

Project – Alexa Integrated with Azure

Description – The project is about creating a Alexa Skill to get the ingredients for a recipe. This was achieved by creating the required Skill in Amazon ecosystem and then having the Alexa App connect to Azure services. The Azure Ecosystem is used to host the business logic as well as the data store. The Business Logic was to be able to parse the Intent and get the results and then send them back to Alexa for speech generation. The Skill that I developed was to get the ingredients for a set of recipes.

Hardware/OS – Window 7/Windows 10, Visual Studio 2017.

Technology – Alexa, JSON, Azure, C#(for creating the logic for parsing the intent and then responding back with result to Alexa)

Overview

- 1) Create the Backend API in Visual Studio with C# for the Intent parsing and response.
- 2) Publish the project in Azure AD
- 3) Create the Skill in Alexa on amazon website at <http://Developer.amazon.com>
- 4) Map the correct intent and slots to code in the Backend API.
- 5) Test the App in Alexa test Simulator and check the results.

Challenges – One of the challenges that I ran into is getting the Alexa request to seamless integrate to the Azure with authentication.

Benefits – Using Native Alexa configuration allows to create the Skills on the fly with rich feature set. We can then use Azure given that Mckesson has a subscription and store all the necessary back end data in azure as well as be able to use Single sign on functionality. This model in turn can be used to connect to any of the backend systems to fetch the data based on keywords. The database or data store can be residing in any repository other than Amazon echo system which can help integrate some legacy systems.

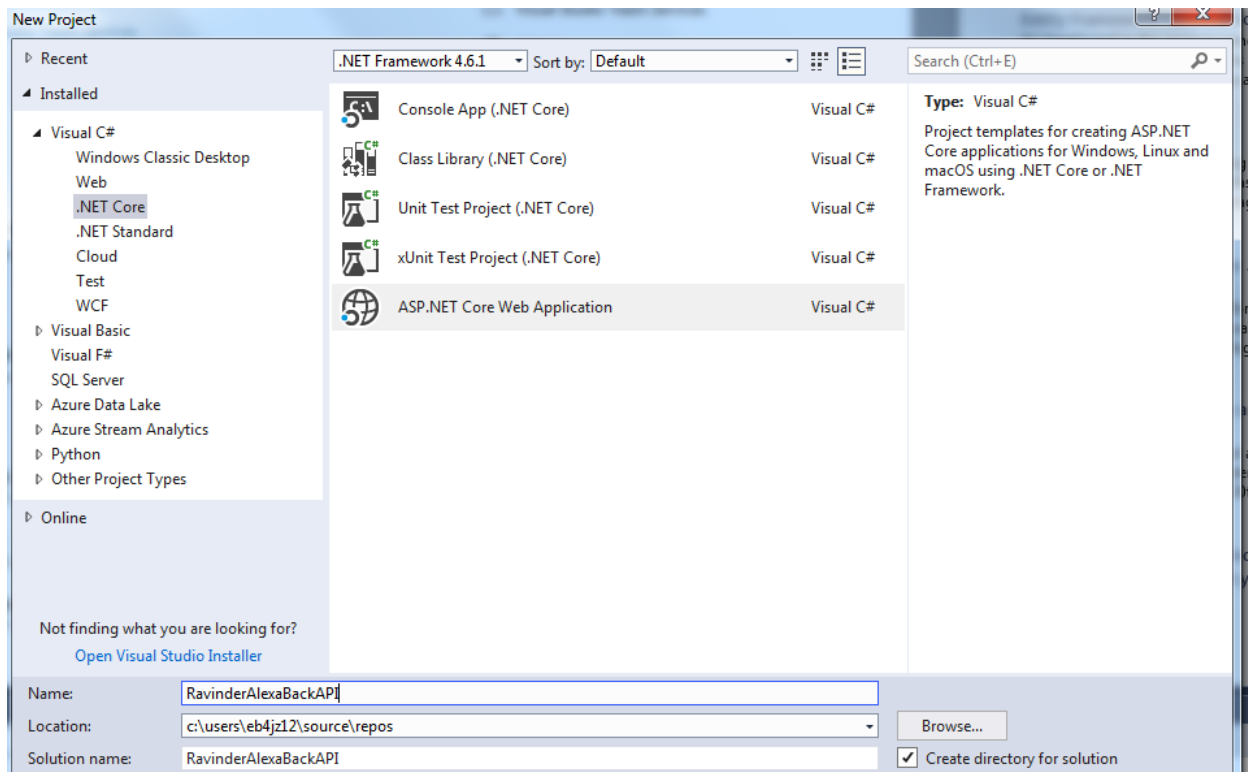
Summary – I was successfully able to integrate the New skill on Alexa to the backend in Azure. This can be enhanced to do full authorization and then securely fetch the data from a in house database. My goal to continue this further and create a rich feature set and give new Skill or APP in Alexa suited to the Pharmacy environment while leveraging the data that is housed securely in Mckesson /Azure within permissible constraints.

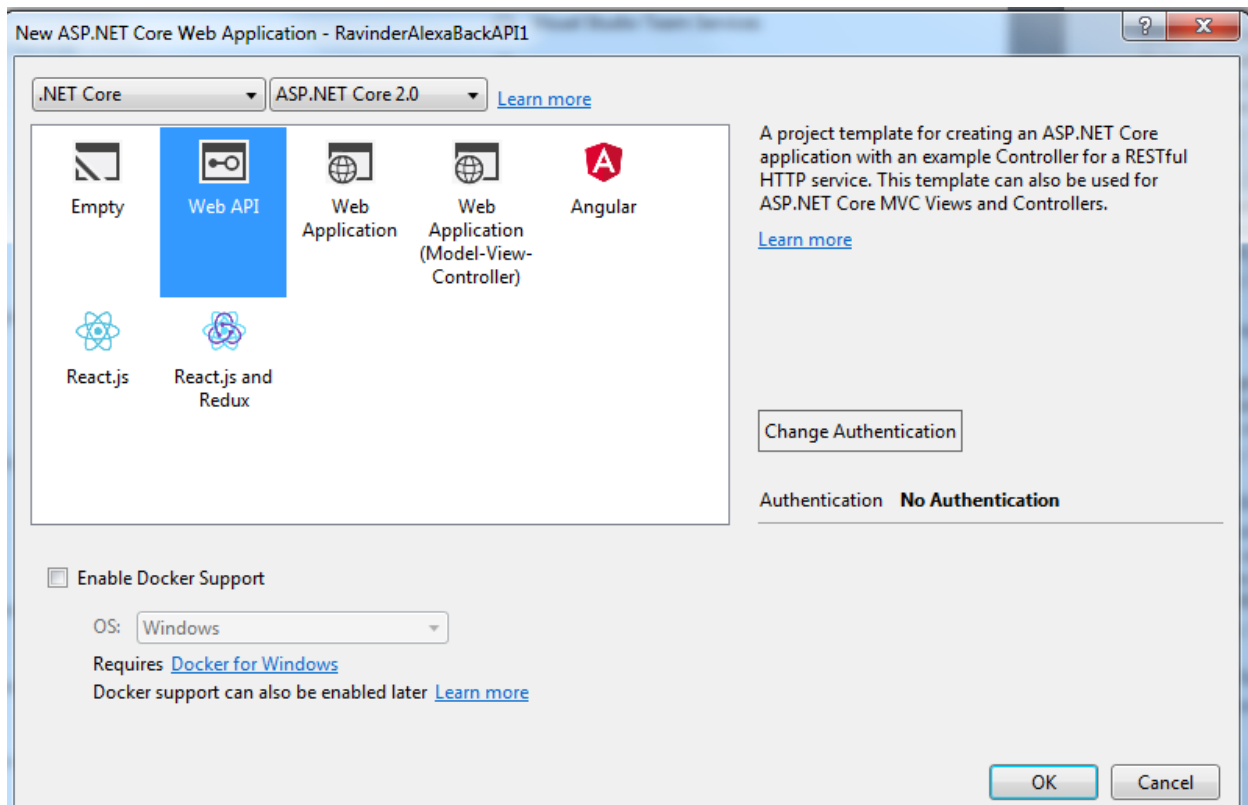
YouTube URL –

GitHub URL –

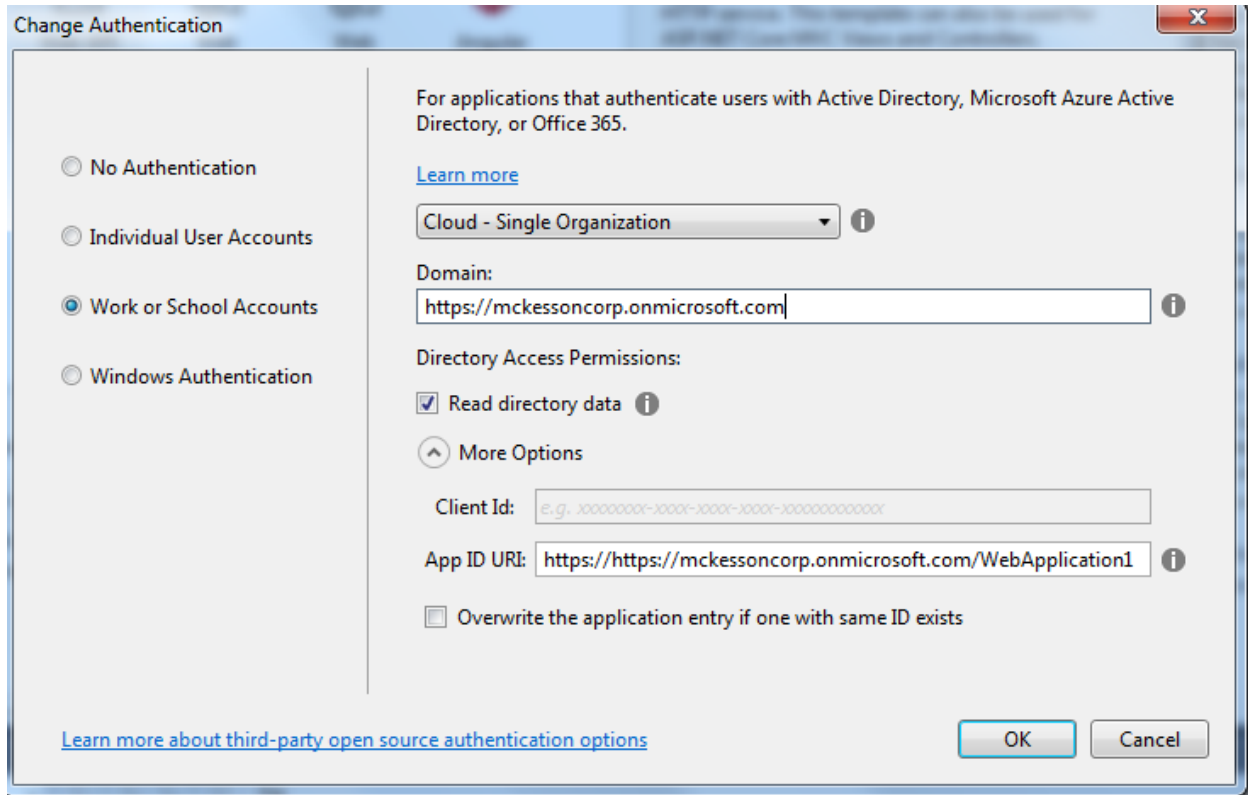
Project Description – I had to create a skill in Alexa and be able to configure it talk to azure where the code to parse the intent lies and then return the proper response which can be read by Alexa as a speech.

1) Creating the ASP.net Core 2.0 Web API application

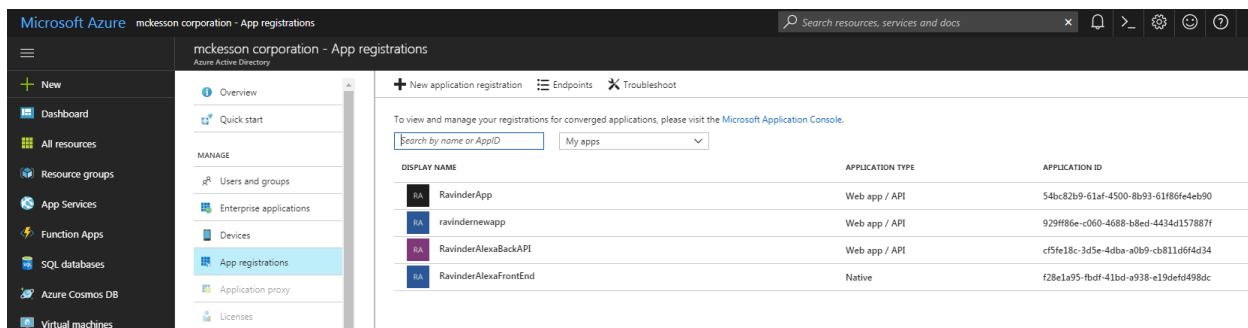




- 2) We will need to change the client authentication and also check the "Read directory Data"



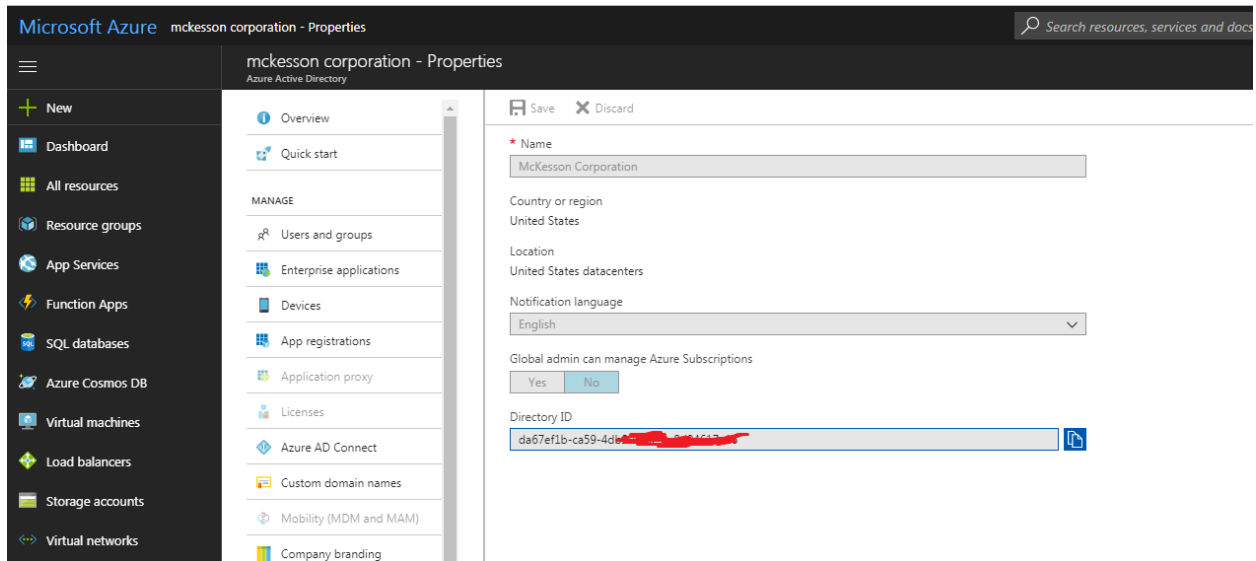
3) At this point the VS will create the project. This then gets deployed to the Azure.



- At this point we will need to check the appsetting.json for the Domain, TenantID and clientID. The clientID will need to be updated with the proper App URI. For the tenantID we need to go to Azure AD resource and then the Properties.
- At this point we need to configure the method to handle the request coming from alexa. For that we will need a package called Alexa.net. We get this package from the NuGet Package manager for the project. This gives the method and properties that are needed to process the request and send the response back to Alexa for speech.

The screenshot displays the Visual Studio interface. The top pane shows the NuGet package manager with the 'Installed' tab selected. It lists several packages: **Alexa.NET** by Tim Heuer (v1.4.2), **Microsoft.AspNetCore.All** by Microsoft (v2.0.3 and v2.0.5), and **Microsoft.NETCore.App** by Microsoft (v2.0.0). The bottom pane shows the `appsettings.json` file with the following schema: `http://json.schemastore.org/appsettings`. The JSON content is as follows:

```
1  {
2    "AzureAd": {
3      "Instance": "https://login.microsoftonline.com/",
4      "Domain": "mckessoncorp.onmicrosoft.com",
5      "TenantId": "da67e18c-3d5e-4dba-a0b9-cb811d6f4d34",
6      "ClientId": "https://mckessoncorp.onmicrosoft.com/RavinderAlexaBackAPI"
7      // "ClientId": "cf5fe18c-3d5e-4dba-a0b9-cb811d6f4d34"
8    },
9    "Logging": {
10     "IncludeScopes": false,
11     "Debug": {
12       "LogLevel": {
13         "Default": "Warning"
14       }
15     },
16     "Console": {
17       "LogLevel": {
18         "Default": "Warning"
19       }
20     }
21   }
22 }
23
```



- 6) At this point we need to handle the request coming in which comes as HTTP Post method. We need to configure and modify the post method in Values controller class. Here we override the post method to as follows.

```

Browse - HTTP 404 Not Found | Policy.html | ValuesController.cs
RavinderAlexaBackAPI | RavinderAlexaBackAPI.Controllers.ValuesController | Post(SkillRequest request)

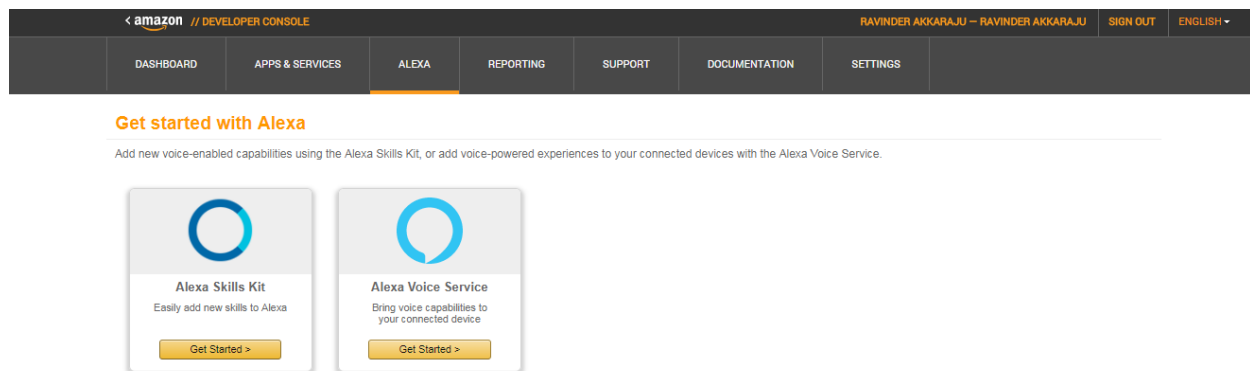
14
15 namespace RavinderAlexaBackAPI.Controllers
16 {
17     // [Authorize]
18     [Route("api/[controller]")]
19     public class ValuesController : Controller
20     {
21
22         // POST api/values
23         // [Authorize]
24         [HttpPost]
25
26         // This is the method that is being invoked. This is the main method that will parse the intent and generate the result.
27         public SkillResponse Post([FromBody]SkillRequest request)
28         {
29
30             SkillResponse response = null;
31             if(request != null)
32             {
33                 PlainTextOutputSpeech outputSpeech = new PlainTextOutputSpeech();
34                 string Item = (request.Request as IntentRequest)?.Intent.Slots.FirstOrDefault(s => s.Key == "Item").Value.Value.ToString();
35                 if (Item.ToUpper() == "Bread".ToUpper() )
36                 { outputSpeech.Text = Item + " Ingredients are wheat flour , water"; }
37
38                 else if (Item.ToUpper() == "Cookie".ToUpper())
39                 {
40                     outputSpeech.Text = Item + " Ingredients are flour ,Sugar, Egg and vanilla";
41                 }
42                 else if (Item.ToUpper() == "Sandwich".ToUpper())
43                 {
44                     outputSpeech.Text = Item + " Ingredients are Bread, Lettuce, Tamato";
45                 }
46                 else if (Item.ToUpper() == "Biryani".ToUpper())
47                 {
48                     outputSpeech.Text = Item + " Ingredients are Rice, Vegetable, Spices";
49                 }
50                 else
51                 {
52                     outputSpeech.Text = " I am not an expert at cooking " + Item + " but i am learning. Will let you know in a few days ";
53                 }
54             }
55             response = ResponseBuilder.Tell(outputSpeech);
56
57             return response;
58         }
59     }
60 }
61
62
63
64

```

The main thing to keep in mind is the Key in the Lambda expression. That will match the slot in the JSON string which comes in when we create the intent in Alexa configuration later. Since I am looking for the ingredients I named it the item. I had mistyped it originally and kept getting an 500 internal error. I have kept the implementation simple but all the key/value pairs or the ingredients can be moved to either a JSON file, any other document or stored in the DB on the Azure.

- 7) Now we can compile and publish the code to Azure. At this juncture we can either publish it to the Azure, local IIS. This is also the place where we assign the app name and either assign to the resource group or create a new one.
- 8) At this point in case we need to authenticate as single sign on then we can create another class. I had trouble being able to integrate it this way so had to comment the code out. We need to comment out the [authorize] line in the values controller class.

- 9) Now at this point we need to create the skill in Alexa.



English (U.S.)

Add a New Language

Skill Information

Interaction Model

Configuration

SSL Certificate

Test

Publishing Information

Privacy & Compliance

Skills Beta Testing

Status Not yet eligible

Skill Type

Define a custom interaction model or use one of the predefined skill APIs. [Learn more](#)

Custom

Language

Language of your skill

English (U.S.)

Application Id

The ID for this skill

amzn1.ask.skill.987f4ce9-0dbc-4268-a3a2-65c159740c03

Name

Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.

Recipe Ingredients

Invocation Name

The name customers use to activate the skill. For example, "Alexa ask Tide Pooler..."

my recipe

For successful Alexa Skills Certification, please review and follow our [Invocation Name Guidelines](#) as well as our [Certification Requirements](#).

Global Fields

These fields apply to all languages supported by the skill.

Audio Player

Does this skill use the audio player directives?

Yes No

Learn more

Video App

Does this skill use the video app directives? [Learn more](#)

Yes No

Render Template

Does this skill use the Render Template directives? [Learn more](#)

Yes No

Amazon Apps & Services

Link an Alexa User with

Lambda Expressions (C#)

Delegates and Events in

PSV-13296 Modify L

Dashboard - Microsoft

biriyani - Google Search

Secure

https://developer.amazon.com/edw/home.html#/skill/amzn1.ask.skill.987f4ce9-0dbc-4268-a3a2-65c159740c03/en_US/intentSchema

Apps

Sharepoint

JIRA

McNet Home Page

Google Maps

VB.NET and C# Com

Documents

Basics

URLS

TeamNotebook - M

Imported

Lunch

CPS

USCIS - I-130, Petiti

U.S. Citizenship and

U.S. Citizenship and

USCIS - Green Card

Back to All Skills

Recipe Ingredients

Custom

ID: amzn1.ask.skill.987f4ce9-0dbc-4268-a3a2-65c159740c03

Getting Started

English (U.S.)

Add a New Language

Skill Information

Interaction Model

Configuration

SSL Certificate

Test

Publishing Information

Privacy & Compliance

Skills Beta Testing

Status Not yet eligible

Try the skill builder (beta), an intuitive interface for building your interaction model and creating dialog prompts

Launch Skill Builder

Intent Schema

The schema of user intents in JSON format. For more information, see [Intent Schema](#). Also see [built-in slots](#) and [built-in intents](#).

```

1 {
2   "intents": [
3     {
4       "slots": [
5         {
6           "name": "Item",
7           "type": "MyList"
8         }
9       ],
10      "intent": "RecipeIngredients"
11    }
12  ]
13}
```

Custom Slot Types (Optional)

Custom slot types to be referenced by the Intent Schema and Sample Utterances. For general information about custom slots, see [Custom Slot Types](#).

Type

Values

MyList

Bread | Cookie | Tortia | Biryani | Sandwich

Delete

Edit

Add Slot Type

Here we need to create a new List which is custom. In the MyList I create the various items like Bread, Cookie etc. which will be the key words which will be gleaned from the intent to be sent to the backend in azure to be processed. Any values in the list can be used to get the response back. This will be used as key's in the backend to send an appropriate response.

Interaction Model

Configuration

SSL Certificate

Test

Publishing Information

Privacy & Compliance

Skills Beta Testing NEW

Status Not yet eligible

These fields apply to all languages supported by the skill.

Endpoint

Service Endpoint Type: ☐ AWS Lambda ARN (Amazon Resource Name) ☒ HTTPS

Recommended
AWS Lambda is a server-less compute service that runs your code in response to events and automatically manages the underlying compute resources for you.
[More info about AWS Lambda](#)
[How to integrate AWS Lambda with Alexa](#)

Default

Provide geographical region endpoints? (Optional) ☐ Yes ☒ No

Account Linking

Do you allow users to create an account or link to an existing account with you? ☐ Yes ☒ No

[More info about Account Linking](#)
[Tips for successful Account Linking](#)

Permissions

Request users to access resources and capabilities
Please request permissions to resources and capabilities that are absolutely core to the customer experience delivered by the skill.
[Learn More](#)

☐ Device Address

☐ Full Address

☐ Country & Postal Code Only

☐ Lists Read

☐ Lists Write

The default URL needs to be set to the method which will handle the call in my case it is <https://ravinderalexabackapi.azurewebsites.net/api/values>.

Account linking is needed to be able to configure and authorize the user before being able to successfully use the Business logic stored in the Azure which in turn will access the data store.

[Back to All Skills](#)

?

Recipe Ingredients
Custom
ID: amzn1.ask.skill.98714ce9-0dbc-4268-a3a2-65c159740c03

English (U.S.) [Add a New Language](#)

Skill Information

Interaction Model

Configuration

SSL Certificate

Test

Publishing Information

Privacy & Compliance

Skills Beta Testing NEW
Status Not yet eligible

Global Fields

These fields apply to all languages supported by the skill.

To protect your security and the security of end users, we require that you use a certificate while developing an Alexa skill. For more information, see [Registering and Managing Alexa Skills - About SSL Options](#).

Certificate for DEFAULT Endpoint:

Please select one of the three methods below for the web service:

- ☐ My development endpoint has a certificate from a trusted certificate authority
- ☒ My development endpoint is a sub-domain of a domain that has a wildcard certificate from a certificate authority
- ☐ I will upload a self-signed certificate in X.509 format. [Learn how to create a self signed certificate.](#)

See [Certification Requirements](#) in our technical documentation as you develop your skills and prepare to submit to Amazon.

[Save](#) [Submit for certification](#) [Next](#)

English (U.S.) [Add a New Language](#)

Skill Information

Interaction Model

Configuration


SSL Certificate

Test

Publishing Information

Privacy & Compliance

Skills Beta Testing NEW
Status Not yet eligible



Try the testing simulator (beta), an intuitive interface for testing multi-turn dialogs and device renderings.
[Go to Test Simulator](#) BETA

Please complete the Interaction Model tab to start testing this skill.

Enabled This skill is enabled for testing on your account.

Once you have completed testing on your device, please complete the Description and Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

The skill is available in "Skills > Your Skills" page of the Alexa App when you select 'Yes' above. You can then enable the skill and test its functionality on your device by asking Alexa, **ask my recipe**

For successful Alexa Skills Certification, please test for our requirements on [Session Management](#) and [Error Handling](#).

Voice Simulator

Hear how Alexa will speak a response entered in plain text or SSML. [Learn more about supported SSML tags.](#)

For example: Here is a word spelled out: `<say-as interpret-as="spell-out">hello</say-as>`.

Listen

10

Ravinder V Akkaraju Project - Alexa

TESTING --

To test the app, enter the name of the ingredients. The response should be what we configured in the Azure Back end method.

Service Simulator

Use Service Simulator to test your HTTPS endpoint: <https://ravinderalexabackapi.azurewebsites.net/api/values>

Note: Service Simulator does not currently support testing audio player directives, dialog model, customer permissions and customer account linking. Text mode does not support launch intents and single interaction phrases.

Text

JSON

Enter Utterance

Bread Ingredients

Ask Recipe Ingredients

Reset

Service Request

```
1 {
2   "session": {
3     "new": true,
4     "sessionId": "SessionId.1988de57-ce40-4620-98a
5     "application": {
6       "applicationId": "amzn1.ask.skill.987f4ce9-6
7     },
8     "attributes": {},
9     "user": {
10      "userId": "amzn1.ask.account.AF04NZDNPYESPP
11    }
12  },
13  "request": {
14    "type": "IntentRequest",
15    "requestId": "EdwRequestId.e02e3e6b-c450-4719-
```

Service Response

```
1 {
2   "version": "1.0",
3   "response": {
4     "outputSpeech": {
5       "text": "bread Ingredients are wheat flour , w
6       "type": "PlainText"
7     },
8     "speechletResponse": {
9       "outputSpeech": {
10        "text": "bread Ingredients are wheat flour ,
11      },
12      "shouldEndSession": true
13    }
14  }
```

Listen

Another test for item cookie.

Service Simulator

Use Service Simulator to test your HTTPS endpoint: <https://ravinderalexabackapi.azurewebsites.net/api/values>

Note: Service Simulator does not currently support testing audio player directives, dialog model, customer permissions and customer account linking. Text mode does not support launch intents and single interaction phrases.

Text

JSON

Enter Utterance

Cookie Ingredients

Ask Recipe Ingredients

Reset

Service Request

Service Response

```
1 {
2   "session": {
3     "new": false,
4     "sessionId": "SessionId.1988de57-ce40-4620-98a1-111111111111",
5     "application": {
6       "applicationId": "amzn1.ask.skill.987f4ce9-0000-4000-9000-000000000000"
7     },
8     "attributes": {},
9     "user": {
10      "userId": "amzn1.ask.account.AF04NZDNPYESPPH000000000000"
11    }
12  },
13  "request": {
14    "type": "IntentRequest",
15    "requestId": "EdwRequestId.53b472ec-6c83-4d3e-9000-000000000000"
16  }
```

```
1 {
2   "version": "1.0",
3   "response": {
4     "outputSpeech": {
5       "text": "cookie Ingredients are flour ,Sugar ,butter ,eggs ,",
6       "type": "PlainText"
7     },
8     "speechletResponse": {
9       "outputSpeech": {
10        "text": "cookie Ingredients are flour ,Sugar ,butter ,eggs ,",
11        "type": "PlainText"
12      },
13      "shouldEndSession": true
14    }
15  }
```

Listen

Testing for the item which is not part of the list

Service Simulator

Use Service Simulator to test your HTTPS endpoint: <https://ravinderalexabackapi.azurewebsites.net/api/values>

Note: Service Simulator does not currently support testing audio player directives, dialog model, customer permissions and customer account linking. Text mode does not support launch intents and single interaction phrases.

Text

JSON

Enter Utterance

pizza Ingredients

Ask Recipe Ingredients

Reset

Service Request

Service Response

```
1 {
2   "session": {
3     "new": false,
4     "sessionId": "SessionId.1988de57-ce40-4620-98a
5     "application": {
6       "applicationId": "amzn1.ask.skill.987f4ce9-6
7     },
8     "attributes": {},
9     "user": {
10      "userId": "amzn1.ask.account.AF04NZDNPYESPP
11    }
12  },
13  "request": {
14    "type": "IntentRequest",
15    "requestId": "EdwRequestId.cd9b37af-222c-4ca1-
```

```
1
2: "1.0",
3: {
4: "Speech": {
5:   "text": " Could not find the ingredients for pizza",
6:   "type": "PlainText"
7: },
8: "letResponse": {
9:   "outputSpeech": {
10:    "text": " Could not find the ingredients for pizza
11:   },
12:   "shouldEndSession": true
13: },
14: }
```

Listen