Final Project
Alexa with Azure Integration
Ravinder Akkaraju

Project 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 recipe.

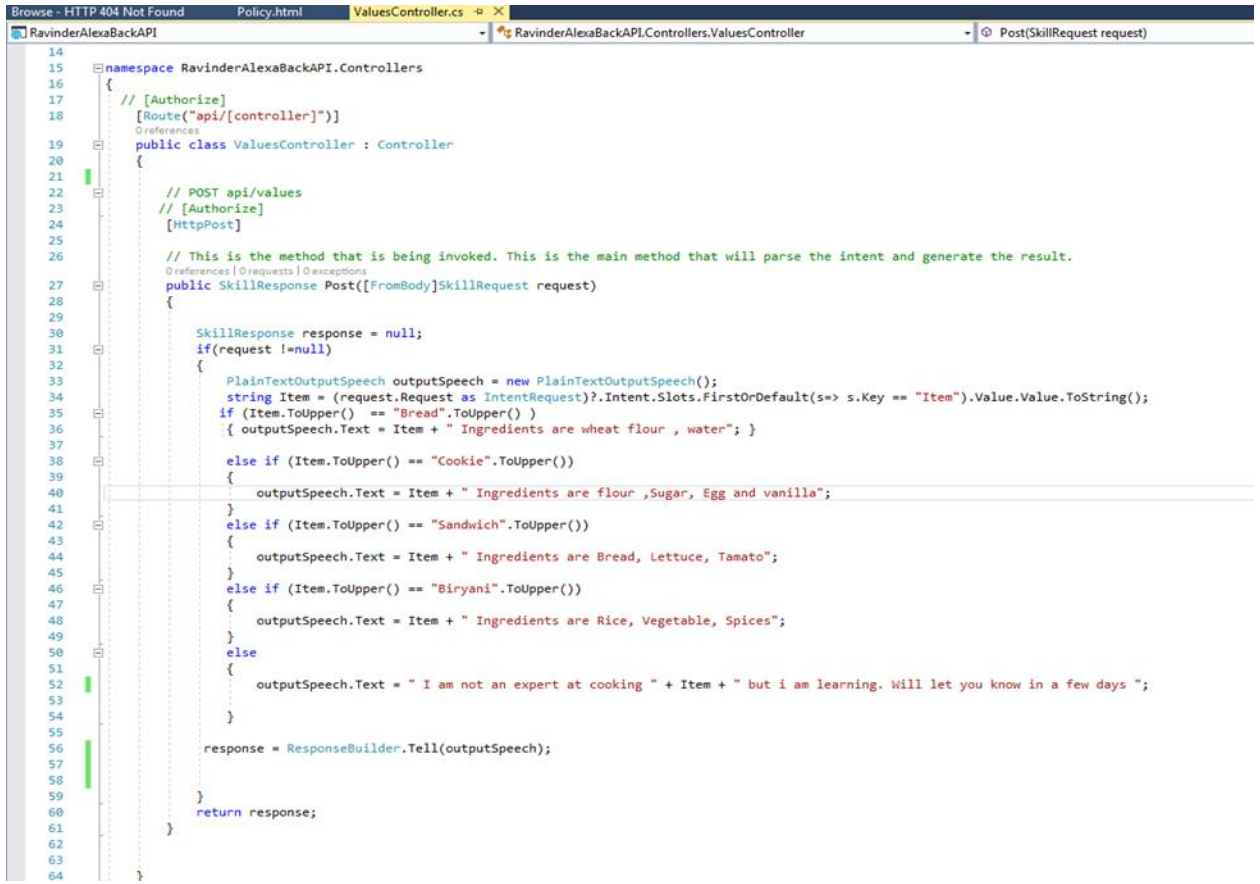
Technology & Hardware

- Window 7/Windows 10, Visual Studio 2017
- Alexa, JSON, Azure, C#(for creating the logic for parsing the intent and then responding back with result to Alexa)

Overview

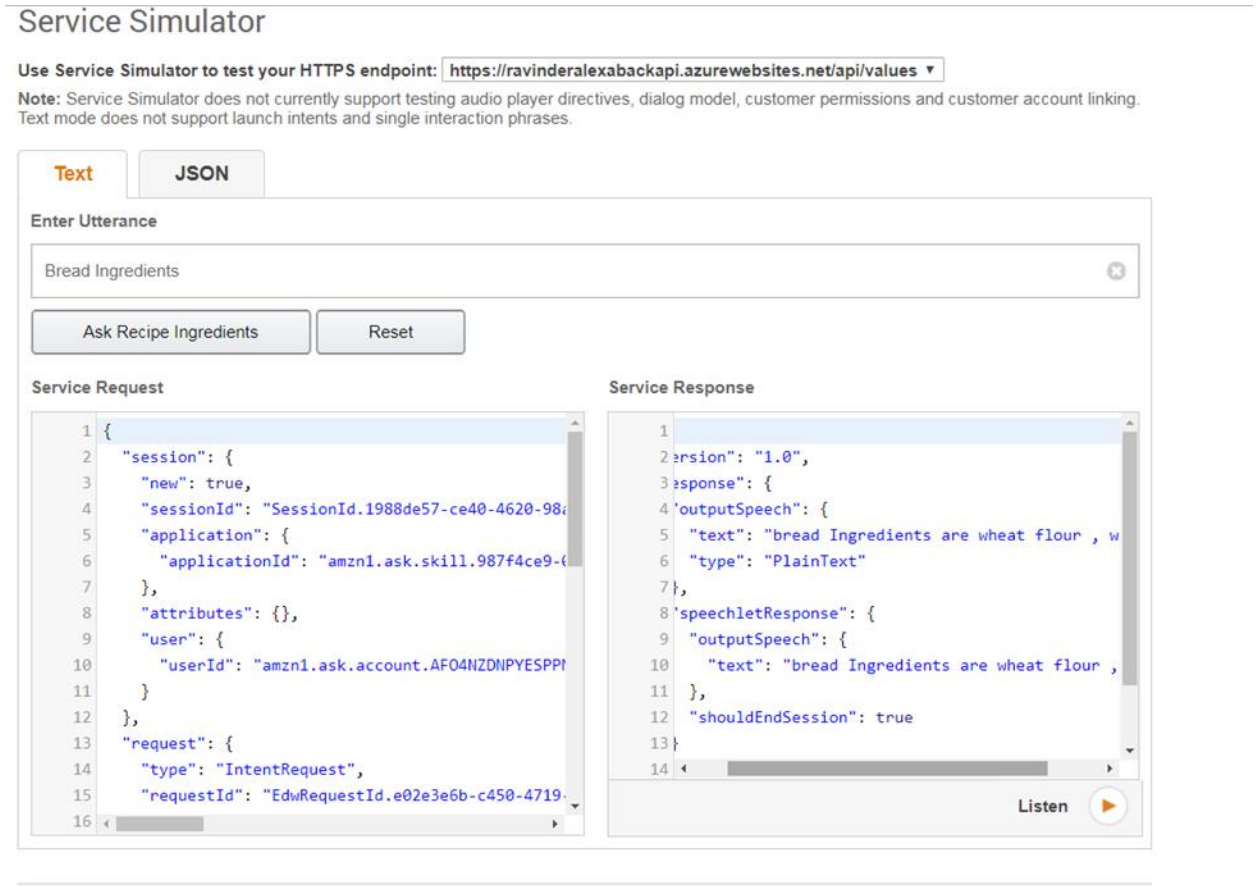
- Create the Backend API in Visual Studio with C# for the Intent parsing and response.
- Publish the project in Azure AD
- Create the Skill in Alexa on amazon website at https://developer.amazon.com
- Map the correct intent and slots to code in the Backend API.
- Test the App in Alexa test Simulator and check the results.

Code Snippet



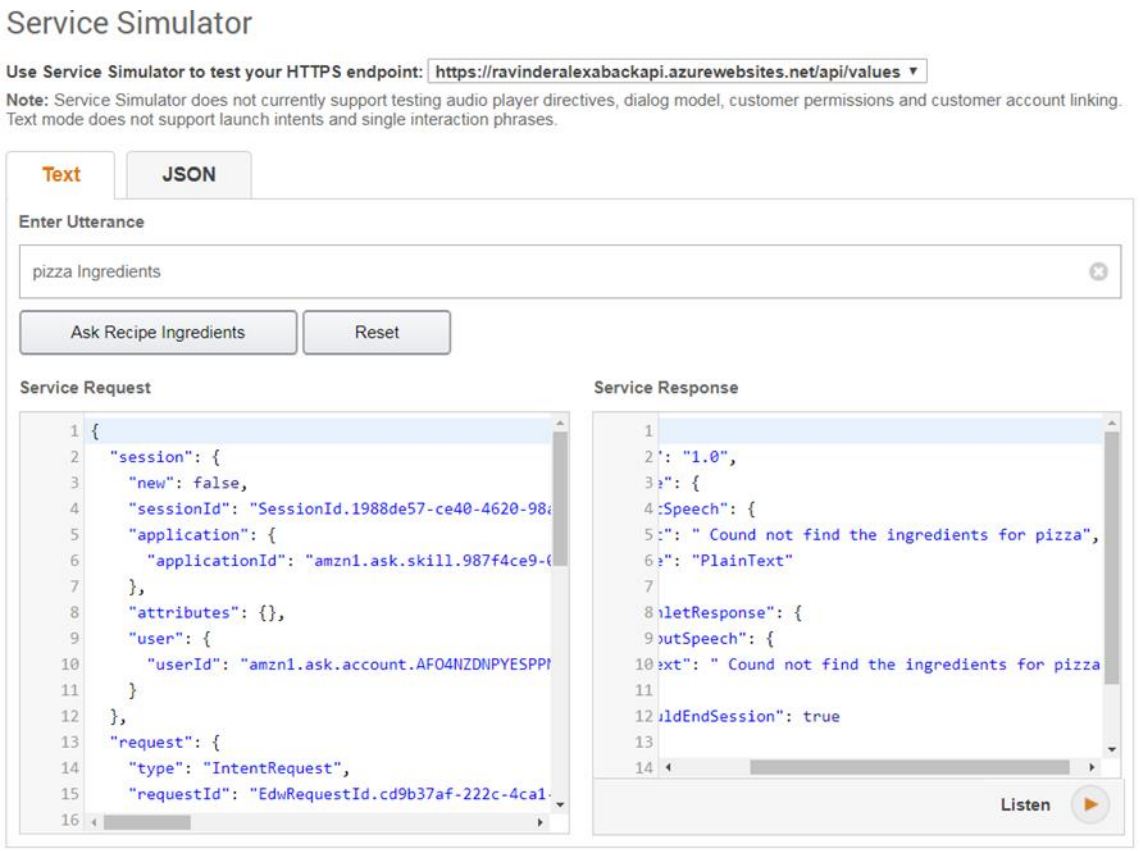

```
14
15  namespace RavinderAlexaBackAPI.Controllers
16  {
17    // [Authorize]
18      [Route("api/[controller]")]
        0 references
19      public class ValuesController : Controller
20      {
21
22          // POST api/values
23         // [Authorize]
24          [HttpPost]
25
26          // This is the method that is being invoked. This is the main method that will parse the intent and generate the result.
            0 references | 0 requests | 0 exceptions
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
56                 response = ResponseBuilder.Tell(outputSpeech);
57
58
59              }
60              return response;
61          }
62
63
64      }
```

Results


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
Service Response
{
"session": {
"new": true,
"sessionId": "SessionId.1988de57-ce40-4620-98
"application": {
"applicationId": "amzn1.ask.skill.987f4ce9-
},
"attributes": {},
"user": {
"userId": "amzn1.ask.account.AFO4NZDNPYESPP
}
},
"request": {
"type": "IntentRequest",
"requestId": "EdwRequestId.e02e3e6b-c450-4719-
ersion": "1.0",
esponse": {
outputSpeech": {
"text": "bread Ingredients are wheat flour , w
"type": "PlainText"
},
speechletResponse": {
"outputSpeech": {
"text": "bread Ingredients are wheat flour ,
},
"shouldEndSession": true
}
Listen

Results

- Negative Test



Challenges and Benefits

- **Benefits** – Using Native Alexa configuration allows to create the Skills on the fly with rich feature set. We can then use Azure given than 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 eco system which can help integrate some legacy systems.
- **Challenges** – One of the challenges that I ran into is getting the Alexa request to seamless integrate to the Azure with authentication.

Future Enhancement

- 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.

URLs

- GitHub URL
- YouTube URL