Building the office of the future with Cortana

GETTING STARTED WITH THE BOT FRAMEWORK

About Me

- Bill Crider
- Director: Microsoft Practice at Ascendum
- Cincinnati Ohio
- Email: Bill.Crider@Ascendum.com
- Twitter: @Bill_Crider_

State of voice/bot tech

- Estimated to grow to 82% of all mobile devices by 2020
- In your home, office and car already
- A new frontier in productivity

Natural uses in the Office 365 world of collaboration

- Replace search with guided bots and voice commands
- Personal assistants – calendar, appointments, tasks
- Connect with people in your organization
- Canned routines for predictable conversations

Setup Cortana on different devices

- Invoke Speaker
- Phone
- PC
- Connect calendars on all 3 devices
- Connect to iHeart and Spotify music services
- Can control basic commands on PC
- Voice recognition still 1B generation – fairly basic interactions

Demo – Cortana OOTB

- Settings – set up voice activation
- Manage what Cortana can access
- Search across devices
- Heads Up
- Invoke speaker
- Open Alexa

Developing Bots and Cortana apps

- Easiest way is to run as Azure service with Bot services providing the middleware
- C#, NodeJS platforms
- V4 release in November 2018 – major rewrite of platform
- https://dev.botframework.com/
- Lot of the documentation is out of date
- All the Cortana docs are V3 as of early December 2018

Setting up Azure Services

- Everything can be done for testing on FREE levels of service
- Be Careful – a lot of quick starts in the docs provision you a paid level of services

- DEMO – Set up App Services

Set Up App Service Plan

Search App Service Plan in all services on Azure Portal

Create a free tier – you will be able to choose this tier for your future services

Setting up your Dev Environments

- Visual Studio 2017 version 15.9 minimum
  - Online samples use .Net Core 2.1 – you will get errors with lower versions
- Visual Studio Code (or other dev tool)
- NodeJS
- Bot Emulator

https://github.com/Microsoft/BotFramework-Emulator/releases/tag/v4.1.0

Setting up your Dev Environments

- GitHub Samples

https://github.com/Microsoft/BotBuilder-Samples

VS 2017 Bot Template

https://botbuilder.myget.org/feed/aitemplates/package/vsix/BotBuilderV4.fbe0fc50-a6f1-4500-82a2-189314b7bea2

CLI Tool

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?view=azure-cli-latest

Demo – Setting up a Bot from Azure

- This is the easiest way to get started – everything is set up for you
- Go to Azure Portal , select BOT Services – AI + machine learning section
- Add " web app bot" , Click create
- Fill out the form
  - Select the Free pricing tier you created earlier if you don't want to pay
  - Doesn't seem to matter which template you pick, they are the same codebase
  - Select V4 and language preference
  - Enterprise BOT – requires separate signup and approval
  - Application insights gives you some usage data

Demo - Set up a Bot from Visual Studio

- https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-howto-deploy-azure?view=azure-bot-service-4.0&tabs=csbotvs
- Build some code in Visual Studio (or start with samples)
- Right click solution – Publish
- Bot Registration

Demo - Add a Cortana channel

- Go to Bot Services and select a BOT
- Under Bot management, select Channels
- Add the Cortana channels
- Test the Bot in web chat or the Emulator

Set up LUIS Service

- Used for accessing the natural language processing service
- https://www.luis.ai/applications
- Apps set up automatically for Bot created from Azure portal
- Intents are a course of action that you can provide logic for
  - Intents are trained from presumed utterance patterns
- Entities are keywords from utterances

Set up QnA Service

- https://www.qnamaker.ai/
- Create question and answer pairs
- A set of questions and answers is a “ knowledge base”

Lets look at some code

- Echo Bot
- Adaptive Card
- Cortana Card
- Debugging online
- Debugging locally