Quickstart: Get intent with REST APIs

05/18/2020 • 25 minutes to read •

Choose a programming language

C#	Go	Java	JavaScript	Python

In this article

Prerequisites
Create Pizza app
Train and publish the Pizza app
Record the app ID, prediction key, and prediction endpoint of your Pizza app
Create the Node.js project
Get intent programmatically
Clean up resources
Next steps

In this quickstart, you will use a LUIS app to determine a user's intention from conversational text. Send the user's intention as text to the Pizza app's HTTP prediction endpoint. At the endpoint, LUIS applies the Pizza app's model to analyze the natural language text for meaning, determining overall intent and extracting data relevant to the app's subject domain.

For this article, you need a free LUIS account.

Reference documentation | Sample

Prerequisites

- Node.js programming language
- Visual Studio Code

Create Pizza app

1. Select pizza-app-for-luis-v6.json to bring up the GitHub page for the `pizza-app-for-luis.json` file.

2. Right-click or long tap the **Raw** button and select **Save link as** to save the `pizza-app-for-luis.json` to your computer.
3. Sign into the LUIS portal.
4. Select My Apps.
5. On the **My Apps** page, select **+ New app for conversation**.
6. Select **Import as JSON**.
7. In the **Import new app** dialog, select the **Choose File** button.
8. Select the `pizza-app-for-luis.json` file you downloaded, then select **Open**.
9. In the **Import new app** dialog **Name** field, enter a name for your Pizza app, then select the **Done** button.

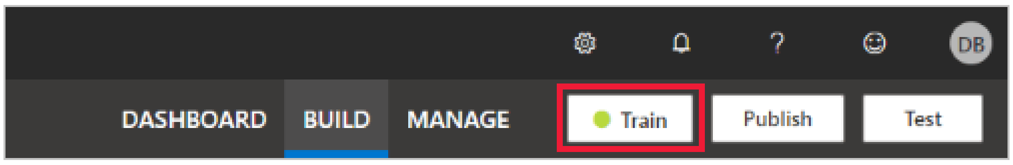
The app will be imported.

If you see the dialog **How to create an effective LUIS app**, close the dialog.

Train and publish the Pizza app

You should see the **Intents** page with a list of the intents in the Pizza app.

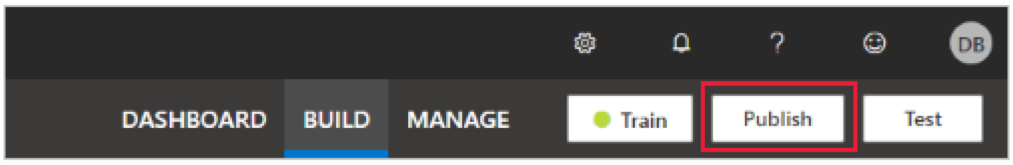
1. In the top-right side of the LUIS website, select the **Train** button.



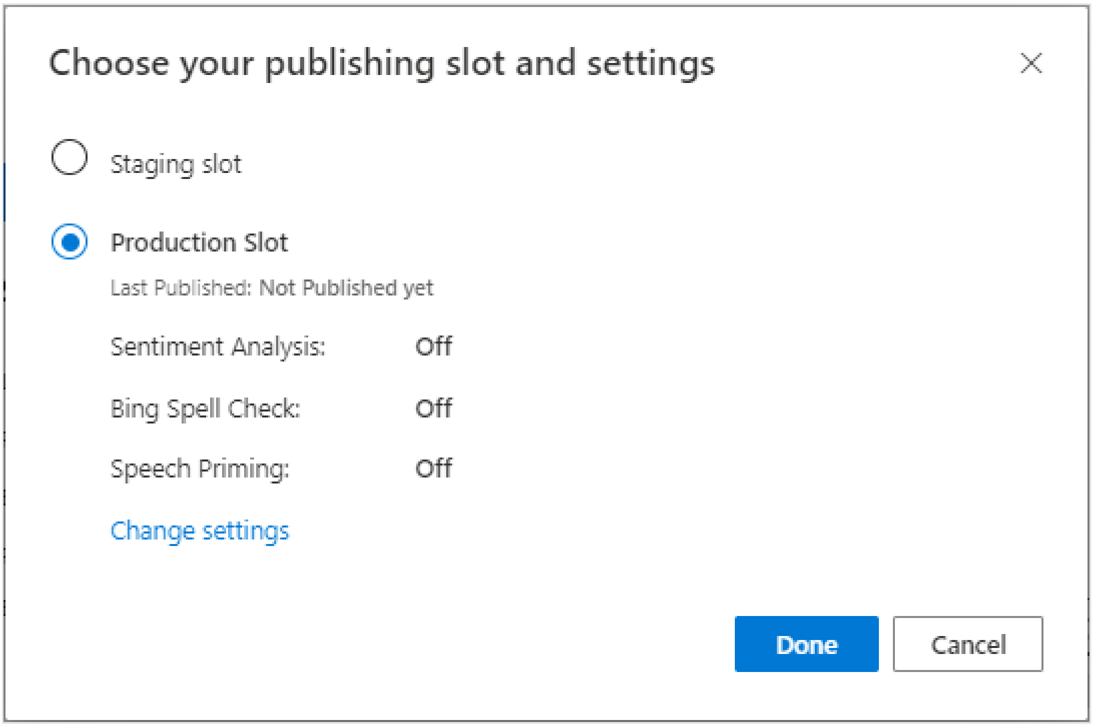
2. Training is complete when status indicator on the **Train** button is green.

In order to receive a LUIS prediction in a chat bot or other client application, you need to publish the app to the prediction endpoint.

1. Select **Publish** in the top-right navigation.



2. Select the **Production** slot, then select **Done**.


Choose your publishing slot and settings

Staging slot

[✓ Production Slot]

Last Published: Not Published yet

Sentiment Analysis:	Off
Bing Spell Check:	Off
Speech Priming:	Off

3. Select the **Access your endpoint URLs** link in the notification to go to the **Azure Resources** page. The endpoint URLs are listed as the **Example Query**.

Your Pizza app is now ready to use.

Record the app ID, prediction key, and prediction endpoint of your Pizza app

To use your new Pizza app, you will need the app ID, prediction key, and prediction endpoint of your Pizza app.

To find these values:

1. From the **Intents** page, select **MANAGE**.
2. From the **Application Settings** page, record the **App ID**.
3. Select **Azure Resources**.
4. From the **Azure Resources** page, record the **Primary Key**. This value is your prediction key.
5. Record the **Endpoint URL**. This value is your prediction endpoint.

Create the Node.js project

1. Create a new folder to hold your Node.js project, such as `node-predict-with-rest`.

2. Open a new Command Prompt, navigate to the folder you created and execute the following command:

Console Copy

```
npm init
```

Press Enter at each prompt to accept the default settings.

3. Install the dependencies by entering the following commands:

Console Copy

```
npm install --save request
npm install --save request-promise
npm install --save querystring
```

Get intent programmatically

Use Node.js to query the prediction endpoint and get a prediction result.

1. Copy the following code snippet to a file named `predict.js`:

JavaScript Copy

```javascript
//
// This quickstart shows how to predict the intent of an utterance by
using the LUIS REST APIs.
//

var requestPromise = require('request-promise');
var queryString = require('querystring');

// Analyze a string utterance.
getPrediction = async () => {

    //////////
    // Values to modify.

    // YOUR-APP-ID: The App ID GUID found on the www.luis.ai
Application Settings page.
    const LUIS_appId = "YOUR-APP-ID";

    // YOUR-PREDICTION-KEY: Your LUIS authoring key, 32 character
value.
    const LUIS_predictionKey = "YOUR-PREDICTION-KEY";

    // YOUR-PREDICTION-ENDPOINT: Replace this with your authoring key
endpoint.
    // For example, "https://westus.api.cognitive.microsoft.com/"
    const LUIS_endpoint = "https://YOUR-PREDICTION-ENDPOINT/";
    // The utterance you want to use.
    const utterance = "I want two large pepperoni pizzas on thin crust
please";
    //////////

    // Create query string
    const queryParams = {
        "show-all-intents": true,
        "verbose":  true,
        "query": utterance,
        "subscription-key": LUIS_predictionKey
    }

    // Create the URI for the REST call.
    const URI =
`${LUIS_endpoint}luis/prediction/v3.0/apps/${LUIS_appId}/slots/producti
on/predict?${queryString.stringify(queryParams)}`

    // Send the REST call.
    const response = await requestPromise(URI);

    // Display the response from the REST call.
    console.log(response);
}

// Pass an utterance to the sample LUIS app
getPrediction().then(()=>console.log("done")).catch((err)=>console.log(
err));
```

2. Replace the values starting with `YOUR-` with your own values.

Information	Purpose
`YOUR-APP-ID`	Your app ID. Located on the LUIS portal, Application Settings page for your app.
`YOUR-PREDICTION-KEY`	Your 32 character prediction key. Located on the LUIS portal, Azure Resources page for your app.
`YOUR-PREDICTION-ENDPOINT`	Your prediction URL endpoint. Located on the LUIS portal, Azure Resources page for your app. For example, `https://westus.api.cognitive.microsoft.com/`.

3. Review the prediction response, which is returned as JSON:

JSON Copy

```
{"query":"I want two large pepperoni pizzas on thin crust
please","prediction":{"topIntent":"ModifyOrder","intents":
{"ModifyOrder":{"score":1.0},"None":{"score":8.55E-09},"Greetings":
{"score":1.82222226E-09},"CancelOrder":{"score":1.47272727E-
09},"Confirmation":{"score":9.8125E-10}},"entities":{"Order":
[{"FullPizzaWithModifiers":[{"PizzaType":["pepperoni pizzas"],"Size":
[["Large"]],"Quantity":[2],"Crust":[["Thin"]],"$instance":{"PizzaType":
[{"type":"PizzaType","text":"pepperoni
pizzas","startIndex":17,"length":16,"score":0.9978157,"modelTypeId":1,"
modelType":"Entity Extractor","recognitionSources":["model"]}],"Size":
[{"type":"SizeList","text":"large","startIndex":11,"length":5,"score":0
.9984481,"modelTypeId":1,"modelType":"Entity
Extractor","recognitionSources":["model"]}],"Quantity":
[{"type":"builtin.number","text":"two","startIndex":7,"length":3,"score
":0.999770939,"modelTypeId":1,"modelType":"Entity
Extractor","recognitionSources":["model"]}],"Crust":
[{"type":"CrustList","text":"thin
crust","startIndex":37,"length":10,"score":0.933985531,"modelTypeId":1,
"modelType":"Entity Extractor","recognitionSources":
["model"]}]}}],"$instance":{"FullPizzaWithModifiers":
[{"type":"FullPizzaWithModifiers","text":"two large pepperoni pizzas on
thin
crust","startIndex":7,"length":40,"score":0.90681237,"modelTypeId":1,"m
odelType":"Entity Extractor","recognitionSources":
["model"]}]}}],"ToppingList":[["Pepperoni"]],"$instance":{"Order":
[{"type":"Order","text":"two large pepperoni pizzas on thin
crust","startIndex":7,"length":40,"score":0.9047088,"modelTypeId":1,"mo
delType":"Entity Extractor","recognitionSources":
["model"]}],"ToppingList":
[{"type":"ToppingList","text":"pepperoni","startIndex":17,"length":9,"m
odelTypeId":5,"modelType":"List Entity Extractor","recognitionSources":
["model"]}]}}}}
    ```
```


```
The JSON response formatted for readability:

```JSON
{
  "query": "I want two large pepperoni pizzas on thin crust please",
  "prediction": {
    "topIntent": "ModifyOrder",
    "intents": {
      "ModifyOrder": {
        "score": 1
      },
      "None": {
        "score": 8.55e-9
      },
      "Greetings": {
        "score": 1.82222226e-9
      },
      "CancelOrder": {
        "score": 1.47272727e-9
      },
      "Confirmation": {
        "score": 9.8125e-10
      }
    },
    "entities": {
```

```
      "Order": [
        {
          "FullPizzaWithModifiers": [
            {
              "PizzaType": [
                "pepperoni pizzas"
              ],
              "Size": [
                [
                  "Large"
                ]
              ],
              "Quantity": [
                2
              ],
              "Crust": [
                [
                  "Thin"
                ]
              ],
              "$instance": {
                "PizzaType": [
                  {
                    "type": "PizzaType",
                    "text": "pepperoni pizzas",
                    "startIndex": 17,
                    "length": 16,
                    "score": 0.9978157,
                    "modelTypeId": 1,
                    "modelType": "Entity Extractor",
                    "recognitionSources": [
                      "model"
                    ]
                  }
                ],
                "Size": [
                  {
                    "type": "SizeList",
                    "text": "large",
                    "startIndex": 11,
                    "length": 5,
                    "score": 0.9984481,
                    "modelTypeId": 1,
                    "modelType": "Entity Extractor",
                    "recognitionSources": [
                      "model"
                    ]
                  }
                ],
                "Quantity": [
                  {
                    "type": "builtin.number",
                    "text": "two",
                    "startIndex": 7,
                    "length": 3,
                    "score": 0.999770939,
                    "modelTypeId": 1,
                    "modelType": "Entity Extractor",
                    "recognitionSources": [
                      "model"
                    ]
                  }
                ],
                "Crust": [
                  {
                    "type": "CrustList",
                    "text": "thin crust",
                    "startIndex": 37,
                    "length": 10,
                    "score": 0.933985531,
                    "modelTypeId": 1,
                    "modelType": "Entity Extractor",
                    "recognitionSources": [
                      "model"
                    ]
                  }
                ]
              }
            }
          ],
          "$instance": {
            "FullPizzaWithModifiers": [
              {
                "type": "FullPizzaWithModifiers",
                "text": "two large pepperoni pizzas on thin crust",
                "startIndex": 7,
                "length": 40,
                "score": 0.90681237,
                "modelTypeId": 1,
                "modelType": "Entity Extractor",
                "recognitionSources": [
                  "model"
                ]
              }
            ]
          }
        }
      ],
      "ToppingList": [
        [
          "Pepperoni"
        ]
      ],
      "$instance": {
        "Order": [
          {
            "type": "Order",
            "text": "two large pepperoni pizzas on thin crust",
            "startIndex": 7,
            "length": 40,
            "score": 0.9047088,
            "modelTypeId": 1,
            "modelType": "Entity Extractor",
            "recognitionSources": [
              "model"
            ]
          }
        ],
        "ToppingList": [
          {
            "type": "ToppingList",
            "text": "pepperoni",
            "startIndex": 17,
            "length": 9,
            "modelTypeId": 5,
            "modelType": "List Entity Extractor",
            "recognitionSources": [
              "model"
            ]
          }
        ]
      }
    }
  }
}
```

Clean up resources

When you are finished with this quickstart, delete the project folder from the file system.

Next steps

Add utterances and train

Is this page helpful?

Yes No