

Source: C# Corner (www.c-sharpcorner.com)[PRINT](#)

Article



Build Travel Agent Application Using LUIS API In Microsoft Cognitive Services - Part Two

By [Abdul Rasheed Feroz Khan](#) on Sep 25 2016

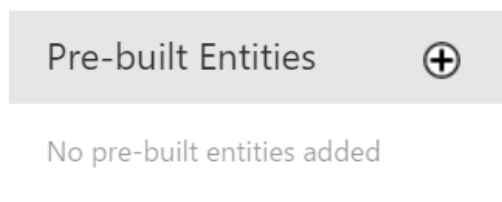
Introduction

This article is a continuation of the previous one, [Build Travel Agent Application Using LUIS API In Microsoft Cognitive Services - Part One](#), in which I was working with building a travel agent application using LUIS API. Here, I will be working with Bing entities and seeding the system with training and hosting them with a URL.

Note - Please make sure that you have worked on my previous article which is a starter one for this demo that we are going to work out.

Step 7 - By the previous demo, we have completed up to the steps of adding a child entity. Now, we have to handle this application with some request, here we will be using Bing entities. Let's say, for an example - "Book me a flight to Boston on May 4". This has to understand the date words like name of the month which is May over here. Now, this can be represented with the help of pre-built entity model called datetime.

You can find the pre-built entities option at the left pane of your LUIS API. Click on the "+" sign next to it.



Select "datetime" under pre-built entities and click on "OK".

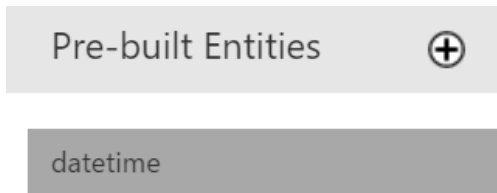
Pre-built entities



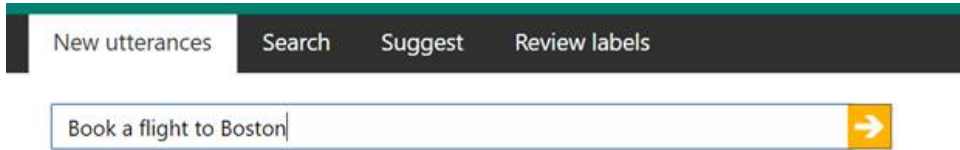
Which Bing entity do you want to add?

**OK**

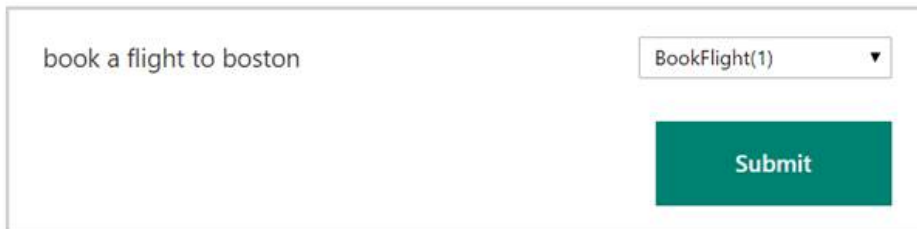
Now, you can find the "datetime" pre-built entity has been added at the left pane.



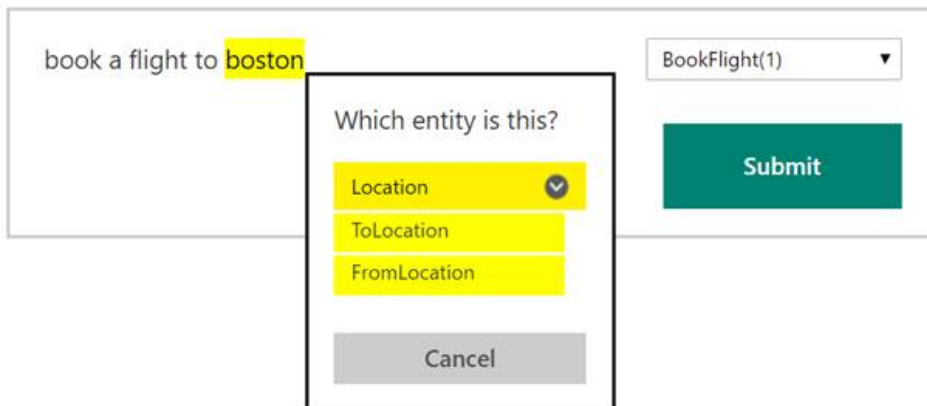
Step 8 - When the pre-built entities are added, we can step on for the labeling process in which we will be seeding the system. Click on the “new utterance” text block and add the new utterance as “Book a flight to London” followed by clicking on enter key.



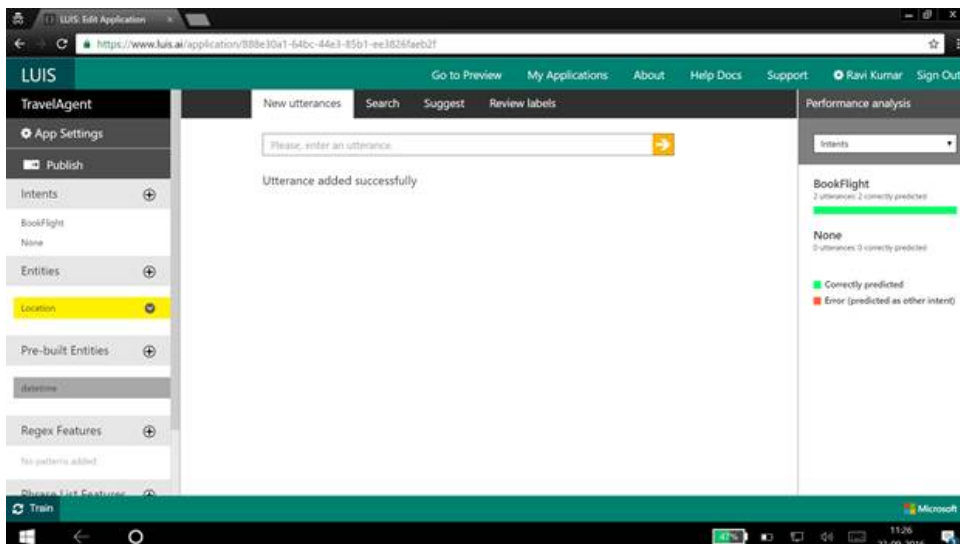
You will be getting the below screen now which we will be using it for labelling.



Now, select the location “Boston” and it will be identified by a yellow label. Select location followed by Fromlocation or Tolocation. Here, I will be selecting “tolocation” and clicking on Submit.

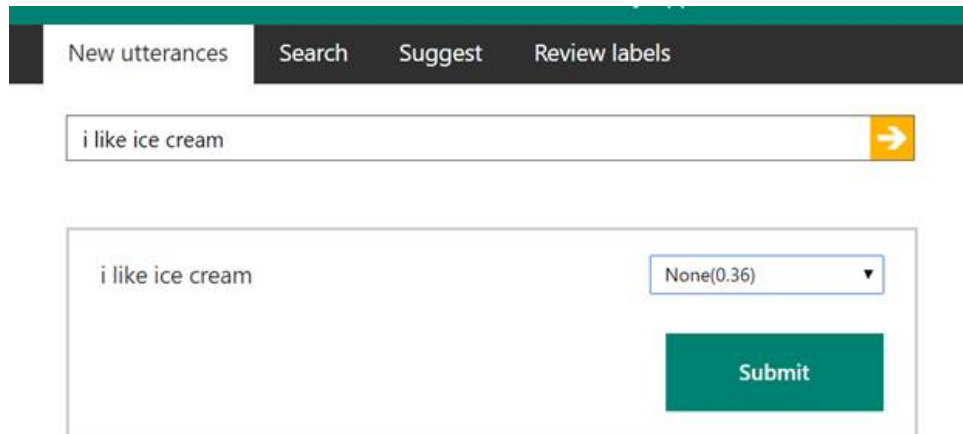


This will let us know that the utterance has been added successfully.



Step 8 - Further, this system has to be seeded with some examples of each entity and intent where it will be a dummy one with none. So, let's add some another utterance as "I like ice cream".

Click on the new utterance text block and add the utterance "I like ice cream".



Select "None" and click on Submit button.

Again, this utterance is also added successfully.

Step 9 - Now, we will be working on Training the system. When you train a model, LUIS identifies the relevant intents and entity which can be used in future. Training a system can be done as simply as by clicking on the "Train" button at the lower left pane (at the end of your window, before start menu).

Click on "train".

This results with the status that the training has been completed with the respective time.

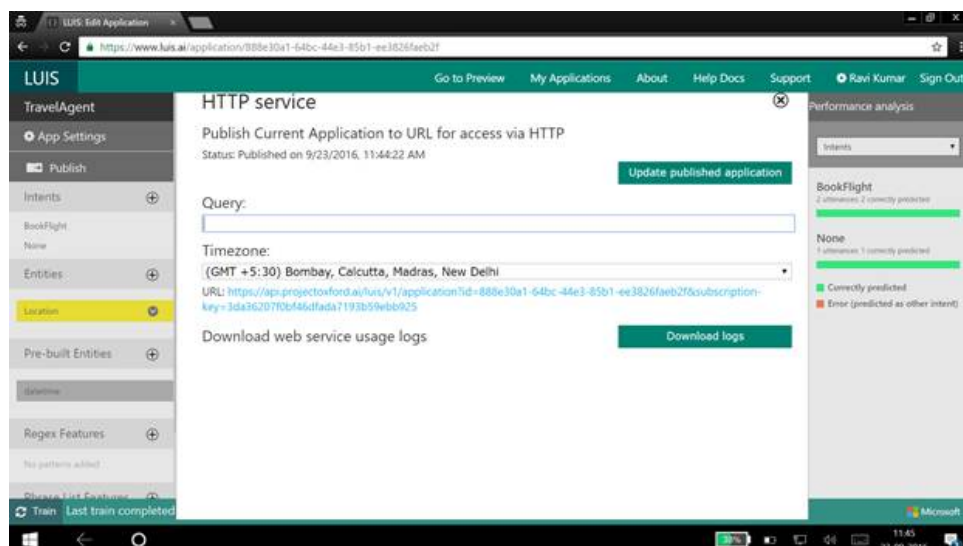
Step 10 - Next, we have to work on publishing this application with an HTTP endpoint.

Click the "Publish" button at the top left pane.

Click on "Publish Web Service" now, this will help us to publish the application with an HTTP endpoint.

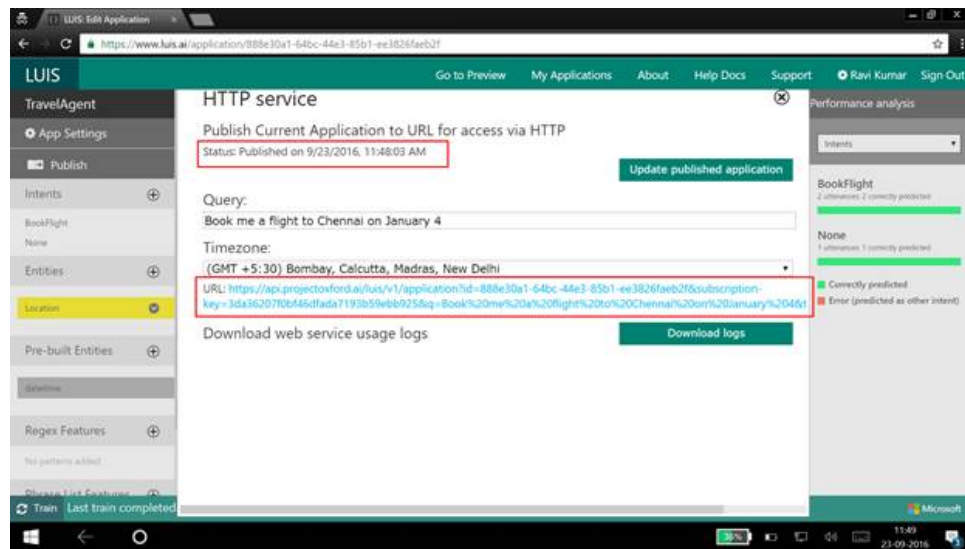
Now let's select the IST zone.

You will be getting a screen now as below.



Step 11 - Enter the query as "Book me a flight to Chennai on January 4" and click on Update Published application.

When the query is updated, you can find the URL has been changed with the query given and the status of the application published also with the time above.



URL- <https://api.projectoxford.ai/luis/v1/application?id=888e30a1-64bc-44e3-85b1-ee3826faeb2f&subscription-key=3da36207f0bf46dfada7193b59ebb925&q=Book%20me%20a%20flight%20to%20Chennai%20on%20January%204&timezoneOffset=5.5>

Step 12 - Click on the URL now. You will be getting a browser page with JSON code, as shown below.



```
{
  "query": "Book me a flight to Chennai on January 4",
  "intents": [
    {
      "intent": "BookFlight",
      "score": 0.993438661
    },
    {
      "intent": "None",
      "score": 0.403279841
    }
  ],
  "entities": [
    {
      "entity": "january 4",
      "type": "builtin.datetime.date",
      "startIndex": 31,
      "endIndex": 39,
      "resolution": {
        "date": "XXXX-01-04"
      }
    }
  ]
}
```

JSON explanation

- The field Query is filled with "Book me a flight to Chennai on January 4" which holds the JSON type of string.
- The field intent over here is "BookFlight, None" with a JSON type of object.
- The field score over here is "0.993438661, 0. 403279841" which is a confidence score between 0 and 1 followed by the JSON type of "number".
- Entities over here is January 4 and we don't have Chennai over here as entity as I didn't add that particular utterance before. In case my query is "Book me a flight to Boston on January 4" than I will be receiving the entity over here as London too. You can surf the below code:

The screenshot displays the LUIS application interface. The main content area shows the 'HTTP service' configuration, including a query input field with the text 'Book me a flight to Boston on January 4', a timezone dropdown menu set to '(GMT +5:30) Bombay, Calcutta, Madras, New Delhi', and a URL for the application. A 'Download logs' button is visible. On the right side, the 'Performance analysis' panel shows the results of the training process, listing two intents: 'BookFlight' (2 utterances, 3 correctly predicted) and 'None' (7 utterances, 1 correctly predicted). The bottom status bar indicates 'Train: Last train completed'.



URL - <https://api.projectoxford.ai/luis/v1/application?id=888e30a1-64bc-44e3-85b1-ee3826faeb2f&subscription-key=3da36207f0bf46dfada7193b59ebb925&q=Book%20me%20a%20flight%20to%20Boston%20on%20January%204&timezoneOffset=5.5>

JSON Code

```

1. {
2.   "query": "Book me a flight to Boston on January 4",
3.   "intents": [
4.     {
5.       "intent": "BookFlight",
6.       "score": 0.9996851
7.     },
8.     {
9.       "intent": "None",
10.      "score": 0.3422906
11.    }
12.  ],
13.  "entities": [
14.    {
15.      "entity": "boston",
16.      "type": "Location::ToLocation",
17.      "startIndex": 20,
18.      "endIndex": 25,
19.      "score": 0.876996458
20.    },
21.    {
22.      "entity": "january 4",
23.      "type": "builtin.datetime.date",
24.      "startIndex": 30,
25.      "endIndex": 38,
26.      "resolution": {
27.        "date": "XXXX-01-04"
28.      }
29.    }
30.  ]
31. }

```

Well!! Now, we have built a basic LUIS Application. Read my next articles to work with the advanced features of LUIS.

Thank you for using C# Corner