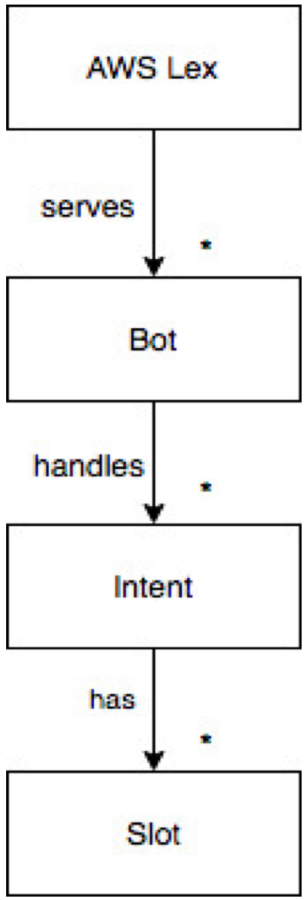


amazon alexa

AWS Lex Bot – Bots, Intents & Slots

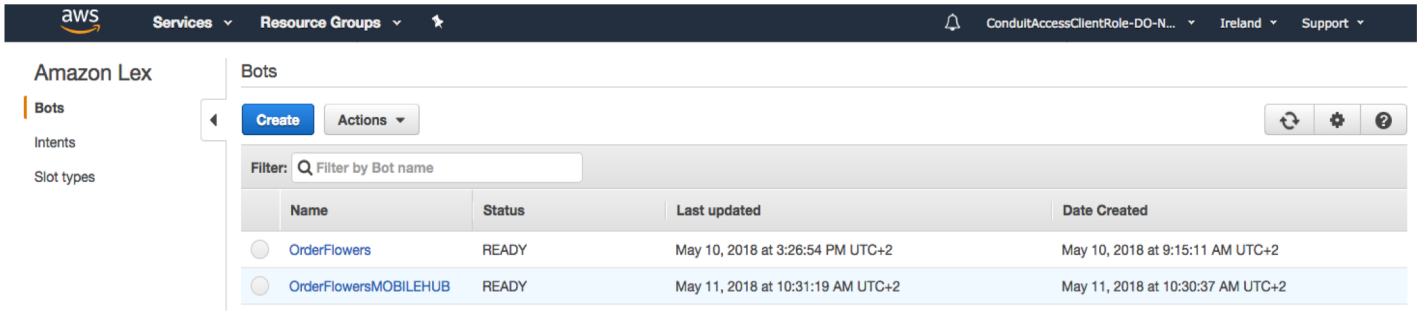
- Bot - a ro(bot) able to carry out conversation with a user (in a textual or voice enabled mode)
- Intent – a particular goal that the user wants to achieve
- Slot – data / parameters that the user need to fulfill in order to understand the goal

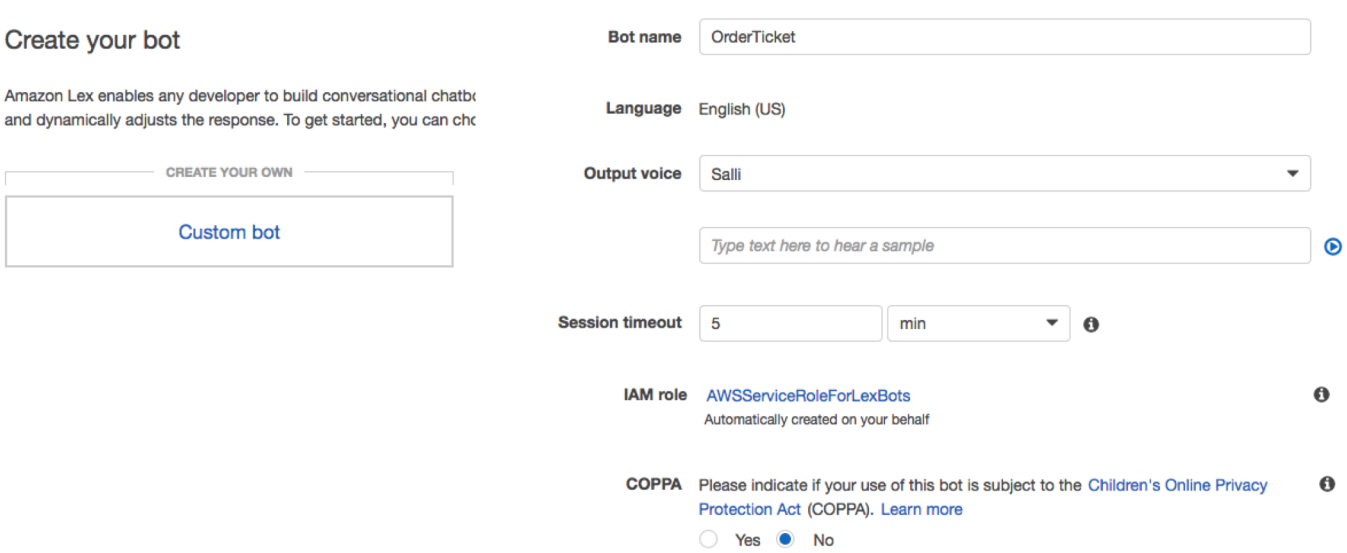
- Examples
  - *Play Hello by Adele*
    - Intent - PlayMusicIntent
    - Slots: [ArtistName="Adele"], [SongName="Hello"]
  - *Check the weather in Gdańsk*
    - Intent: ForecastedWeatherIntent
    - Slots: [place="Gdańsk"]

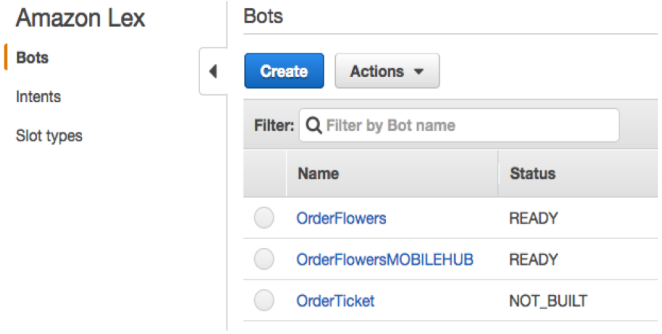


amazon alexa

Support in ordering a SKM ticket – Lex Bot

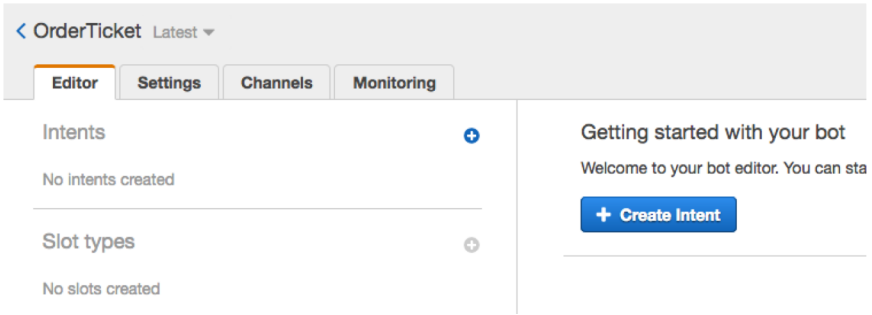


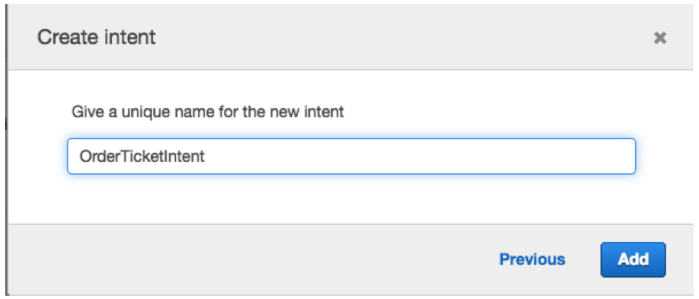




Ordering a ticket – Intent (1)

- Intent – OrderTicket
- Slots
  - fromStation
  - toStation
  - dateOfTravel
  - timeOfTravel
- Built-in Intents
  - HelpIntent
  - CancelIntent
  - …

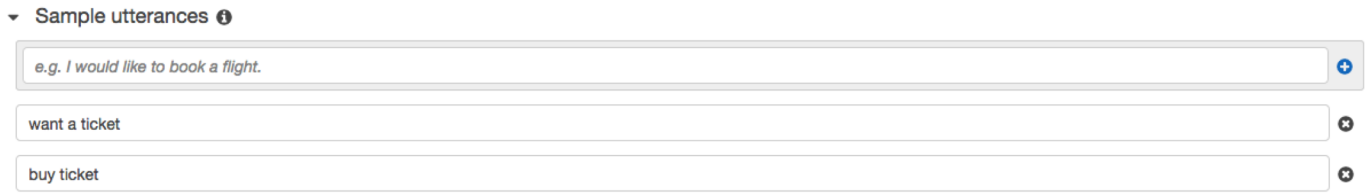




amazon alexa

Ordering a ticket – Intent – Sample utterances

- Utterance – spoken or typed phrases that invoke the customer's intent



Ordering a ticket – Intent – Slots

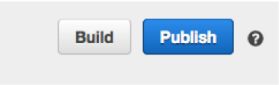
- Slot type
  - Built-in slots [Alexa Skills Kit]
    - Date, Time, Numbers, Cities
    - …

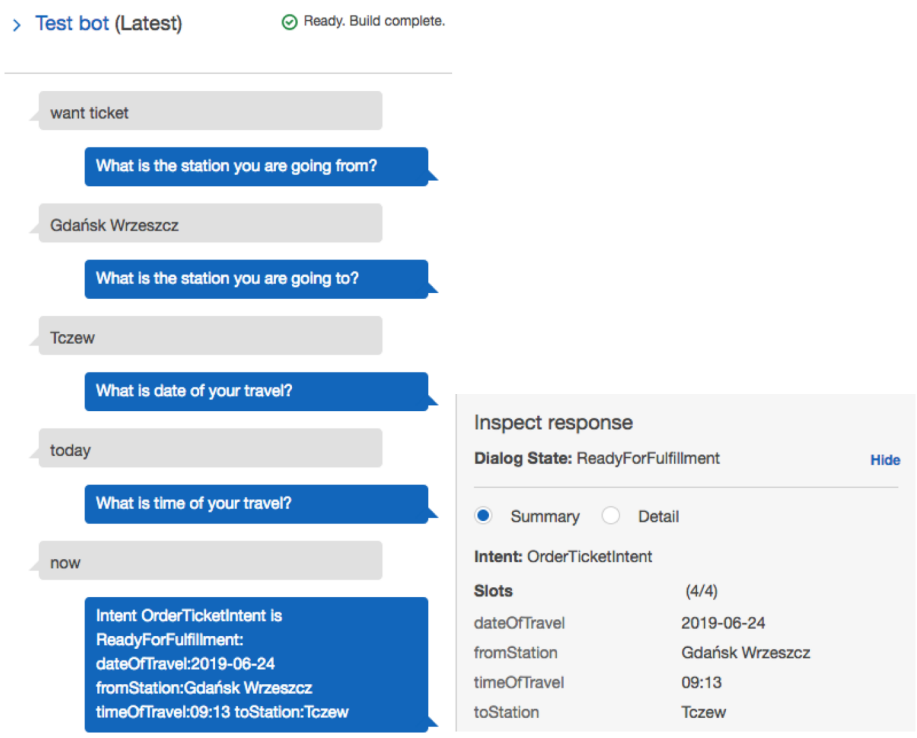
Slots

Priority	Required	Name	Slot type	Version	Prompt	Settings
		e.g. Location	e.g. AMAZON.US_CITY		e.g. What city?	
1.	✓	fromStation	AMAZON.EUROPE_CITY	Built-in	What is the station you are going from?	
2.	✓	toStation	AMAZON.EUROPE_CITY	Built-in	What is the station you are going to?	
3.	✓	dateOfTravel	AMAZON.DATE	Built-in	What is date of your travel?	
4.	✓	timeOfTravel	AMAZON.TIME	Built-in	What is time of your travel?	

Ordering a ticket – first try

- First try (after Build)
  - Interaction
  - Summary (ReadyForFulfillment)
  - Inspect response
- Bot which was built can be published
- Bot is versioned

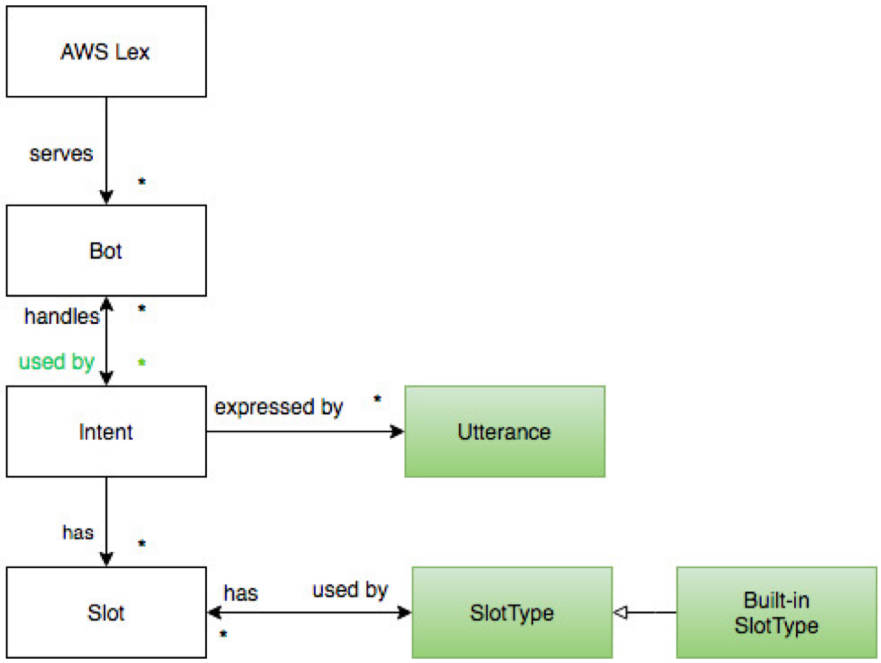




amazon alexa

Part 1 - Summary

- We have a working chat bot
  - Recognize customer utterances
  - Ask for customer choices (slots)
  - Provide final response
- BUT
  - Customer can provide fake strain stations, we cannot validate them
  - It is not possible to cancel customer order
  - Final information is unreadable



amazon alexa

Part 2 – Build a useful AWS Lex chat bot

Adding confirmation prompts

Validation via customized slot type

Formatting the bot's response

Error Handling

amazon alexa

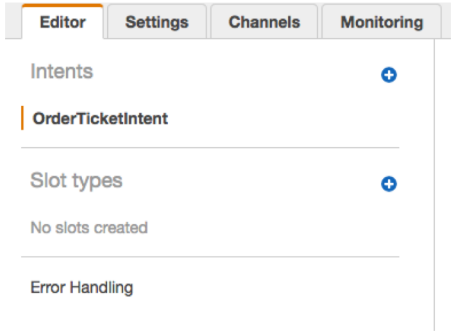
Adding confirmation prompts

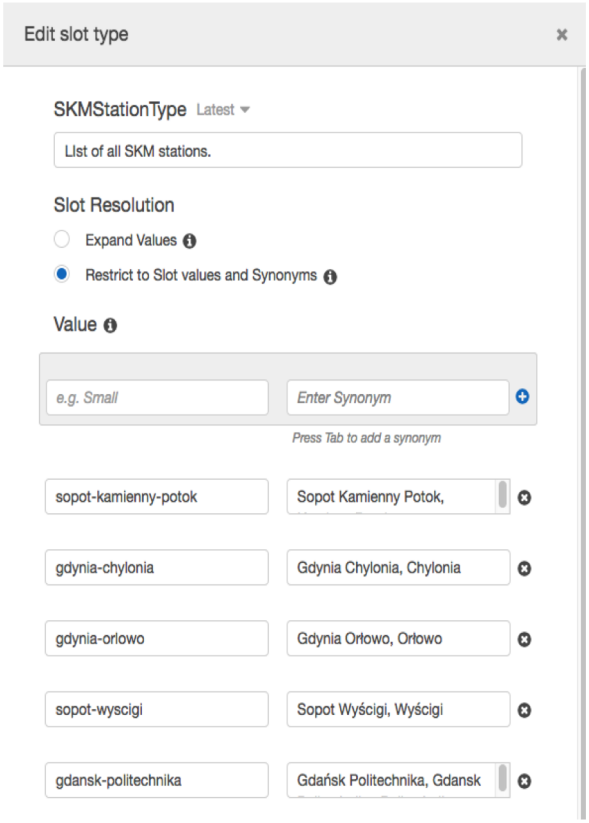
- Confirm – a question to confirm your intent and choices (slots). It appears at the end of the conversation.
- Cancel – a message which will appear when you cancel your intent.

⚠ Before testing - remember to save the intent definition and build it.

Define new slot type for SKM stations

- Customized slot type for SKM stations
  - Via Lex GUI (synonyms only in English)
  - Via Export/Import feature



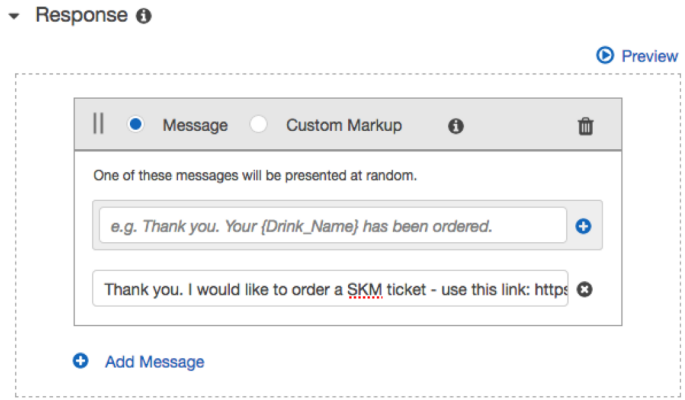


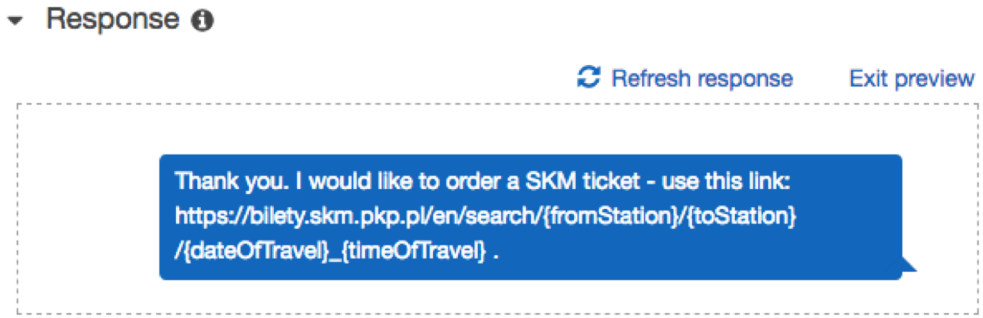
⚠ Remember to select proper version of the slot type in slots definition.

To import already defined features for this slot type – use [this file](#).

Formatting the bot’s response

- Response
  - final element of a bot’s intent, and are displayed to users after the fulfillment of the intent is complete
- Textual response
  - Slot values: {slotName}, e.g. {fromStation}
  - Session attributes value (will discuss it a bit later)

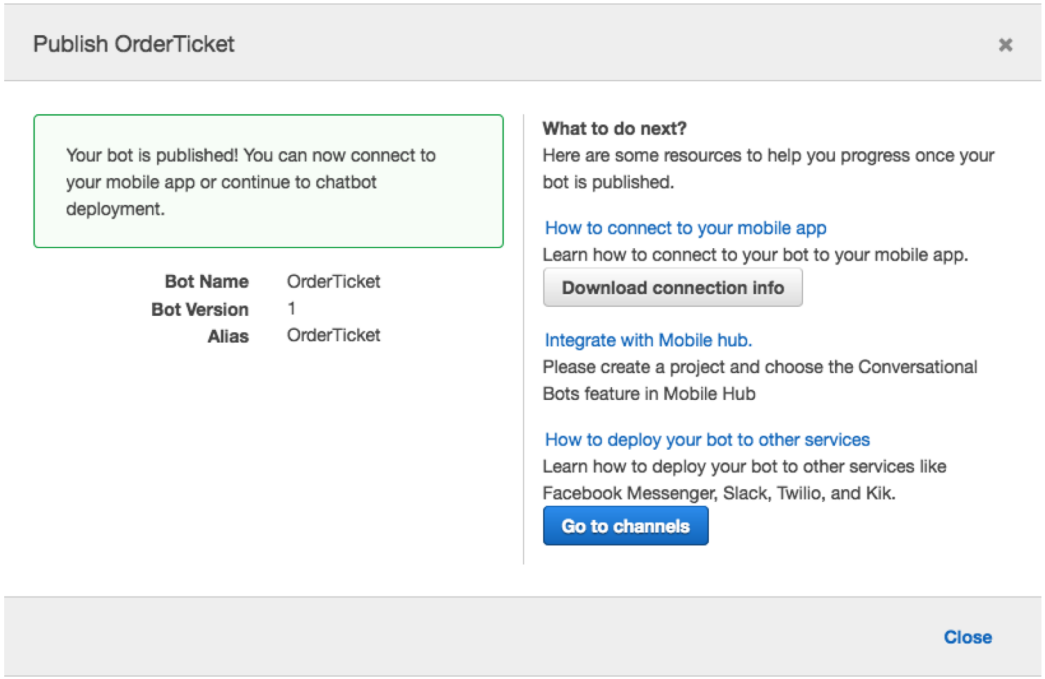




amazon alexa

Error handling

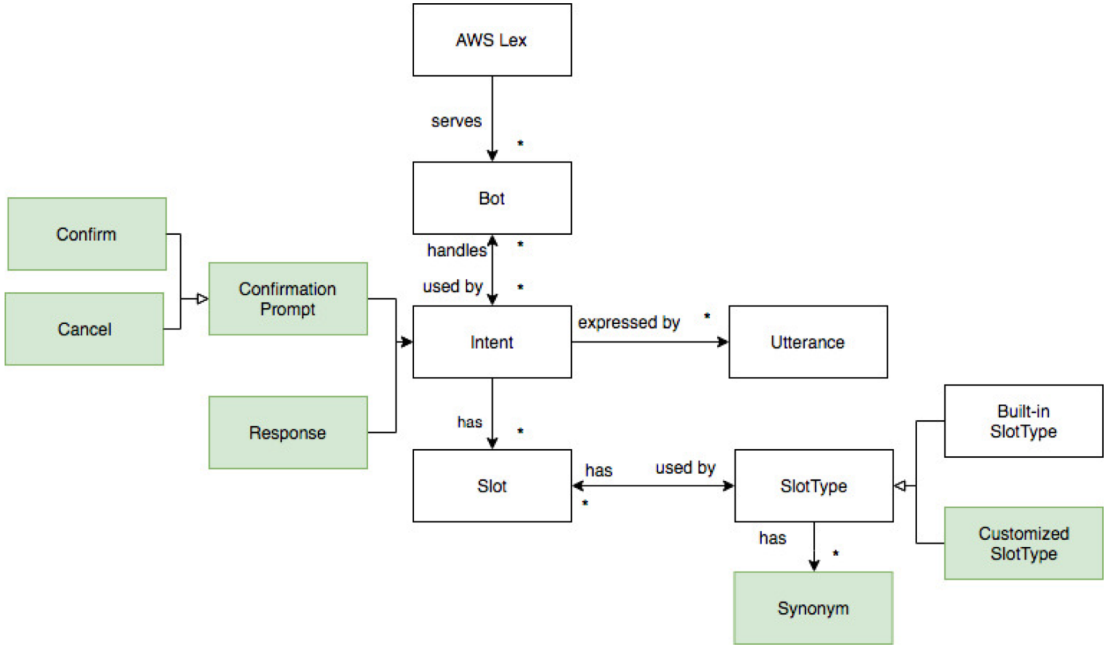
Publish the bot



amazon alexa

Part 2 - Summary

- We are able to:
  - provide proper from / to SKM stations
  - ask for the final confirmation of the customer intent
  - Provide final message (SKM url)
  - handle errors
- BUT
  - Customer may provide the same from and to stations
  - The final url is not correct (date format)



amazon alexa

Part 3 – Building **even more useful** bot

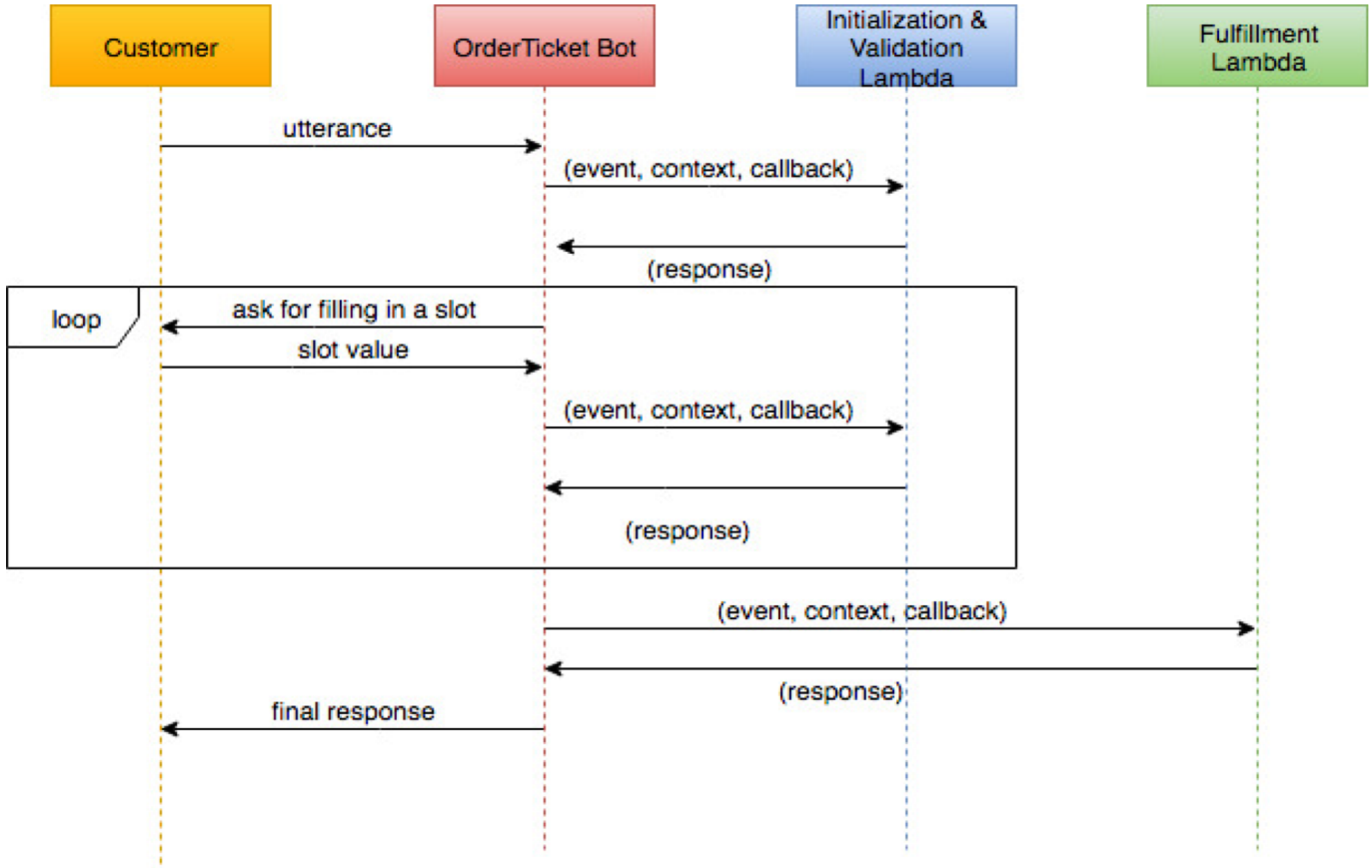
Advanced validation (lambda)

Advanced response formatting (lambda)

Deploy a demo application

amazon alexa

Lex Bot Interaction Model



amazon alexa

Lex Data Structures - Input

- CurrentIntent
  - Slots – if not filled in – empty
- Bot metadata
- Request metadata
- Session attributes
- Request attributes

```
{
  "currentIntent": {
    "name": "intent-name",
    "slots": {
      "slot name": "value",
      "slot name": "value"
    },
    "slotDetails": {
      "slot name": {
        "resolutions" : [
          { "value": "resolved value" },
          { "value": "resolved value" }
        ],
        "originalValue": "original text"
      },
      "slot name": {
        "resolutions" : [
          { "value": "resolved value" },
          { "value": "resolved value" }
        ],
        "originalValue": "original text"
      }
    },
    "confirmationStatus": "None, Confirmed, or Denied (intent confirmation, if configured)"
  },
  "bot": {
    "name": "bot name",
    "alias": "bot alias",
    "version": "bot version"
  },
  "userId": "User ID specified in the POST request to Amazon Lex.",
  "inputTranscript": "Text used to process the request",
  "invocationSource": "FulfillmentCodeHook or DialogCodeHook",
  "outputDialogMode": "Text or Voice, based on ContentType request header in runtime API request",
  "messageVersion": "1.0",
  "sessionAttributes": {
     "key": "value",
     "key": "value"
  },
  "requestAttributes": {
     "key": "value",
     "key": "value"
  }
}
```

amazon alexa

Lex Data Structures - Output

- (selected) Dialog actions
  - Close – not expecting a response from the user
  - ConfirmIntent
  - Delegate – continue conversation with the user
  - ElicitSlot – the user is expected to provide a value for a given slot

```
{
    "sessionAttributes": {
    "key1": "value1",
    "key2": "value2"
    ...
  },
  "dialogAction": {
    "type": "ElicitIntent, ElicitSlot, ConfirmIntent, Delegate, or Close",
    Full structure based on the type field. See below for details.
  }
}
```

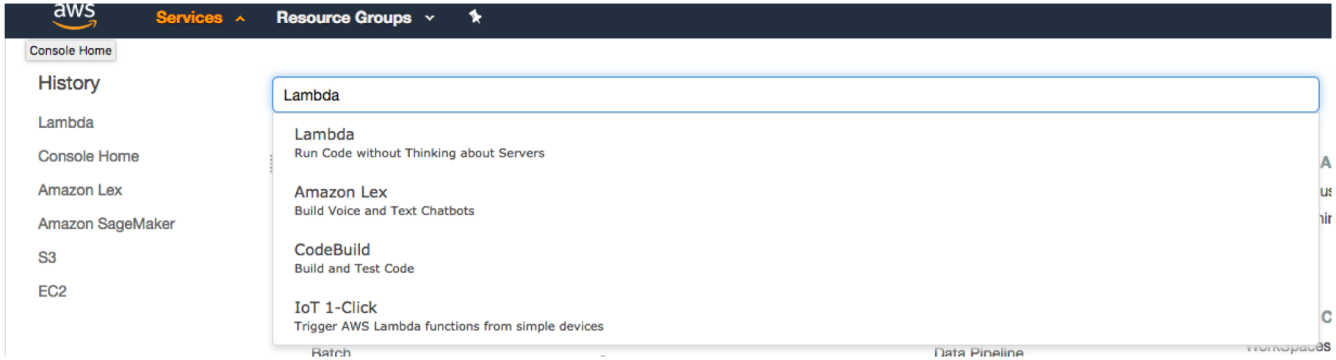
amazon alexa

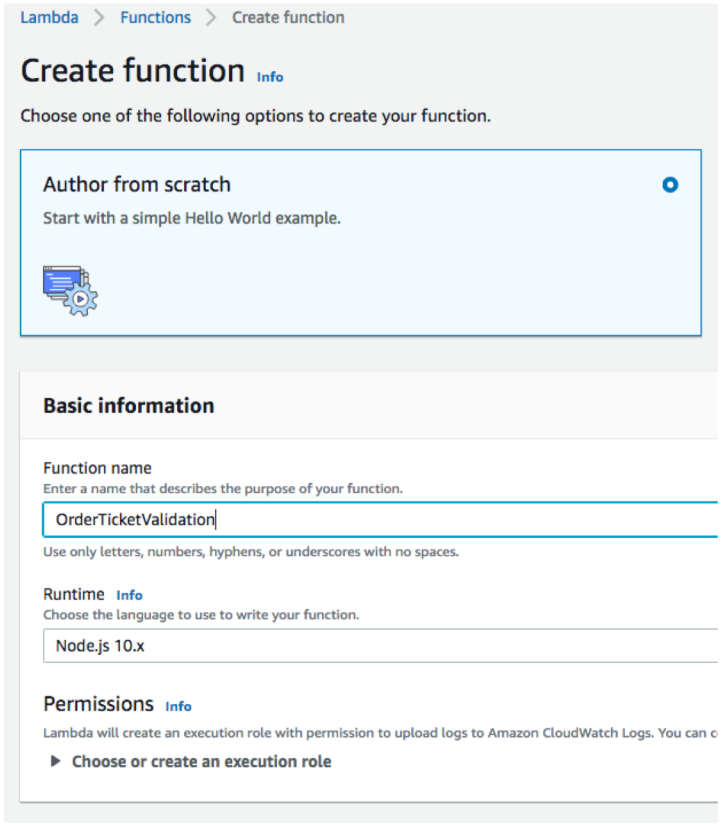
Lambda function implemented in Node.js

- Handler – the method in Lambda function that processes events
  - Types
    - Async handler
    - Non-async handler (recommended by Lex)
  - Inputs
    - Event – chatbot data compliant with the Lex input structure
    - Callback – a handler to call Lex
  - Output
    - Chatbot command and other chatbot data compliant with the Lex output structure

amazon alexa

Create a Lambda function

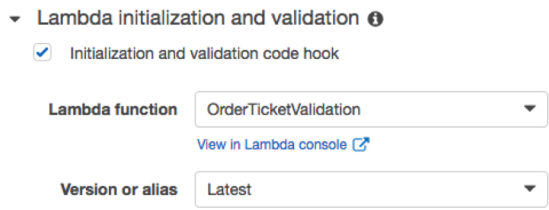




Initialization and validation using Lambda OrderTicketValidation

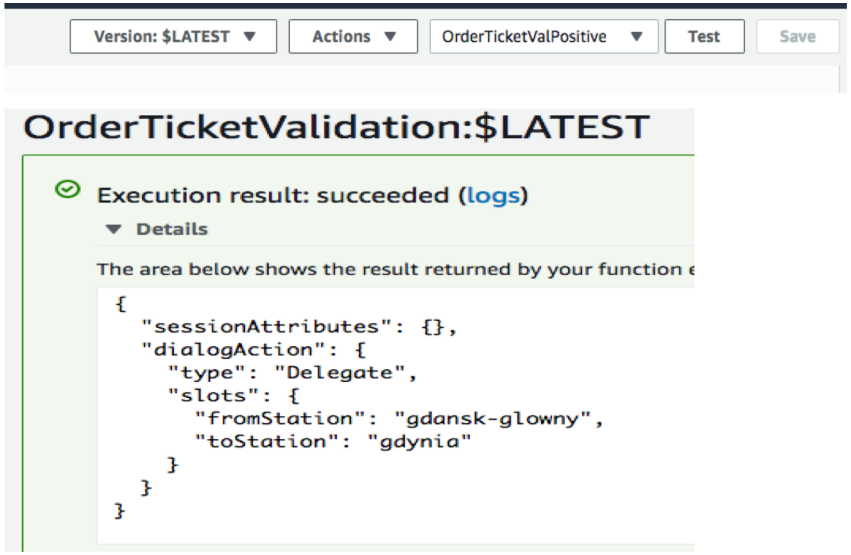
- Step 1 - Import lambda function – OrderTicketValidation.zip
- Step 2 – analyze validation rule for checking from/to stations
  - Get the request
  - Get fromStation slot value (empty)
  - Get toStation slot value (empty)
  - Compare from/to stations
  - Return proper action
    - Delegate – if everything is OK
    - ElicitSlot – if toStation slot needs to be corrected
- Step 3 – test the function

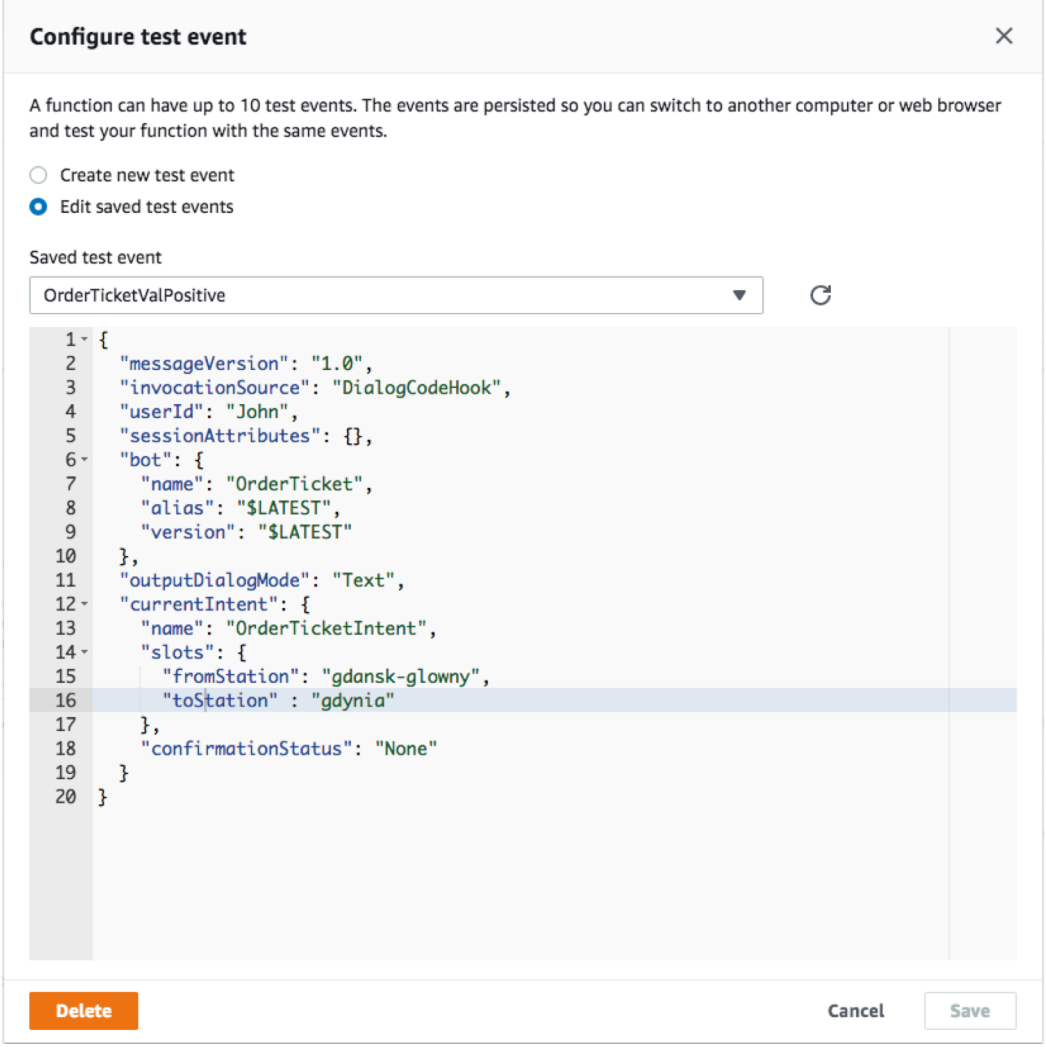
To import lambda code of this function – use this file.
Tests can be imported separately – they are available here.



Test the function

- Provide input with slots
- Verify the result





Formatting the result using Lambda OrderTicketFulfillment

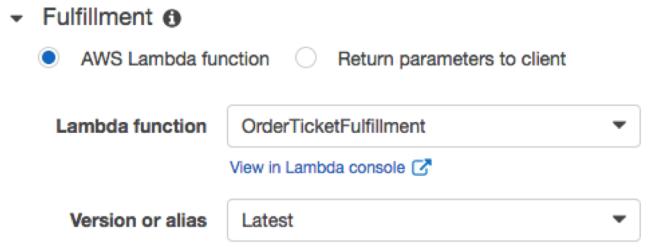
amazon alexa

To order SKM ticket we need to generate proper url, an example:

https://bilety.skm.pkp.pl/en/search/gdansk-glowny/gdansk-wrzeszcz/26-06-2019_18:00

The expected pattern:

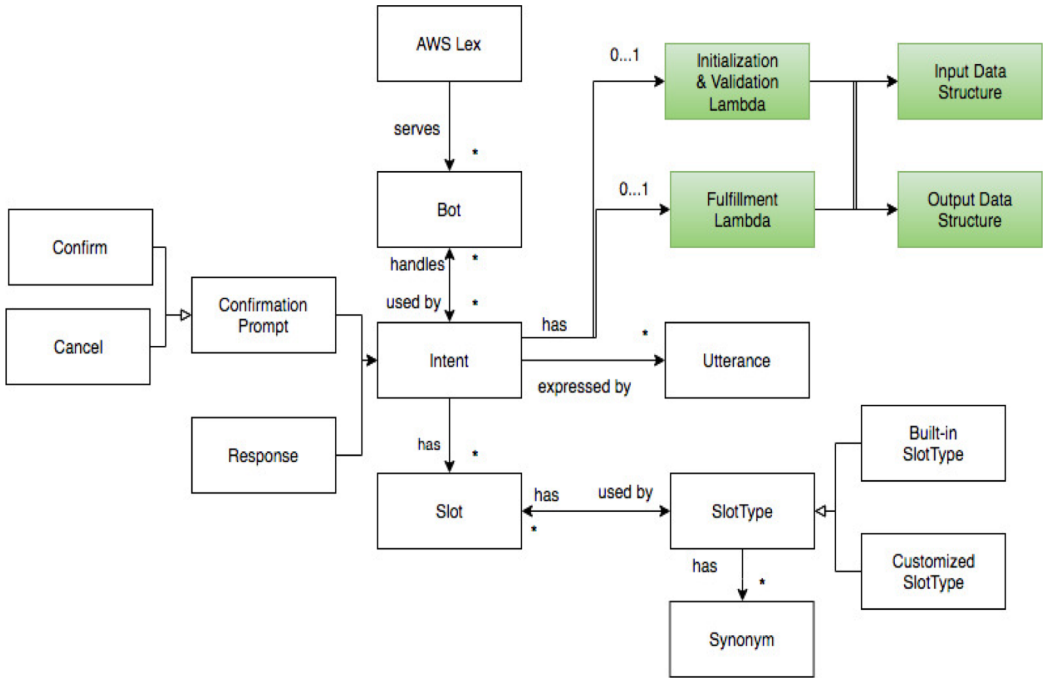
https://bilety.skm.pkp.pl/en/search/<fromStation>/<toStation>/<DD>-<MM>-<YYYY>_<HH:MI>



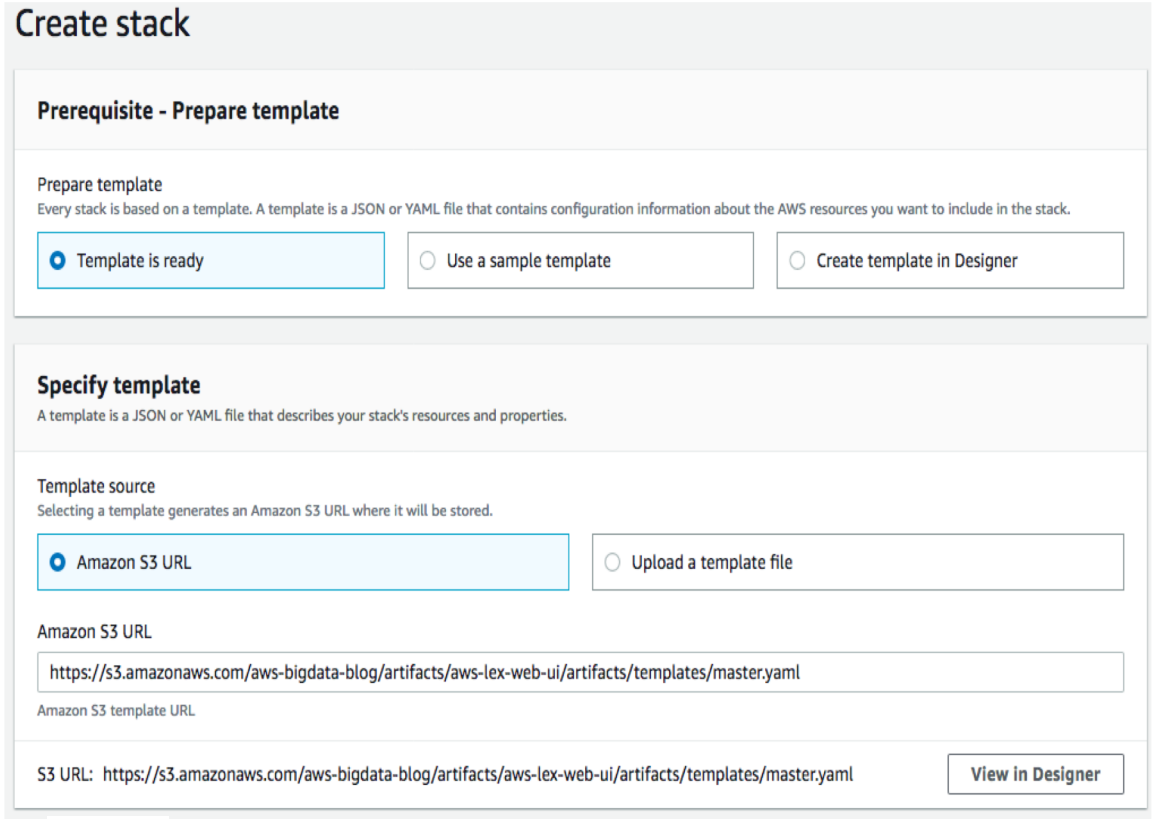
To import lambda code of this function – use this file.
Tests can be imported separately – they are available here.

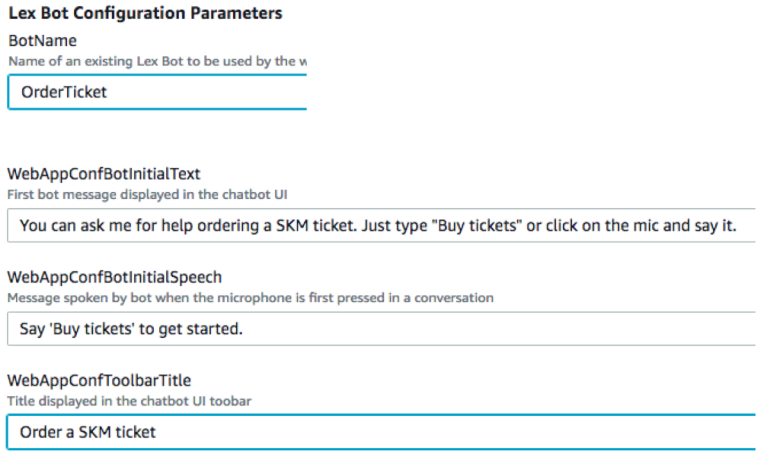
Part 3 - Summary

- We are able to
  - Provide data and validate them
  - Get proper url that routes us to SKM web page
- BUT
  - This work for a single ticket only
  - Still there are many parts of ticket ordering which have to be done manually



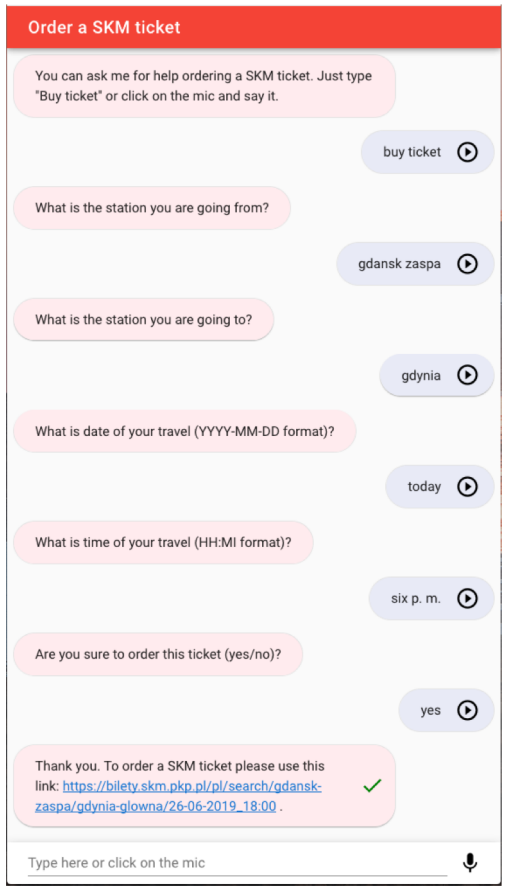
CloudFormation - Deploy a WebUI application

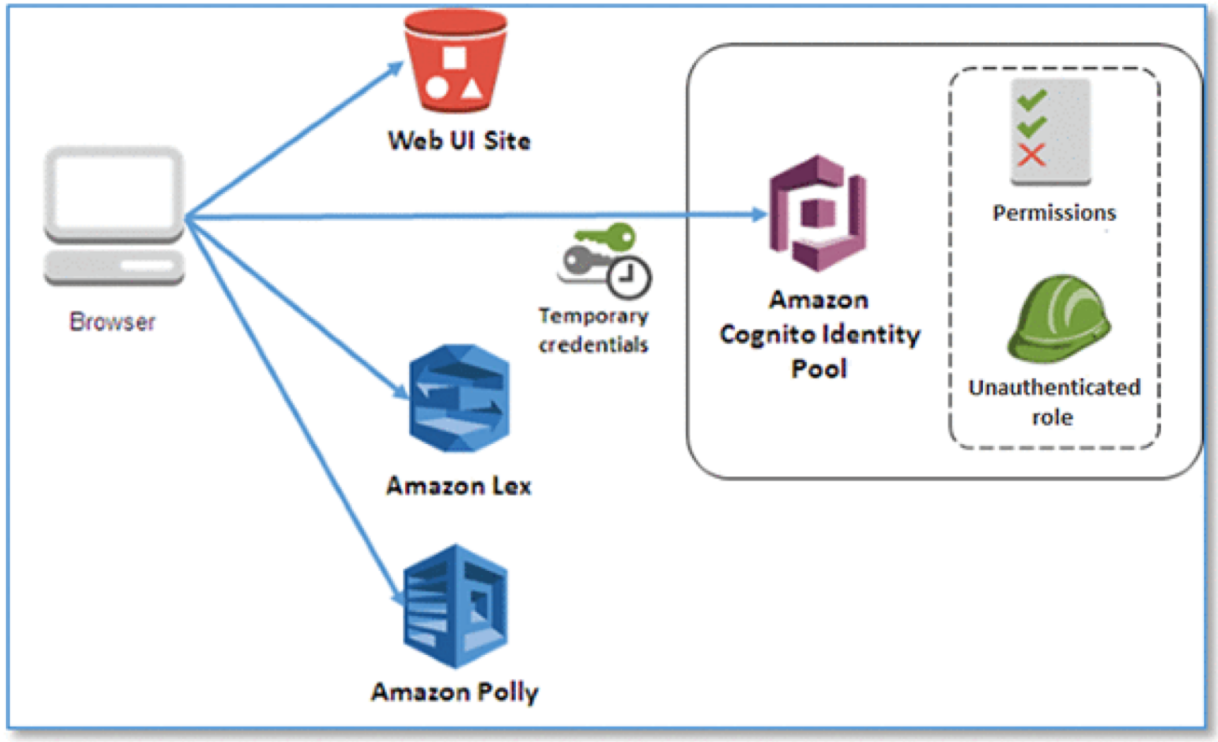




A preconfigured template is available here.

Demo - voice and textual interfaces





amazon alexa

Part 3 - Summary

- We are able to:
  - Understand the intent and all required slots
  - Validate the provided slots
  - Provide final message (SKM url)
  - Handle errors
- BUT
  - We do not cope with other types of train tickets (e.g. seasons tickets)
  - We do not have a mobile interface
  - …

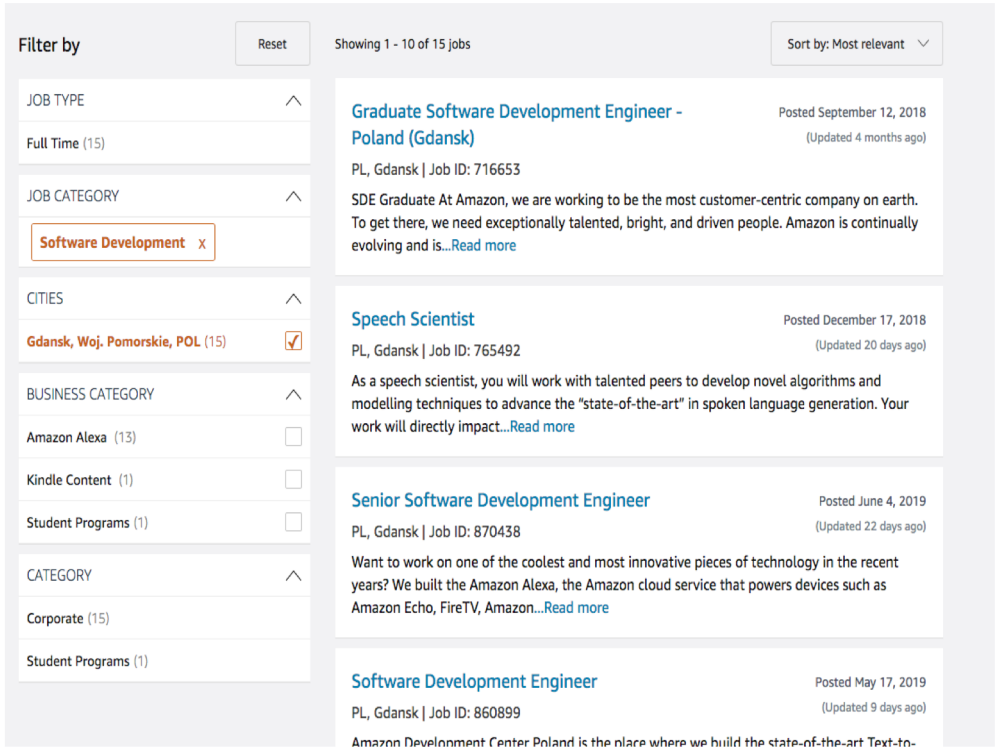
BUT for today – the plan we had for this workshop has been implemented ☺

Amazon Development Center in Gdańsk
We are hiring!

Currently, there are 15 positions open on Software Development

- Software Development Engineers
- Technical Program Managers

- Check this page [Amazon Hiring page]



amazon alexa

Thank you for your attention