

Microsoft Azure Functions

Dino Wang

tw MVC

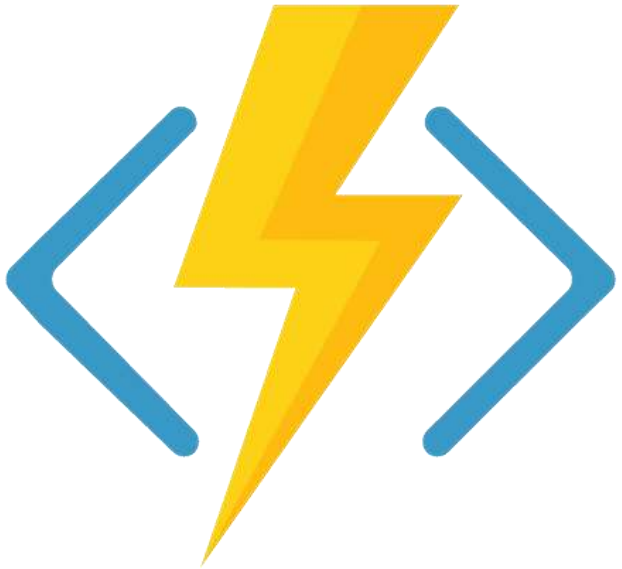
Dino Wang

- 禾盛數碼科技 hexdigits
- twMVC 共同創辦人
- 現任 Microsoft Azure MVP
- 微軟 TechDay & MSDN 講師
- 合著「ASP.NET MVC 5 網站開發美學」
- https://www.facebook.com/dinowang
- dino@hexdigits.com



Agenda

- Azure Functions 概觀
- 瞭解 Function App 結構
- C# Developer 須知
- 應用場景範例



什麼是 Azure Functions

- Azure 提供的雲端計算服務
- 非常易於開發的、事件驅動的架構
- 第一個能 pay-per-use 的雲端計算服務
- 延伸 PaaS 服務
- Serverless 架構

適合 Azure Functions 做的事情

- 定時排程工作
- Azure 服務的各種事件處理
- 其他 SaaS 服務的事件處理，例如 OneDrive
- Single Page App
- 行動應用的後端系統
- IoT 的即時流處理
- Bot

關於名稱

- Azure Functions

  微軟的產品名稱

- Function App

  我們所開發的應用程式實例

目前的 Azure Functions

- 2016 年 5 月於 //Build 大會中釋出 Preview
- 目前為 Preview 0.3 版

  7 月有個 0.2 → 0.3 升級改版，囧無 Release Note
- 活躍的討論區
  - http://stackoverflow.com/questions/tagged/azure-functions
  - https://social.msdn.microsoft.com/Forums/azure/en-US/home?forum=AzureFunctions

Azure Functions 計價方式

- 執行時間 以及 執行次數。
- 每月 一百萬次 或 總執行時數於 400,000 GB-s 內免費。

  Gibabyte Seconds
- 在 Preview 期間不提供 SLA。
- 或以 App Service 的方式執行 Function App。
- https://azure.microsoft.com/en-us/pricing/details/functions/

Serverless 架構

- "Serverless" 沒有伺服器也太神了吧？
  - 不，你的 code 當然都是在機器上執行的。
- 意味一個不需要你花心思的執行環境。
  - 上傳你的程式碼 (Bring your own code) 就是了。
  - 不需要考慮如何管理他們，煩惱全交給雲端。

Azure Functions 兩個重點

- **Triggers**
  - 事件驅動的來源
  - 來源事件訊息
  - 所有的 Function 必有一個 Trigger 事件作為起始
- **Binding**
  - 綁定的資源，包含事件訊息
  - 包含輸入、輸出的資源
  - 利用預先設定綁定的雲端資源，簡化開發專注流程

Azure Functions Bindings

類型	相關服務	Trigger	Input	Output
Schedule	Azure Functions	✔		
HTTP Rest/WebHook	Azure Functions	✔		✔
Blob	Azure Storage	✔	✔	✔
Queue	Azure Storage	✔		✔
Table	Azure Storage		✔	✔
NoSQL	Azure DocumentDB		✔	✔
Streams	Azure EventHub	✔		✔
Push Notifications				✔

Azure Functions 特性

- 整合

  可輕易與 Azure 多項服務進行整合

- 多種佈署方式

  整合 GitHub、VSTS、Git、BitBucket、OneDrive、Dropbox、… 為部署來源

- 可選的授權服務提供者

  Azure AD、Facebook、Google、Twitter、Microsoft Account (OAuth providers)

Azure Functions 特性

- 開發語言

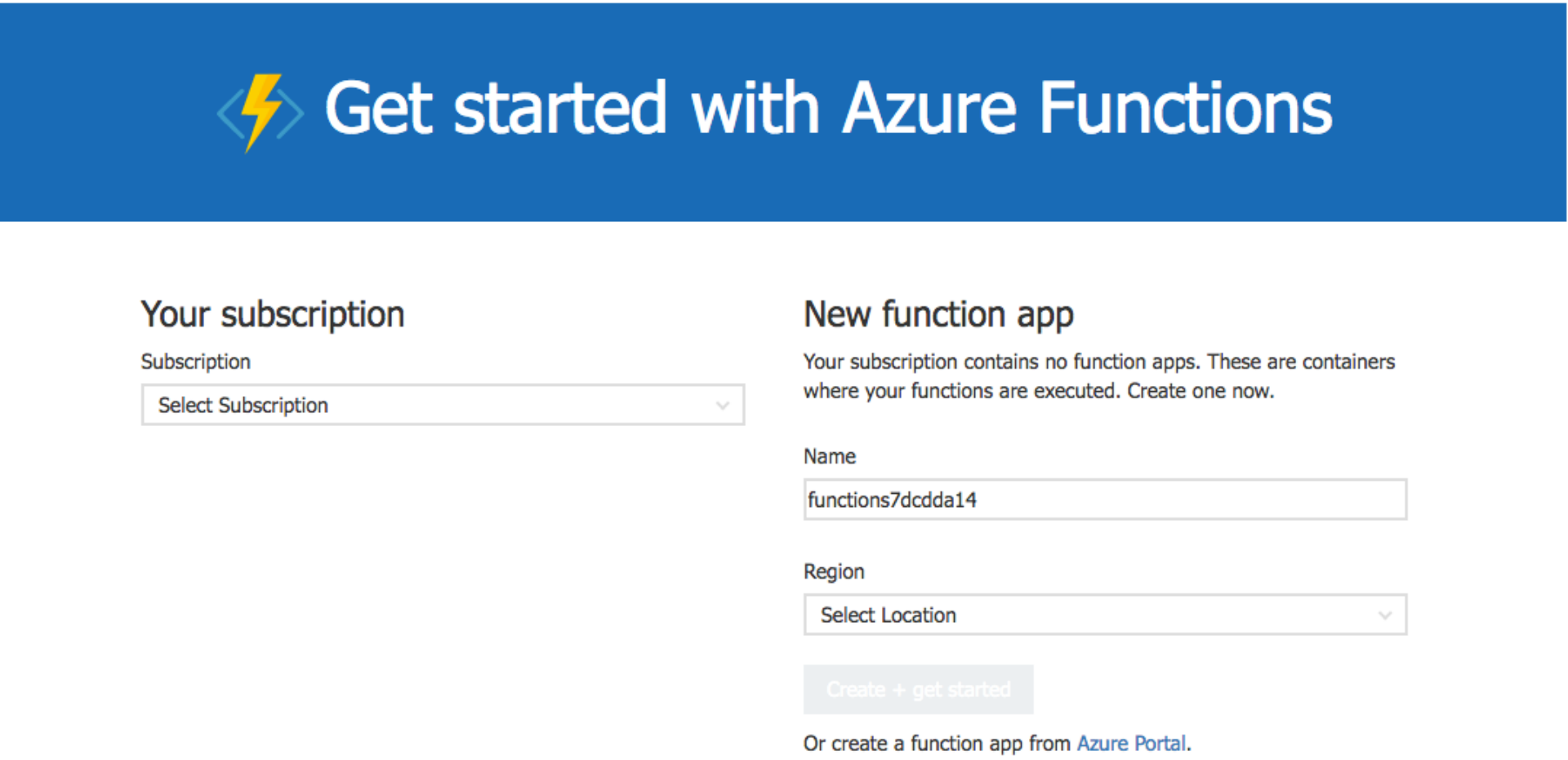
  C#、Node、Python、F#、PHP、Batch、Bash

- 可定義所需安裝元件

  NuGet、NPM

新增 Function App

- https://functions.azure.com



建立 Function App

- http://portal.azure.com



注意 App Service Plan 的選擇

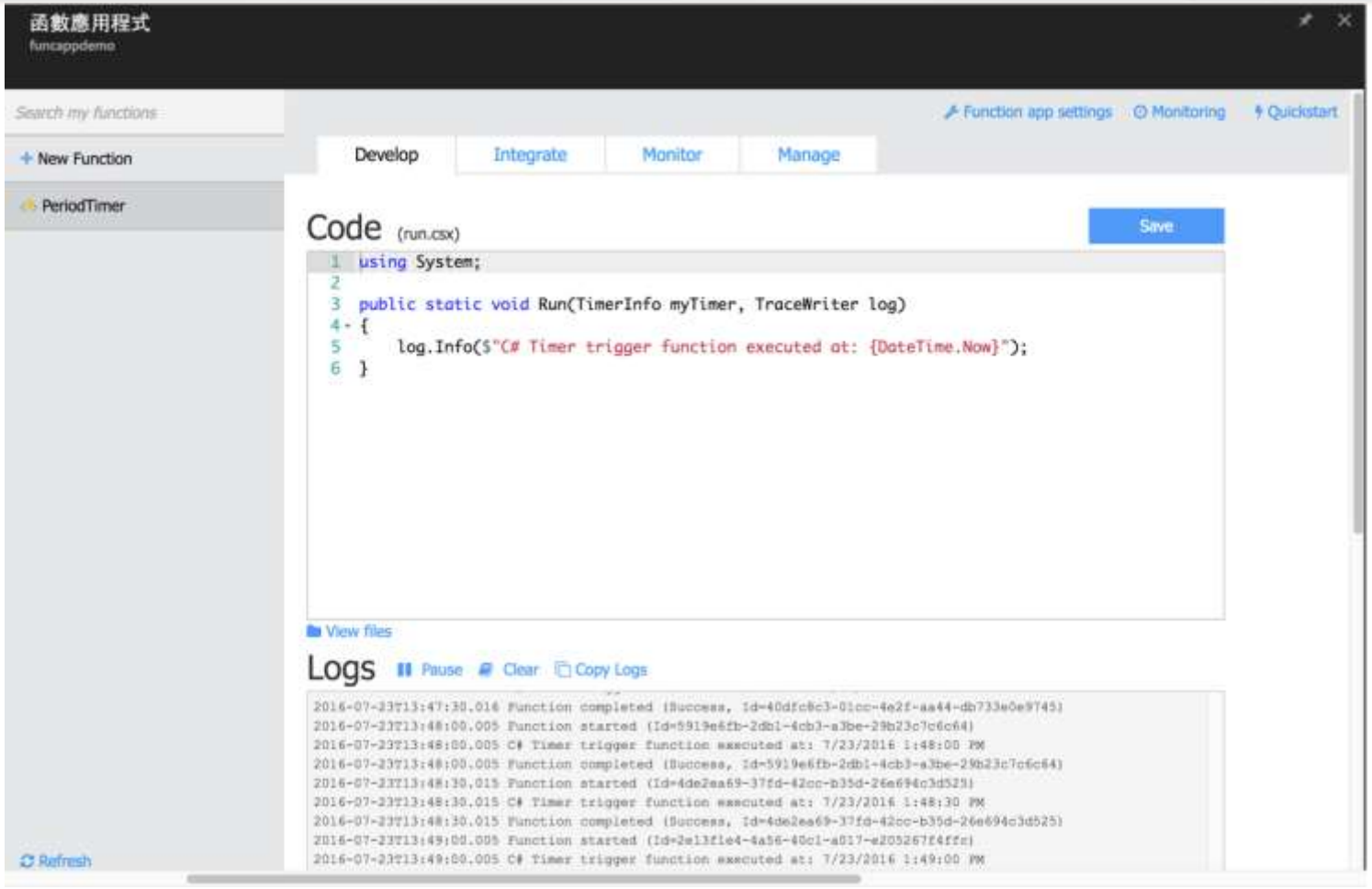
- 傳統 Classic

  採用專責 App Service VM 執行 Function App。

- 動態 Dynamic

  由 Azure 自動分配指定執行 Function App 的實例，盡可能地利用 Azure 的資源，以執行時間計算費用 (pay-per-use)。

Function App 控制台





第一個 Function App

Hello World

關於 TimerTrigger

- CRON 表示式 (Azure WebJob 與 Function App 相同)
- 注意 CRON 表示式由 6 個欄位組成:

```
{second} {minute} {hour} {day} {month} {day of the week}
```

- 支援的標記符號有 * , - /

http://goo.gl/Fqn58F

關於 TimerTrigger

- 一些 CRON 的線上服務
  - http://www.cronmaker.com/
  - http://crontab.guru/
  - http://crontab-generator.org/
- 以上都只提供 5 個欄位的 CRON 表示式，沒有到秒級

OK! 在 Hello World 裡學習到

- 人工建立第一個 Function 以及設定的界面
- Trigger、Binding 在 Function 中的角色
- 以 C# 為例的程式本體 run.csx (C# Script)

進一步地探索 Function App

- Function App 是基於 Azure App Service 的架構
- 你可以用熟知的 Kudu 服務來瞭解 Function App
  - 專案結構
  - 執行紀錄
  - API Keysyo



demo

Explore Function App with Kudu

關於 API Keys

- 由 App Service 中取得 API Key

  D:\home\data\Functions\secrets\host.json

```
{
  "masterKey": "                                                    ",
  "functionKey": "                                                  "
}
```

- API Key 的作用

  能驅動 Function app 中所有 function

  但不包含 disabled function

關於 API Keys

- 自訂 API Key

  可以為個別的 function 定義額外的 API Key，做法是在 secret (D:\home\data\Functions\secrets\) 目錄下新增與 function 同名的 JSON 檔案。

- D:\home\data\Functions\secrets\{FuncName}.json

```
{
  "key":"                                                                    "
}
```

OK! 在 Hello World 裡學習到

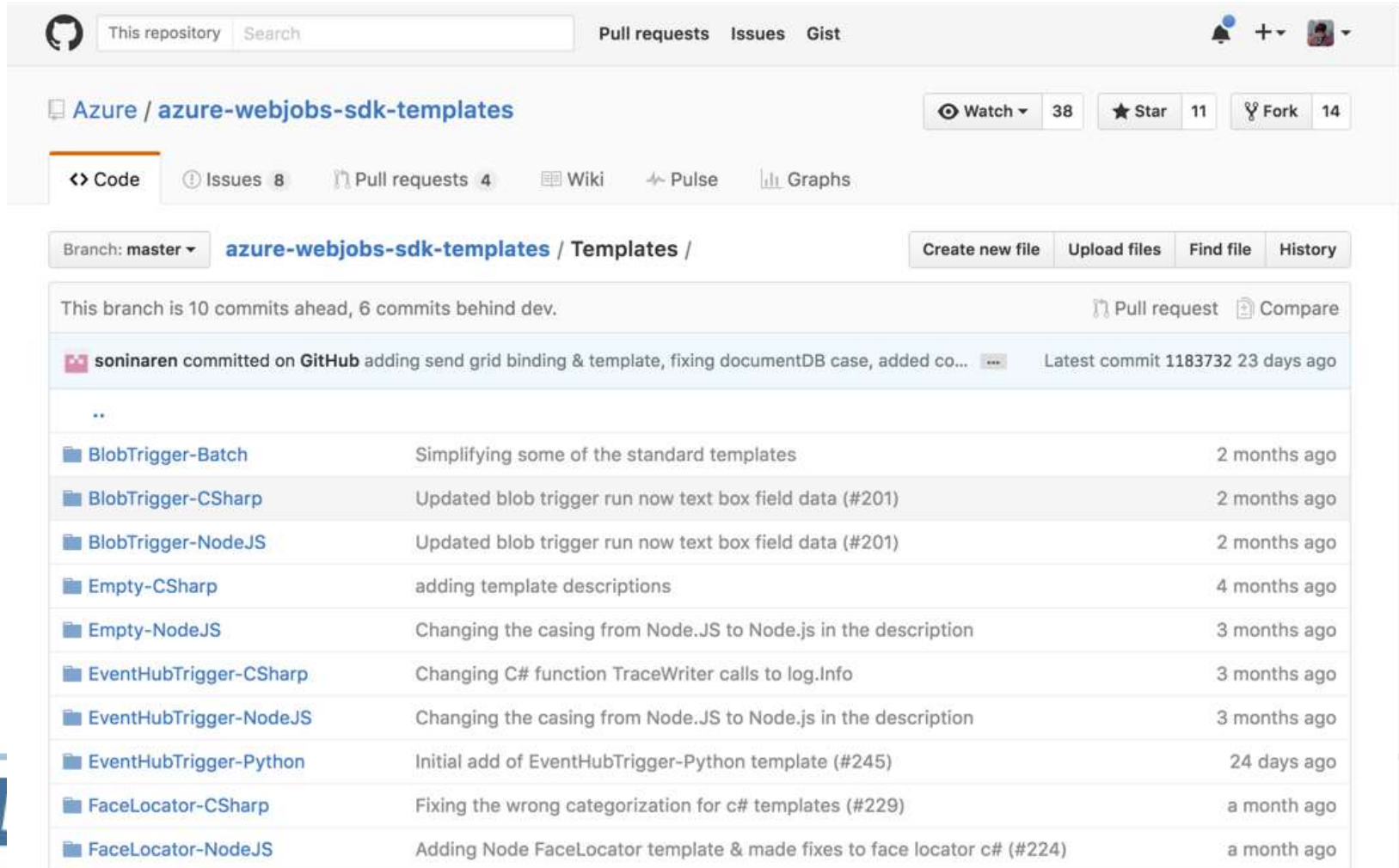
- 人工建立第一個 Function 以及設定的界面
- Trigger、Binding 在 Function 中的角色
- Function 程式本體 (以 C# Scripts 為例)
- function.js →定義了 Trigger、Binding

在自己的編輯器裡寫 Function App

- 剛開始寫時 function.json 真的是個很卡的東西 …

在自己的編輯器裡寫 Function App

- 參考官方的 templates



在自己的編輯器裡寫 Function App

- Yeoman generator 是你的好朋友 

First, install Yeoman and generator-azurefunctions using npm (we assume you have pre-installed node.js).

```
npm install -g yo
npm install -g generator-azurefunctions
```

Then generate your new Azure Functions project:

```
yo azurefunctions
```

.NET Developer

C#

- 了解 C# Script (.csx) 的開發要領
- 如何由 Azure Portal 進行開發

  如同開發 Period Timer 範例一樣使用線上編輯器
- 如何由 Source Control (如 GitHub) 進行開發

  由版本控制部署的 Function 無法在 Portal 進行修改

  當然，也有例外

Azure Functions C# developer reference
https://azure.microsoft.com/en-us/documentation/articles/functions-reference-csharp/

Function App from CI

C#

- 建立 GitHub repository (可用 private)
- 本地端產生專案目錄，例如 demo，並建立本地 repo.
- 使用 yeoman 產生一個 Function
- 完成 Function 所需的程式碼
- Commit 以及 push 到 GitHub repo.
- 設定 Function App 自動由 GitHub 部署



demo

Function App from CI

VSCode / Intellisense

C#

- 如果你的 VSCode 無法順利提供 Intellisense，試著在編輯期間暫時將副檔名 Run.csx 更名為 Run.cs 即可。
- 若您使用 ATOM 編輯器亦相同，問題源自於提供語法解析 OmniSharp 套件，記得安裝。
- 送上版控前請確保 Run 副檔名為 .csx。

Function App Structures

C#

- 獨立 Function 目錄

  有 function.json

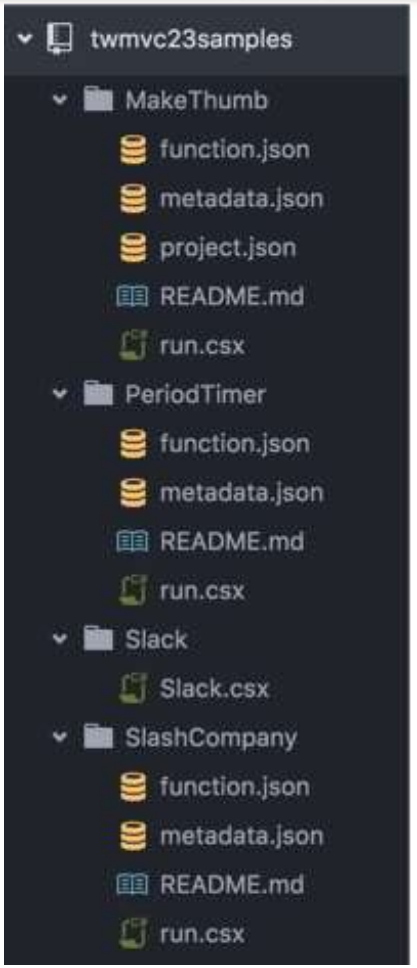
  有 run.csx

- 可以存在非 Function 的目錄

  沒有 function.json 的目錄

  存放共用類別或其他資源檔案

  如右圖中的 Slack\



整合情境



demo

以 Function App 接力縮圖

C# Script

- #r "AssemblyName"

  自 GAC 增加參考的組件

- #load "AnotherScript"

  載入其他的 C# Script 檔案

- C# Script 內不能含有 Namespace

注意句尾
不必加上 ";"

整合情境 2

凡事都有個 Bot



偽 Slack Bot Session

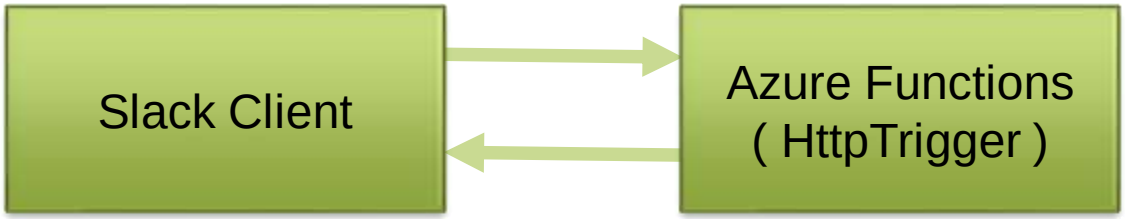
合適的使用情境

- Bot 經常不是全時服務
- 以使用次數、執行時間計算的 Azure Functions 很搭

整合 Slack 的多種方式

- Outgoing Webhooks
- Incoming Webhooks
- Slash Commands
- Bot Users

Slack Command + Functions



Slash Command + Functions

- Slash Command 的形式

  /{命令名稱} {命令參數}

- Slash Command 送出參數至定義的 Function URL (HTTP Post)

- 回應 Slash Command

Azure Functions URL

- 一個 Function 對外暴露的 URL 結構

```
https://{function app name}.azurewebsites.net/api/{function name}
```

- 另外應加上 API Key



demo

Slack + Azure Functions

今天的 LAB

- GitHub Repository

  https://github.com/dinowang/twmvc23samples

- Fork repo 然後部署到你自己的 Azure!

史上最強的注音輸入法



千呼萬喚

MAC 平台上，最聰明的注音輸入法
現在立即前往購買 ！

tw MVC
樂在Coding 愛上MVC

完全支援

請放心在 Windows 8 上使用 新自然輸入法 10 系統升級免煩惱，平板模式也沒問題！

Blog 是記錄知識的最佳平台



感謝 Jetbrains 贊助贈品



ReSharper

https://www.jetbrains.com/resharper/

感謝 OzCode 贊助贈品



http://www.oz-code.com/

感謝 ALIVE 贊助贈品



https://comealive.io/

業界師資、實戰教學



http://skilltree.my

謝謝各位

http://mvc.tw

- 本投影片所包含的商標與文字皆屬原著作者所有。
- 本投影片使用的圖片皆從網路搜尋。
- 本著作係採用姓名標示-非商業性-相同方式分享 3.0 台灣授權。閱讀本授權條款，請到 http://creativecommons.org/licenses/by-nc-sa/3.0/tw/，或寫信至Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.