Final Project

Alexa

James Francis

Deep Azure@McKesson
Dr. Zoran B. Djordjević

Introduction

Problem Statement

Consumers are making a tectonic shift away from “Brick-and-Mortar” stores to online shopping using various technologies: Intelligent Personal Assistants, Web Apps, Phone Apps, etc. Pharmacies have been lagging in this new frontier and are beginning to understand that it is no longer a question of if their industry will be transformed but it is a question of who will lead this transformation and when it will happen. In this project, we will leverage Alexa as an interactive assistant to a Pharmacy Management System hosted in Azure to automate the process of filling prescriptions for consumers.

Disruptive Technologies affecting “Brick-and-Mortar” Pharmacies

- Artificial Intelligence (A.I.) and Robotics
- Autonomous Vehicles and Drones
- Cloud Technologies

Disruptive Competitive Forces

- Amazon
- Amazon/Berkshire Hathaway/J.P. Morgan Health Care Company
- CVS–Aetna Deal

amazon alexa

Microsoft Azure

Enter PharmAid: a cloud-based Alexa-Enabled Pharmacy front-end

PharmAid is a Pharmacy Solution which pharmacies can leverage as an adjunct to their existing “Brick-and-Mortar” stores or aid in consolidating pharmacies to a central fill site to reduce costs and increase customer satisfaction.

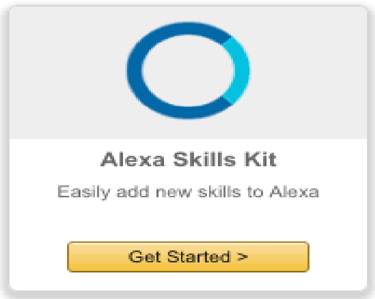
How?

- Leverages Cloud Technologies to reduce infrastructure costs and improve reporting and analytics.
- Integration with Alexa to ease ordering of prescriptions and to allow tracking of prescriptions.
- Use robotics and central fill technologies to automatically fill prescriptions.
- Integration with a Pharmacy Application API to perform pharmacy functions such as DUR (Drug Utilization Review), PPI (Pharmacy Prescriber Interface), CSR (Controlled Substance Reporting) and other pharmacy functions.
- Leverage A.I. to perform common pharmacy functions.
- Deliver products to customers more efficiently.
- Increases customer satisfaction by eliminating the need to call or travel to a pharmacy.

Azure and Amazon Technologies

- Alexa Skills Kit

  Alexa Skills Kit is used to develop the interface to be used with Alexa. It uses an Invocation word to invoke your app service and an utterance (what you want the service to do).



- Azure Web API Application Service

  An Azure Web API is used to establish an https endpoint for Alexa to use. It is responsible for servicing the requests.

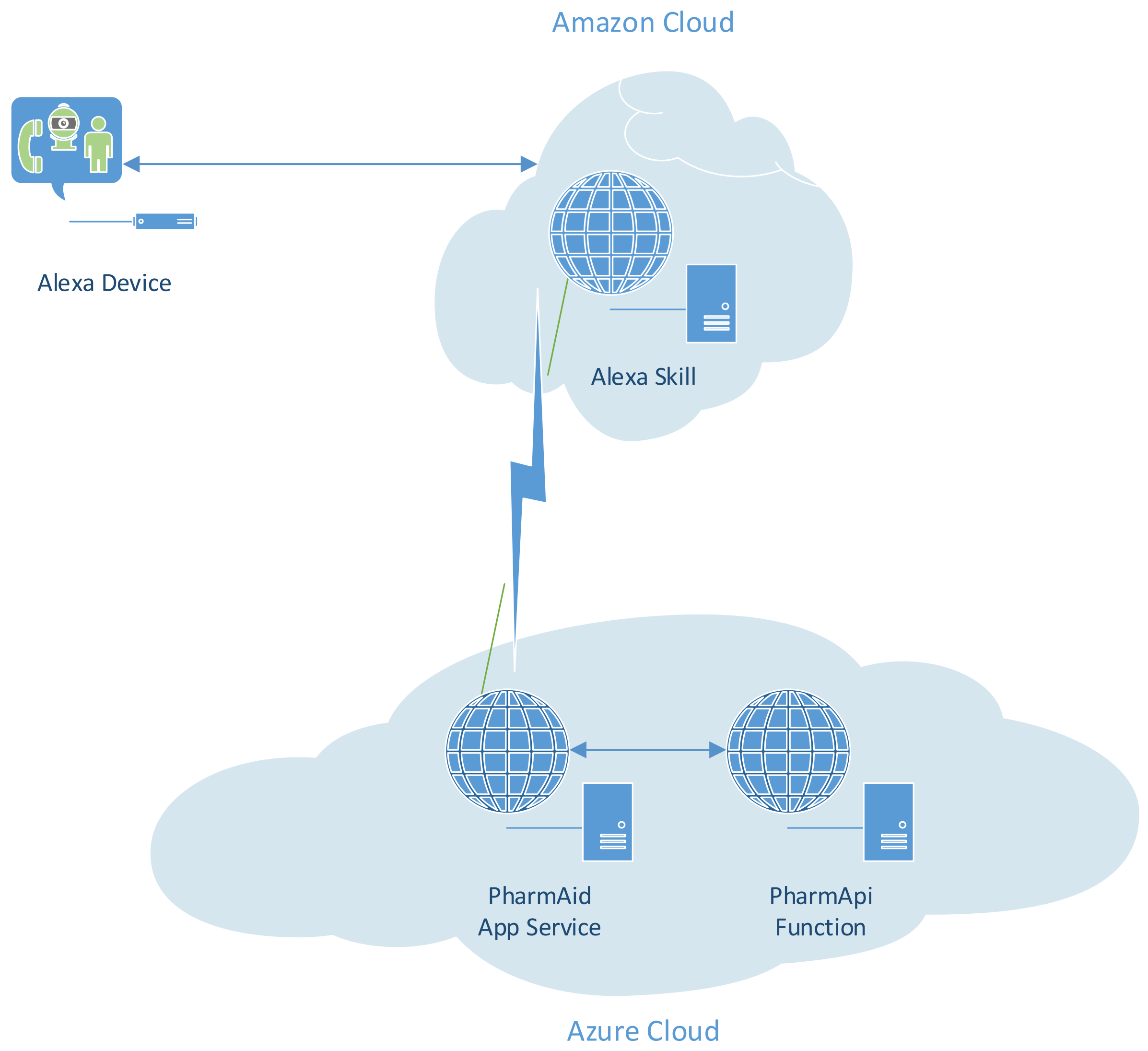
- Azure Functions

  An Azure HTTP Get Function was used to simulate a Pharmacy Management System API.

- Visual Studio .NET Application

  The entire Azure Application was developed in Visual Studio. The excellent AlexaSkillsKit.NET package was used for Alexa interface compatibility.

PharmAid Data Flow



Data Description

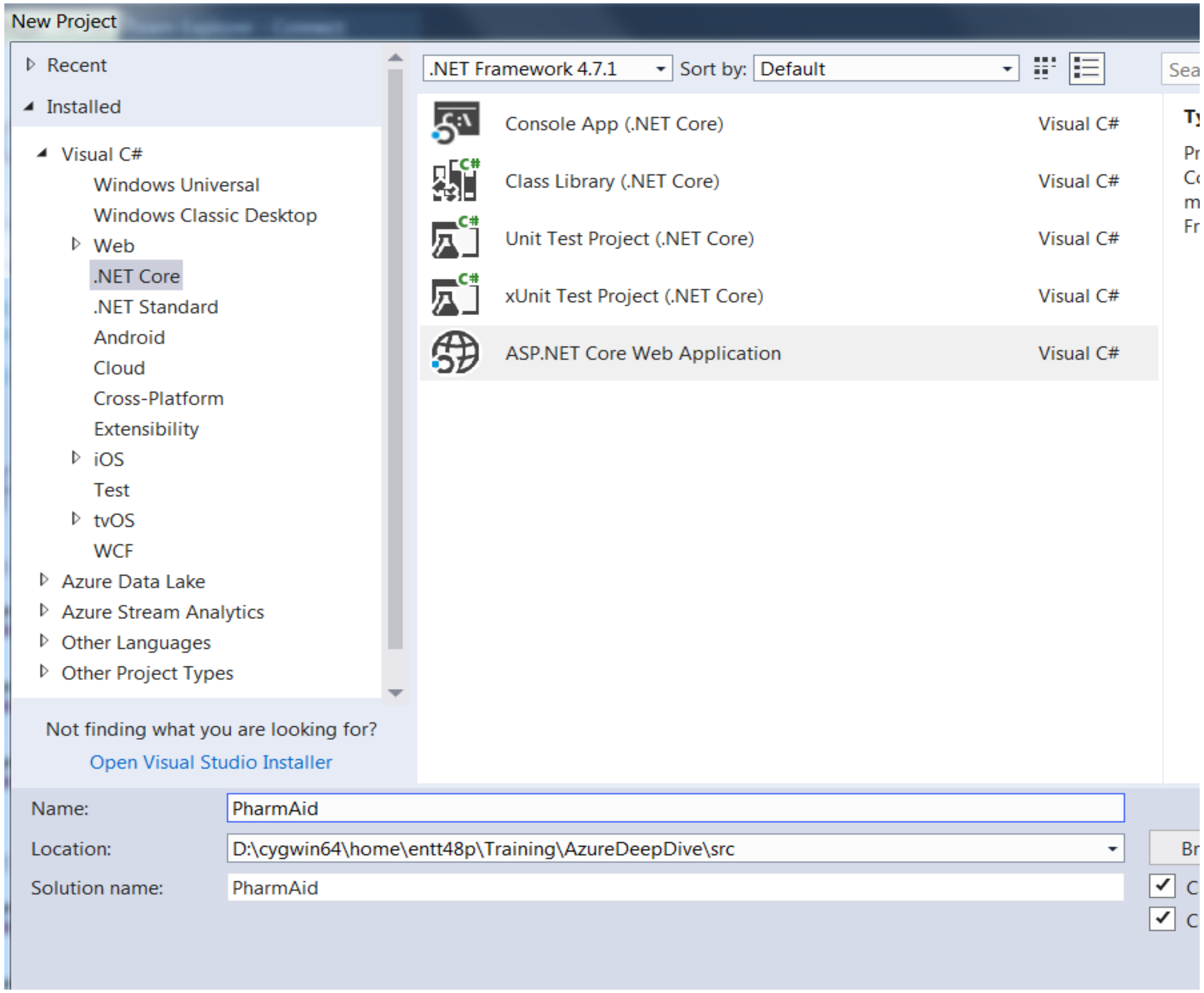
Since I was developing an Alexa integration, I did not have or need a large data set.

- PharmAid was the Application Invocation Phrase that I selected. Alexa is invoked with a simple “Ask PharmAid”.
- A JSON Object which defines the intents that Alexa can use when the application is invoked.
- Utterances which define the phrases that can be serviced by PharmAid. I only implemented four phrases for this project but this could easily be extended.

Intent	Utterance
FillRxIntent	to fill my prescription
WhereRxIntent	to find my prescription
RxReadyIntent	when will my prescription arrive
CallDoctorIntent	to call my doctor

PharmAid Configuration

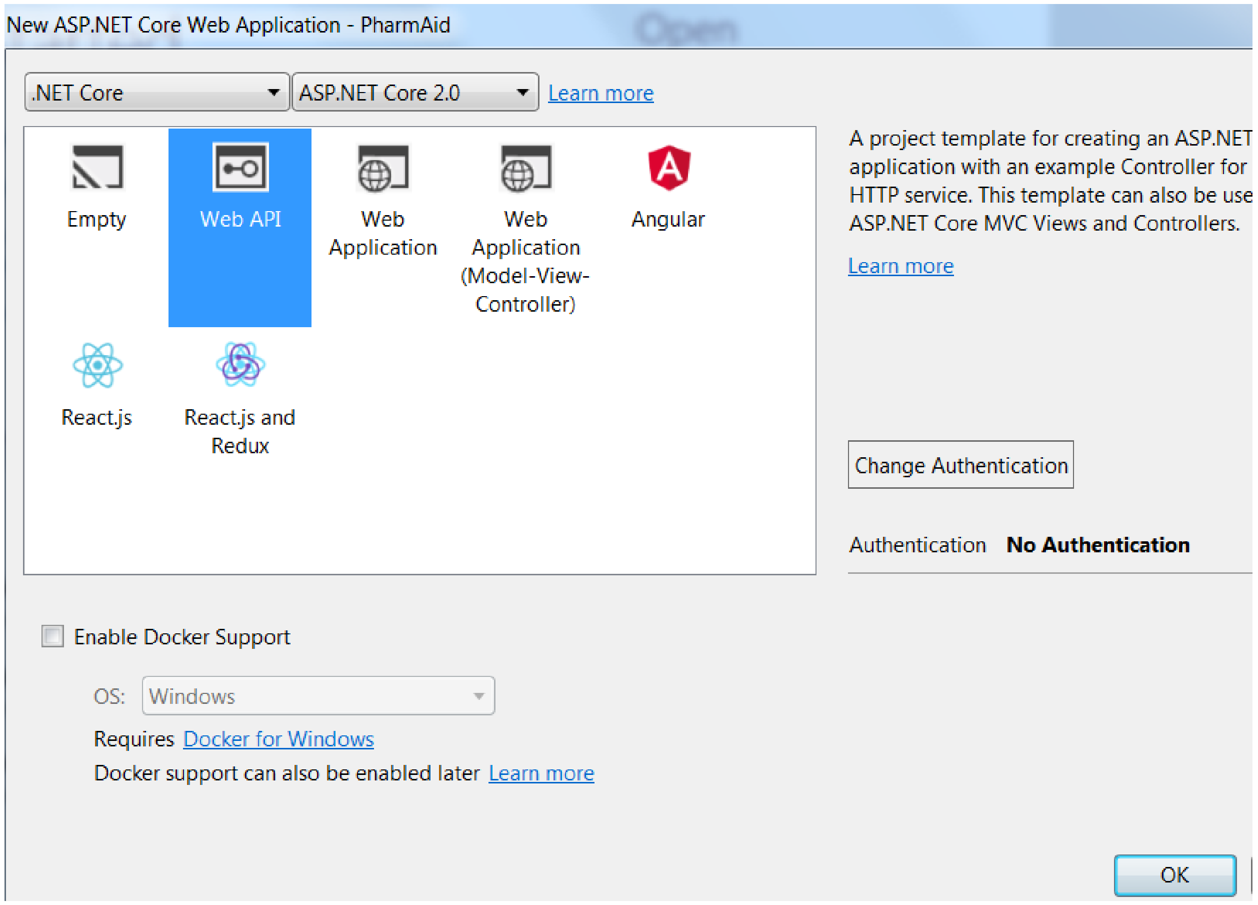
- Start by creating and ASP.NET Core Application in Visual Studio 2017



Name: PharmAid
Location: D:\cygwin64\home\entt48p\Training\AzureDeepDive\src
Solution name: PharmAid

PharmAid Configuration (cont)

- Select the Web API template which will function as the Alexa Skill API



New ASP.NET Core Web Application - PharmAid

.NET Core | ASP.NET Core 2.0 | Learn more

A project template for creating an ASP.NET application with an example Controller for HTTP service. This template can also be use ASP.NET Core MVC Views and Controllers.

Learn more

Authentication **No Authentication**

Enable Docker Support

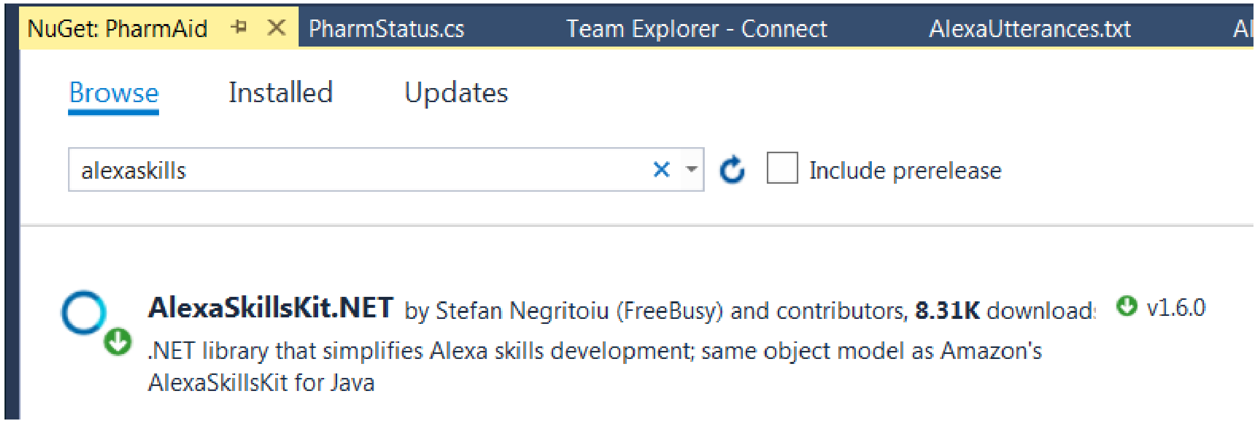
OS: Windows

Requires Docker for Windows

Docker support can also be enabled later Learn more

PharmAid Configuration (cont)

- Install the AlexaSkills.NET package and Newtonsoft.JSON package



NuGet: PharmAid | PharmStatus.cs | Team Explorer - Connect | AlexaUtterances.txt | Al

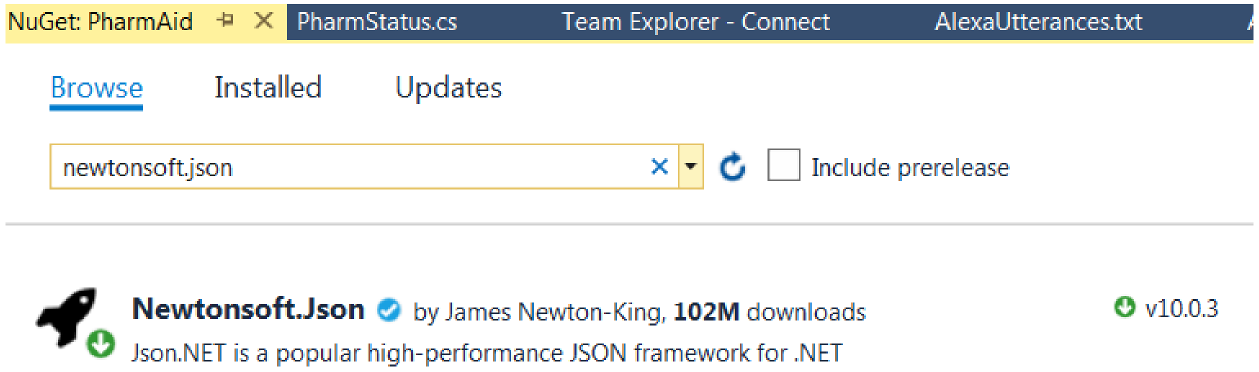
Browse | Installed | Updates

alexaskills

Include prerelease

AlexaSkillsKit.NET by Stefan Negritoiu (FreeBusy) and contributors, **8.31K** download v1.6.0

.NET library that simplifies Alexa skills development; same object model as Amazon's AlexaSkillsKit for Java



NuGet: PharmAid | PharmStatus.cs | Team Explorer - Connect | AlexaUtterances.txt

Browse | Installed | Updates

newtonsoft.json

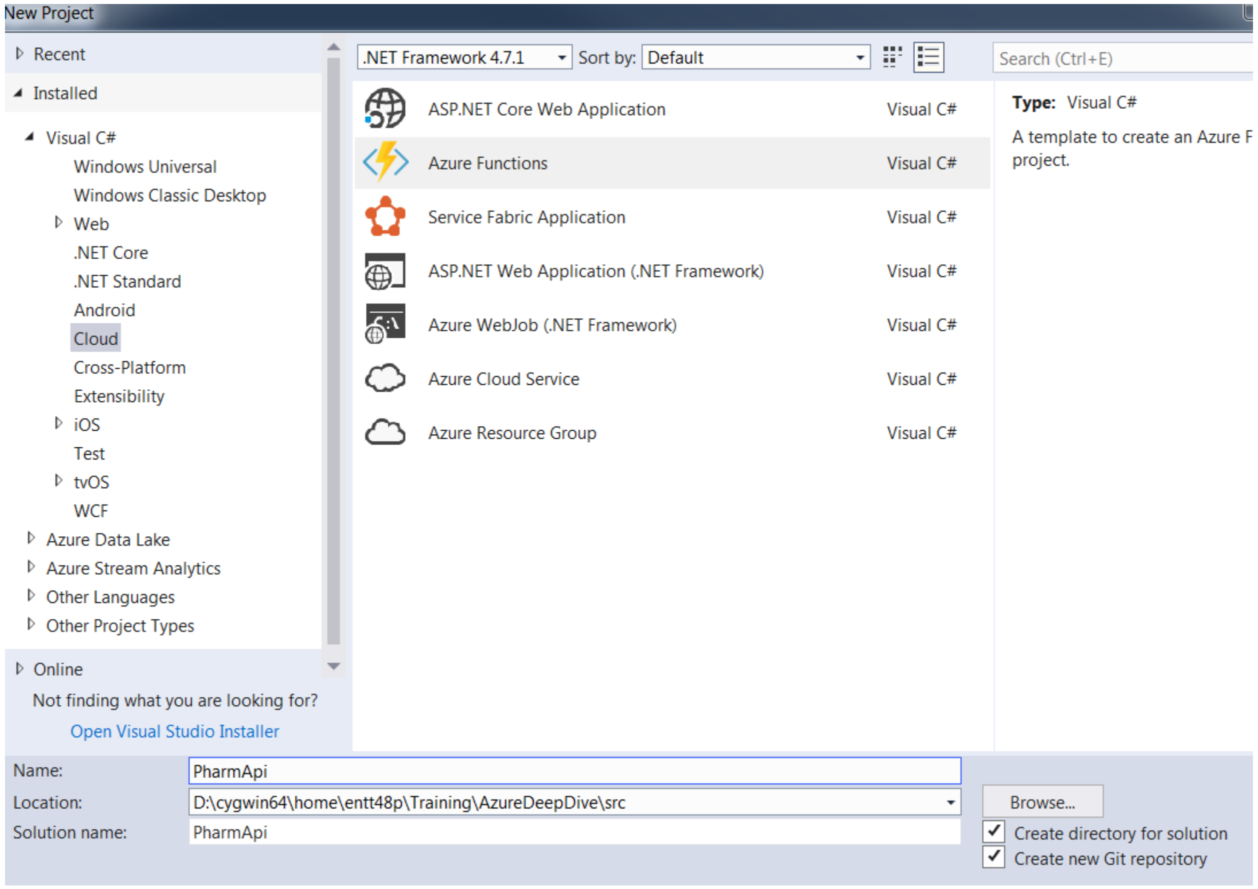
Include prerelease

Newtonsoft.Json by James Newton-King, **102M** downloads v10.0.3

Json.NET is a popular high-performance JSON framework for .NET

PharmApi Configuration

- Create an Azure Function (simulated pharmacy management system) in Visual Studio 2017 to respond to my PharmAid application.


New Project

.NET Framework 4.7.1 Sort by: Default

Search (Ctrl+E)

ASP.NET Core Web Application	Visual C#
Azure Functions	Visual C#
Service Fabric Application	Visual C#
ASP.NET Web Application (.NET Framework)	Visual C#
Azure WebJob (.NET Framework)	Visual C#
Azure Cloud Service	Visual C#
Azure Resource Group	Visual C#

Type: Visual C#

A template to create an Azure F project.

Name: PharmApi

Location: D:\cygwin64\home\entt48p\Training\AzureDeepDive\src

Solution name: PharmApi

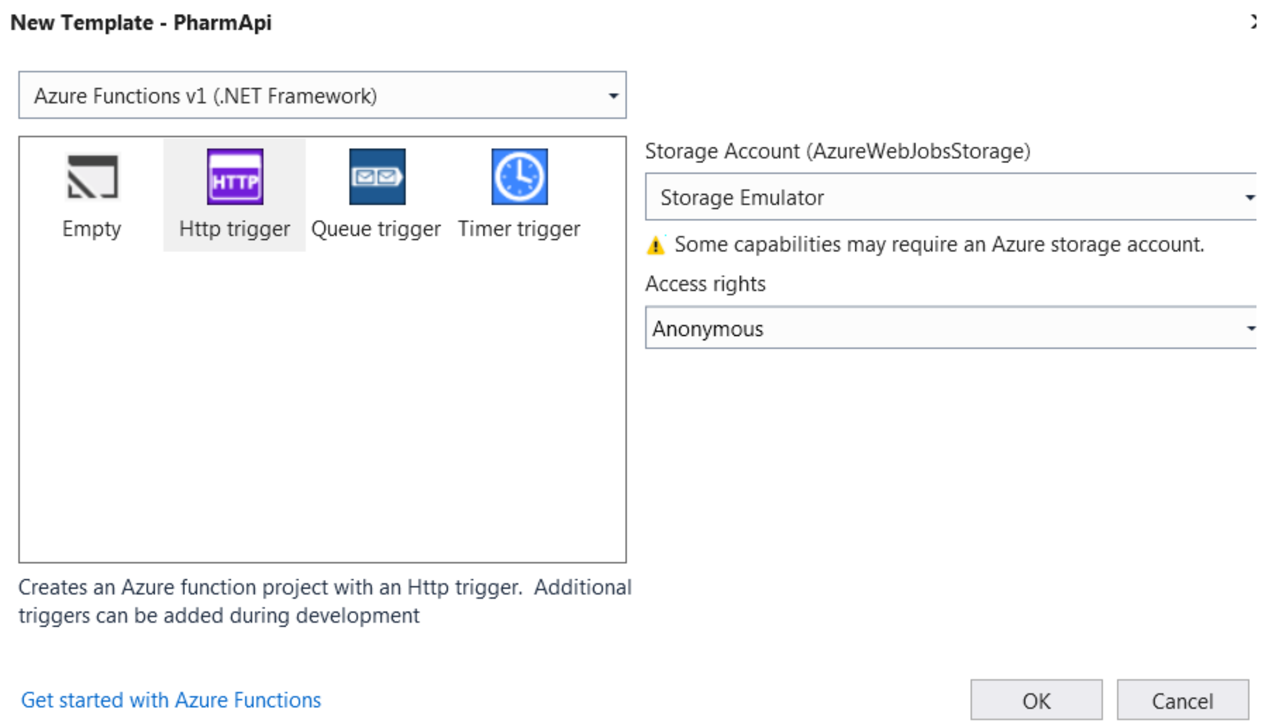
Browse...

✓ Create directory for solution

✓ Create new Git repository

PharmApi Configuration (cont)

- Use Http trigger template so PharmApi responds to Get requests from PharmAid.



New Template - PharmApi

Azure Functions v1 (.NET Framework)

Creates an Azure function project with an Http trigger. Additional triggers can be added during development

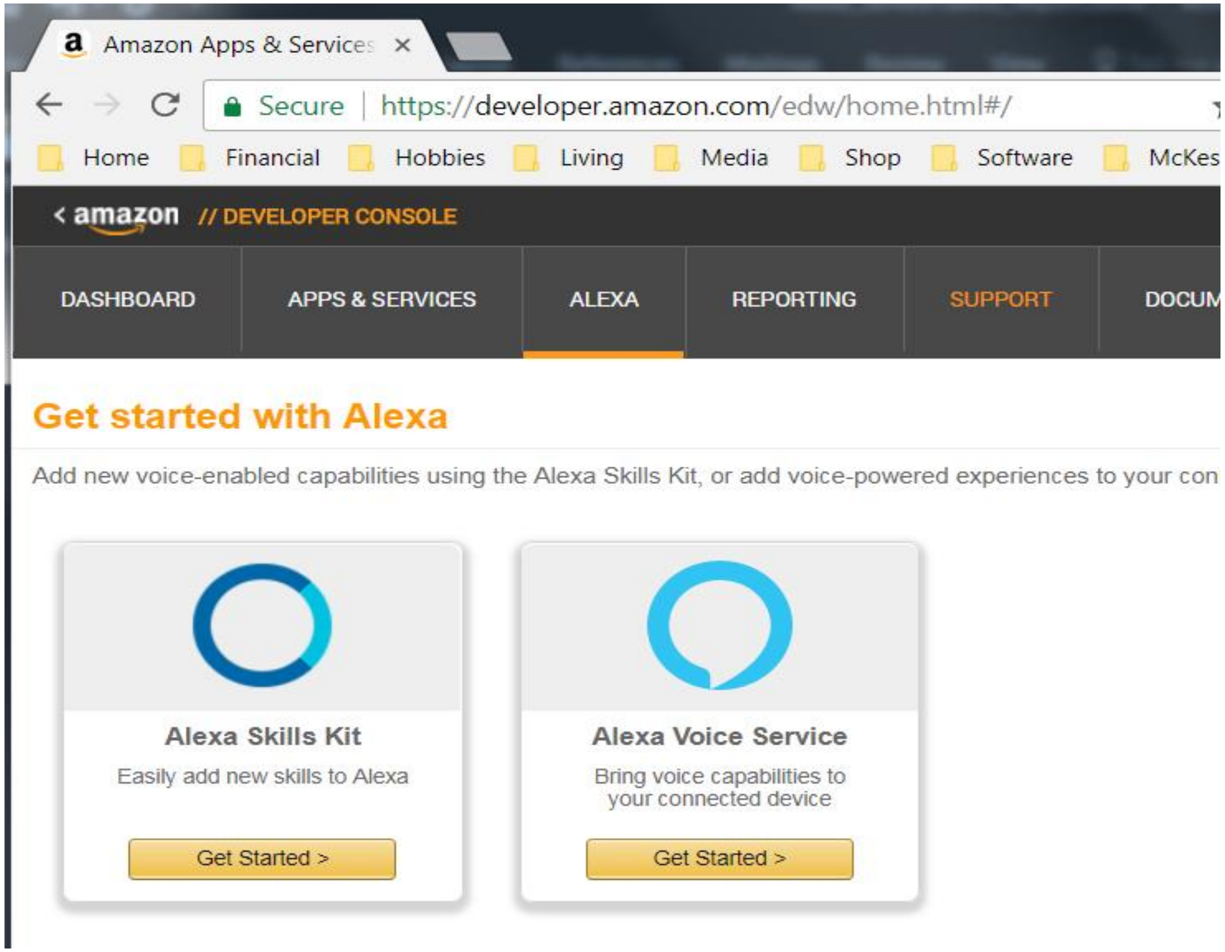
Storage Account (AzureWebJobsStorage): Storage Emulator

⚠ Some capabilities may require an Azure storage account.

Access rights: Anonymous

Alexa Skills Configuration

- To create an Alexa Skill, you must have an Amazon Developer Account which is free. After registering, you can login to the Alexa Skills Developer Portal to create your skill. Click on Alexa Skills Kit to get started.



Get started with Alexa

Add new voice-enabled capabilities using the Alexa Skills Kit, or add voice-powered experiences to your con

Alexa Skills Kit

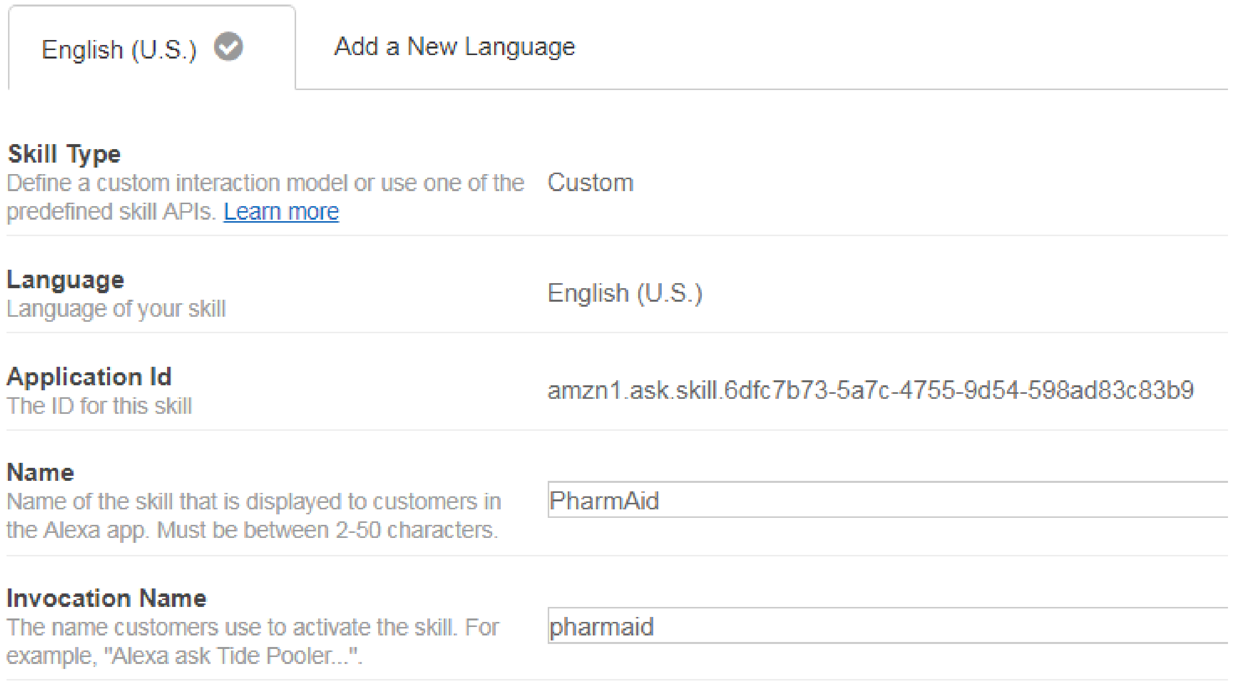
Easily add new skills to Alexa

Alexa Voice Service

Bring voice capabilities to your connected device

Alexa Skills Configuration (cont)

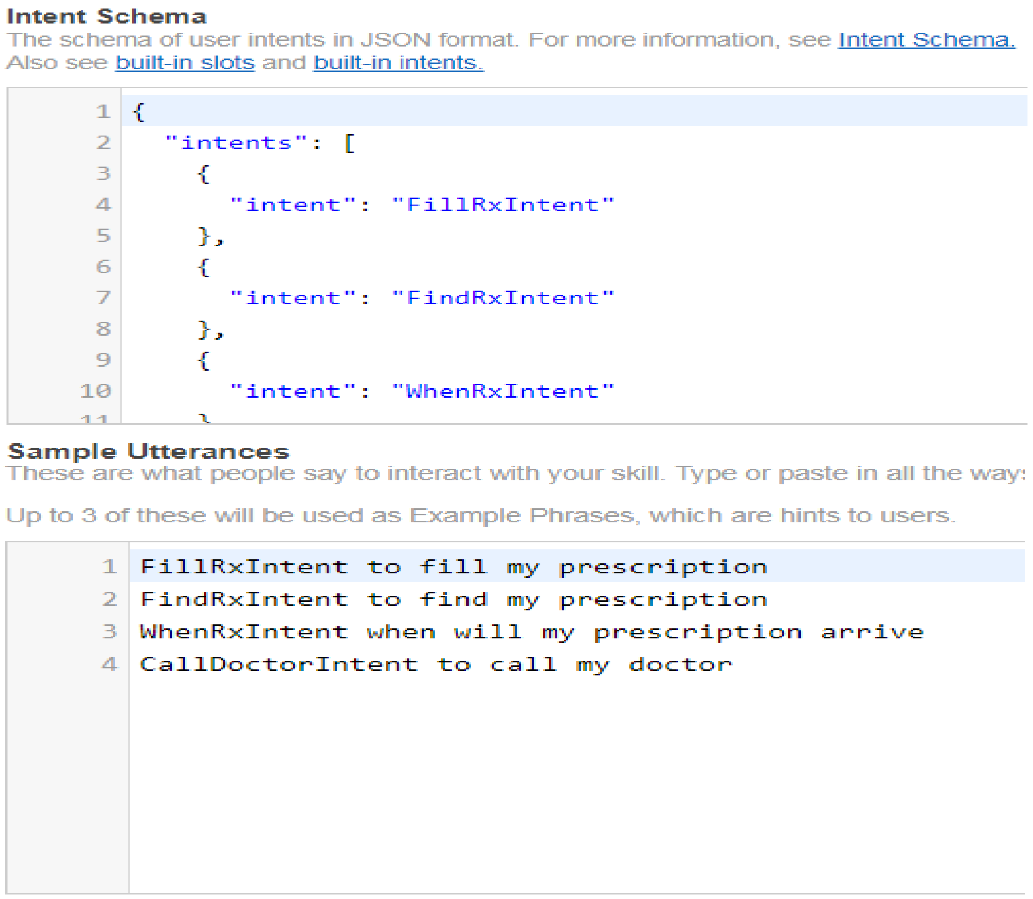
- Enter your skill information. My skill will be invoked by “Ask PharmAid…”.



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.6dfc7b73-5a7c-4755-9d54-598ad83c83b9
Name Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.	PharmAid
Invocation Name The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...".	pharmaid

Alexa Skills Configuration (cont)

- Create an interaction model which is a combination of an Intent Schema in JSON format and Utterances. Utterances are phrases you say after invoking your application.



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.

```
{
  "intents": [
    {
      "intent": "FillRxIntent"
    },
    {
      "intent": "FindRxIntent"
    },
    {
      "intent": "WhenRxIntent"
```

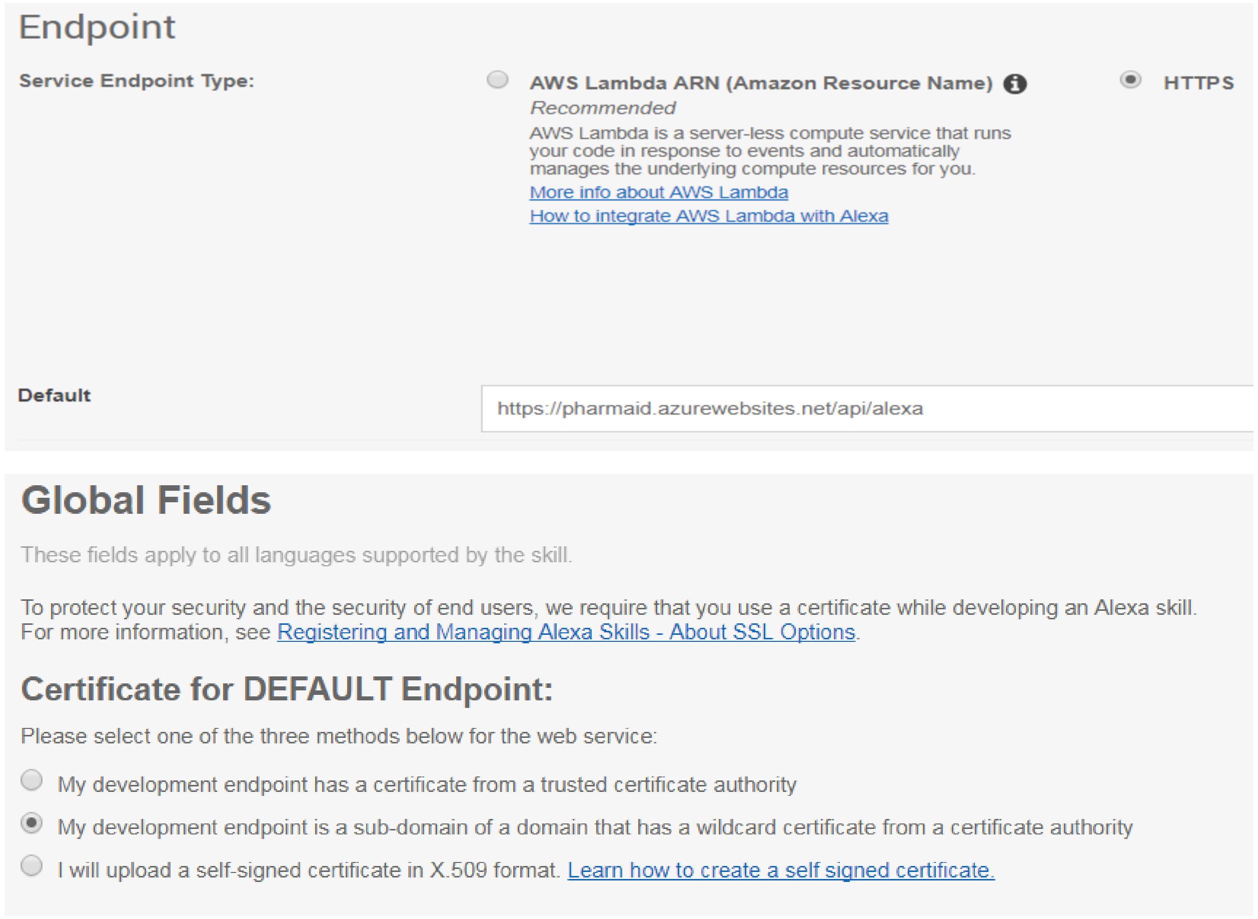
Sample Utterances
These are what people say to interact with your skill. Type or paste in all the way

Up to 3 of these will be used as Example Phrases, which are hints to users.

```
FillRxIntent to fill my prescription
FindRxIntent to find my prescription
WhenRxIntent when will my prescription arrive
CallDoctorIntent to call my doctor
```

Alexa Skills Configuration (cont)

- The next step is critical to connect to Azure. Select an HTTPS endpoint, not AWS Lambda, and enter the Azure endpoint.
- Make sure to enter the correct certificate for the DEFAULT Endpoint.



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: https://pharmaid.azurewebsites.net/api/alexa

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.

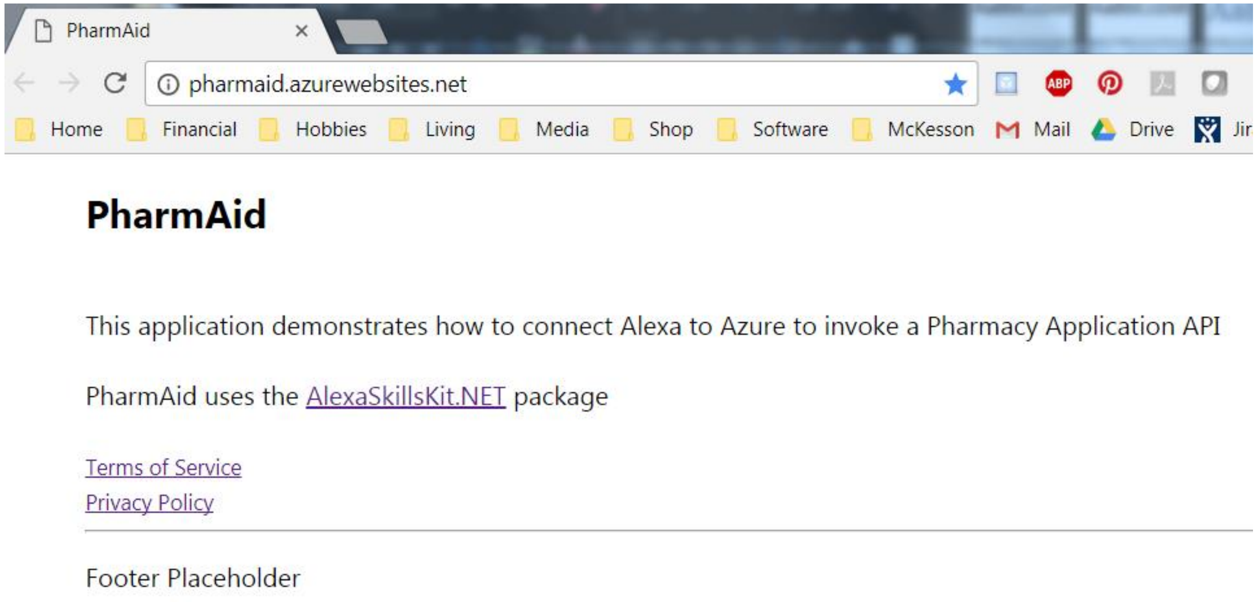
Certificate for DEFAULT Endpoint:

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.

PharmAid Visualization

- Enter pharmaid.azurewebsites.net in a browser and verify it works.



PharmAid

This application demonstrates how to connect Alexa to Azure to invoke a Pharmacy Application API

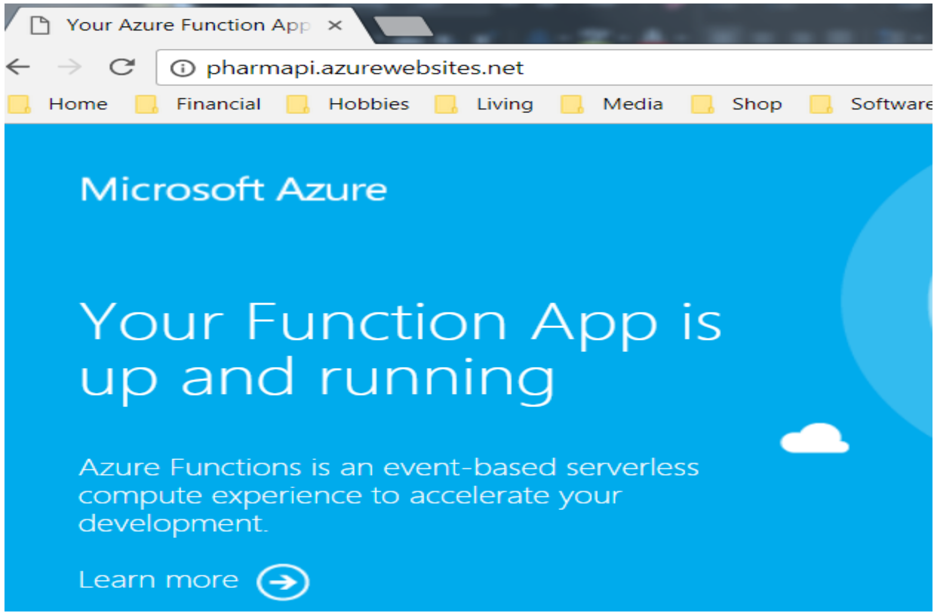
PharmAid uses the AlexaSkillsKit.NET package

Terms of Service
Privacy Policy

Footer Placeholder

PharmApi Visualization (cont)

- Enter pharmapi.azurewebsites.net in a browser and verify it works.
- Verify that the function returns JSON code.


Your Azure Function App

pharmapi.azurewebsites.net

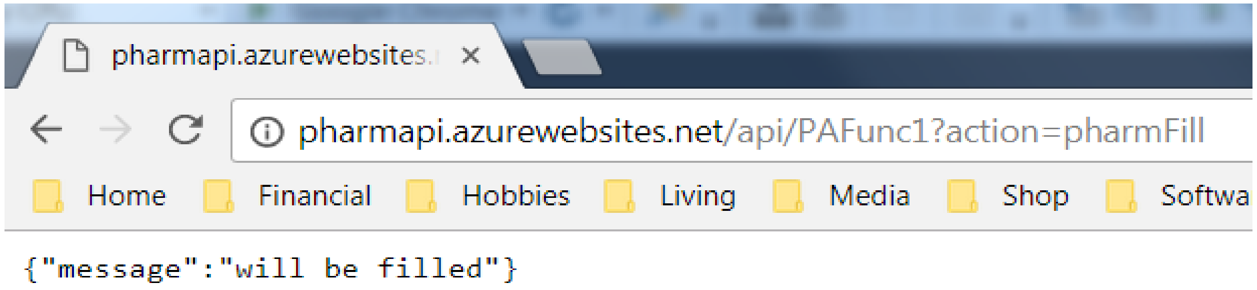
Home Financial Hobbies Living Media Shop Software

Microsoft Azure

Your Function App is up and running

Azure Functions is an event-based serverless compute experience to accelerate your development.

Learn more


pharmapi.azurewebsites.

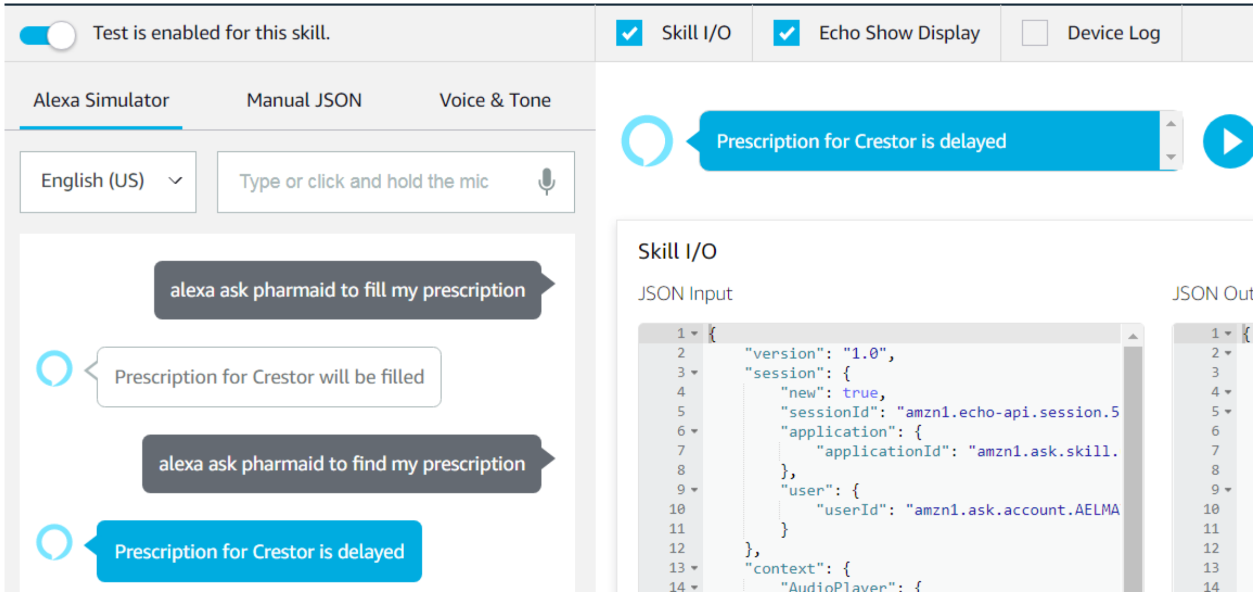
pharmapi.azurewebsites.net/api/PAFunc1?action=pharmFill

Home Financial Hobbies Living Media Shop Softwa

```
{"message":"will be filled"}
```

Alexa Skills Results

- The first way to test your skill end-to-end is to enter the Beta Test Simulator. Hold down the microphone and speak your invocation word and your utterance. The dark grey is my voice input and the Alexa response is spoken as well as displayed.



alexa ask pharmaid to fill my prescription

Prescription for Crestor will be filled

alexa ask pharmaid to find my prescription

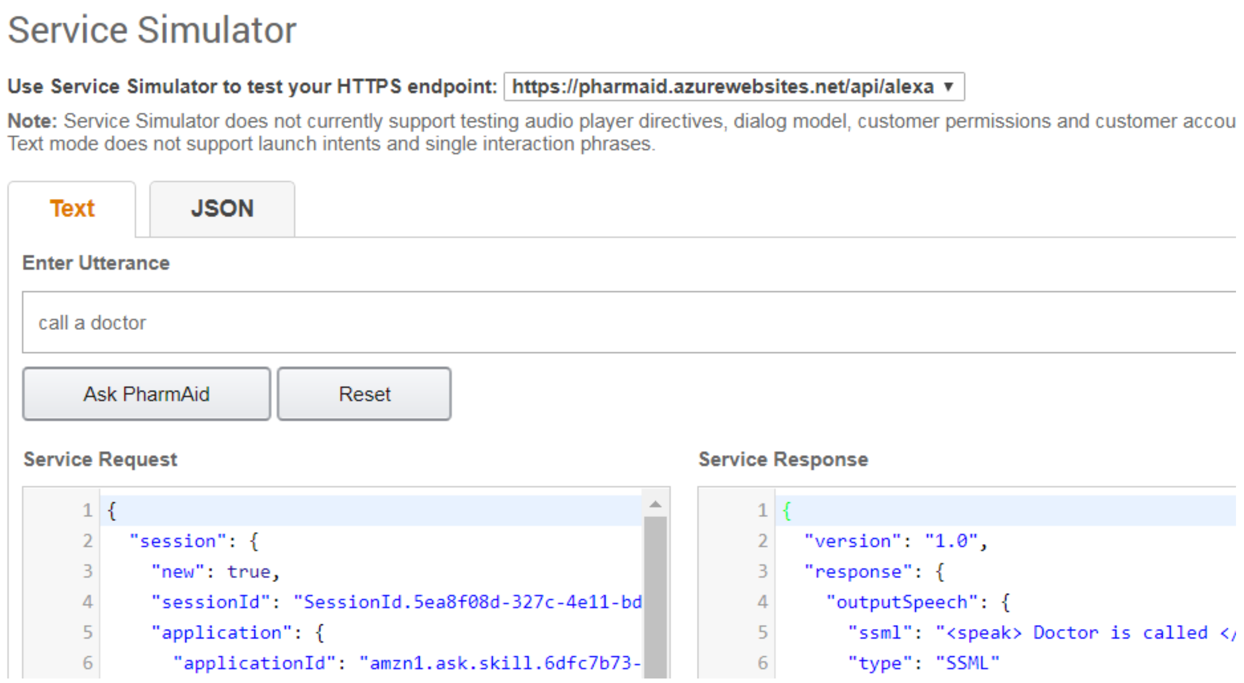
Prescription for Crestor is delayed

Prescription for Crestor is delayed

```
{
    "version": "1.0",
    "session": {
        "new": true,
        "sessionId": "amzn1.echo-api.session.5
        "application": {
            "applicationId": "amzn1.ask.skill.
        },
        "user": {
            "userId": "amzn1.ask.account.AELMA
        }
    },
    "context": {
        "AudioPlayer": {
```

Alexa Skills Result (cont)

- The second way to test your skill is to use the Developer Console. Enter your invocation and utterance to display the response.



Service Simulator

Use Service Simulator to test your HTTPS endpoint: https://pharmaid.azurewebsites.net/api/alexa ▼

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

Enter Utterance

call a doctor

Service Request

```
{
  "session": {
    "new": true,
    "sessionId": "SessionId.5ea8f08d-327c-4e11-bd
    "application": {
      "applicationId": "amzn1.ask.skill.6dfc7b73-
```

Service Response

```
{
  "version": "1.0",
  "response": {
    "outputSpeech": {
      "ssml": "<speak> Doctor is called </
      "type": "SSML"
```

Summary

Lessons Learned

Likes

- The AlexaSkillsKit.NET package was excellent. It made implementing the Alexa calls very easy.
- Visual Studio is simply great to program in when building Azure applications due to its tight integration. It also works seamlessly with github. It is simply a matter of a few mouse clicks to compile, test, and publish the application.
- The Alexa Skills portal was very nice. After becoming an Alexa developer, you login and create your skill in a matter of minutes. You give it an invocation word, a JSON schema, and your utterances. From there, you pass an https endpoint to Azure and you are done. The portal makes it easy to test your application. You can test it via verbal commands and listening to the response or by typing in your commands and seeing the response.
- Alexa has a concept of slots where you can define preset values. For example, you could use slots to hold the 10 most common medications to make it easier to order them.

Summary (cont)

Lessons Learned

Benefits/Pros

- The entire application is server-less using Azure App Services so there is no infrastructure to maintain. The application should also scale rather easily.
- Alexa is already prevalent in a lot of households and is leading in market share so it would be useful for individuals that are comfortable with the technology.
- It shouldn’t be too hard to extend Alexa to work with an existing Pharmacy Application Management System that has a decent API using this structure. You could easily implement as many intents as you wanted and retrieve valid information for the consumer.

Summary (cont)

Lessons Learned

Dislikes

- The verbal commands are tricky. Alexa is pretty good at understanding speech but it doesn’t always get it right which can be frustrating.
- It would be nice if Azure was more tightly integrated with Amazon Alexa. The AlexaSkillsKit is nice but it feels like it should be easier to integrate your Alexa skill.

Issues/Cons

- Security/PHI would be a big concern. We would have to be very careful about what data is being submitted to Alexa.
- Privacy is another concern. Users might not want to be shouting out prescriptions they are taking to an active listening device.
- I’m not sure how easy this would be to port to other voice assistants. I just don’t know enough about them.
- There are a lot of medications and I don’t think Alexa would get them right most of the time.

What Next?

Alexa and other digital assistants are here to stay. The devices will become more and more prevalent. I also am pretty confident that the pharmacy industry will look dramatically different a few years from now than it does today.

In a few years, the millennial generation, whom are very comfortable with technology, will demand that their prescriptions are delivered to them. It is no longer a question of when this transformation will happen but it is simply a matter of when and whom will lead it.

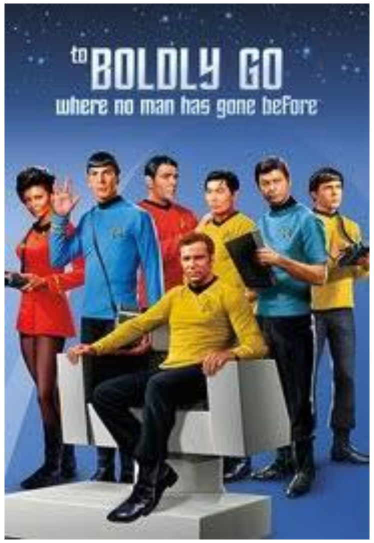
We are at a cross-roads with technology. ..

- The cloud has proven an effective strategy to reduce and eliminate infrastructure costs.
- The cloud is evolving rapidly with technology such as machine learning, analytics, server-less functions, etc. which hosted applications could ever afford to manage or support.
- IOT (Internet of Things) has arrived and will connect everything. Medications will have RFID devices that track their distribution and ensure they are being taken.
- A.I. (Artificial Intelligence) has arrived. A.I. will replace a lot of pharmacy functions.
- Autonomous Vehicles/Drones are coming.

What Next? (cont)

I could envision a world where I can simply ask Alexa to fill my prescription and a few hours later the medication is delivered via an autonomous vehicle or drone that I walk up to which verifies my identity via facial recognition and then opens a door to allow me to retrieve it.

Tomorrows pharmacy will look dramatically different than the pharmacy of today.



PharmAid was written as a proof-of-concept to see if I could integrate Alexa with Azure. The idea could be extended for pharmacy-to-doctor communications, pharmacy-to-pharmacy, pharmacy-to-distributor, and pharmacy to consumer.

I would like to see what would happen if Alexa was integrated with a real Pharmacy Management System. This would be interesting and could help lead pharmacies into the future.

YouTube URLs, GitHub URL

- 2 minute (short): https://youtu.be/QwHZInw3xYE
- 15 minutes (long): https://youtu.be/KFFh9k4tmng
- GitHub Repositories with all artifacts:

  https://github.com/james-francis/DeepAzure-FinalProject

Separate GitHub URLs for cloning

- PharmAid (includes Alexa JSON): https://github.com/james-francis/PharmAid
- PharmApi: https://github.com/james-francis/PharmApi