Microsoft

SmartShopping Sample Overview

Windows Partner Enablement Team
Microsoft, Taiwan

Agenda

- Scenario & Solution Overview
- Design Overview
- Backend Services
- Frontend Apps

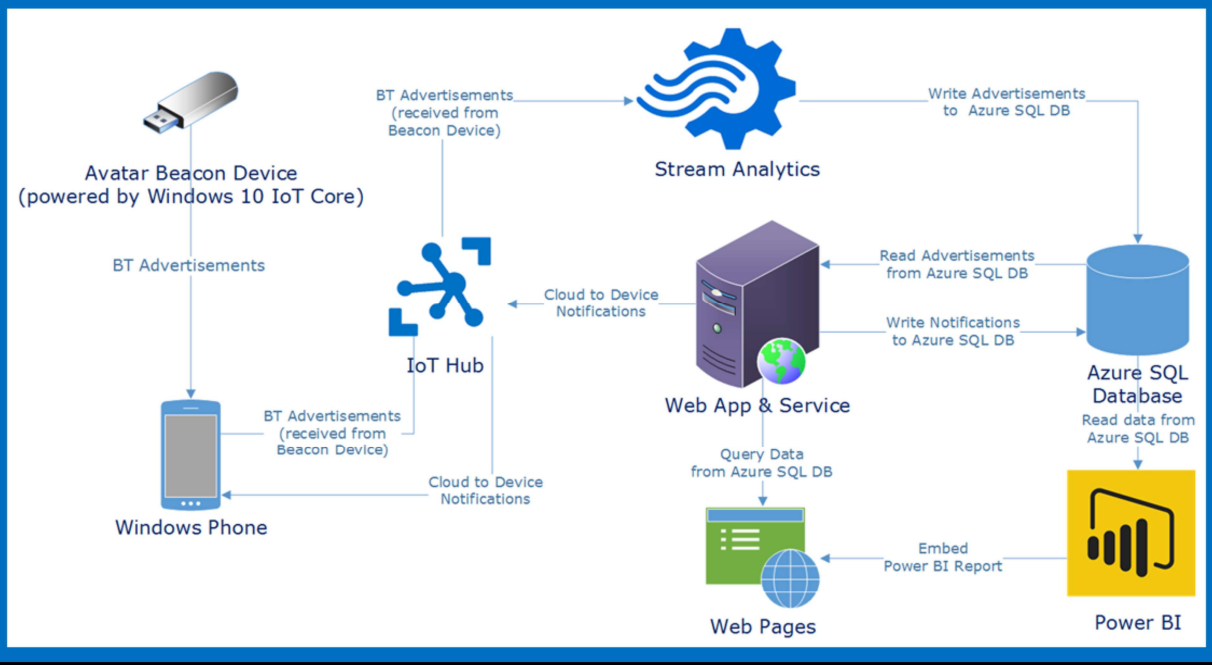
Scenario Overview - Retail

- To do targeting promotion to guests that come close to the hot spot setup on retail floor.
- To collect guests visit data on retail floor or booth.
- Use BLE beacon to declare a hot spot.
- A hand-held device for guests to report position and to receive promotion notification.
- Keep data on Azure for advanced use.

Solution Overview – Technologies

- BLE Advertisement
- Azure – IoT Hub, Stream Analytics job
- Azure - Web App / API App
- Azure – SQL Database
- Windows 10 Mobile – Notification
- UWP App & Background Tasks
- Power BI

Solution Overview – Architecture



Design Overview

- Avatar BLE Beacon (skip in this slides)
- Beacon emulator (test tool for development use)
- Frontend Phone App (UWP App)
- Backend Web Apps (ASP.NET / MVC)

UWP Sample
https://github.com/Microsoft/Windows-universal-samples

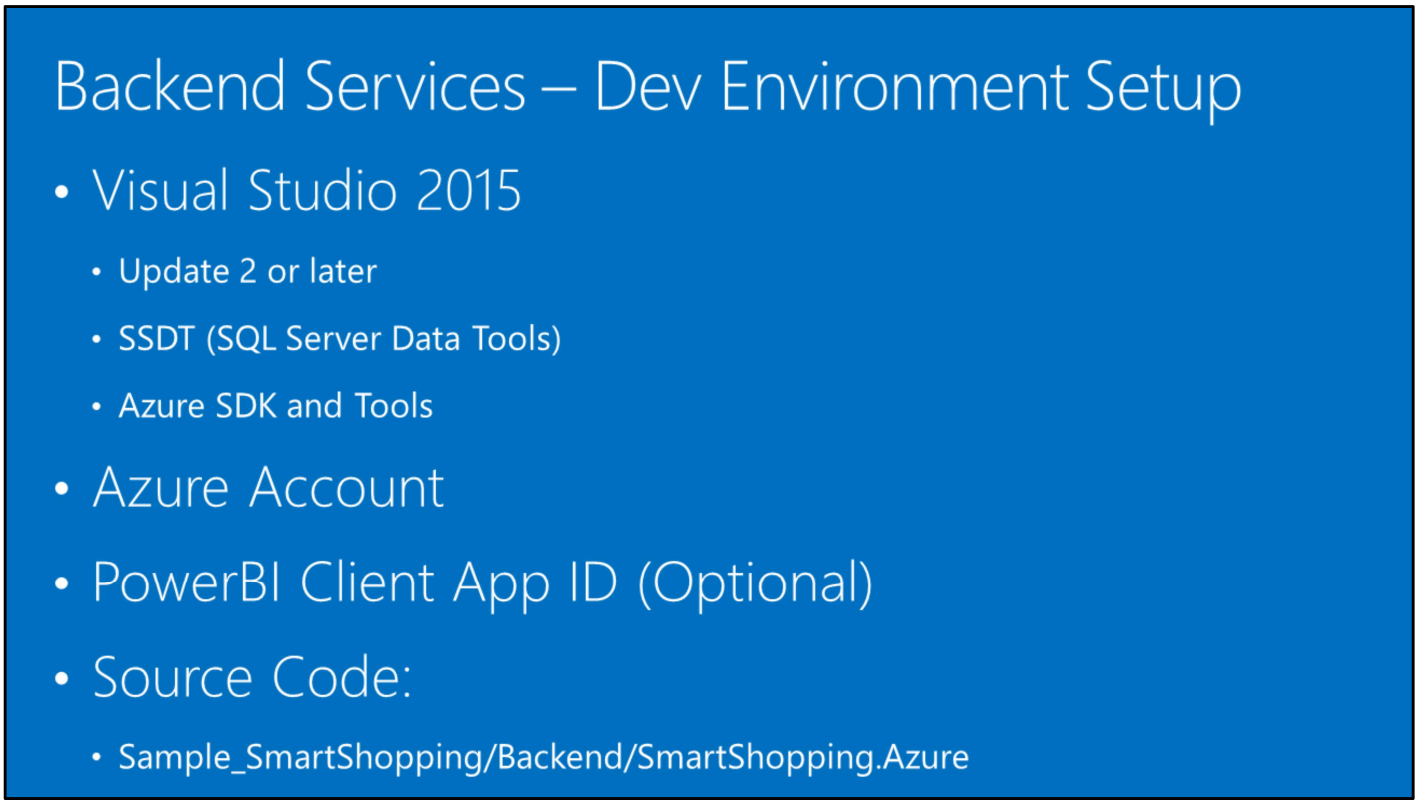
UWP Sample – Bluetooth Advertisement
https://github.com/Microsoft/Windows-universal-samples/tree/master/Samples/BluetoothAdvertisement

Design Overview - Frontend Phone App

- A UWP App
  - To watch BLE Advertisement message broadcasted from beacon devices.
  - To send BLE signal information to backend IoT Hub.
  - To receive product promotion message from backend IoT Hub.
  - To toast a notification with product promotion highlights.
  - To show promotion content on screen.
- For Windows 10 mobile devices

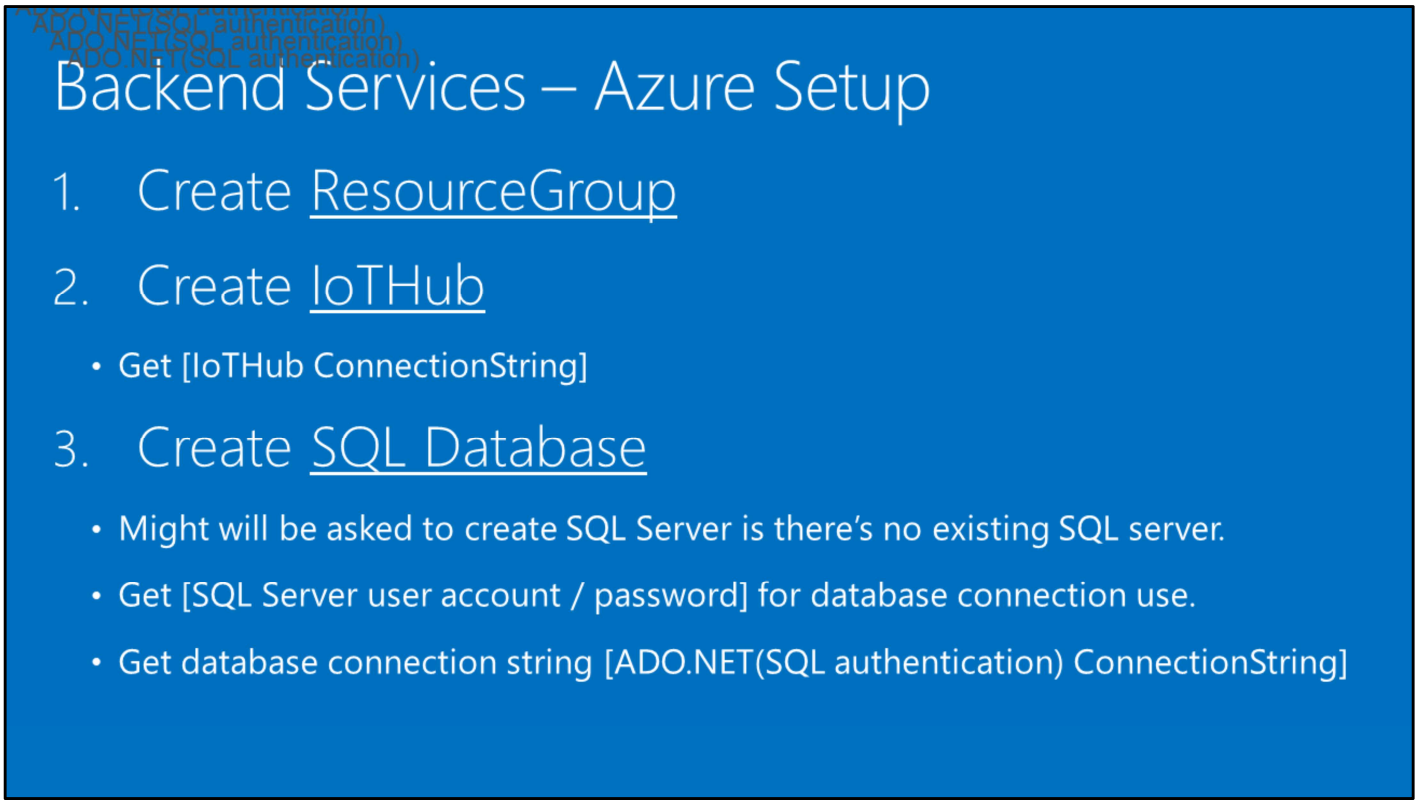
Design Overview - Backend Services

- Provide Web UI :
  - For Management / Monitoring / Reports
- Provide RESTful API:
  - For mobile devices (phone app) to get IoT device Identity
- Process messages from/to devices
- Hosted on Azure
  - ASP.NET / MVC / Access to IoTHub / Access to SQL Database



Download SQL Server Data Tools (SSDT)
https://msdn.microsoft.com/library/mt204009.aspx

Azure SDK for VS 2015
https://www.visualstudio.com/en-us/features/azure-tools-vs.aspx



ADO.NET(SQL authentication) Connection String format:

Server=tcp:{server hostname},1433;Initial Catalog={database name};Persist Security Info=False;User ID={your_username};Password={your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;

-- When you copy the connection string from Azure portal, the "User ID" and "Password" are not configured in the connection string, you need to fill in your SQL account name and password,

For example :

your SQL server host name: mysqlservername.database.windows.net

your SQL account name: mysqlaccount

your SQL account password: mysqlpassword

Set in connection string like this:

"User ID=mysqlaccount@mysqlservername;Password=mysqlpassword;"

Backend Services – Azure Setup

4. Create tables in SQL Database

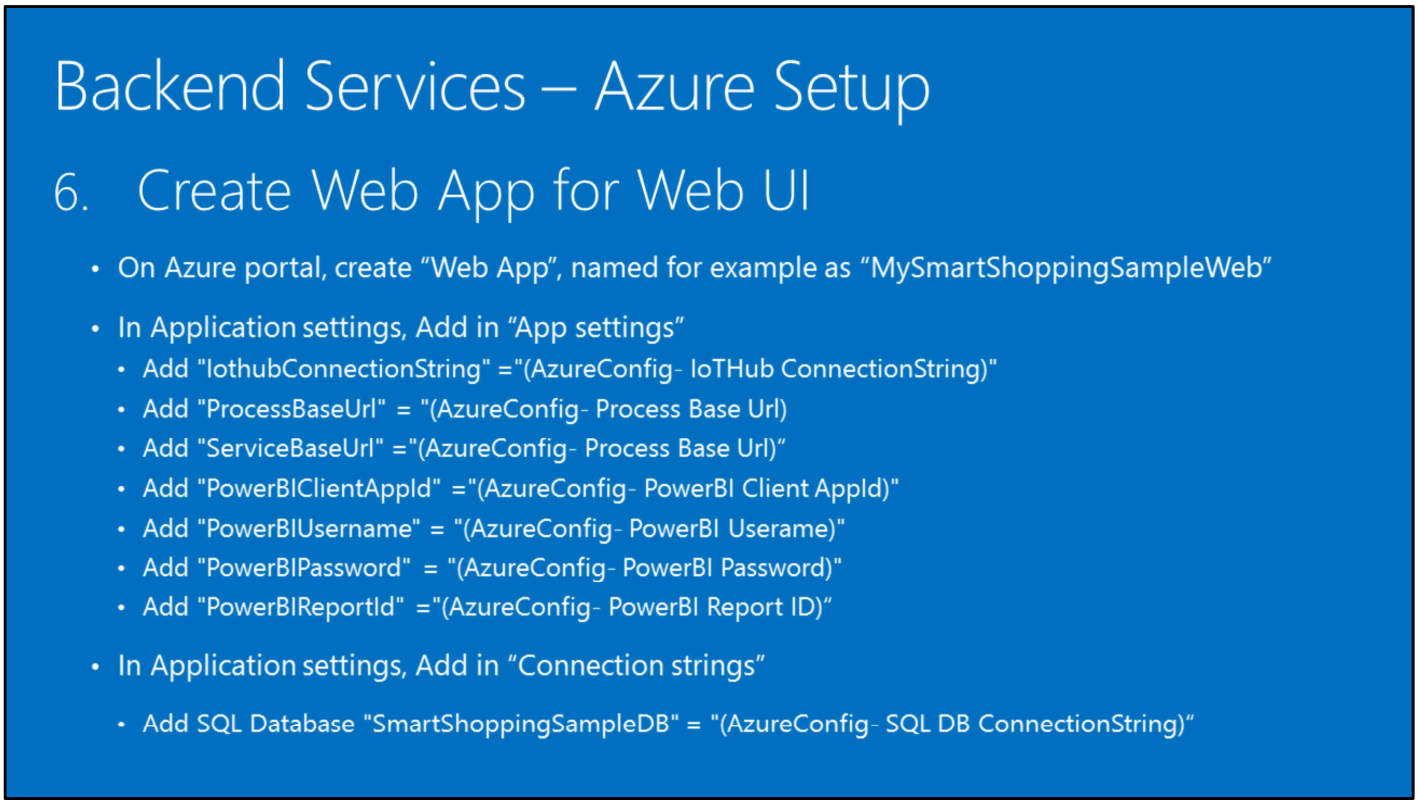
- Run SQL script to create tables (using VisualStudio SSDT):
  Sample_SmartShopping/Backend/SmartShopping.Azure/SQLScript/Table
- Run SQL script to create store procedures (using VisualStudio SSDT):
  Sample_SmartShopping/Backend/SmartShopping.Azure/SQLScript/Proc
- Run BCP command to populate sample data (optional):
  Sample_SmartShopping/Backend/SmartShopping.Azure/SQLScript/Data/bcpin.cmd

SSDT: Visual Studio > View > SQL Server Object Explorer

Backend Services – Azure Setup

5. Create Stream Analytics job

- On Azure portal, set Input from IoTHub, named as "InputIoTHub"
- On Azure portal, set output to SQL Database table, named as "OutputSQL"
- On Azure portal, set query copied from: Sample_SmartShopping/Backend/SmartShopping.Azure/SQLScript/StreamAnalytics/StreamAnalytics_Query.sql
- Start job manually



Storing Your Secrets in Azure Websites App Settings
https://blogs.msdn.microsoft.com/cdndevs/2015/03/20/storing-your-secrets-in-azure-websites-app-settings/

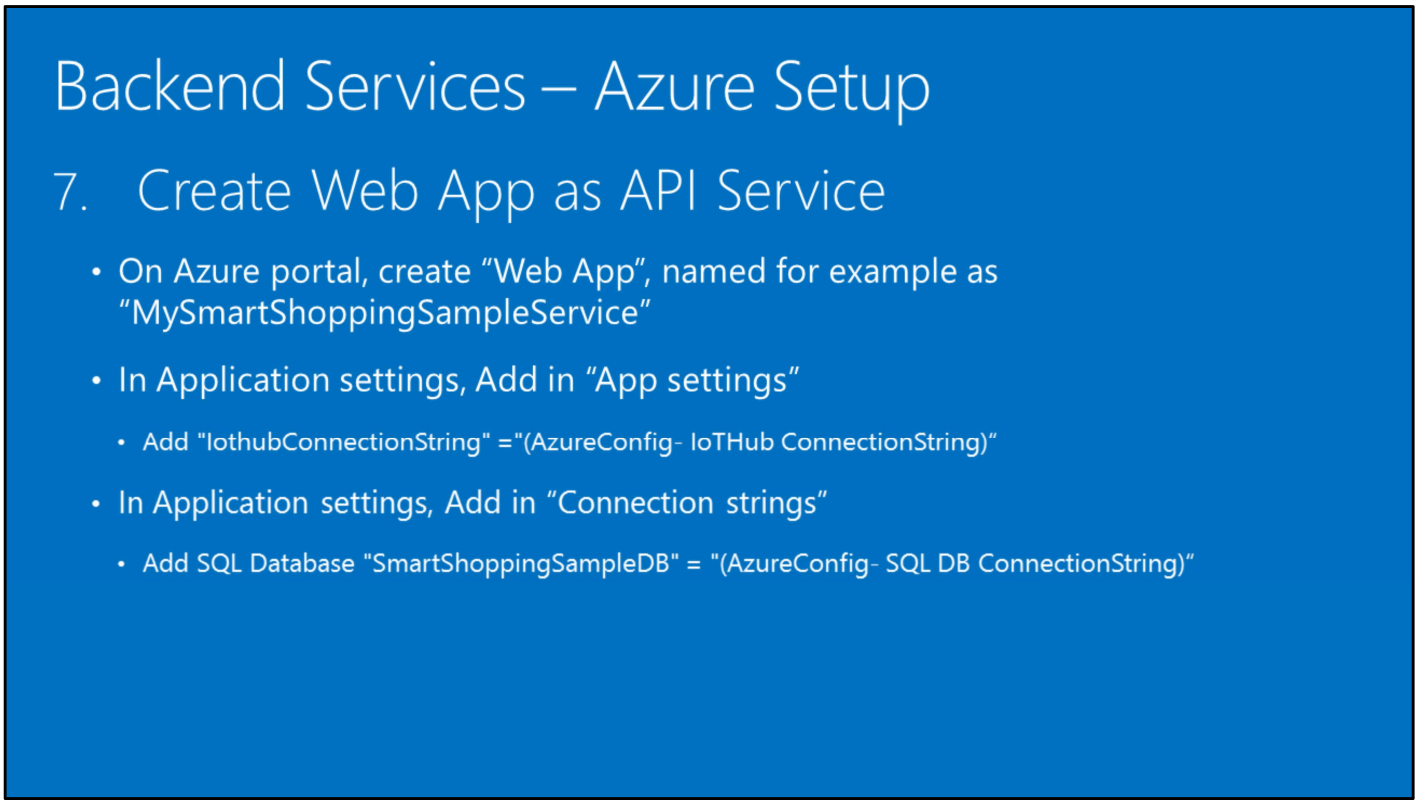
Best practices for deploying passwords and other sensitive data to ASP.NET and Azure App Service
http://www.asp.net/identity/overview/features-api/best-practices-for-deploying-passwords-and-other-sensitive-data-to-aspnet-and-azure

Authentication and authorization in Azure App Service
https://azure.microsoft.com/en-us/documentation/articles/app-service-authentication-overview/

Create a REST service using ASP.NET Web API and SQL Database in Azure App Service
https://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-rest-service-aspnet-api-sql-database/

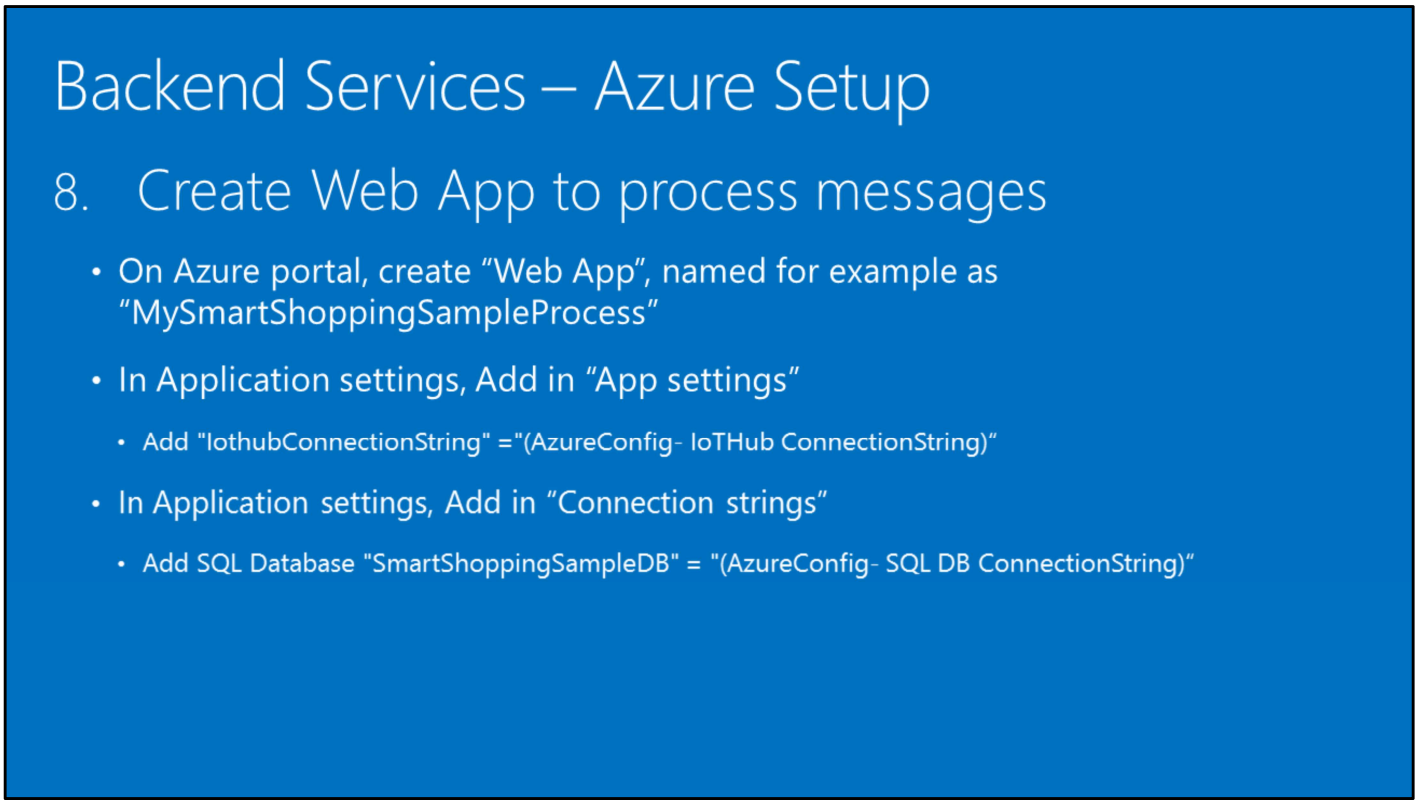
-- BaseUrl examples:
"ProcessBaseUrl" = "https://mysmartshoppingsampleprocess.azurewebsites.net"

-- PowerBI report Id, browse to PowerBI Report, Find Report ID (GUID) in URL:
https://msit.powerbi.com/groups/me/reports/<report-guid>/ReportSection



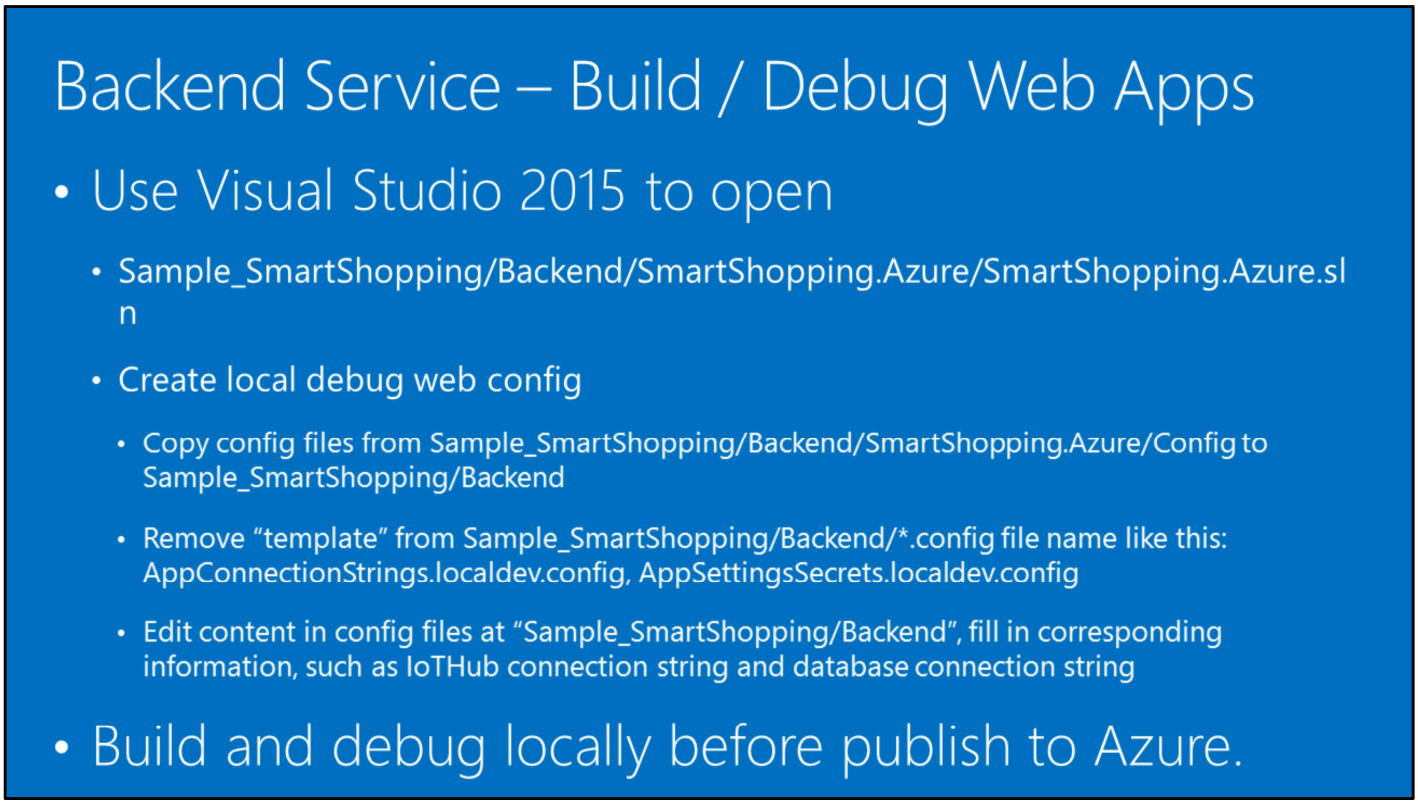
Create a REST service using ASP.NET Web API and SQL Database in Azure App Service

https://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-rest-service-aspnet-api-sql-database/



Create a REST service using ASP.NET Web API and SQL Database in Azure App Service

https://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-rest-service-aspnet-api-sql-database/

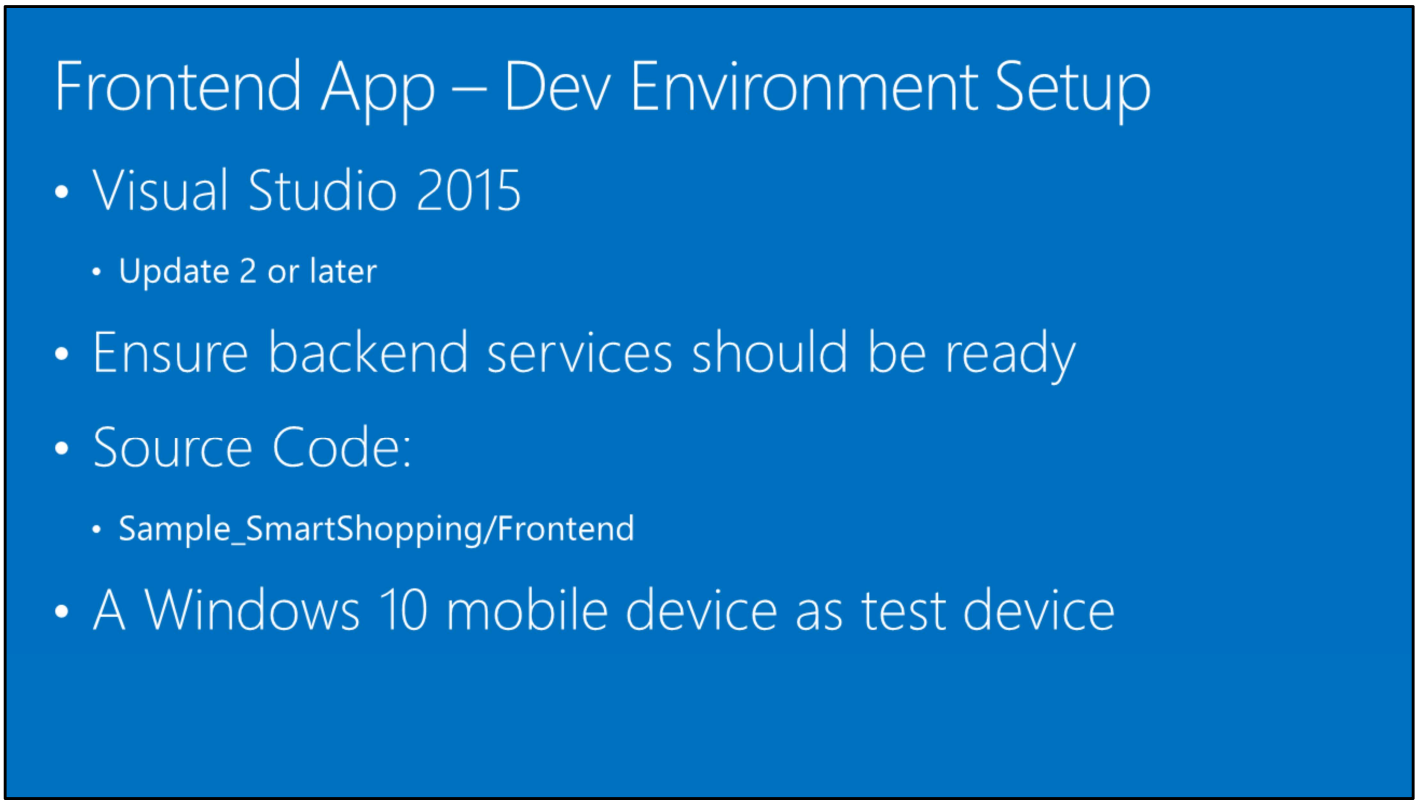


Best practices for deploying passwords and other sensitive data to ASP.NET and Azure App Service
http://www.asp.net/identity/overview/features-api/best-practices-for-deploying-passwords-and-other-sensitive-data-to-aspnet-and-azure

-- The settings in local config file are identical to those set in Azure application settings.

Backend Service – Publish Web Apps

- Each project is a Web App.
  - Publish to corresponding Azure Web App:
    - Project "SmartShoppingService" -> Azure Web App "MySmartShoppingSampleService"
    - Project "SmartShoppingProcess" -> Azure Web App "MySmartShoppingSampleProcess"
    - Project "SmartShoppingWeb" -> Azure Web App "MySmartShoppingSampleWeb"



-- This UWP app could be run on Windows phone (Windows 10) or Windows 10 tablet

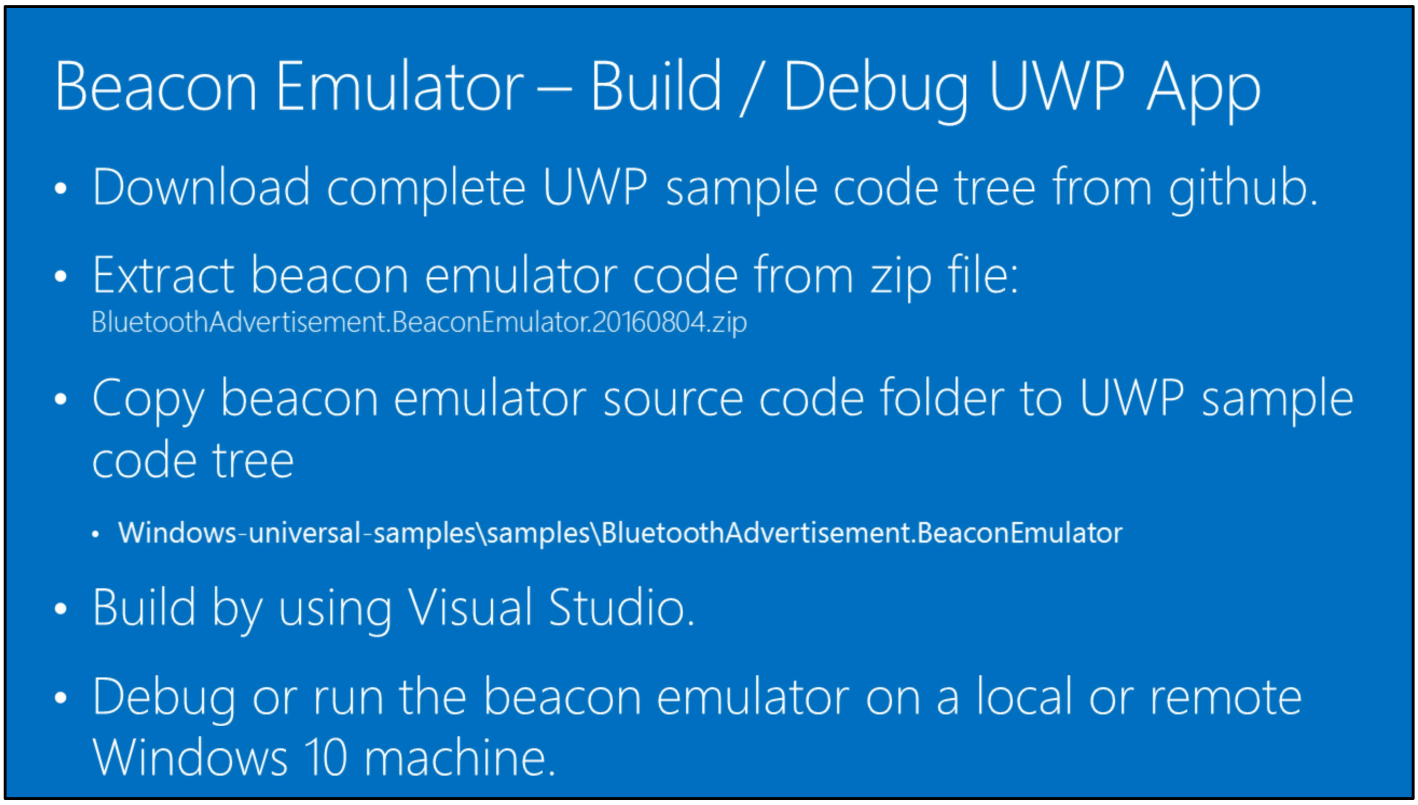
Frontend App – Build / Debug UWP App

- Use Visual Studio 2015 to open
  - Sample_SmartShopping\Frontend\SmartShopping.PhoneApp.sln
- Edit configuration file at
  - Sample_SmartShopping\FrontEnd\SmartShopping.PhoneApp\Data\SmartShoppingSample.Config.xml : "ServiceBaseUrl"
- Build "SmartShopping.PhoneApp" project.
- Connect phone device via USB.
- Debug/Deploy phone app to phone device to test.

<config Key="ServiceBaseUrl" Value="https://mysmartshoppingsampleservice.azurewebsites.net/" />

-- If target device is a tablet, the device should be connected via wired ethernet or Wifi, and using remote debugging to debug the app running on target tablet device:

https://msdn.microsoft.com/en-us/library/y7f5zaaa.aspx



UWP Sample

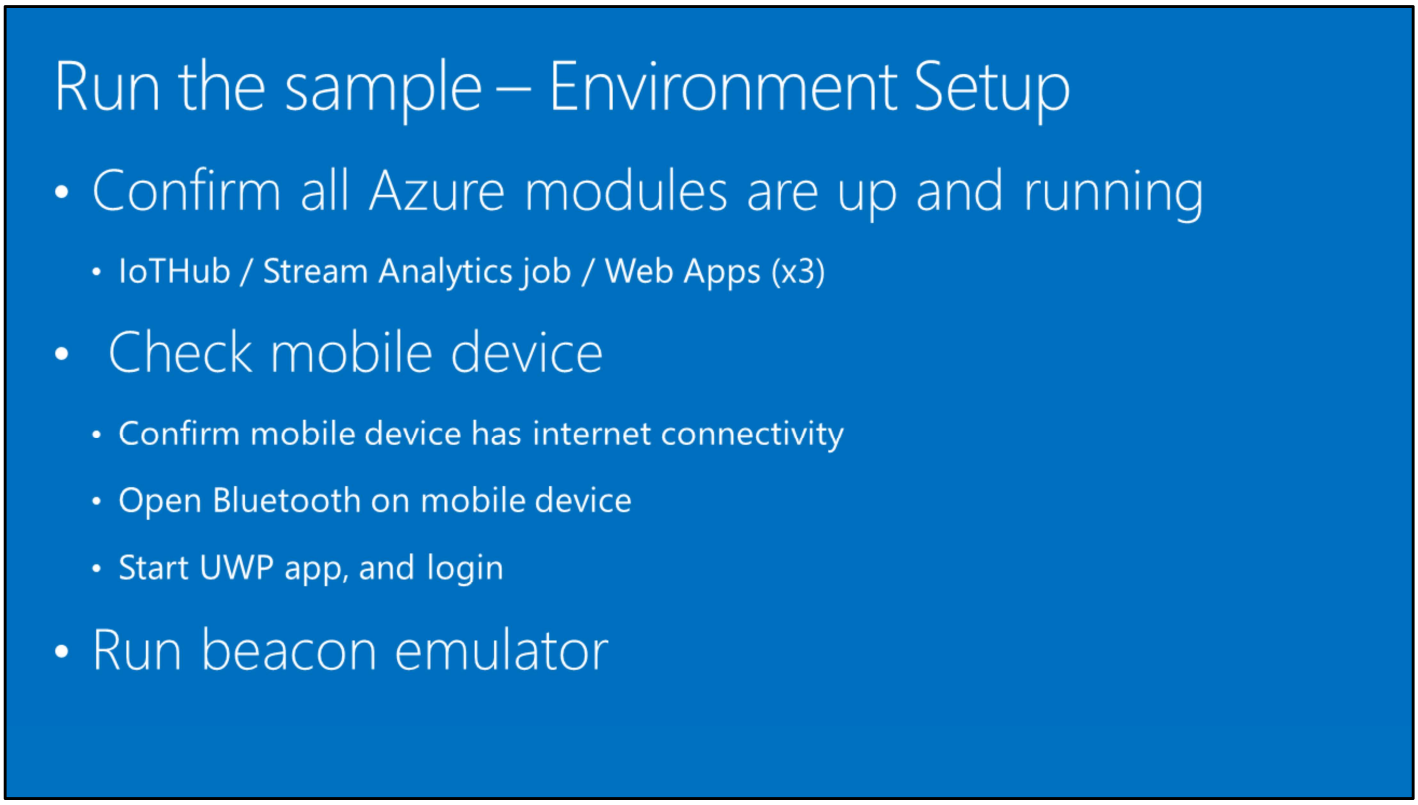
https://github.com/Microsoft/Windows-universal-samples

UWP Sample – Bluetooth Advertisement

https://github.com/Microsoft/Windows-universal-samples/tree/master/Samples/BluetoothAdvertisement

-- If your local machine is not windows 10, you need to use remote debugging. The target machine should be windows 10. The device should be connected via wired ethernet or Wifi, and using remote debugging to debug the app running on target machine.

https://msdn.microsoft.com/en-us/library/y7f5zaaa.aspx



-- You could generate UWP app packages (.appx, similar to Android apk) to facilitate app installation, so you may install the app to devices without the use of Visual Studio. However, If target device is winodws phone, we still suggest to use Visual Studio to deploy the app to phone device.

-- Please notice that this app is not going to publish to Windows Store, so when you generate the app package using

Visual Studio -> (select a project and right click to show menu) -> Store->Create App Packages

You need to choose "No" if you are asked for "Do you want to build packages to upload to the Windows Store?"

Packaging UWP apps

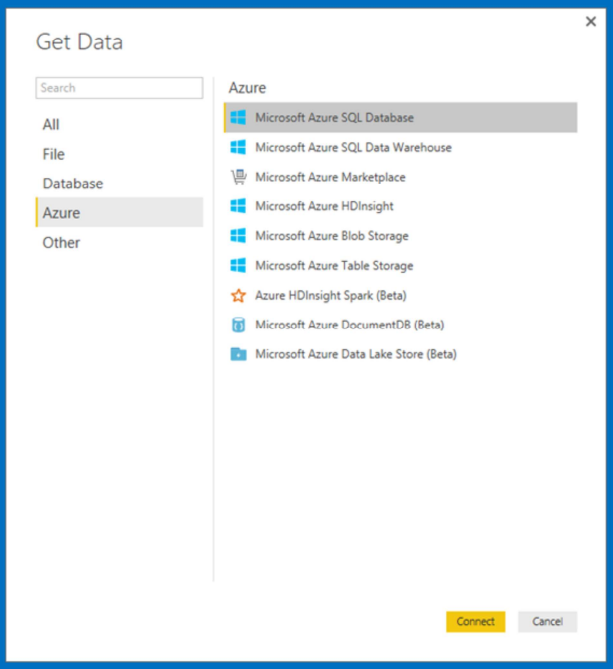
https://msdn.microsoft.com/en-us/windows/uwp/packaging/packaging-uwp-apps

Run the sample

- Put mobile device closer to beacon emulator until promotion notification arrived.
- Open web pages to verify results

Power BI – Share Report

- Dataset (Direct Query):
  - Get Data > Azure > Microsoft Azure SQL database
- Edit Report
- Publish to PowerBI Portal
  - Pin Report to Dashboard and Share
  - Find Report Id



PowerBI Portal
https://powerbi.microsoft.com/en-us/landing/signin/

Browse to PowerBI Report, Find Report ID (GUID) in URL
https://msit.powerbi.com/groups/me/reports/<report-guid>/ReportSection