



中原大學 雲端計算平台實務

12/03-作業報告

Create serverless applications

資訊四乙 10727211 林彥輝

授課教師：鍾武君 教授

中華民國一一〇年十二月

# 1. Learning Path Intro

## Create serverless applications

<https://docs.microsoft.com/en-us/learn/patterns/create-serverless-applications/>

## 2. Summary Homework Assignment

### Model 2: Create serverless logic with Azure Functions

#### 1. Create a function app

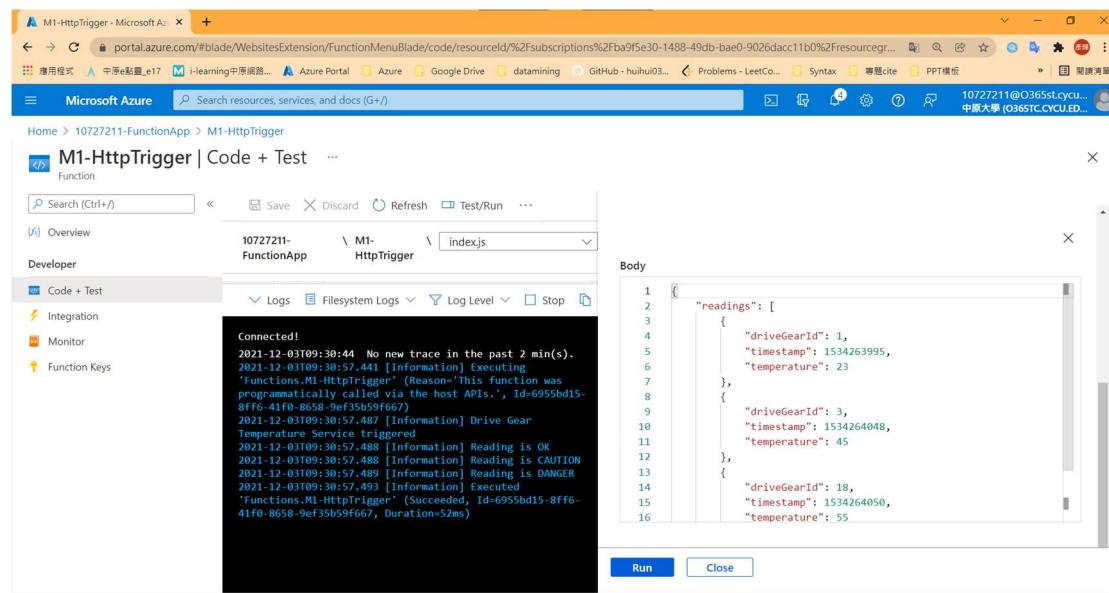
The screenshot shows the Microsoft Azure portal interface. A new function app named "M1-HttpTrigger" is being created under the "Function" category. The "Overview" tab is selected, displaying basic information such as the function app name, status (Enabled), resource group (110-1-CS456), subscription (中原大學), and subscription ID. There are tabs for "Code + Test", "Integration", "Monitor", and "Function Keys". Action buttons include "Enable", "Disable", "Delete", "Get Function Url", and "Refresh". A "JSON View" link is also present.

#### 2. Test Function - 1

The screenshot shows the Azure Functions test interface for the "M1-HttpTrigger" function. The "Logs" tab is active, showing the logs for the function's execution. The log output indicates a successful execution of the function with a duration of 220ms. Below the logs, a terminal window shows the command used to trigger the function via curl, followed by a message confirming the successful execution.

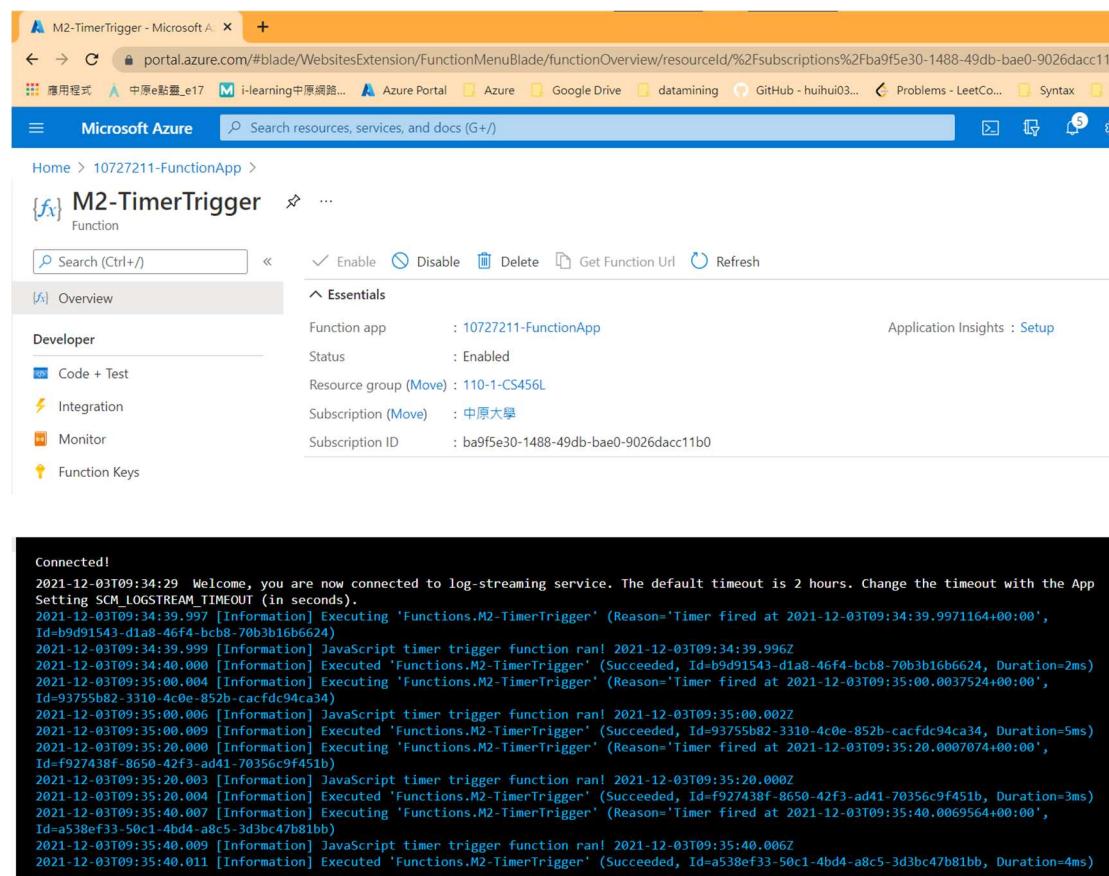
The screenshot shows the Azure Functions test interface for the "M1-HttpTrigger" function. The "Output" tab is active, showing the HTTP response code (200 OK) and content. The response content is a placeholder message: "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response."

### 3. Test Function – 2 (IOT temperature)



## Model 3: Execute an Azure Function with triggers

### 1. Create TimerTrigger & Test



## 2. Create HttpTrigger & Test

The screenshot shows the Microsoft Azure portal interface. At the top, there is a navigation bar with links to 'portal.azure.com', 'Azure Portal', 'Azure', 'Google Drive', 'datamining', and 'GitHub - huihui03...'. Below the navigation bar, the main header says 'Microsoft Azure' with a search bar that reads 'Search resources, services, and docs (G+/)'. The breadcrumb navigation shows 'Home > 10727211-FunctionApp > M2-HttpTrigger'.

The main content area displays the 'M2-HttpTrigger' function details. It includes a 'Search (Ctrl+/' input field and buttons for 'Enable', 'Disable', 'Delete', 'Get Function Url', and 'Refresh'. The 'Developer' section lists 'Code + Test', 'Integration', 'Monitor', and 'Function Keys'. The 'Essentials' section provides the following information:

Function app	: <a href="#">10727211-FunctionApp</a>
Status	: Enabled
Resource group (Move)	: 110-1-CS456L
Subscription (Move)	: 中原大學
Subscription ID	: ba9f5e30-1488-49db-bae0-9026dacc11b0

Below this, there is a browser window showing the function's URL: <https://10727211-functionapp.azurewebsites.net/api/M2-HttpTrigger?>. The page content is: "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response."

Finally, another browser window shows the result of a query parameter: <https://10727211-functionapp.azurewebsites.net/api/M2-HttpTrigger?name=Pooh>. The page content is: "Hello, Pooh. This HTTP triggered function executed successfully."

### 3. Create BlobTrigger & Test

The screenshot shows the Azure portal interface with the following details:

**M2-BlobTrigger - Microsoft Azure** | **Upload blob - Microsoft Azure**

**Microsoft Azure** | Search resources, services, and docs (G+/)

**M2-BlobTrigger** Function

**Overview** (selected) | **Developer** | **Code + Test** | **Integration** | **Monitor** | **Function Keys**

**Essentials**

Function app	: 10727211-FunctionApp
Status	: Enabled
Resource group (Move)	: 110-1-CS456L
Subscription (Move)	: 中原大學
Subscription ID	: ba9f5e30-1488-49db-bae0-9026dacc11b0

Application Insights : [Setup](#)

**Upload blob**

Drag and drop files here  
or  
[Browse for files](#)

Overwrite if files already exist

**Advanced**

**Upload**

**Current uploads**

ReadMe.md	✓ 1.42 KiB / 1.42 KiB
GitHubDesktop.exe	✓ 383.67 KiB / 383.67 KiB

Dismiss: [Completed](#) [All](#)

**Logs**

Connected!

```
2021-12-03T09:42:24 Welcome, you are now connected to log-streaming service. The default timeout is 2 hours. Change the timeout with the App Setting SCM_LOGSTREAM_TIMEOUT (in seconds).
2021-12-03T09:43:02.838 [Information] Executing 'Functions.M2-BlobTrigger' (Reason='New blob detected: samples-workitems/GitHubDesktop.exe', Id=eb56fb7b-11d7-4cf6-9000-6f73175da9bc)
2021-12-03T09:43:02.838 [Information] Trigger Details: MessageId: 443f53e0-ef8c-424c-9fdf-978f0e083874, DequeueCount: 1, InsertionTime: 2021-12-03T09:43:02.000+00:00, BlobCreated: 2021-12-03T09:43:00.000+00:00, BlobLastModified: 2021-12-03T09:43:00.000+00:00
2021-12-03T09:43:02.849 [Information] JavaScript blob trigger function processed blobblob: samples-workitems/GitHubDesktop.exeBlob Size: 392880 Bytes
2021-12-03T09:43:02.851 [Information] Executed 'Functions.M2-BlobTrigger' (Succeeded, Id=eb56fb7b-11d7-4cf6-9000-6f73175da9bc, Duration=44ms)
```

# Model 4: Chain Azure Functions together using input and output bindings

## 1. Create a HttpTrigger

The screenshot shows the Microsoft Azure portal interface for the 'M3-HttpTrigger' function. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main content area displays the 'Overview' tab for the function app. Key details shown include:

- Function app: 10727211-FunctionApp
- Status: Enabled
- Resource group (Move): 110-1-CS456L
- Subscription (Move): 中原大學
- Subscription ID: ba9f5e30-1488-49db-bae0-9026dacc11b0

## 2. Create a cosmosDB

The screenshot shows the Microsoft Azure portal interface for the '10727211cosmosdb' Azure Cosmos DB account. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main content area displays the 'Overview' tab for the account. Key details shown include:

- Status: Online
- Resource group (Move): 110-1-CS456L
- Subscription (Move): 中原大學
- Subscription ID: ba9f5e30-1488-49db-bae0-9026dacc11b0
- Total throughput limit: No total throughput limit

A deployment status message at the top right indicates: Deployment 'Microsoft.Azure.CosmosDB-20211203T175244' to resource group '110-1-CS456L' was successful.

## 3. Add test data

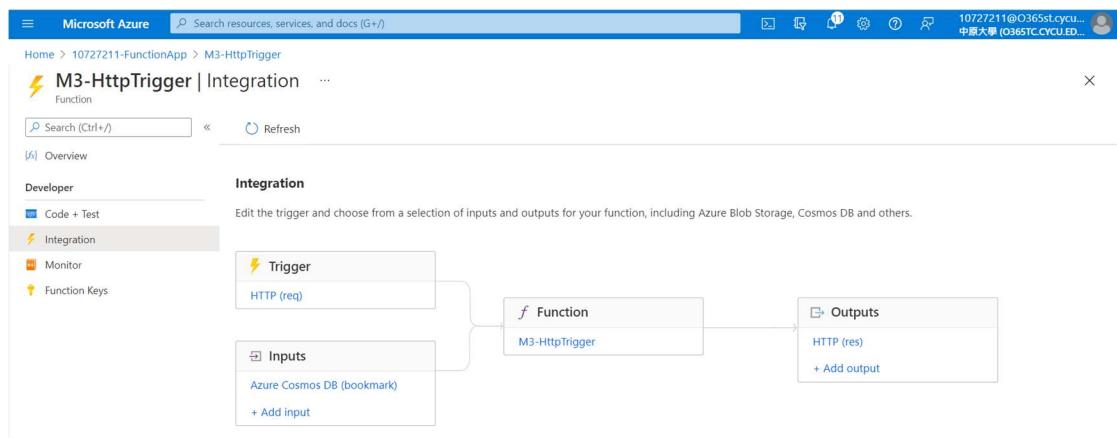
The screenshot shows the Microsoft Azure portal interface for the '10727211cosmosdb' Data Explorer. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main content area displays the 'Data Explorer' tab for the account. Key details shown include:

- Container: func-io-learn-db
- Items: docs, portal, learn, marketplace, blog

The 'docs' item is selected, and its JSON representation is displayed in the preview pane:

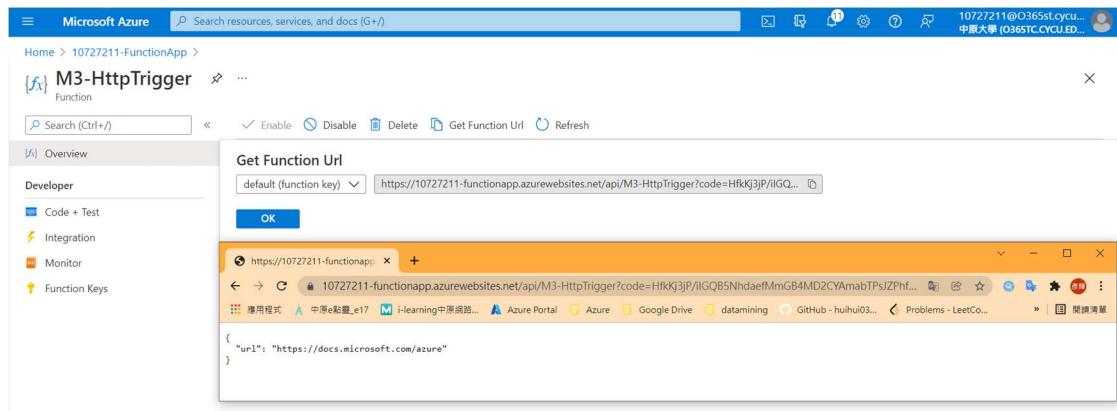
```
1  [
2   {
3     "id": "docs",
4     "url": "https://docs.microsoft.com/azure",
5     "_rid": "Uk0jNbdAggBAAAAAAA4AaA=",
6     "_self": "dbs/Uk0jNbdAggB/colls/ie0jNbdAggB/docs/Ue0jNbdAggBAAAAAAA4AaA=",
7     "_etag": "V:0000750b-0000-0000-0000-61a9eb660000",
8     "_attachments": "attachments/",
9     "_ts": 1638525798
}
```

## 4. Add Azure cosmosDB binding



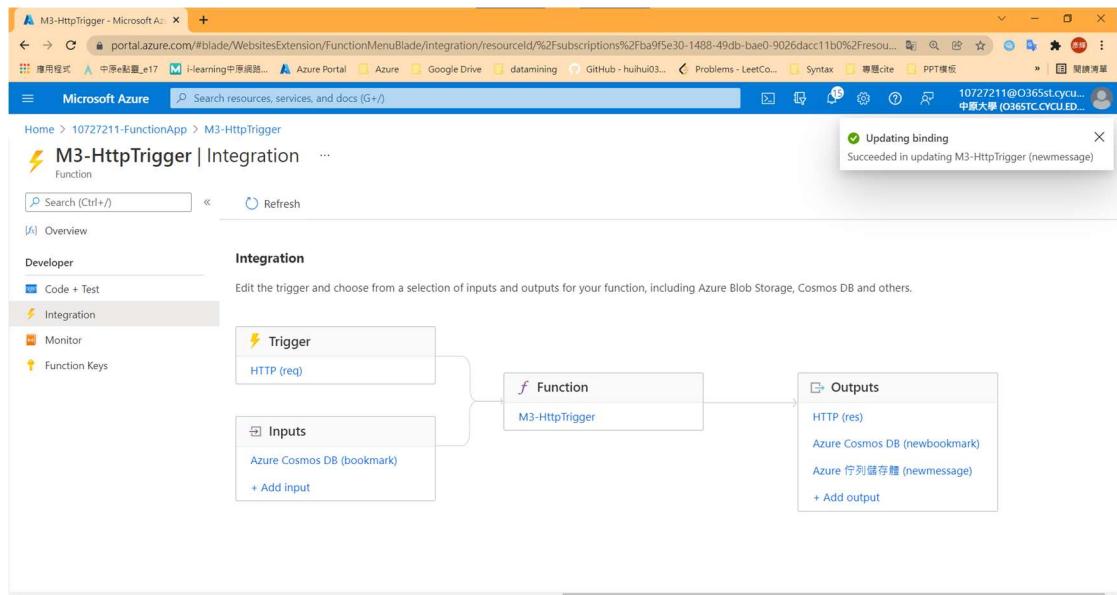
The screenshot shows the Azure portal's 'Integration' blade for the 'M3-HttpTrigger' function. The 'Integration' tab is active. On the left, there are tabs for 'Developer' (selected), 'Code + Test', 'Integration' (selected), 'Monitor', and 'Function Keys'. The main area contains a flowchart: 'Trigger (HTTP req)' points to 'Function (M3-HttpTrigger)', which then points to 'Outputs (HTTP res)'. Under 'Inputs', 'Azure Cosmos DB (bookmark)' is listed with a '+ Add input' button. The URL in the browser bar is <https://10727211-functionapp.azurewebsites.net/api/M3-HttpTrigger?code=HfkKj3jP/lGQB5NhdaefMmGB4MD2CYAmabTPsjZPhf...>.

## 5. Result



The screenshot shows the Azure portal's 'Overview' blade for the 'M3-HttpTrigger' function. The 'Get Function Url' section shows the URL: <https://10727211-functionapp.azurewebsites.net/api/M3-HttpTrigger?code=HfkKj3jP/lGQB5NhdaefMmGB4MD2CYAmabTPsjZPhf...>. Below this, a browser window is open at the same URL, displaying a JSON response: { "url": "https://docs.microsoft.com/azure" }.

## 6. Add an Azure Queue Storage and cosmosDB output binding



The screenshot shows the Azure portal's 'Integration' blade for the 'M3-HttpTrigger' function. The 'Integration' tab is active. The 'Outputs' section now includes 'HTTP (res)', 'Azure Cosmos DB (newbookmark)', and 'Azure 行列儲存體 (newmessage)'. A message box indicates 'Updating binding' and 'Succeeded in updating M3-HttpTrigger (newmessage)'. The URL in the browser bar is <https://10727211-functionapp.azurewebsites.net/api/M3-HttpTrigger?code=HfkKj3jP/lGQB5NhdaefMmGB4MD2CYAmabTPsjZPhf...>.

## 7. Push message to queue storage

The screenshot shows the Microsoft Azure Function blade for the 'M3-HttpTrigger' function. The code editor displays the 'index.js' file with the following content:

```
// Create a JSON string of our bookmark.
var bookmarkString = JSON.stringify({
  id: req.body.id,
  url: req.body.url
});

// Write this bookmark to our database.
context.bindings.newbookmark = bookmarkString

// Push this bookmark onto our queue for further processing.
context.bindings.messageQueue = { id: 'github', url: 'https://www.github.com' }
```

The 'Test/Run' tab is selected, showing the 'HTTP response code' as 200 OK and the 'HTTP response content' as 'bookmark added!'. Below the code editor, the logs show the execution of the function with the message 'bookmark added!'.

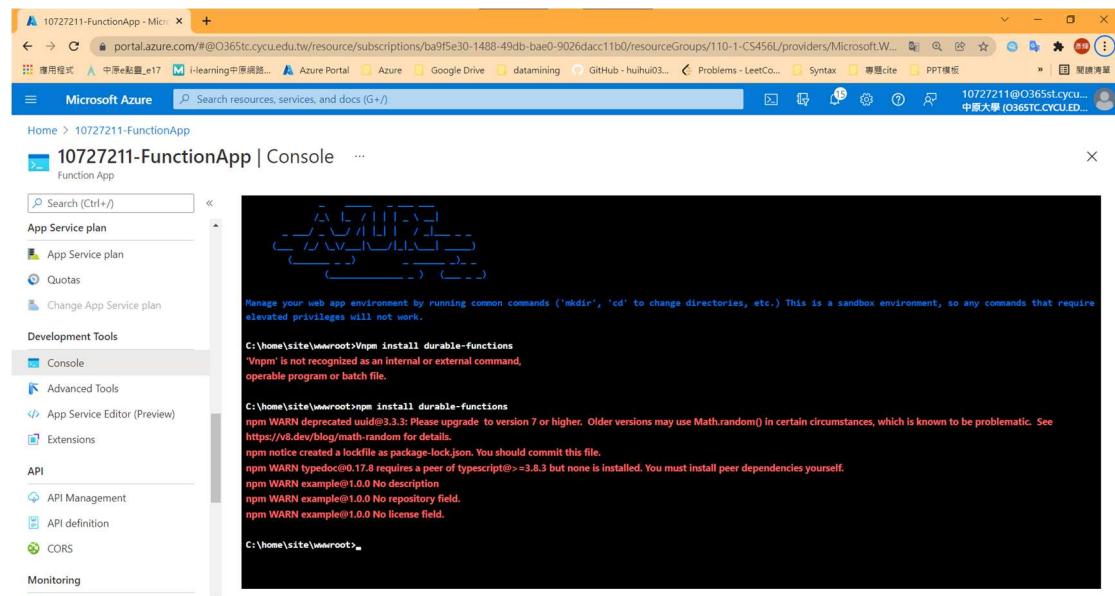
## 8. Result

The screenshot shows the Microsoft Azure Queue storage blade for the 'bookmarks-post-process' queue. The queue table displays one message:

Id	Message text	Insertion time	Expiration time	Dequeue count
e0183ec5-5e44-43f2...	{ "id": "github", "url": "https://www.github.com" }	12/3/2021, 6:15:48 PM	12/10/2021, 6:15:48 PM	0

# Model 5: Create a long-running serverless workflow with Durable Functions

## 1. Install the durable-functions npm package



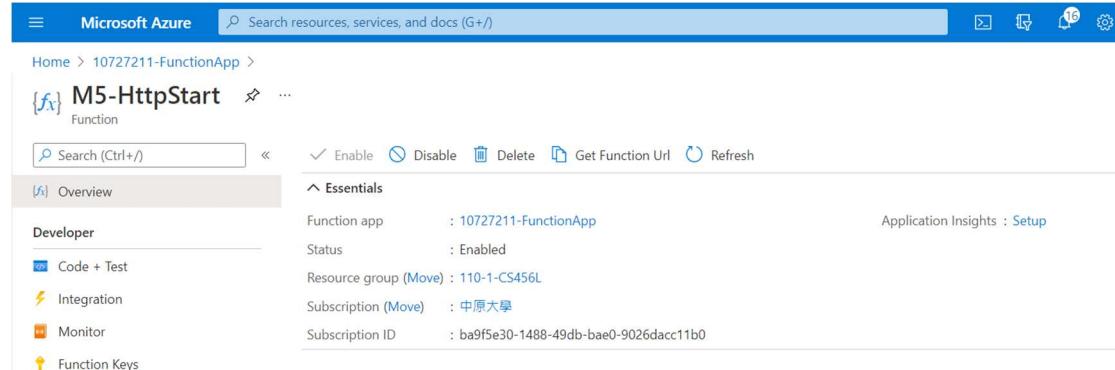
The screenshot shows the Microsoft Azure portal's console interface for a Function App named "10727211-FunctionApp". The left sidebar lists "Development Tools" with "Console" selected. The right pane displays a terminal window with the following command and output:

```
C:\home\site\wwwroot>npm install durable-functions
'npm' is not recognized as an internal or external command,
operable program or batch file.

C:\home\site\wwwroot>npm install durable-functions
npm WARN deprecated uid@3.3.3: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blogs/math-random-for-details.
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN typedoc@0.17.8 requires a peer of typescript@>=3.8.3 but none is installed. You must install peer dependencies yourself.
npm WARN example@1.0.0 No description
npm WARN example@1.0.0 No repository field.
npm WARN example@1.0.0 No license field.

C:\home\site\wwwroot>
```

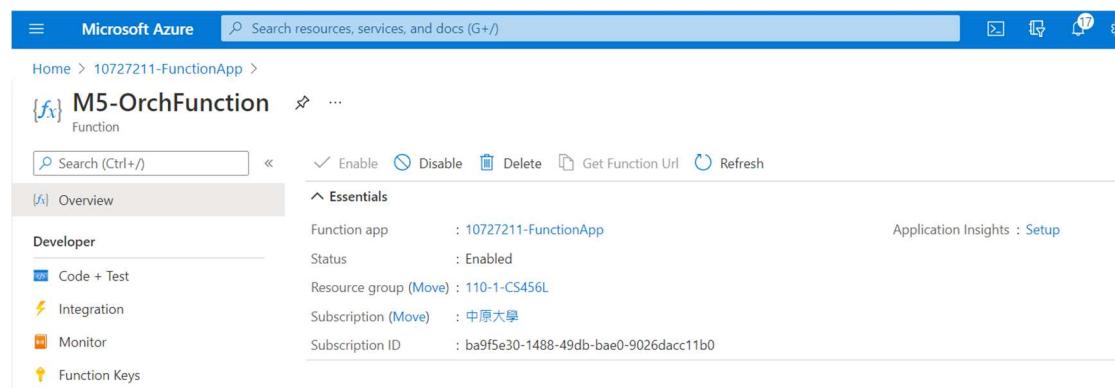
## 2. Create Durable Functions Http start



The screenshot shows the Microsoft Azure portal's "Overview" page for a Function App named "M5-HttpStart". The left sidebar lists "Developer" sections: "Code + Test", "Integration", "Monitor", and "Function Keys". The right pane displays the function's details:

Developer	Function app	Application Insights
Code + Test	: 10727211-FunctionApp	: Setup
Integration	Status : Enabled	
Monitor	Resource group (Move) : 110-1-CS456L	
Function Keys	Subscription (Move) : 中原大學	
	Subscription ID : ba9f5e30-1488-49db-bae0-9026dacc11b0	

## 3. Create the orchestrator function



The screenshot shows the Microsoft Azure portal's "Overview" page for a Function App named "M5-OrchFunction". The left sidebar lists "Developer" sections: "Code + Test", "Integration", "Monitor", and "Function Keys". The right pane displays the function's details:

Developer	Function app	Application Insights
Code + Test	: 10727211-FunctionApp	: Setup
Integration	Status : Enabled	
Monitor	Resource group (Move) : 110-1-CS456L	
Function Keys	Subscription (Move) : 中原大學	
	Subscription ID : ba9f5e30-1488-49db-bae0-9026dacc11b0	

```

1 const df = require("durable-functions");
2
3 module.exports = df.orchestrator(function*(context) {
4     const outputs = [];
5
6     /*
7     * We will call the approval activity with a reject and an approved to simulate both
8     */
9
10    outputs.push(yield context.df.callActivity("M5-Approval", "Approved"));
11    outputs.push(yield context.df.callActivity("M5-Approval", "Rejected"));
12
13    return outputs;
14 });
15

```

#### 4. Create the activity function

**Essentials**

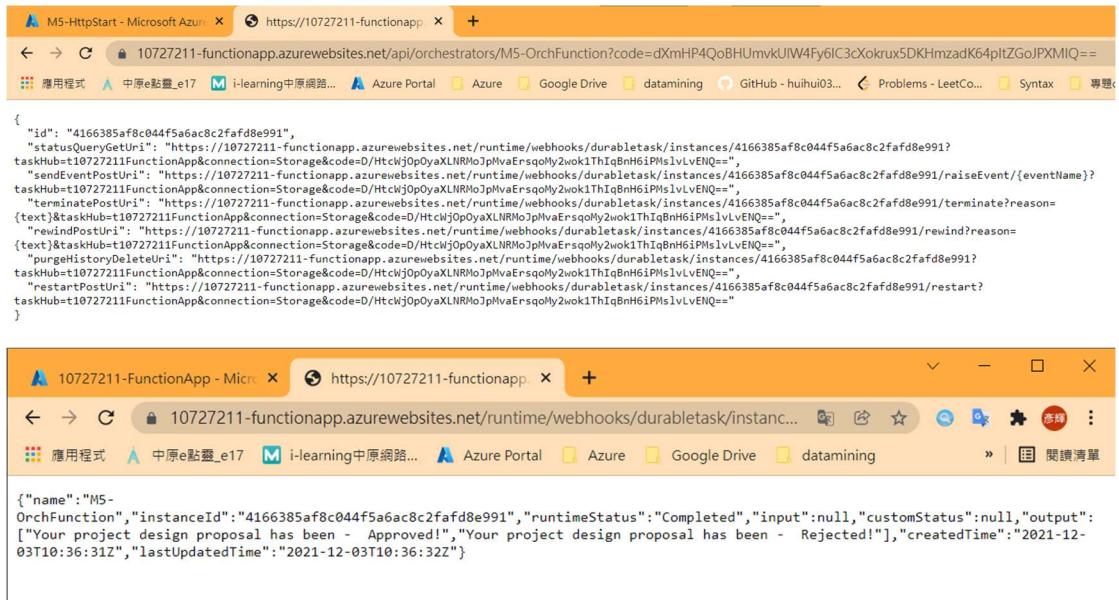
Function app	: 10727211-FunctionApp	Application Insights : Setup
Status	: Enabled	
Resource group (Move)	: 110-1-CS456L	
Subscription (Move)	: 中原大學	
Subscription ID	: ba9f5e30-1488-49db-bae0-9026dacc11b0	

```

1 module.exports = async function(context) {
2     return `Your project design proposal has been - ${context.bindings.name}!`;
3 };
4
5

```

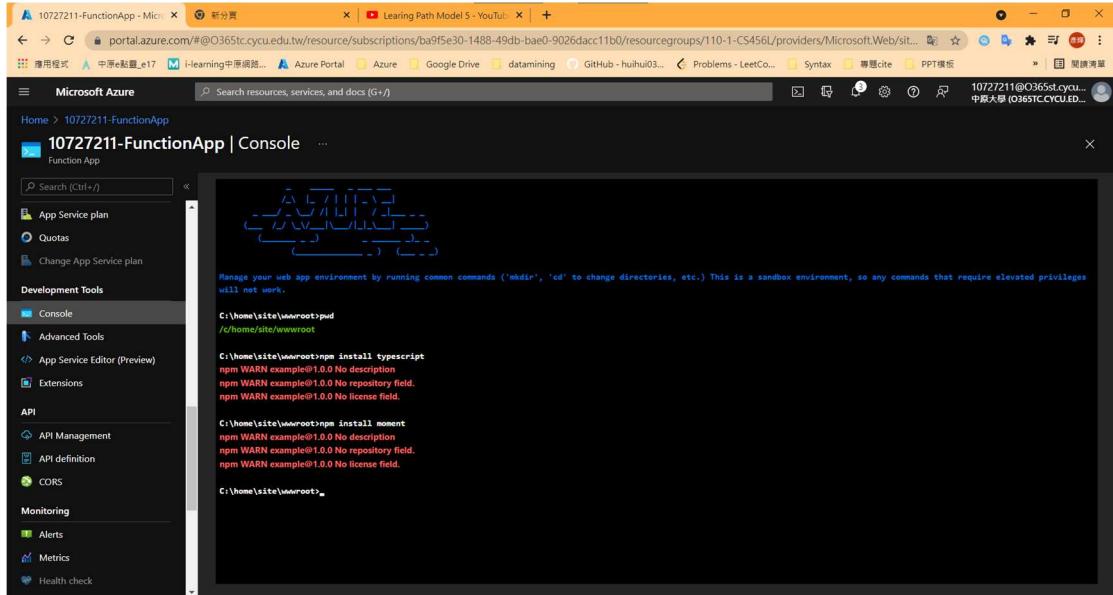
## 5. Verify that the durable functions workflow starts



```
{ "id": "4166385af8c044f5a6ac8c2fafd8e991", "statusQueryGetUri": "https://10727211-functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/4166385af8c044f5a6ac8c2fafd8e991?taskHub=10727211FunctionApp&connection=Storage&code=D/HtcWj0pOyaxLNRMoJpMvaErsqoMy2woK1ThIqBnH6iPMs1vLvENQ==", "sendEventPostUri": "https://10727211-functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/4166385af8c044f5a6ac8c2fafd8e991/raiseEvent/{eventName}?taskHub=10727211FunctionApp&connection=Storage&code=D/HtcWj0pOyaxLNRMoJpMvaErsqoMy2woK1ThIqBnH6iPMs1vLvENQ==", "terminatePostUri": "https://10727211-functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/4166385af8c044f5a6ac8c2fafd8e991/terminate?reason={text}&taskHub=10727211FunctionApp&connection=Storage&code=D/HtcWj0pOyaxLNRMoJpMvaErsqoMy2woK1ThIqBnH6iPMs1vLvENQ==", "rewindPostUri": "https://10727211-functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/4166385af8c044f5a6ac8c2fafd8e991/rewind?reason={text}&taskHub=10727211FunctionApp&connection=Storage&code=D/HtcWj0pOyaxLNRMoJpMvaErsqoMy2woK1ThIqBnH6iPMs1vLvENQ==", "purgeHistoryDeleteUri": "https://10727211-functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/4166385af8c044f5a6ac8c2fafd8e991?taskHub=10727211FunctionApp&connection=Storage&code=D/HtcWj0pOyaxLNRMoJpMvaErsqoMy2woK1ThIqBnH6iPMs1vLvENQ==", "restartPostUri": "https://10727211-functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/4166385af8c044f5a6ac8c2fafd8e991/restart?taskHub=10727211FunctionApp&connection=Storage&code=D/HtcWj0pOyaxLNRMoJpMvaErsqoMy2woK1ThIqBnH6iPMs1vLvENQ==" }
```

```
{"name":"M5-OrchFunction","instanceId": "4166385af8c044f5a6ac8c2fafd8e991", "runtimeStatus": "Completed", "input": null, "customStatus": null, "output": "[Your project design proposal has been - Approved!, Your project design proposal has been - Rejected!]", "createdTime": "2021-12-03T10:36:31Z", "lastUpdatedTime": "2021-12-03T10:36:32Z"}
```

## 6. Add moment npm package to your function app



```
Manage your web app environment by running common commands ('mkdir', 'cd' to change directories, etc.) This is a sandbox environment, so any commands that require elevated privileges will not work.

C:\home\site\wwwroot>pwd
C:\home\site\wwwroot>

C:\home\site\wwwroot>npm install typescript
npm WARN example@1.0.0 No description
npm WARN example@1.0.0 No repository field.
npm WARN example@1.0.0 No license field.

C:\home\site\wwwroot>npm install moment
npm WARN example@1.0.0 No description
npm WARN example@1.0.0 No repository field.
npm WARN example@1.0.0 No license field.

C:\home\site\wwwroot>
```

## 7. Add an escalation activity to your function app

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and a navigation bar with icons for Home, Search resources, services, and docs (G+), and notifications. Below the navigation bar, the URL is "Home > 10727211-FunctionApp >". The main title is "M5-Escalation" with a "Function" icon. On the left, a sidebar titled "Developer" has "Code + Test" selected. The "Overview" tab is active. In the center, under "Essentials", it shows the function app details: Function app : 10727211-FunctionApp, Status : Enabled, Resource group (Move) : 110-1-CS456L, Subscription (Move) : 中原大學, Subscription ID : ba9f5e30-1488-49db-bae0-9026dacc11b0, and Application Insights : Setup. To the right, there's a "Code + Test" section for the "M5-Escalation" function. The code editor shows the file "index.js" with the following content:

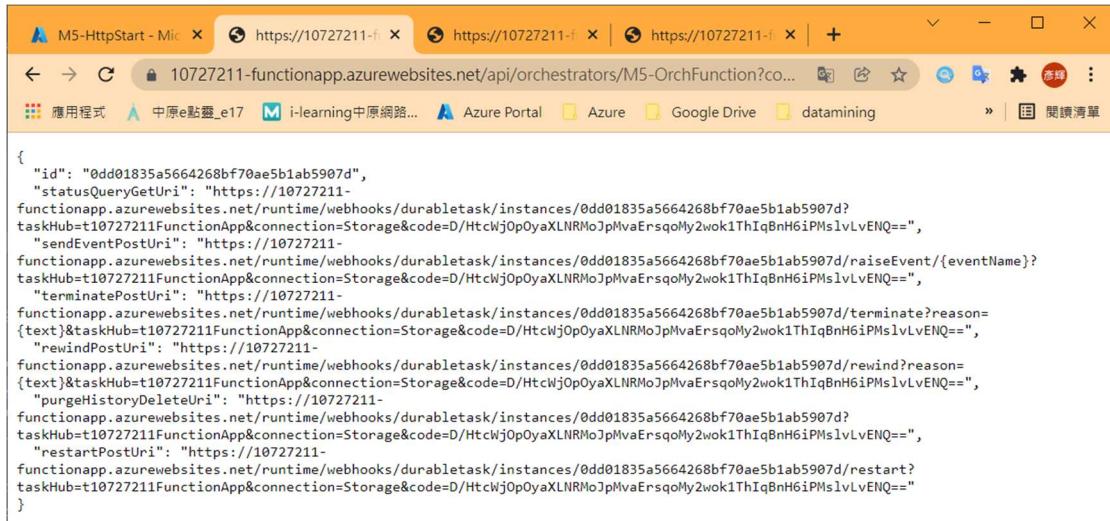
```
1 module.exports = async function (context) {
2     ...
3     return `ESCALATION : You have not approved the project design proposal - reassigning to your Manager! - ${context.bindings.name}!`;
4 }
```

## 8. Update the orchestration function to use the escalation function

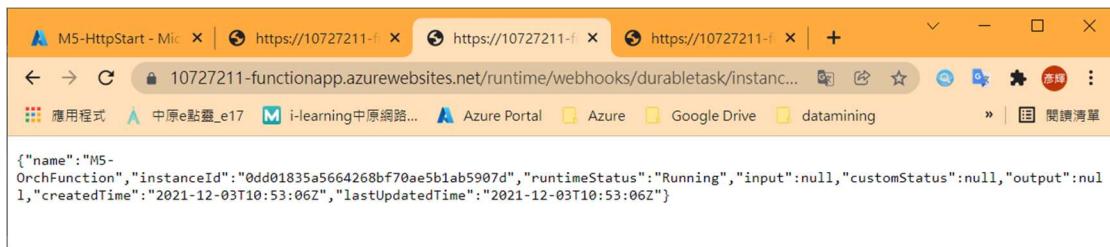
The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and a navigation bar with icons for Home, Search resources, services, and docs (G+), and notifications. Below the navigation bar, the URL is "Home > 10727211-FunctionApp >". The main title is "M5-OrchFunction" with a "Function" icon. On the left, a sidebar titled "Developer" has "Code + Test" selected. The "Overview" tab is active. In the center, under "Code + Test", it shows the file "index.js" for the "M5-OrchFunction" function. The code editor shows the following content:

```
1 const df = require("durable-functions");
2 const moment = require("moment");
3 module.exports = df.orchestrator(function*(context) {
4     const outputs = [];
5     const deadline = moment.utc(context.df.currentUtcDateTime).add(20, "s");
6     const activityTask = context.df.waitForExternalEvent("M5-Approval");
7     const timeoutTask = context.df.createTimer(deadline.toDate());
8     const winner = yield context.df.Task.any([activityTask, timeoutTask]);
9     if (winner === activityTask) {
10         outputs.push(yield context.df.callActivity("M5-Approval", "Approved"));
11     }
12     else {
13         outputs.push(yield context.df.callActivity("M5-Escalation", "Head of department"));
14     }
15     if (!timeoutTask.isCompleted) {
16         // All pending timers must be complete or canceled before the function exits.
17         timeoutTask.cancel();
18     }
19 }
20 return outputs;
21 });
22
23 
```

## 9. Verify that the Durable Functions workflow starts

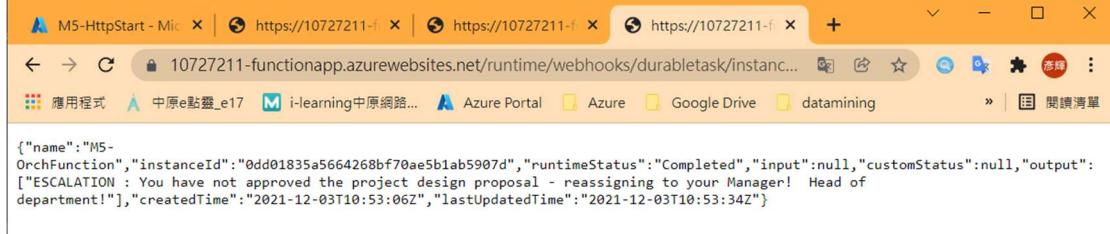


```
{  
    "id": "0dd01835a5664268bf70ae5b1ab5907d",  
    "statusQueryGetUri": "https://10727211-",  
    "functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/0dd01835a5664268bf70ae5b1ab5907d?  
taskHub=t10727211FunctionApp&connection=Storage&code=D/HtcWjOpOyaXLNRMoJpMvaErsqoMy2wok1ThIqBnH6iPMs1vLvENQ==",  
    "sendEventPostUri": "https://10727211-",  
    "functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/0dd01835a5664268bf70ae5b1ab5907d/raiseEvent/{eventName}?  
taskHub=t10727211FunctionApp&connection=Storage&code=D/HtcWjOpOyaXLNRMoJpMvaErsqoMy2wok1ThIqBnH6iPMs1vLvENQ==",  
    "terminatePostUri": "https://10727211-",  
    "functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/0dd01835a5664268bf70ae5b1ab5907d/terminate?reason=  
{text}&taskHub=t10727211FunctionApp&connection=Storage&code=D/HtcWjOpOyaXLNRMoJpMvaErsqoMy2wok1ThIqBnH6iPMs1vLvENQ==",  
    "rewindPostUri": "https://10727211-",  
    "functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/0dd01835a5664268bf70ae5b1ab5907d/rewind?reason=  
{text}&taskHub=t10727211FunctionApp&connection=Storage&code=D/HtcWjOpOyaXLNRMoJpMvaErsqoMy2wok1ThIqBnH6iPMs1vLvENQ==",  
    "purgeHistoryDeleteUri": "https://10727211-",  
    "functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/0dd01835a5664268bf70ae5b1ab5907d?  
taskHub=t10727211FunctionApp&connection=Storage&code=D/HtcWjOpOyaXLNRMoJpMvaErsqoMy2wok1ThIqBnH6iPMs1vLvENQ==",  
    "restartPostUri": "https://10727211-",  
    "functionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/0dd01835a5664268bf70ae5b1ab5907d/restart?  
taskHub=t10727211FunctionApp&connection=Storage&code=D/HtcWjOpOyaXLNRMoJpMvaErsqoMy2wok1ThIqBnH6iPMs1vLvENQ=="  
}
```



```
{"name":"M5-  
OrchFunction","instanceId":"0dd01835a5664268bf70ae5b1ab5907d","runtimeStatus":"Running","input":null,"customStatus":null,"output":nul  
l,"createdTime":"2021-12-03T10:53:06Z","lastUpdatedTime":"2021-12-03T10:53:06Z"}
```

After 20s...



```
{"name":"M5-  
OrchFunction","instanceId":"0dd01835a5664268bf70ae5b1ab5907d","runtimeStatus":"Completed","input":null,"customStatus":null,"output":  
["ESCALATION : You have not approved the project design proposal - reassigning to your Manager! Head of  
department!"],"createdTime":"2021-12-03T10:53:06Z","lastUpdatedTime":"2021-12-03T10:53:34Z"}
```

## Model 6: Develop, test, and publish Azure Functions by using Azure Functions Core Tools

1. Create a local Azure Functions project

```
azureuser@Azure:~$ mkdir ~/loan-wizard
azureuser@Azure:~$ cd ~/loan-wizard
azureuser@Azure:~/loan-wizard$ func init
Select a number for worker runtime:
1. dotnet
2. node
3. python
4. powershell
5. custom
Choose option: 2
node
Select a number for language:
1. javascript
2. typescript
Choose option: 1
javascript
Writing package.json
Writing .gitignore
Writing host.json
Writing local.settings.json
Writing /home/azureuser/loan-wizard/.vscode/extensions.json
```

2. Create an HTTP-triggered function

```
azureuser@Azure:~/loan-wizard$ func new
Select a number for template:
1. Azure Blob Storage trigger
2. Azure Cosmos DB trigger
3. Durable Functions activity
4. Durable Functions HTTP starter
5. Durable Functions orchestrator
6. Azure Event Grid trigger
7. Azure Event Hub trigger
8. HTTP trigger
9. IoT Hub (Event Hub)
10. Kafka output
11. Kafka trigger
12. Azure Queue Storage trigger
13. RabbitMQ trigger
14. SendGrid
15. Azure Service Bus Queue trigger
16. Azure Service Bus Topic trigger
17. SignalR negotiate HTTP trigger
18. Timer trigger
Choose option: 8
HTTP trigger
Function name: [HttpTrigger] simple-interest
Writing /home/azureuser/loan-wizard/simple-interest/index.js
Writing /home/azureuser/loan-wizard/simple-interest/function.json
```

### 3. Replace index.js

Azure Cloud Shell interface showing the contents of index.js in VS Code and the command 'code .' in the terminal.

```
index.js
module.exports = async function(context, req) {
    // Try to grab principal, rate, and term from the query string
    // parse them as numbers
    const principal = parseFloat(req.query.principal);
    const rate = parseFloat(req.query.rate);
    const term = parseFloat(req.query.term);

    if ([principal, rate, term].some(isNaN)) {
        // If any empty or non-numeric values, return a 400 response
        // error message
        context.res = {
            status: 400,
            body: "Please supply principal, rate and term in the query
        };
    } else {
        // Otherwise set the response body to the product of the three
        context.res = { body: principal * rate * term };
    }
}

azureuser@Azure:~/loan-wizard$ code .
```

### 4. Run the function locally

Azure Cloud Shell interface showing the output of 'func start' and a local curl request.

```
func start
Azure Functions Core Tools (2.7.2936 Commit hash: c00b06616a741dcdf70a3061fc415adde8
4010f8)
Function Runtime Version: 2.0.14494.0
AZURE_FUNCTIONS_ENVIRONMENT: Development
Hosting environment: Development
Content root path: /home/azureuser/loan-wizard
Now listening on: http://0.0.0.0:7071
Application started. Press Ctrl+C to shut down.

Functions:

    simple-interest: [GET,POST] http://localhost:7071/api/simple-interest

For detailed output, run func with --verbose flag.
[2021-11-24T14:52:09.860] Worker process started and initialized.
[2021-11-24T14:52:14.627] Host lock lease acquired by instance ID '0000000000000000
0000000EAFF55D'.
^CApplication is shutting down...

func start &> ~/output.txt &
[1] 261
curl "http://localhost:7071/api/simple-interest" -w "
\n"
Please supply principal, rate and term in the query string
curl "http://localhost:7071/api/simple-interest?principal=5000&rate=.035&term=36" -w "\n"
6300
pkill func
[1]+ Done func start &> ~/output.txt
code ~/output.txt
```

## 5. See output.txt

Azure Functions Core Tools (2.7.2936 Commit hash: c00b06616a741dc...)

Function Runtime Version: 2.0.14494.0

AZURE\_FUNCTIONS\_ENVIRONMENT: Development

Hosting environment: Development

Content root path: /home/azureuser/loan-wizard

Now listening on: http://0.0.0.0:7071

Application started. Press Ctrl+C to shut down.

Functions:

simple-interest: [GET,POST] http://localhost:7071/api/simple-interest

For detailed output, run func with --verbose flag.

[2021-11-24T14:53:02.421] Worker process started and initialized.

[2021-11-24T14:53:03.266] Executing 'Functions.simple-interest' (S)

[2021-11-24T14:53:03.501] Executed 'Functions.simple-interest' (S)

[2021-11-24T14:53:07.259] Host lock lease acquired by instance ID

[2021-11-24T14:53:08.759] Executing 'Functions.simple-interest' (S)

[2021-11-24T14:53:08.766] Executed 'Functions.simple-interest' (S)

## 6. Create a function app

```
azureuser@Azure:~/loan-wizard$ RESOURCEGROUP="learn-0f6b9075-14e1-4f58-9cb9-cd8a0f43c8a"
azureuser@Azure:~/loan-wizard$ STORAGEACCT=learnstorage$(openssl rand -hex 5)
azureuser@Azure:~/loan-wizard$ FUNCTIONAPP=learnfunctions$(openssl rand -hex 5)
azureuser@Azure:~/loan-wizard$ az storage account create \
>   --resource-group "$RESOURCEGROUP" \
>   --name "$STORAGEACCT" \
>   --kind StorageV2 \
>   --location centralus

azureuser@Azure:~/loan-wizard$ az functionapp create \
>   --resource-group "$RESOURCEGROUP" \
>   --name "$FUNCTIONAPP" \
>   --storage-account "$STORAGEACCT" \
>   --runtime node \
>   --consumption-plan-location centralus \
>   --functions-version 3
```

## 7. Publish to Azure

```
azureuser@Azure:~/loan-wizard$ cd ~/loan-wizard
azureuser@Azure:~/loan-wizard$ func azure functionapp publish "$FUNCTIONAPP" ←
You're trying to publish to a non-v2 function app from v2 tooling. ←
You can pass --force to force update the app to v2, or switch to v1 or v3 tooling for ←
publishing
azureuser@Azure:~/loan-wizard$ func azure functionapp publish "$FUNCTIONAPP" --force
Getting site publishing info...
Creating archive for current directory...
Uploading 1.35 KB [#####
Upload completed successfully.
Deployment completed successfully.
Syncing triggers...
Functions in learnfunctionsd40a1687b5:
    simple-interest - [httpTrigger]
        Invoke url: https://learnfunctions40a1687b5.azurewebsites.net/api/simple-interest?code=HgKBN7L9EA0emzo58bgZqUyaMAfuPmB3JWlgWBdEmDwKxQG1y/xxg==
```

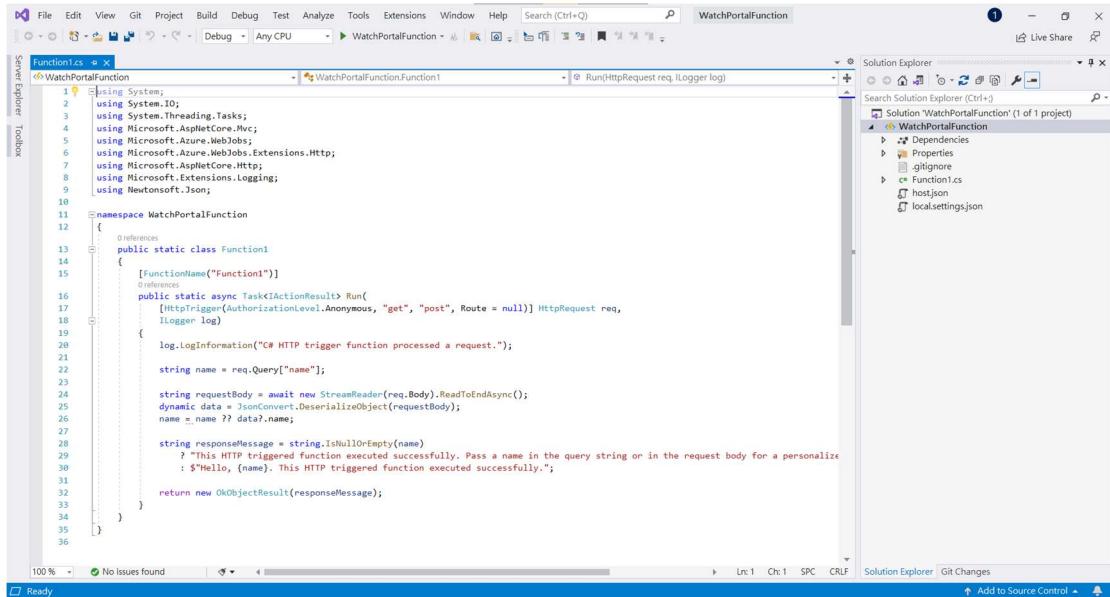
## 8. Run the function



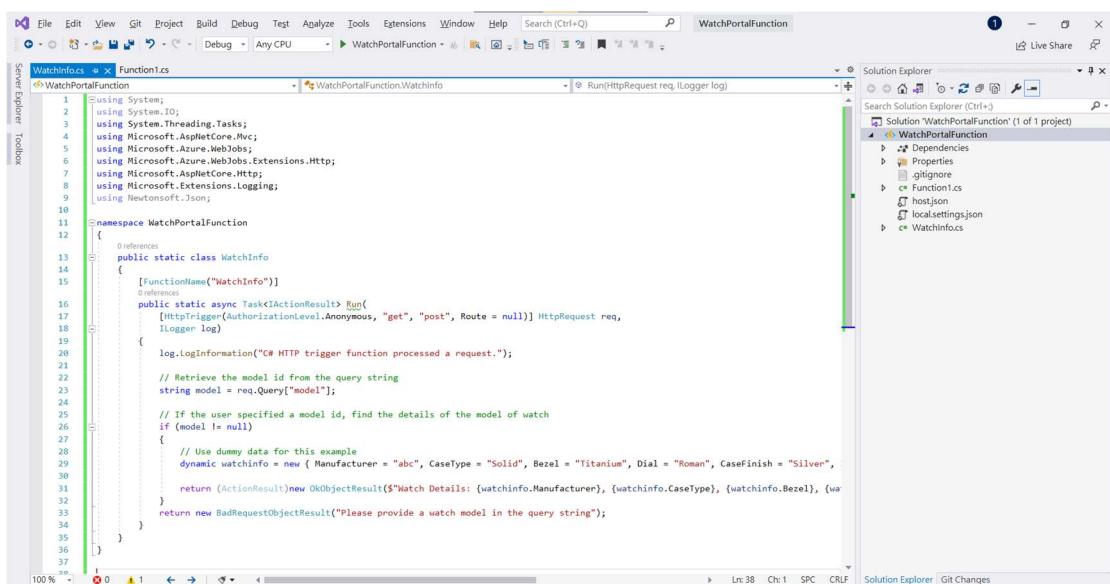
# (Model 7: Develop, test, and deploy an Azure Function with Visual Studio)

## Visual Studio

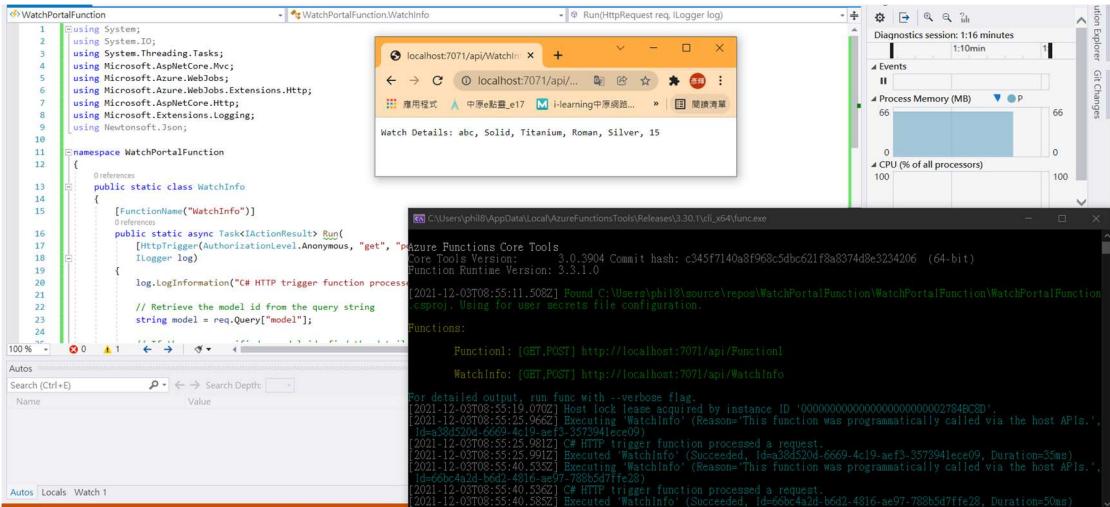
### 1. Create an Azure Function app



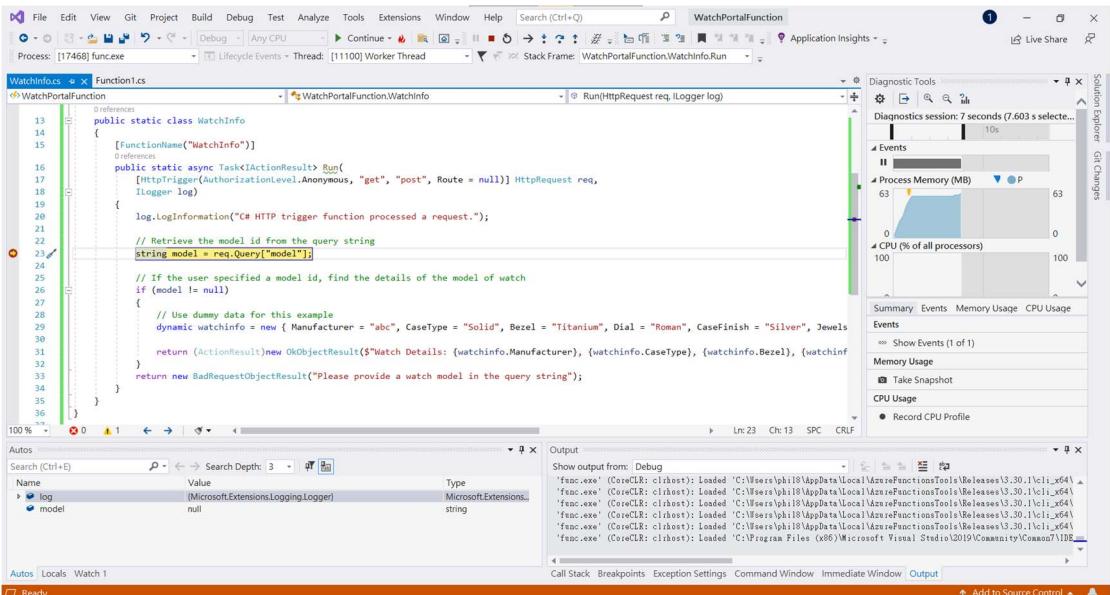
### 2. Create WatchInfo Function



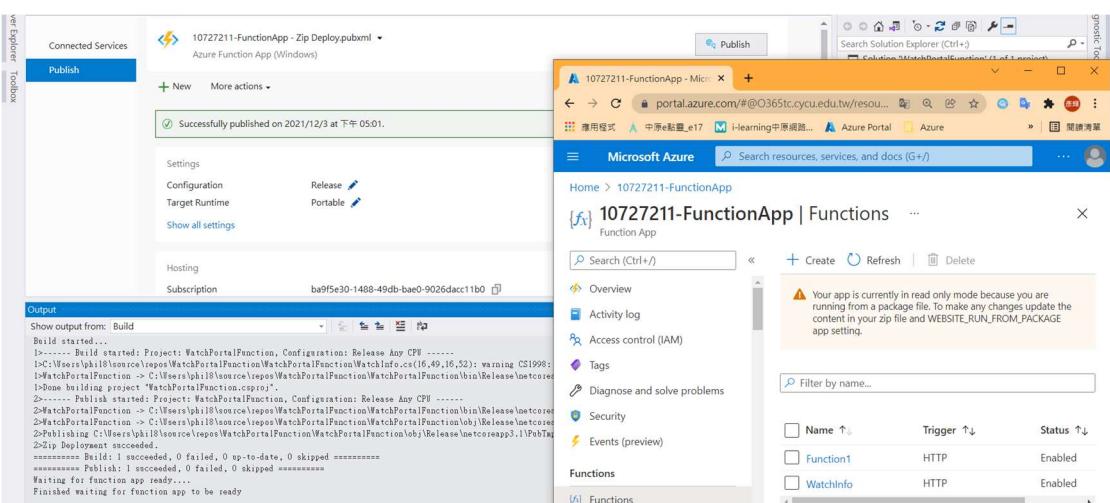
### 3. Test the Azure Function locally



### 4. Add breakpoint & Test Azure Function locally



### 5. Deploy the WatchInfo function to the Azure Function App



## 6. Verify the functions have been deployed

The screenshot shows the Azure portal interface for a function named 'WatchInfo'. The 'Get Function Url' section displays the URL <https://10727211-functionapp.azurewebsites.net/api/WatchInfo?&model=abc>. Below the URL, resource group, subscription, and subscription ID information are listed. A preview window shows the JSON response: "Watch Details: abc, Solid, Titanium, Roman, Silver, 15".

## 7. Unit test an Azure Function (success version)

The Test Explorer shows three successful tests under the 'WatchFunctionsTests' group. The results are summarized in the Group Summary pane.

Test	Duration	Traits	Error Message
WatchFunctionsTests (3)	57 ms		
WatchFunctionUnitTests (3)	57 ms		
TestWatchFunctionFailureNo...	4 ms		
TestWatchFunctionFailureNo...	10 ms		
TestWatchFunctionSuccess	43 ms		

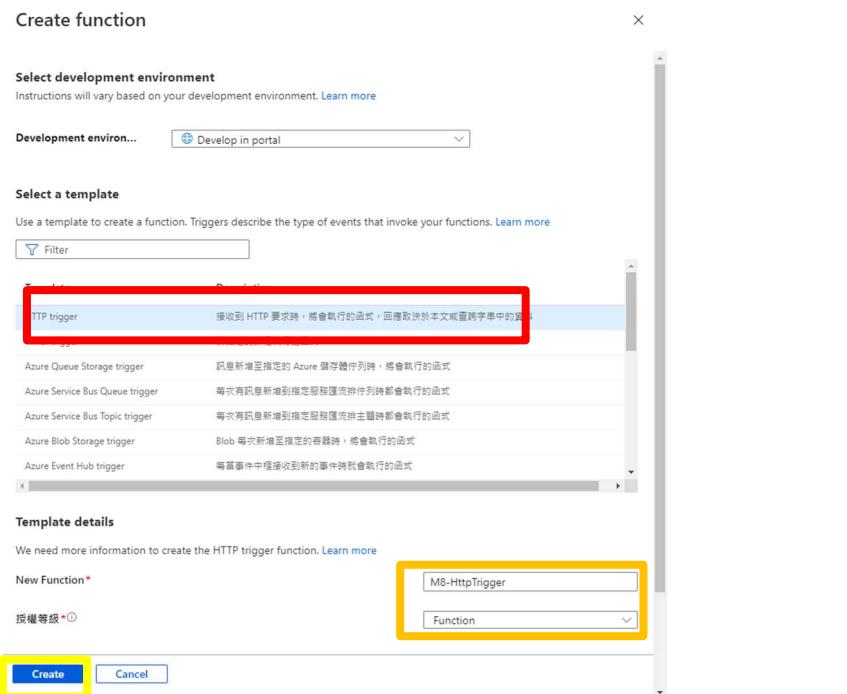
## 8. Unit test an Azure Function (error version)

The Test Explorer shows two passed tests and one failed test under the 'WatchFunctionsTests' group. The failed test is 'TestWatchFunctionSuccess' with an error message: 'Assert.IsNotNull() Failure E...'. The results are summarized in the Group Summary pane.

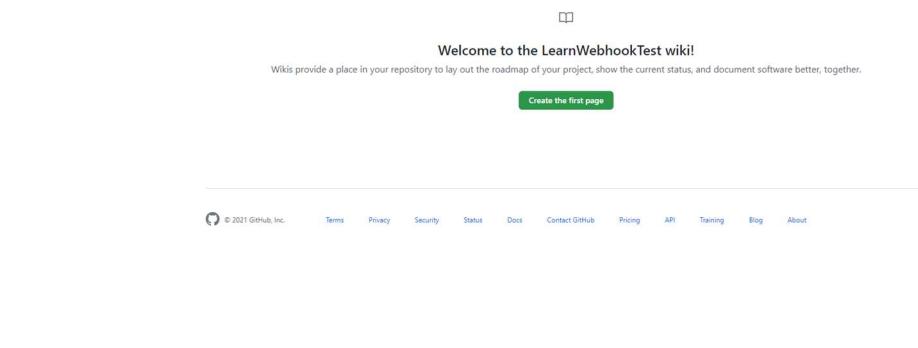
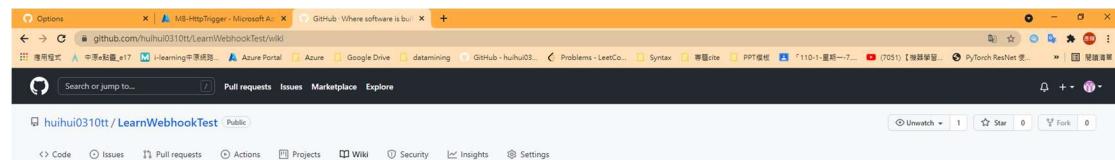
Test	Duration	Traits	Error Message
WatchFunctionsTests (3)	25 ms		
WatchFunctionUnitTests (3)	25 ms		
TestWatchFunctionFailureNo...	4 ms		
TestWatchFunctionFailureNo...	10 ms		
TestWatchFunctionSuccess	11 ms		Assert.IsNotNull() Failure E...

# Model 8: Monitor GitHub events by using a webhook with Azure Functions

## 1. Create a Webhook Trigger function



## 2. Set up a webhook for a GitHub repository



[https://github.com/huihui0310t/LearnWebhookTest/wiki/\\_new](https://github.com/huihui0310t/LearnWebhookTest/wiki/_new)

### 3. Add a webhook and Test

The screenshot shows the GitHub settings page for the repository 'huihui0310tt/LearnWebhookTest'. The 'Webhooks' section is selected. A single webhook is listed with the URL 'https://10727211-functionapp.azurewebsites.net/github'. There are 'Edit' and 'Delete' buttons next to it.

### 4. Replace index.js in Azure Function

The screenshot shows the Azure Functions developer tools for the function 'M8-HttpTrigger'. The code editor displays the following JavaScript code:

```
module.exports = async function (context, req) {
    context.log('JavaScript HTTP trigger function processed a request.');

    const name = (req.query.name || (req.body && req.body.name));
    const responseMessage = `Hello, ${name}! This HTTP triggered function executed successfully.`;
    const response = {
        body: `${responseMessage}`,
        status: 200,
        headers: { 'Content-Type': 'text/plain; charset=utf-8' }
    };

    if (req.body.pages[0].title) {
        response.body = `Page is ${req.body.pages[0].title}, Action is ${req.body.pages[0].action}, Event Type is ${req.headers['x-github-event']}`;
    } else {
        response.status = 400;
        response.body = ("Invalid payload for Wiki event");
    }

    context.res = response;
}
```

### 5. 2<sup>nd</sup> Trigger Webhook

The screenshot shows the GitHub settings page for the repository 'huihui0310tt/LearnWebhookTest'. The 'Webhooks / Manage webhook' section is selected. A list of webhook deliveries is shown, with one entry highlighted:

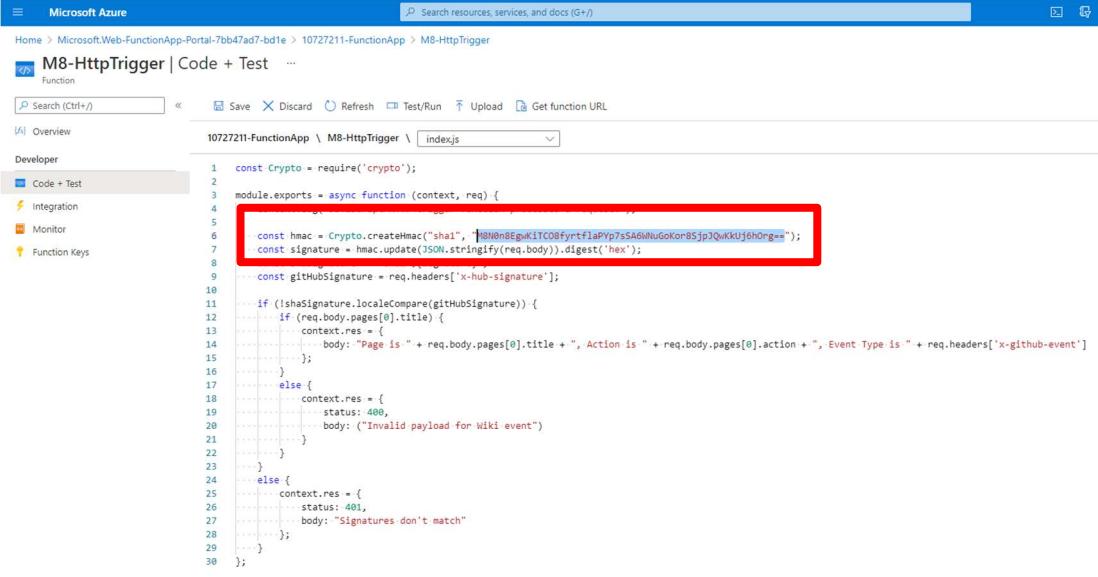
Delivery ID	Timestamp	Status
544b87e8-4eb2-11ec-9055-0e9581be86d1	2021-11-25 23:18:21	Completed in 0.59 seconds
Request	200	

Details for the delivery show the headers and body of the received request:

Headers:  
Content-Type: text/plain; charset=utf-8  
Date: Thu, 25 Nov 2021 15:18:21 GMT

Body:  
Page is Home, Action is edited, Event Type is gollum

## 6. Webhook encryption



**Webhooks / Manage webhook**

**Recent Deliveries**

Delivery ID	Request	Response	Timestamp	Actions
544b87e0-4e02-11ec-9055-0e9581be86d1			2021-11-25 23:23:52	...

**Request** **Response 200** **Redeliver** **Completed in 0.55 seconds.**

**Headers**

```
Request URL: https://10727211-functionapp.azurewebsites.net/api/M8-HttpTrigger?code=M8N0n8EgwKiTC08fyrtflaPYo7sA6WtuGoKor85jpJQikUj6hOrg=
Request method: POST
Accept: */*
Content-Type: application/json
User-Agent: GitHub-Hookshot/e32936c
X-GitHub-Delivery: 544b87e0-4e02-11ec-9055-0e9581be86d1
X-GitHub-Event: gollum
X-GitHub-Hook-ID: 330221597
X-GitHub-Hook-Installation-Target-ID: 431885804
X-GitHub-Hook-Installation-Target-Type: repository
X-Hub-Signature: sha1=9c1a48c1b4eaf14d26b2974f1b36117d0c7f4dd
X-Hub-Signature-256: sha256=53fe862b052157ea105a2333a63a6bc4f3320ec0905fd920e0aea6b56935bbb3
```

**Payload**

```
{
  "pages": [
    {
      "page_name": "Home",
      "title": "Home",
      "summary": null,
      "action": "edited",
      "sha": "b86a52f62dece2135d637f1787b70cf02bd380",
      "html_url": "https://github.com/huihui0310tt/LearnWebhookTest/wiki/Home"
    }
  ]
}
```

# Model 9: Enable automatic updates in a web application using Azure Functions and SignalR Service

## 1. Create Resource in Azure

```
export STORAGE_ACCOUNT_NAME=10727211model9storage
echo "Storage Account Name: $STORAGE_ACCOUNT_NAME"

az storage account create \
    --name $STORAGE_ACCOUNT_NAME \
    --resource-group 110-1-CS456L \
    --kind StorageV2 \
    --sku Standard_LRS

az cosmosdb create \
    --name 10727211cosmosdb \
    --resource-group 110-1-CS456L
```

## 2. Download sample app code & switch directory

```
問題 輸出 檢錄主控台 終端機
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\phil18\Desktop\Azure\1125_demo_serverless_workspace> git clone https://github.com/MicrosoftDocs/mslearn-advocates.azure-functions-and-signalr.git serverless-demo
Cloning into 'serverless-demo'...
remote: Enumerating objects: 151, done.
remote: Counting objects: 100% (132/132), done.
remote: Compressing objects: 100% (33/33), done.
Receiving objects: 87% (132/151) (delta 0), pack-reused 119 receiving objects: 86% (130/151)
Receiving objects: 100% (151/151), 97.79 KiB | 370.00 KiB/s, done.
Resolving deltas: 100% (83/83), done.
PS C:\Users\phil18\Desktop\Azure\1125_demo_serverless_workspace> code .\serverless-demo\start
PS C:\Users\phil18\Desktop\Azure\1125_demo_serverless_workspace>
```

## 3. Update local settings

```
STORAGE_CONNECTION_STRING=$(az storage account show-connection-string \
    --name $(az storage account list \
        --resource-group 110-1-CS456L \
        --query [0].name -o tsv) \
    --resource-group 110-1-CS456L \
    --query "connectionString" -o tsv)

COSMOSDB_ACCOUNT_NAME=$(az cosmosdb list \
    --resource-group 110-1-CS456L \
    --query [0].name -o tsv)
```

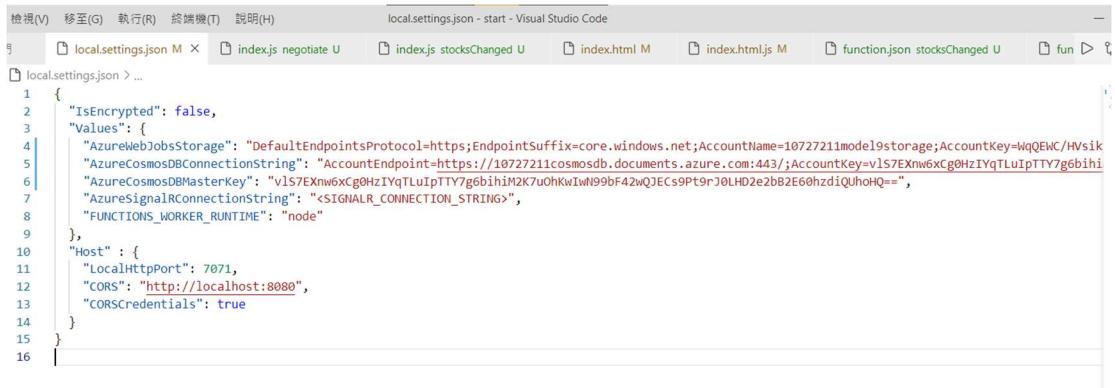
```

COSMOSDB_CONNECTION_STRING=$(az cosmosdb list-connection-strings \
--name $COSMOSDB_ACCOUNT_NAME \
--resource-group 110-1-CS456L \
--query "connectionStrings[?description=='Primary SQL Connection
String'].connectionString" -o tsv)

COSMOSDB_MASTER_KEY=$(az cosmosdb list-keys \
--name $COSMOSDB_ACCOUNT_NAME \
--resource-group 110-1-CS456L \
--query primaryMasterKey -o tsv)

printf "\n\nReplace <STORAGE_CONNECTION_STRING>
with:\n$STORAGE_CONNECTION_STRING\n\nReplace
<COSMOSDB_CONNECTION_STRING>
with:\n$COSMOSDB_CONNECTION_STRING\n\nReplace
<COSMOSDB_MASTER_KEY> with:\n$COSMOSDB_MASTER_KEY\n\n"

```

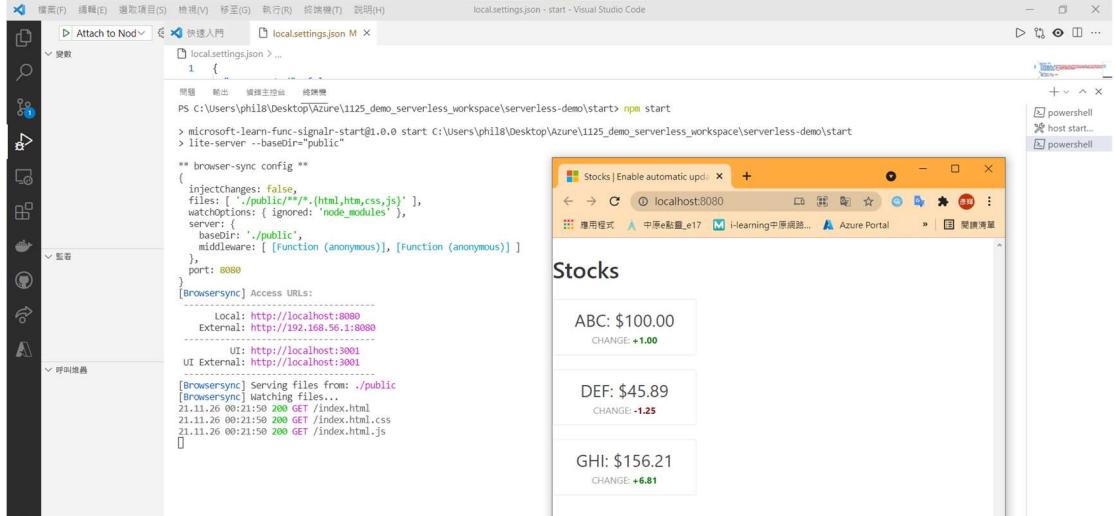


```

{
  "IsEncrypted": false,
  "Values": {
    "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=10727211model9storage;AccountKey=WqQEWC/HVsik",
    "AzureCosmosDBConnectionString": "AccountEndpoint=https://10727211cosmosdb.documents.azure.com:443/;AccountKey=v157EXnw6xCg0HzIYqTLuIpTTY7g6bihi",
    "AzureCosmosDBMasterKey": "v1s7Exnw6xCg0HzIYqTLuIpTTY7g6bihiM2K7u0hKwIwN99bF42wQJECs9Pt9rJ0LHD2e2bE60hzdiQuhoHQ==",
    "AzureSignalRConnectionString": "<SIGNALR_CONNECTION_STRING>",
    "FUNCTIONS_WORKER_RUNTIME": "node"
  },
  "Host": {
    "LocalHttpPort": 8071,
    "CORS": "http://localhost:8080",
    "CORSCredentials": true
  }
}

```

#### 4. Run the application



## 1. Create a SignalR account

```
SIGNALR_SERVICE_NAME=s10727211signalr

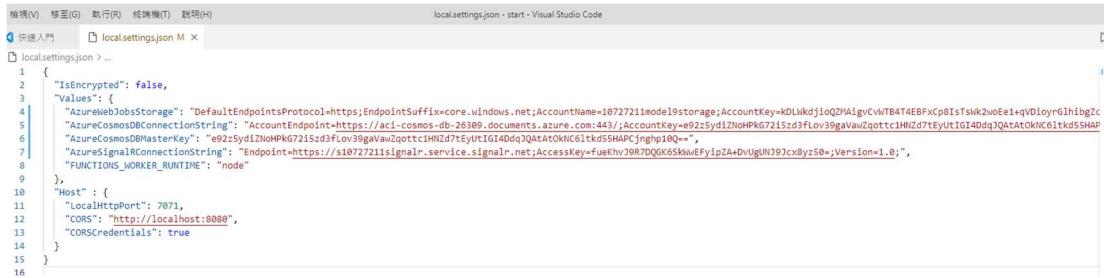
az signalr create \
    --name ${SIGNALR_SERVICE_NAME} \
    --resource-group 110-1-CS456L \
    --sku Free_DS2 \
    --unit-count 1

az resource update \
    --resource-type Microsoft.SignalRService/SignalR \
    --name ${SIGNALR_SERVICE_NAME} \
    --resource-group 110-1-CS456L \
    --set properties.features[flag=ServiceMode].value=Serverless

SIGNALR_CONNECTION_STRING=$(az signalr key list \
    --name $(az signalr list \
        --resource-group 110-1-CS456L \
        --query [0].name -o tsv) \
    --resource-group 110-1-CS456L \
    --query primaryConnectionString -o tsv)

printf "\n\nReplace <SIGNALR_CONNECTION_STRING>
with:\n$SIGNALR_CONNECTION_STRING\n\n"
```

## 2. Update local settings

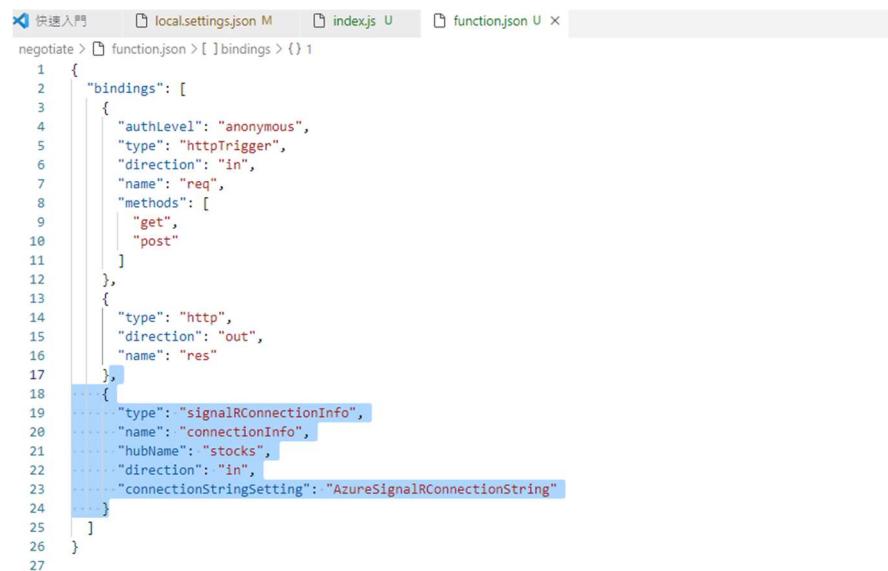


```
local.settings.json - start - Visual Studio Code
本地文件夹 快速入门 local.settings.json M X
local.settings.json > ...
1 {
2     "IsEncrypted": false,
3     "Values": {
4         "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=10727211mode19storage;AccountKey=kDLwkKdjoQ2VAigvCvITB4T4EBFxcp8IsTsk2w0Ee1+qVDioy6lhbgZc
5         "AzureCosmosDBConnectionString": "AccountEndpoint=https://aci-cosmos-db-26309.documents.azure.com:443/;AccountKey=e92z5ydi1Zh0HPKG721szd3fLov39gavawZqottc1H2d7tEyUtg140dgjQAtOKNC61tkd5SHAP
6         "AzureCosmosDBMasterKey": "e92z5ydi1Zh0HPKG721szd3fLov39gavawZqottc1H2d7tEyUtg140dgjQAtOKNC61tkd5SHAPjngh10q==",
7         "AzureSignalRConnectionString": "Endpoint=https://s10727211signalr.service.signalr.net;AccessKey=fueKhvJ9R7DQ0K65kwiEfYipZA+DvUgUlJ93cx8yz50=;Version=1.0;",
8         "FUNCTIONS_WORKER_RUNTIME": "node"
9     },
10    "Host": {
11        "LocalHttpPort": 7071,
12        "CORS": "http://localhost:8080",
13        "CORScredentials": true
14    }
15 }
```

## 3. Create Azure Functions

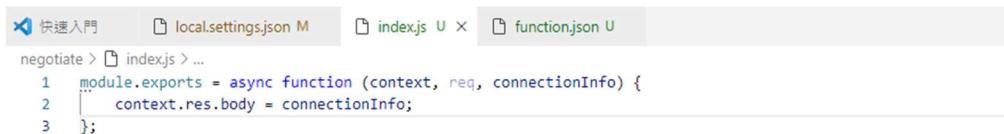
	Azure Functions: Create Function
Template	HTTP Trigger
Name	negotiate
Authorization level	Anonymous

#### 4. Update client connections : negotiate > function.json



```
1  {
2    "bindings": [
3      {
4        "authLevel": "anonymous",
5        "type": "httpTrigger",
6        "direction": "in",
7        "name": "req",
8        "methods": [
9          "get",
10         "post"
11       ],
12     },
13     {
14       "type": "http",
15       "direction": "out",
16       "name": "res"
17     },
18     {
19       "type": "signalRConnectionInfo",
20       "name": "connectionInfo",
21       "hubName": "stocks",
22       "direction": "in",
23       "connectionStringSetting": "AzureSignalRConnectionString"
24     }
25   ]
26 }
27
```

#### 5. Update client connections : negotiate > index.js



```
1 module.exports = async function (context, req, connectionInfo) {
2   context.res.body = connectionInfo;
3 };
```

#### 6. Create Azure Functions

Azure Functions: Create Function	
Template	Azure Cosmos DB Trigger
Name	stocksChanged
App setting for your Azure Cosmos DB account	AzureCosmosDBConnectionString
Database name	stocksdb
Collection name	stocks
Collection name for leases	leases
Create lease collection if not exists	true

## 7. Update Detect and broadcast database changes : stocksChanged > function.json

```

1  {
2      "bindings": [
3          {
4              "type": "cosmosDBTrigger",
5                  "name": "documents",
6                  "direction": "in",
7                  "leaseCollectionName": "leases",
8                  "connectionStringSetting": "AzureCosmosDBConnectionString",
9                  "databaseName": "stocksd",
10                 "collectionName": "stocks",
11                 "createLeaseCollectionIfNotExists": "true",
12                 "feedPollDelay": 500
13             },
14             {
15                 "type": "signalR",
16                     "name": "signalRMessages",
17                     "connectionString": "AzureSignalRConnectionString",
18                     "hubName": "stocks",
19                     "direction": "out"
20             }
21         ]
22     }

```

## 8. Update Detect and broadcast database changes : stocksChanged > index.js

```

1  module.exports = async function (context, documents) {
2      const updates = documents.map(stock => ({
3          target: 'updated',
4          arguments: [stock]
5      }));
6
7      context.bindings.signalRMessages = updates;
8      context.done();
9  }

```

## 9. Update the web application : public > index.html

```

14  <!-- BEGIN: Replace markup in this section -->
15  <div id="app" class="container">
16      <h1>Stocks</h1>
17      <div id="stocks">
18          <div v-for="stock in stocks" class="stock">
19              <transition name="fade" mode="out-in">
20                  <div class="list-item" key="stock.price">
21                      <div class="lead">{{ stock.symbol }}: ${{ stock.price }}</div>
22                      <div class="change">
23                          <span
24                              :class="`is-up`: stock.changeDirection === '+', `is-down`: stock.changeDirection === '-'">
25                              {{ stock.changeDirection }}{{ stock.change }}
26                          </span>
27                      </div>
28                  </div>
29              </transition>
30          </div>
31      </div>
32  <!-- END -->
33
34
35  <script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.10/vue.min.js" integrity="sha256-ch1NFSx3TdcQ2Xla7SvmbLAAvAQL00Y/LB1UX04vIY=" crossorigin="anonymous"></script>
36  <script src="https://cdnjs.cloudflare.com/ajax/libs/axios/0.19.0/axios.min.js" integrity="sha256-5134GVHHD1r1r9qxsXhC8Zhw74PHMaFpsMpSPXtjTs=" crossorigin="anonymous"></script>
37  <script src="https://cdn.jsdelivr.net/npm/@aspnet/signalr@1.1.0/dist/browser/signalr.js"></script>
38  <script src="index.html.js" type="text/javascript"></script>
39

```

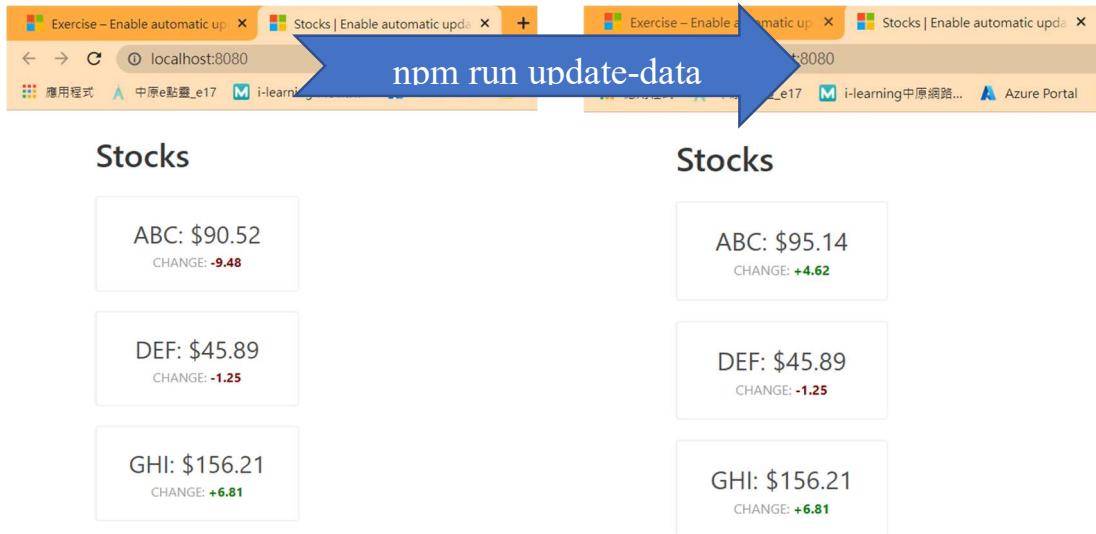
## 10. Update the web application : public > index.html.js

```

1  const LOCAL_BASE_URL = 'http://localhost:7071';
2  const REMOTE_BASE_URL = '<FUNCTION_APP_ENDPOINT>';
3
4  const getAPIBaseUrl = () => {
5      const isLocal = /localhost/.test(window.location.href);
6      return isLocal ? LOCAL_BASE_URL : REMOTE_BASE_URL;
7  }
8
9  const app = new Vue({
10     el: '#app',
11     data() {
12         return {
13             stocks: []
14         }
15     },
16     methods: {
17         async getStocks() {
18             try {
19                 const apiUrl = `${getAPIBaseUrl()}/api/getStocks`;
20                 const response = await axios.get(apiUrl);
21                 app.stocks = response.data;
22             } catch (ex) {
23                 console.error(ex);
24             }
25         }
26     }
27 }

```

## 11. Run the application



### 1. Create Azure Functions

	Azure Functions: Deploy to Function App
Template	Create new Function App in Azure... Advanced
Name	10727211-deployfunctionapp
Running Stack	Node.js 8 LTS
Resource Group	110-1-CS456L
Location for new resource	東南亞 (for CS456L)
Hosting Plan	Consumption
storage	10727211model9storage
Application Insights resource	Skip for now

### 2. Update the function URL

```
快速入門 local.settings.json M index.js negotiate U index.js stocksChanged U index.html M
public > index.htmljs > [?] REMOTE_BASE_URL
1 const LOCAL_BASE_URL = 'http://localhost:7071';
2 const REMOTE_BASE_URL = 'https://10727211-deployfunctionapp.azurewebsites.net';
3
4 const setAPTBaseUrl = () => {
```

### 3. Upload Function Settings

Azure Functions: Upload local settings	
Function App in Azure	10727211-deployfunctionapp

### 4. Configure static websites in Azure Storage

Azure Storage: Configure Static Website	
Storage account	10727211model9storage
Default file	Index.html
Error document	Index.html

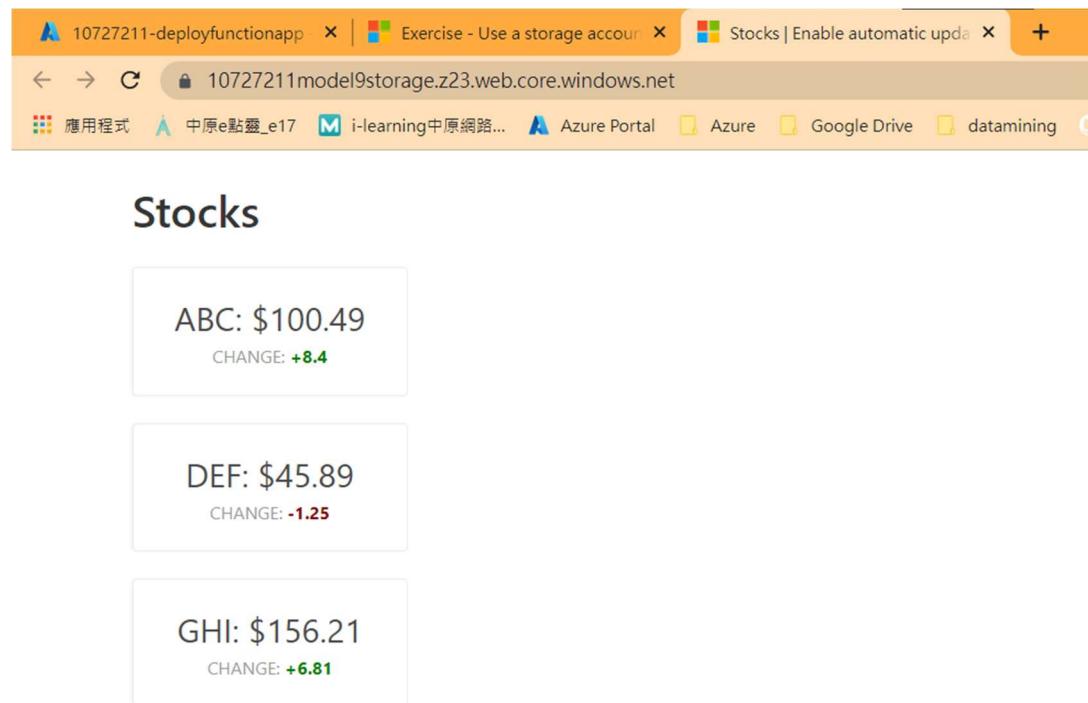
### 5. Deploy the web application to Azure Storage

Azure Storage: Deploy to static website via Azure Storage	
Storage account	10727211model9storage
Select the folder to deploy	Select <b>browse</b> and choose the <i>public</i> subfolder containing the web app.

### 6. Set up CORS in the function app

The screenshot shows the Azure portal interface for managing a Function App named "10727211-FunctionApp". The left sidebar lists various monitoring and logging options. The main content area is titled "CORS" under the "API" section. It includes a note about enabling CORS for JavaScript code running in a browser. A checkbox for "Enable Access-Control-Allow-Credentials" is checked. Below this, a section for "Allowed Origins" contains a single entry: "https://10727211model9storage.z23.web.core.windows.net".

## 7. Run the deployed application



The screenshot shows a web browser window with three tabs open:

- 10727211-deployfunctionapp
- Exercise - Use a storage account
- Stocks | Enable automatic update

The address bar displays the URL: 10727211model9storage.z23.web.core.windows.net. Below the address bar, there is a toolbar with various icons and links.

The main content area is titled "Stocks" and displays three stock price cards:

- ABC: \$100.49**  
CHANGE: **+8.4**
- DEF: \$45.89**  
CHANGE: **-1.25**
- GHI: \$156.21**  
CHANGE: **+6.81**

# Model 10: Expose multiple Azure Function apps as a consistent API by using Azure API Management

## 1. Create functions

```
Bash Requesting a Cloud Shell. Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

azureuser@Azure:~$ git clone https://github.com/MicrosoftDocs/mslearn-apim-and-functions.git
~/OnlineStoreFuncs
Cloning into '/home/azureuser/OnlineStoreFuncs'...
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 39 (delta 16), reused 9 (delta 4), pack-reused 0
Unpacking objects: 100% (39/39), done.
azureuser@Azure:~$ cd ~/OnlineStoreFuncs
azureuser@Azure:~/OnlineStoreFuncs$ bash setup.sh

The resource group is called learn-cbfff884e-d559-4845-b052-78a4882fad4e and is located in westus

Creating a storage account for the functions...
[- Running ...]
```

## 2. Test ProductDetails function

The screenshot shows the Azure portal interface for the ProductDetails function. On the left, the function's code editor displays the function.json configuration file. On the right, a 'Test' panel is open, showing the 'Input' tab with a JSON object and the 'Output' tab with the resulting JSON response. A red box highlights the 'Output' tab.

```
ProductFunction9146e8ab3a \ ProductDetails \ function.json
```

```
HTTP response code
200 OK

HTTP response content
{
  "ID": 3,
  "Name": "Smart Dimmer Switch",
  "Price": 59.99,
  "PartNumber": "DS728"
}
```

The screenshot shows a browser window displaying the JSON response from the Azure function. The URL is https://productfunction9146e8ab3a.azurewebsites.net/api/ProductDetails?code=DQCoENYTge5RTK1s0A0HpEZ4pJiUvVQ723/palu5YoAgQj0sbaeFQ==&id=3. The response body contains the same JSON object as the test results in the previous screenshot.

```
{"ID":3,"Name":"Smart Dimmer Switch","Price":59.99,"PartNumber":"DS728"}
```

### 3. Expose function app as an API using Azure API Management

The screenshot shows two side-by-side Azure management pages. On the left, the 'API Management service' blade is open, where a new service named 'ProductFunction9146e8ab3a-apim' is being created under the 'azure-api.net' subscription, 'Concierge Subscription'. It's assigned to the 'learn-cfbf884e-d559-4845-b052-78a4882fad4e' resource group and located in '(US) West US'. The 'Organization name' is set to 'OnlineStore'. The 'Administrator email' is '10727211@O365st.cycu.edu.tw'. The 'Pricing tier' is 'Consumption (99.95% SLA)'. On the right, the 'API Management' blade is open, showing the configuration for the 'Function App' 'ProductFunction9146e8ab3a'. The 'Display name' is 'ProductFunction9146e8ab3a', 'Name' is 'productfunction9146e8ab3a', 'API URL suffix' is 'products', and the 'Base URL' is 'https://productfunction9146e8ab3a-apim.azure-api.net/products'. Both blades have a 'Create' button at the bottom.

### 4. Test the OnlineStore products endpoint

The screenshot shows the 'ProductFunction9146e8ab3a | API Management' blade. Under the 'Test' tab, the 'Revision 1' section is selected. A red box highlights the 'GET ProductDetails' request, which returns a JSON response: 

```
vary: Accept-Encoding,Origin
{
  "ID": 3,
  "Name": "Smart Dimmer Switch",
  "Price": 59.99,
  "PartNumber": "DS728"
}
```

### 5. Test the OrderDetails function

The screenshot shows the 'OrderDetails | Code + Test' blade. Under the 'Developer' tab, the 'Code + Test' option is selected. A red box highlights the 'Logs' section, which displays the execution logs for the 'OrderFunction30124ea1de \ OrderDetails \ function.json' trigger: 

```
2021-11-29T10:12:03.378 [Information] Executing 'OrderDetails' (Reason="This function was programmatically called via the host APIs.", Id=e6fe1030-7bc1-4ffc-97f7-4fec2aafa5c5)
2021-11-29T10:12:03.378 [Information] CR HTTP trigger function processed a request.
2021-11-29T10:12:03.378 [Information] Executed 'OrderDetails' (Succeeded, Id=e6fe1030-7bc1-4ffc-97f7-4fec2aafa5c5, Duration=0ms)
```

 Below the logs, the 'Output' tab shows the 'HTTP response content': 

```
{
  "ID": 72945,
  "CustomerFirstName": "Yuki",
  "CustomerLastName": "Chiba",
  "Total": 442.5,
  "Shipped": false
}
```

## 6. Add a function to an existing API

The screenshot shows the Microsoft Azure API Management service configuration page. The top navigation bar includes the Microsoft Azure logo, a search bar, and a 'Home > All resources > OrderFunction30124ea1de >' breadcrumb trail. The main title is 'API Management service'.

The configuration form fields are as follows:

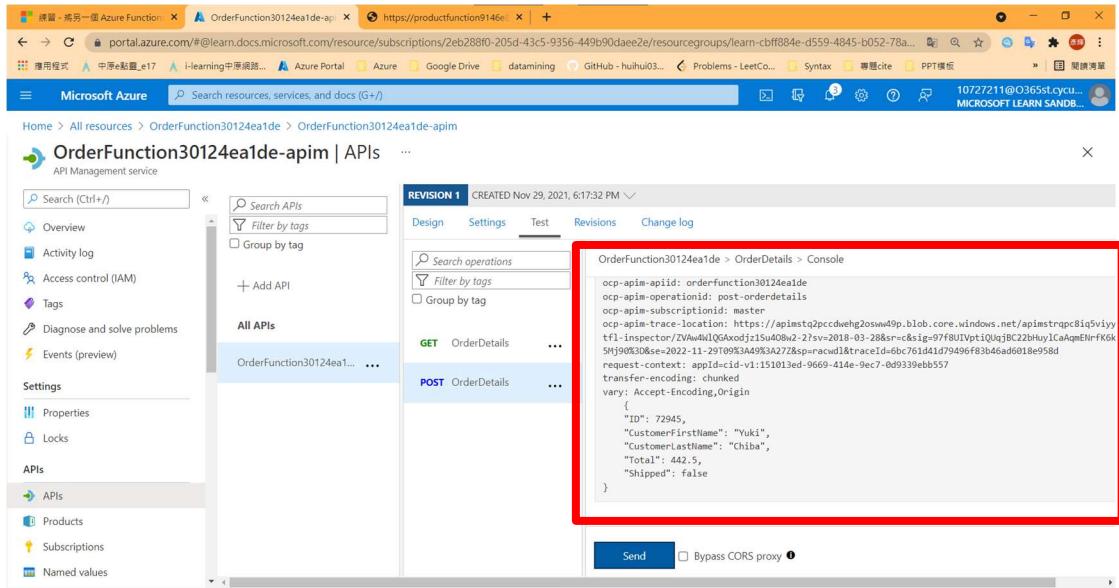
- Name:** OrderFunction30124ea1de-apim (with a green checkmark)
- Subscription:** Concierge Subscription (with a dropdown arrow)
- Resource group:** learn-cbff884e-d559-4845-b052-78a4882fad4e (with a dropdown arrow)
- Create new:** (link)
- Location:** (US) West US (with a dropdown arrow)
- Organization name:** OnlineStore (with a green checkmark)
- Administrator email:** 10727211@O365st.cycu.edu.tw (with a green checkmark)
- Pricing tier:** Consumption ( 99.95% SLA) (with a dropdown arrow)

A blue 'Export' button is located at the bottom left of the configuration area.

The bottom section shows the 'OrderFunction30124ea1de-apim | APIs' dashboard. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events (preview), Properties, Locks, and APIs (selected). The main content area displays a 'Create from Function App' dialog box. The dialog fields are:

- Basic** tab selected
- Function App:** OrderFunction30124ea1de
- Display name:** OrderFunction30124ea1de
- Name:** orderfunction30124ea1de
- API URL suffix:** orders
- Base URL:** https://orderfunction30124ea1de-apim.azure-api.net/orders
- Create** and **Cancel** buttons

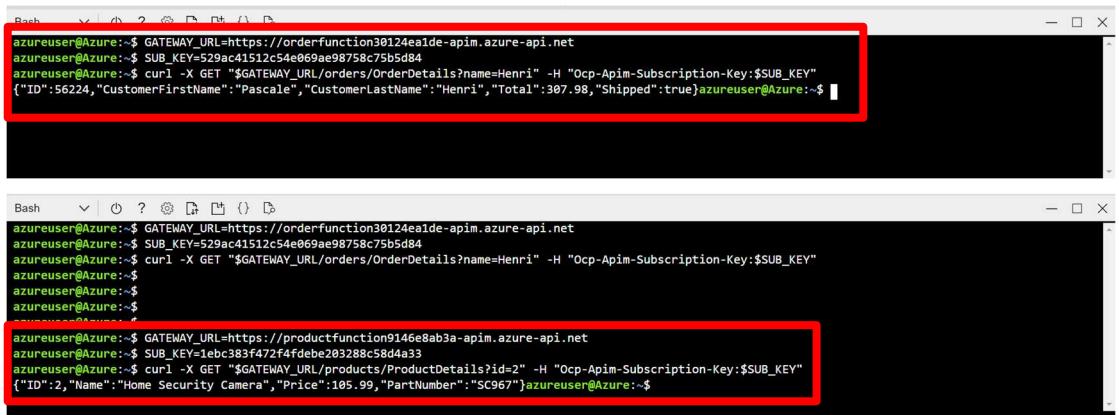
## 7. Test the OnlineStore orders endpoint in the portal



The screenshot shows the Microsoft Azure API Management service interface. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events (preview), Settings, APIs, Products, Subscriptions, and Named values. The main area is titled "OrderFunction30124ea1de-apim | APIs". It shows a revision history with one entry ("REVISION 1", created Nov 29, 2021, 6:17:32 PM). The "Test" tab is selected, showing a "Search operations" dropdown and a list of operations: "GET OrderDetails" and "POST OrderDetails". The "POST OrderDetails" operation is highlighted with a red box. The response body is displayed as a JSON object:

```
OrderFunction30124ea1de > OrderDetails > Console
{
  "ocp-apim-apid": "orderfunction30124ea1de",
  "ocp-apim-operationid": "post-orderdetails",
  "ocp-apim-subscriptionid": "master",
  "ocp-apim-trace-location": "https://apimsto2pcdwehg2oswew49p.blob.core.windows.net/apimstrqpc8ig5vyyf1l-inspector/ZWAwNlQGKxodjz1su409v2-27svs-2018-03-28&sra=c&sig=97f8UJvtIQUqjBC22hKuy1CaAqmENrFK6k5M190930&se=2022-11-29T09%3A09%3A27Z&sp=racwdl&ttraceId=d6bc761d41d7d9496f83b46ad6018e958d",
  "request-context": "appId=cid-v1:151013ed-9669-414e-9ec7-0d9339ebb557",
  "transfer-encoding": "chunked",
  "vary": "Accept-Encoding,Origin",
  "Content-Type": "application/json",
  "ID": 72945,
  "CustomerFirstName": "Yuki",
  "CustomerLastName": "Chiba",
  "Total": 442.5,
  "Shipped": false
}
```

## 8. Test the combined API



```
azureuser@Azure:~$ GATEWAY_URL=https://orderfunction30124ea1de-apim.azure-api.net
azureuser@Azure:~$ SUB_KEY=529ac41512c54e069ae98758c75b5d84
azureuser@Azure:~$ curl -X GET "$GATEWAY_URL/orders/OrderDetails?name=Henri" -H "Ocp-Apim-Subscription-Key:$SUB_KEY"
{"ID":56224,"CustomerFirstName":"Pascale","CustomerLastName":"Henri","Total":307.98,"Shipped":true}azureuser@Azure:~$
```

```
Bash
azureuser@Azure:~$ GATEWAY_URL=https://orderfunction30124ea1de-apim.azure-api.net
azureuser@Azure:~$ SUB_KEY=529ac41512c54e069ae98758c75b5d84
azureuser@Azure:~$ curl -X GET "$GATEWAY_URL/orders/OrderDetails?name=Henri" -H "Ocp-Apim-Subscription-Key:$SUB_KEY"
azureuser@Azure:~$ 
azureuser@Azure:~$ 
azureuser@Azure:~$ 
azureuser@Azure:~$ GATEWAY_URL=https://productfunction9146e8ab3a.apim.azure-api.net
azureuser@Azure:~$ SUB_KEY=1ebc383f472f4fdebe203288c58d4a33
azureuser@Azure:~$ curl -X GET "$GATEWAY_URL/products/ProductDetails?id=2" -H "Ocp-Apim-Subscription-Key:$SUB_KEY"
{"ID":2,"Name":"Home Security Camera","Price":105.99,"PartNumber":"SC967"}azureuser@Azure:~$
```

# Model 11: Build serverless apps with Go

## 1. Create a function app

Azure Functions: Create New Project	
function project workspace	{Part1}
Select a language	Custom Handler
Select a template for your first function	HttpTrigger
Provide a function name	hello
授權等級	Anonymous

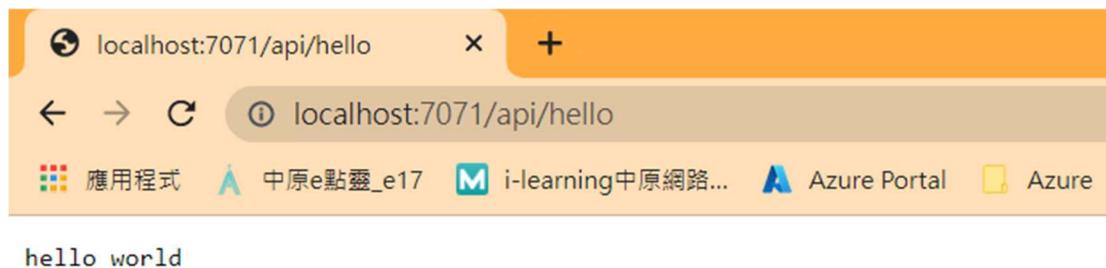
```
server.go 3  helloHandler
server.go U  server.go X  host.json U
server.go >  helloHandler
1 package main
2
3 import (
4     "fmt"
5     "io/ioutil"
6     "log"
7     "net/http"
8     "os"
9 )
10
11 func main() {
12     customHandlerPort, exists := os.LookupEnv("FUNCTIONS_CUSTOMHANDLER_PORT")
13     if !exists {
14         customHandlerPort = "8080"
15     }
16     mux := http.NewServeMux()
17     mux.HandleFunc("/api/hello", helloHandler)
18     fmt.Println("Go server Listening on: ", customHandlerPort)
19     log.Fatal(http.ListenAndServe(": "+customHandlerPort, mux))
20 }
21
22 func helloHandler(w http.ResponseWriter, r *http.Request) {
23     w.Header().Set("Content-Type", "application/json")
24     if r.Method == "GET" {
25         w.Write([]byte("hello world"))
26     } else {
27         body, _ := ioutil.ReadAll(r.Body)
28         w.Write(body)
29     }
30 }
```

## 2. config host.json & build execution

```
host.json - Part1 - Visual Studio Code
host.json U  server.go U  host.json X
host.json > ...
1 {
2     "version": "2.0",
3     "logging": {
4         "applicationInsights": {
5             "samplingSettings": {
6                 "isEnabled": true,
7                 "excludedTypes": "Request"
8             }
9         }
10    },
11    "extensibility": {
12        "id": "Microsoft.Azure.Functions.ExtensionBundle",
13        "version": "[2.0, 3.0.0)"
14    },
15    "customHandler": {
16        "description": {
17            "defaultExecutablePath": "./server.exe",
18            "workingDirectory": "",
19            "arguments": []
20        },
21        "enableForwardingHttpRequest" : true
22    }
23 }
```

```
問題  輸出  標準主控台  終端機
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/powershell
PS C:\Users\philip\Desktop\Azure\1129-go-demo\Part1> go build server.go
```

### 3. Run app



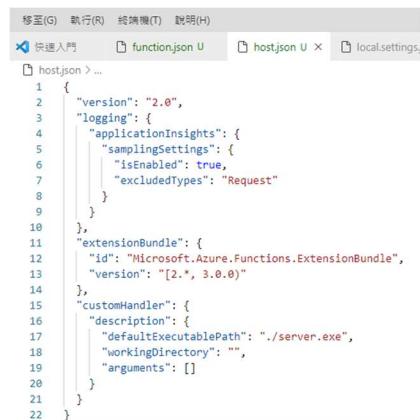
#### 1. Create a function app

Azure Functions: Create New Project	
Select the folder that will contain your function project	{Part2}
Select a language	Custom Handler
Select a template for your first function	HttpTrigger
Provide a function name	queueTrigger
授權等級	Anonymous

```
server.go - Part2 - Visual Studio
server.go > queueHandler
8
9 )
10
11 type InvokeRequest struct {
12     Data map[string]json.RawMessage
13     Metadata map[string]interface{}
14 }
15
16 func queueHandler(w http.ResponseWriter, r *http.Request) {
17     var invokeRequest InvokeRequest
18
19     d := json.NewDecoder(r.Body)
20     d.Decode(&invokeRequest)
21
22     var reqData map[string]interface{}
23     json.Unmarshal(invokeRequest.Data["queueItem"], &reqData)
24
25     var parsedMessage string
26     json.Unmarshal(invokeRequest.Data["queueItem"], &parsedMessage)
27     fmt.Println(parsedMessage) // your message
28 }
29
30
31 func main() {
32     customHandlerPort, exists := os.LookupEnv("FUNCTIONS_CUSTOMHANDLER_PORT")
33     if !exists {
34         customHandlerPort = "8080"
35     }
36     mux := http.NewServeMux()
37     mux.HandleFunc("/queueTrigger", queueHandler)
38     fmt.Println("Go server Listening on: ", customHandlerPort)
39     log.Fatal(http.ListenAndServe(": "+customHandlerPort, mux))
40 }
```

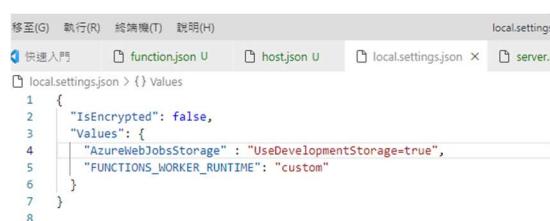
The image shows a screenshot of the Visual Studio Code editor. The title bar says 'server.go - Part2 - Visual Studio'. The editor window displays Go code for an Azure Function named 'queueHandler'. The code defines a struct 'InvokeRequest' with fields 'Data' and 'Metadata'. It then defines a function 'queueHandler' that takes an http.ResponseWriter and an http.Request. Inside the function, it decodes the request body into an 'InvokeRequest' struct, extracts the 'queueItem' from its 'Data' field, and then unmarshals it into a 'parsedMessage' string. Finally, it prints the message to the console. The 'main' function starts an http server on port 'FUNCTIONS\_CUSTOMHANDLER\_PORT' or '8080' if not specified. The code uses standard Go imports like 'os', 'http', and 'json'.

## 2. config host.json & build execution



```
host.json > ...
1  {
2    "version": "2.0",
3    "logging": {
4      "applicationInsights": {
5        "samplingSettings": {
6          "isEnabled": true,
7          "excludedTypes": "Request"
8        }
9      },
10     "extensionBundle": {
11       "id": "Microsoft.Azure.Functions.ExtensionBundle",
12       "version": "[2.*, 3.0.0)"
13     },
14     "customHandler": {
15       "description": {
16         "defaultExecutablePath": "./server.exe",
17         "workingDirectory": "",
18         "arguments": []
19       }
20     }
21   }
```

## 3. config local.settings.json



```
local.settings.json > {} Values
1  {
2    "IsEncrypted": false,
3    "Values": {
4      "AzureWebJobsStorage": "UseDevelopmentStorage=true",
5      "FUNCTIONS_WORKER_RUNTIME": "custom"
6    }
7  }
```

## 4. Start Queue Service



## 5. config function.json



```
queueTrigger > function.json > ...
queueTrigger > function.json > ...
1  {
2    "bindings": [
3      {
4        "name": "queueItem",
5        "type": "queueTrigger",
6        "direction": "in",
7        "queueName": "items",
8        "connection": "AzureWebJobsStorage"
9      },
10     {
11       "type": "http",
12       "direction": "out",
13       "name": "res"
14     }
15   ]
16 }
```

## 6. Create Queues in Azurite



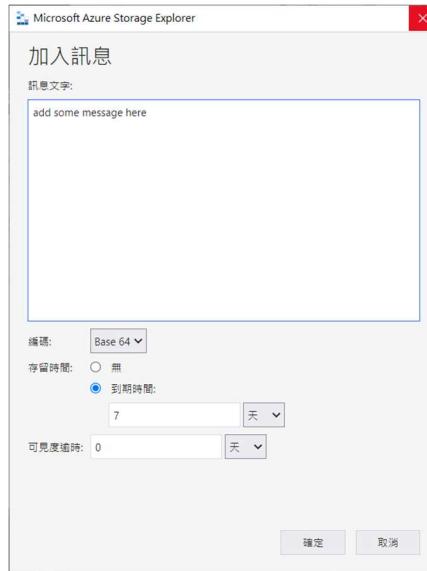
## 7. Run the app

```
問題 退出 指揮主控台 終端機
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/pscore6
PS C:\Users\phil8\Desktop\Azure\1129-go-demo\Part2> func start
Azure Functions Core Tools
Core Tools Version: 3.0.3904 Commit hash: c345f7140a8f968c5dbc621f8a8374d8e3234206 (64-bit)
Function Runtime Version: 3.3.1.0

Functions:
queueTrigger: queueTrigger

For detailed output, run func with --verbose flag.
[2021-11-29T14:24:51.231Z] Go server Listening on: 56075
[2021-11-29T14:24:51.639Z] Worker process started and initialized.
```

## 8. Add message in queue



## 9. Results

```
direction": "in",
"queueName": "items",
"connection": "AzureWebJobsStorage"
},
{
  "type": "http",
  "direction": "out",
  "name": "res"
}
]

[...]
dous PowerShell
right (C) Microsoft Corporation. 著作權所有，並保留一切權利。
試新的跨平台 PowerShell https://aka.ms/pscore6
C:\Users\phil8\Desktop\Azure\1129-go-demo\Part2> func start
e Tools Version: 3.0.3904 Commit hash: c345f7140a8f968c5dbc621f8a8374d8e32
ction Runtime Version: 3.3.1.0

ctions:
queueTrigger: queueTrigger

detailed output, run func with --verbose flag.
[2021-11-29T14:24:51.231Z] Go server Listening on: 56075
[2021-11-29T14:24:51.639Z] Worker process started and initialized.
[2021-11-29T14:26:13.475Z] Trigger Details: MessageId: 53bb71eb-02e6-4fce-be9a-747c94c2d5bf on "New queue message"
[2021-11-29T14:26:13.475Z] "add some message here"
```

## Take screenshots of Badges and Trophies

The screenshot shows a user profile for Yan-Hui, Lin. At the top, there's a circular placeholder for a profile picture, followed by the name 'Yan-Hui, Lin' and the email '10727211@O365st.cycu.edu.tw'. Below the name is a blue progress bar labeled 'LEVEL 7' with a green segment indicating 40450/48499 XP. Underneath the profile information, there are statistics: 28 Badges, 4 Trophies, 1 Reputation, 0 Answers accepted, 0 Following, and 0 Followers. The main area is divided into sections: 'Activity' (Training, Certifications, Q&A, Achievements), 'Trophies' (Store data in Azure, Create serverless applications, Administer containers in Azure, Azure Fundamentals part 1: Describe core Azure concepts), and 'Badges'.

Activity	Trophies	Badges
Training		
Certifications		
Q&A	<b>TROPHY</b> Store data in Azure Completed on 11/27/2021	<b>TROPHY</b> Create serverless applications Completed on 11/23/2021
Achievements		
Collections		
	<b>TROPHY</b> Administer containers in Azure Completed on 11/21/2021	<b>TROPHY</b> Azure Fundamentals part 1: Describe core Azure concepts Completed on 9/18/2021
Badges		

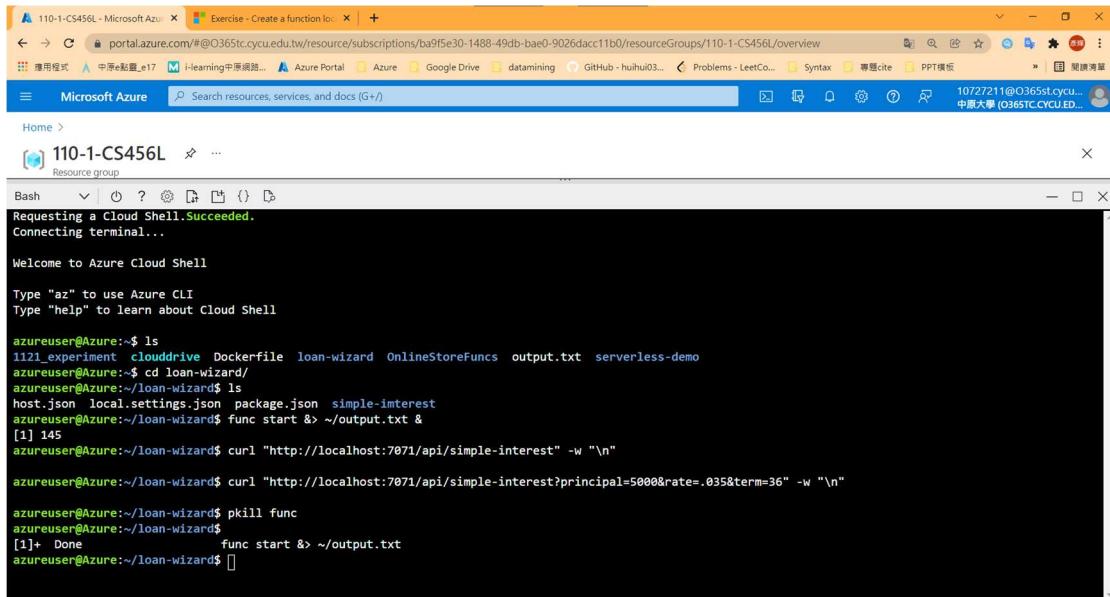
## Learned from the Learning Path

在此 Learning Path 中充分實作了 Azure Function 的各種功能，了解 Serverless 的概念，也從 Azure Function 的 Trigger 中看到 Azure 中的其他資源，透過不同的 Trigger 方法搭配不同的 Azure Resource，甚至可以搭配 IDE 做資源的佈署，使得 Azure Function 能實作的功能越來越多，進而看見 Microsoft Azure 的強大生態圈。

# Problems

## 1. Model6

在 Model6 中，使用 Azure Cloud Shell 在背景執行 Azure Functions 時發現錯誤，同樣的方法使用在沙盒中卻能正常運行。

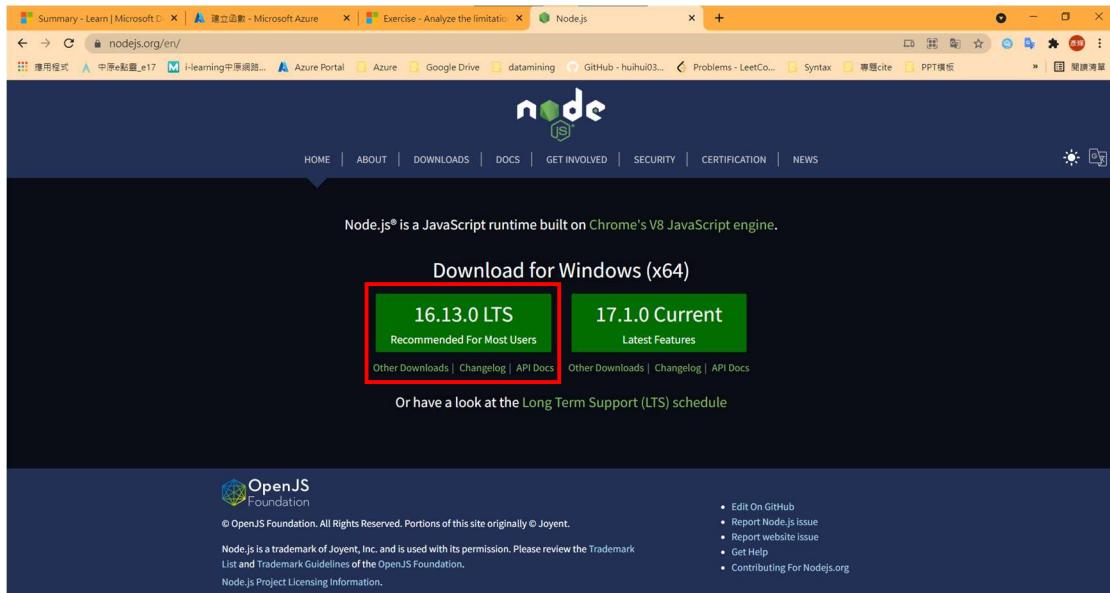


The screenshot shows the Microsoft Azure Cloud Shell interface. At the top, it says "Requesting a Cloud Shell. Succeeded." followed by "Connecting terminal...". Below that, it says "Welcome to Azure Cloud Shell". It then displays a terminal session with the following commands and output:

```
azurouser@Azure:~$ ls
1121_experiment clouddrive Dockerfile loan-wizard OnlineStoreFuncs output.txt serverless-demo
azurouser@Azure:~$ cd loan-wizard/
azurouser@Azure:~/loan-wizard$ ls
host.json local.settings.json package.json simple-interest
azurouser@Azure:~/loan-wizard$ func start &> ~/output.txt &
[1] 145
azurouser@Azure:~/loan-wizard$ curl "http://localhost:7071/api/simple-interest" -w "\n"
azurouser@Azure:~/loan-wizard$ curl "http://localhost:7071/api/simple-interest?principal=5000&rate=.035&term=36" -w "\n"
azurouser@Azure:~/loan-wizard$ pkill func
azurouser@Azure:~/loan-wizard$ [1]+ Done func start &> ~/output.txt
azurouser@Azure:~/loan-wizard$ 
```

## 2. Model9

需要下載 Node.js 以接續教程。但官方給的 Node.js 載點中 <https://nodejs.org/en/> 會安裝到最新版本的 Node.js、npm（如下圖為 16.13LTS）



但此版本將無法與 Azure Core Functions 相容，在 Start Local Function 將出現 Error。

```

public > index.html.js > ...
    ...
46
47     connection.start().then(() => {
48         console.log("SignalR connection established");
49     });
50 };
51
52 connect();

```

正在執行: func host start <

Azure Functions Core Tools  
Core Tools Version: 3.0.3904 Commit hash: c345f7140a0ff968c5dbc621fb8a874d8e3234206 (64-bit)  
Function Runtime Version: 3.3.1.0

[2021-11-23T13:23:59.798Z] Unsupported service transport type: . Use default Transient instead.  
[2021-11-23T13:23:59.798Z] Debugger listening on ws://127.0.0.1:9229/f2049fb8-d009-4a61-869c-853e3d5c4063  
[2021-11-23T13:23:59.802Z] For help, see: https://nodejs.org/en/docs/inspector  
C:\Program Files\Microsoft\Azure Functions Core Tools\workers\node\dist\nodejs\worker.js:15  
[2021-11-23T13:23:59.802Z] throw message;  
[2021-11-23T13:23:59.802Z] C:\Program Files\Microsoft\Azure Functions Core Tools\workers\node\dist\src\nodejsWorker.js:35  
[2021-11-23T13:23:59.802Z] throw message;  
[2021-11-23T13:23:59.802Z] Incompatible Node.js version (v16.13.0). Refer to our documentation to see the Node.js versions supported by each version of Azure Functions: https://aka.ms/functions-node-versions  
[2021-11-23T13:24:03.850Z] Host lock lease acquired by instance ID '00000000000000000000000000000000'.

依序網址來到相容性說明網頁

<https://docs.microsoft.com/zh-tw/azure/azure-functions/functions-reference-node?tabs=v2#node-version>

節點版本		
Functions 版本	節點版本 (Windows)	Linux (節點版本)
4.x	~14	node 14
3.x (建議的)	~14 (建議的) ~12 ~10	node 14 (建議的) node 12 node 10
2.x	~12 ~10 ~8	node 10 node 8
1.x	6.11.2 (由執行階段鎖定)	n/a

將 Node.js 降版後即可解決問題(v14) <https://nodejs.org/download/release/v14.18.1/>

### 3. Model 9

在拿 SingalR key String 時，範例程式碼中的 query[0]有可能導致 Error（將 index 寫死，將拿到第一個 SignalR Resource 的 key，但「第一個 SignalR Resource」並非剛創立的 Resource）

```

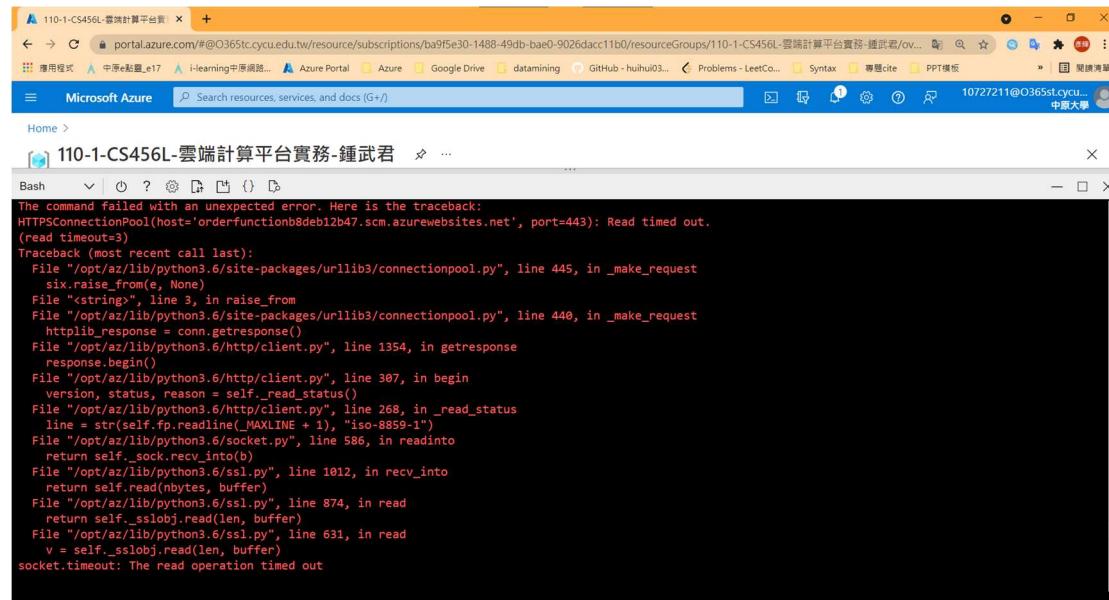
SIGNALR_CONNECTION_STRING=$(az signalr key list \
--name $az_signalr_list \
--resource-group 110-1-CS456 \
--query [0].name -o tsv \
--resource-group 110-1-CS456 \
--query primaryConnectionString -o tsv)

printf "\n\nReplace <SIGNALR_CONNECTION_STRING> with:\n$SIGNALR_CONNECTION_STRING\n\n"

```

#### 4. Model 10

在 110-1-456L 使用官方提供之腳本時出現 Error，原因不明，環境更換為沙盒時即可恢復運作。



```
The command failed with an unexpected error. Here is the traceback:  
HTTPConnectionPool(host='orderfunction88debb12b47.scm.azurewebsites.net', port=443): Read timed out.  
(read timeout=3)  
Traceback (most recent call last):  
  File "/opt/az/lib/python3.6/site-packages/urllib3/connectionpool.py", line 445, in _make_request  
    six.raise_from(e, None)  
  File "cStringIO", line 3, in raise_from  
  File "/opt/az/lib/python3.6/site-packages/urllib3/connectionpool.py", line 440, in _make_request  
    httpplib_response = conn.getresponse()  
  File "/opt/az/lib/python3.6/http/client.py", line 1354, in getresponse  
    response.begin()  
  File "/opt/az/lib/python3.6/http/client.py", line 307, in begin  
    version, status, reason = self._read_status()  
  File "/opt/az/lib/python3.6/http/client.py", line 268, in _read_status  
    line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-1")  
  File "/opt/az/lib/python3.6/socket.py", line 586, in readinto  
    return self._sock.recv_into(b)  
  File "/opt/az/lib/python3.6/ssl.py", line 1012, in recv_into  
    return self.read(nbytes, buffer)  
  File "/opt/az/lib/python3.6/ssl.py", line 874, in read  
    return self._sslobj.read(len, buffer)  
  File "/opt/az/lib/python3.6/ssl.py", line 631, in read  
    v = self._sslobj.read(len, buffer)  
socket.timeout: The read operation timed out
```

#### 5. Model11 – Unit4

在 Windows 環境下，路徑使用'\'將 Error，應改為'/'。

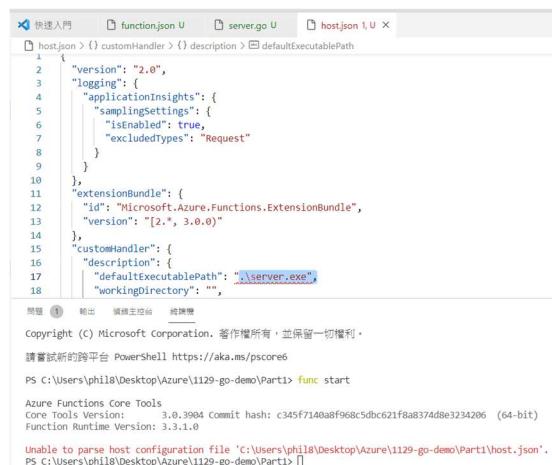
- From a terminal, run `go build server.go` in the project root:



```
Go  
Copy  
go build server.go
```

This step creates an executable file that's called `server` on macOS and Linux, or `server.exe` on a Windows OS.

- Open the `host.json` file and find the `defaultExecutablePath` element inside the `customHandler` one. Specify `./server` on macOS and Linux, or `.\server.exe` on a Windows OS.



```
1  {  
2      "version": "2.0",  
3      "logging": {  
4          "applicationInsights": {  
5              "samplingSettings": {  
6                  "isEnabled": true,  
7                  "excludedTypes": "Request"  
8              }  
9          }  
10     },  
11     "extensionBundle": {  
12         "id": "Microsoft.Azure.Functions.ExtensionBundle",  
13         "version": "[2.+", 3.0.0)"  
14     },  
15     "customHandler": {  
16         "description": {  
17             "defaultExecutablePath": ".\server.exe",  
18             "workingDirectory": ""  
19         }  
20     }  
21 }
```

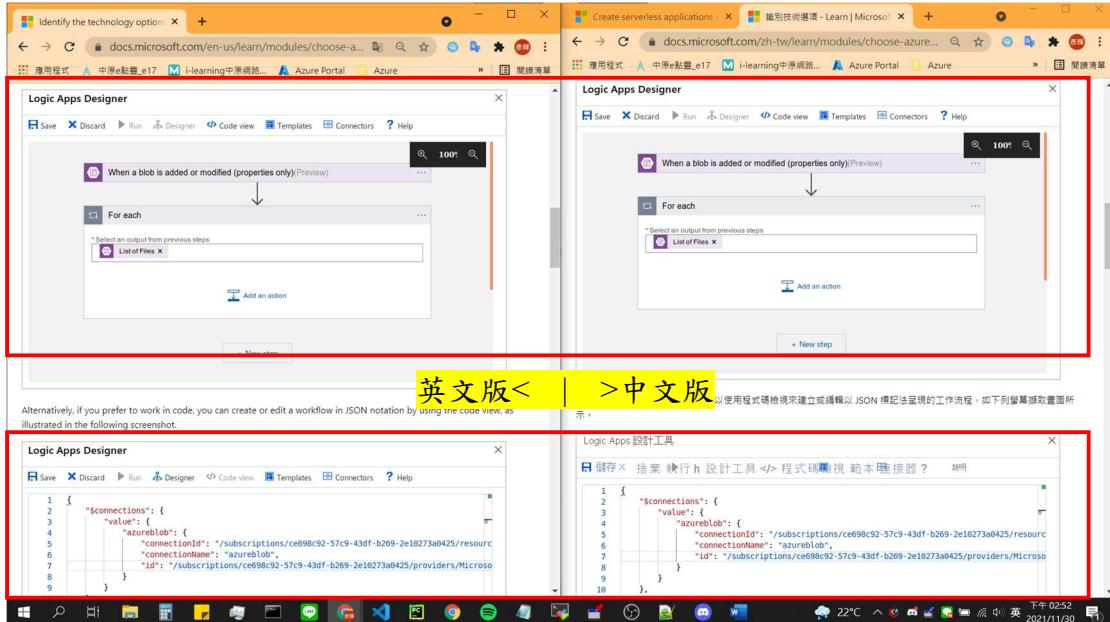
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。  
請嘗試新的跨平台 PowerShell <https://aka.ms/powershell>  
PS C:\Users\philip\Desktop\Azure\1129-go-demo\Part1> func start  
Azure Functions Core Tools  
Core Tools Version: 3.0.3994 Commit hash: c345f7140a0ff968c5dbc621f8a8374d8e3234206 (64-bit)  
Function Runtime Version: 3.3.1.0  
Unable to parse host configuration file 'C:\Users\philip\Desktop\Azure\1129-go-demo\Part1\host.json'.  
PS C:\Users\philip\Desktop\Azure\1129-go-demo\Part1>

# FeedBack

## 中文翻譯改進

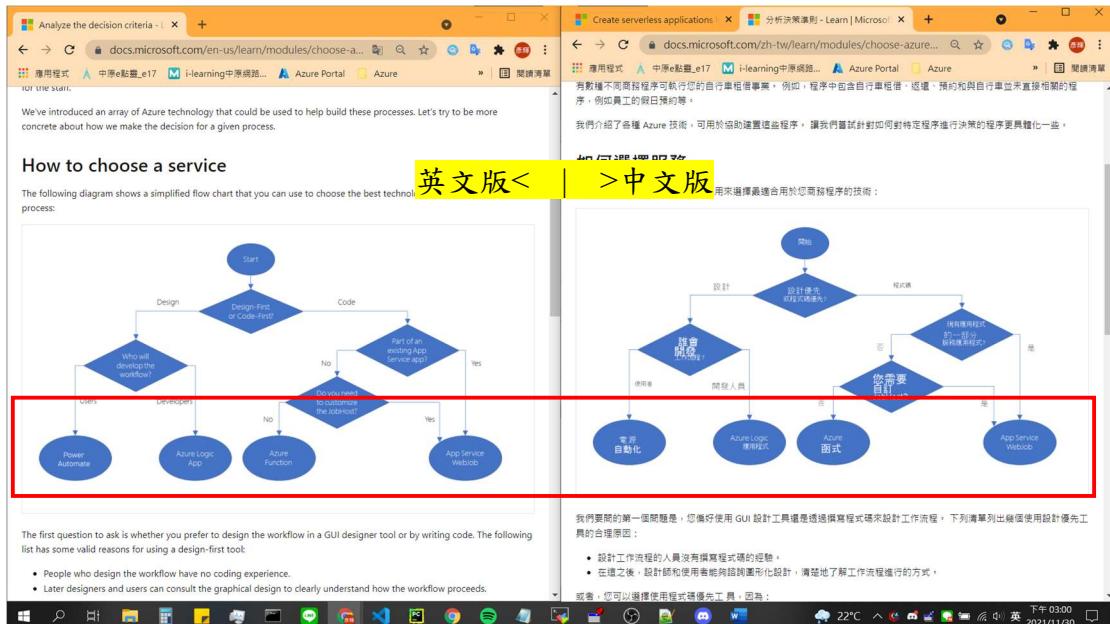
### 1. Model1 – Unit2

左邊為英文版，右邊為中文版，過程中看到教程會穿插圖片做解說，但解說圖片的文字翻譯程度卻不統一，翻譯過後的可讀性甚至更低。



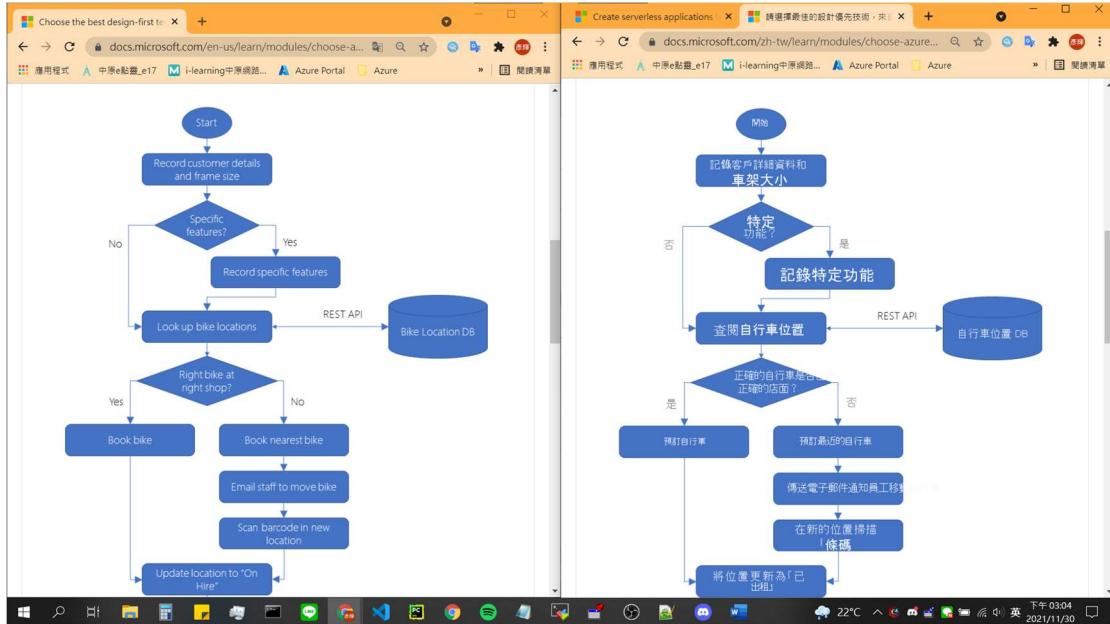
### 2. Model1 – Unit3

產品名稱經過翻譯可讀性降低。



### 3. Model1 – Unit4

排版。



### 4. Model2 – Unit4

服務名稱要統一英文版本，在 Azure Portal 上創建名稱時，服務為英文版本。

Service	Trigger description
Blob Storage	Starts a function when a new or updated blob is detected.
Azure Cosmos DB	Start a function when inserts and updates are detected.
Event Grid	Starts a function when an event is received from Event Grid.
HTTP	Starts a function with an HTTP request.
Microsoft Graph Events	Starts a function in response to an incoming webhook from the Microsoft Graph. Each instance of this trigger can react to one Microsoft Graph resource type.
Queue Storage	Starts a function when a new item is received on a queue. The queue message is provided as input to the function.
Service Bus	Starts a function in response to messages from a Service Bus queue.
Timer	Starts a function on a schedule.

觸發器

函式是由事件驅動，這表示它們是為回應事件而執行。啟動函式的事件種類稱為「觸發器」（trigger）。每個函式都必須只設定一個觸發器。

Azure 支援下列服務的觸發器。

觸發器	觸發器說明
Blob 儲存體	偵測到新的或更新的 Blob 時，啟動函式。
Azure Cosmos DB	偵測到插入及更新時，啟動函式。
Event Grid	接收到来自「事件网格」的事件時，啟動函式。
HTTP	使用 HTTP 要求啟動函式。
Microsoft Graph 事件	啟動函式以回應從 Microsoft Graph 導入的 Webhook。僅座標觸發程序的每個執行個體都可回應一種 Microsoft Graph 事件類別。
佇列儲存體	在佇列上收到新項時，啟動函式。佇列訊息會當成函式輸入提供。
服務匯流排	啟動函式以回應來自服務匯流排佇列的訊息。
計時器	定期啟動函式。

建立函數

選取範本

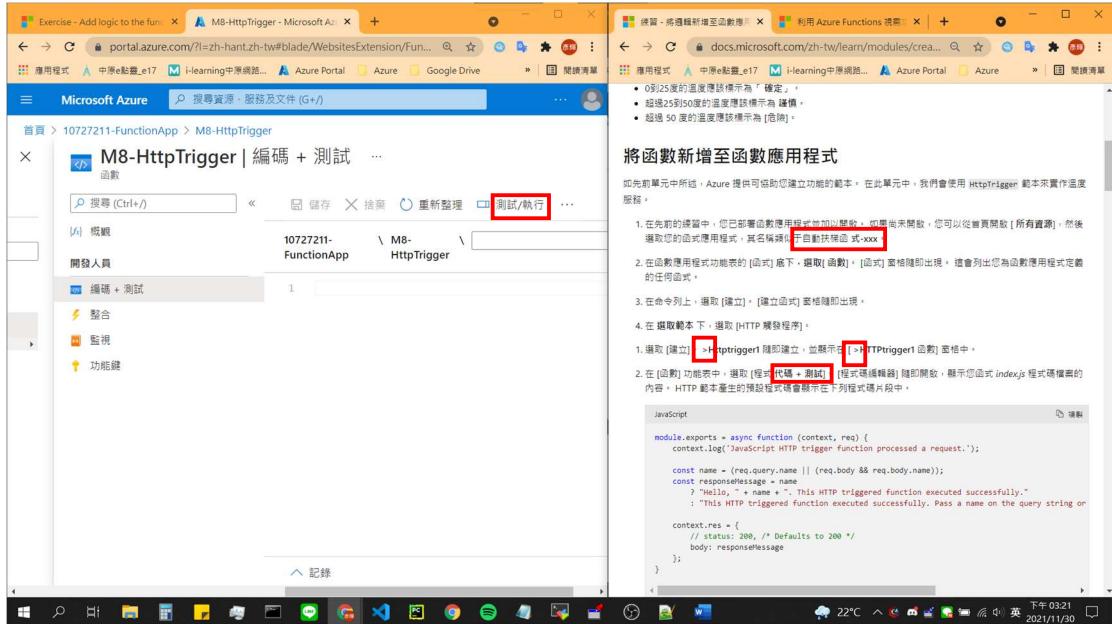
使用範本建立函式。觸發器描述叫用函式之事件的類型。深入了解

範本

HTTP trigger  
Timer trigger  
Azure Queue Storage trigger  
Azure Service Bus Queue trigger  
Azure Service Bus Topic trigger  
Azure Blob Storage trigger  
Azure Event Hub trigger

## 5. Model2 – Unit5

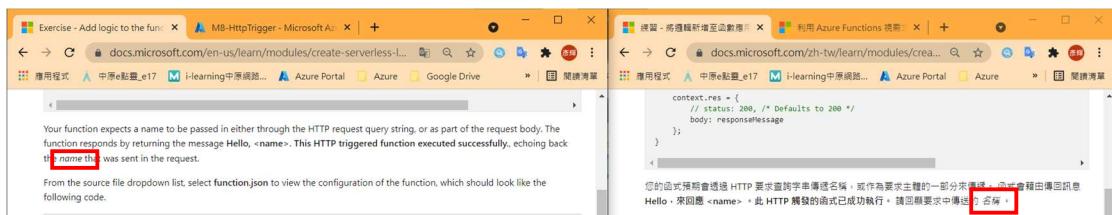
名稱不統一。



The screenshot shows two browser windows. The left window is the Microsoft Azure portal showing the 'M8-HttpTrigger' function configuration. The right window is a Microsoft Learn article titled '將函數新增至函數應用程式' (Add a function to a function app). The article contains instructions and a code editor showing a sample JavaScript file for an HTTP trigger function.

## 6. Model2 – Unit5

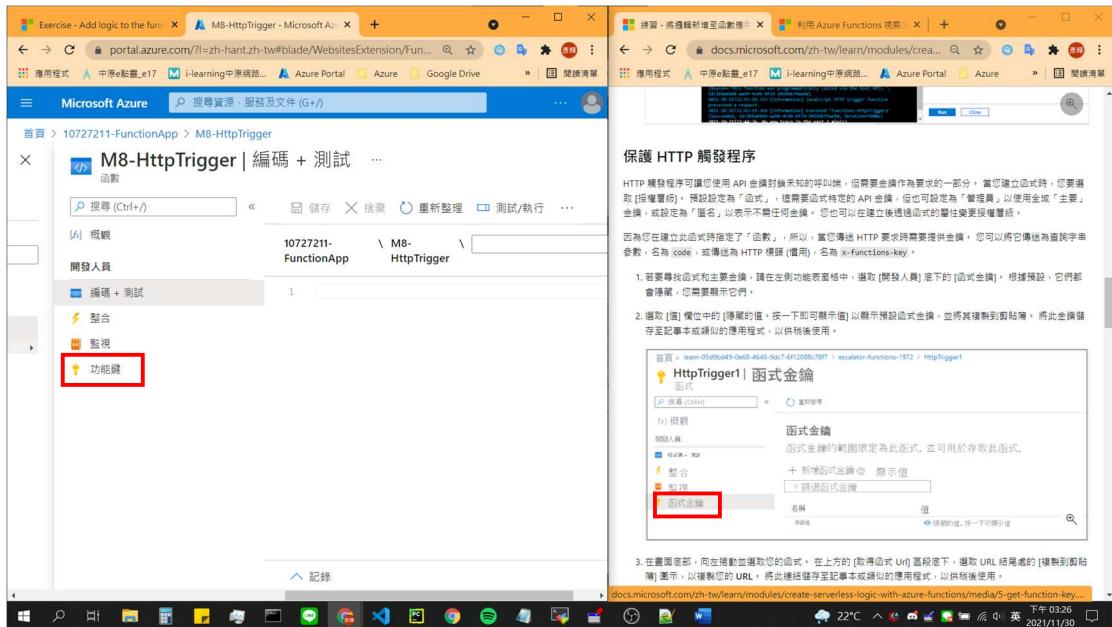
變數名稱被翻譯。



The screenshot shows two browser windows. The left window is a Microsoft Learn article showing a note about function parameters. The right window is another Microsoft Learn article with a specific section highlighted, showing a note about the function returning the HTTP request query string or as part of the request body.

## 7. Model2 – Unit5

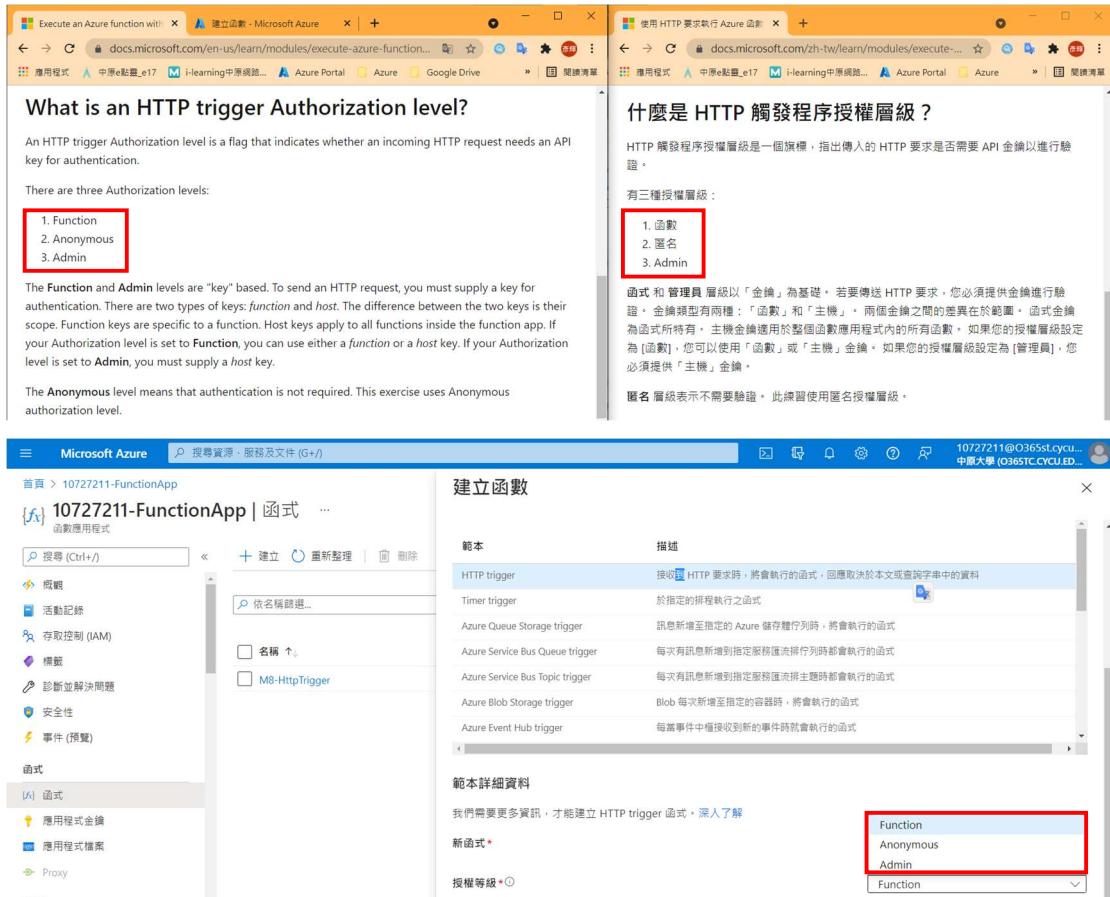
名稱不統一。



The screenshot shows two browser windows. The left window is the Microsoft Azure portal showing the 'M8-HttpTrigger' function configuration with the 'Function Keys' tab highlighted. The right window is a Microsoft Learn article titled '保護 HTTP 觸發器' (Protect an HTTP trigger), which includes a link to 'HTTP觸發器' (HTTP trigger).

## 8. Model3 – Unit2

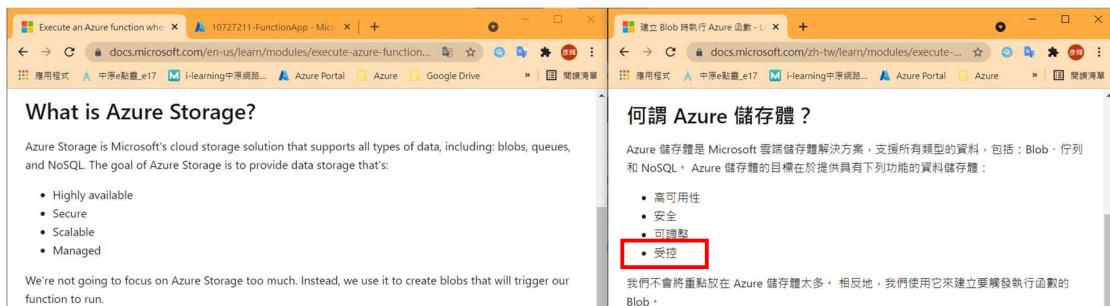
服務名稱要統一英文版本，在 Azure Portal 上創建名稱時，服務為英文版本。



The screenshot shows the Microsoft Azure portal interface. On the left, the left-hand navigation pane is visible with various service icons like Functions, Storage, and App Services. The main workspace is titled '建立函數' (Create Function). The '範本' (Template) section has 'HTTP trigger' selected. Below it, there's a detailed description of what an HTTP trigger does. To the right, there's a '範本詳細資料' (Template Details) section with a note about needing more information to create the function. At the bottom, there's a '權等級' (Authorization Level) dropdown menu where the 'Function' option is highlighted with a red box.

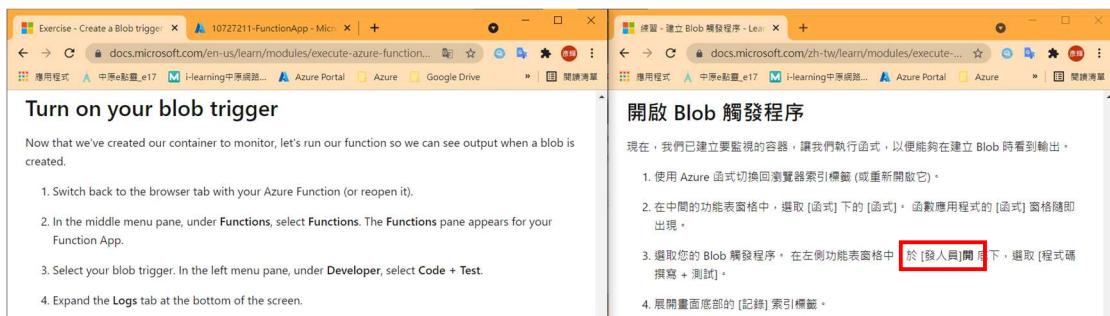
## 9. Model 3 – Unit7

「管理」也許是個更好的翻譯。



The screenshot shows the Microsoft Azure portal interface. On the left, the left-hand navigation pane is visible with various service icons like Functions, Storage, and App Services. The main workspace is titled '建立儲存體' (Create Storage Account). The '何謂 Azure 儲存體?' (What is Azure Storage?) section contains a brief introduction to Azure Storage. Below it, there's a '受控' (Controlled by) dropdown menu where the '受控' (Controlled) option is highlighted with a red box.

## 10. Model3 – Unit8



The screenshot shows the Microsoft Azure portal interface. On the left, the left-hand navigation pane is visible with various service icons like Functions, Storage, and App Services. The main workspace is titled 'Turn on your blob trigger'. It contains a numbered list of steps to run the function. To the right, there's a '開啟 Blob 觸發程序' (Enable Blob Trigger) section with detailed instructions. Step 3 in the list is highlighted with a red box.

## 11. Model4 – Unit5

變數名稱不能被翻譯&有部分內容未被翻譯

3. Enter the following values for each setting.

Setting	Value	Description
Database id	Select Create new, and enter func-io-learn-db for the database id	Database names can be 1 to 255 characters long, and cannot contain /, #, ?, or a trailing space.
Database Max RU/s	4000	Accept the default throughput of 4000 request units per second (RU/s). To reduce latency, you can scale up the performance later.
Container id	Bookmarks	Container IDs have the same character requirements as database names.
Partition key	/id	The partition key specifies how the documents in Azure Cosmos DB collections are distributed across logical data partitions. You'll use the Partition key setting as a convenience because you're not concerned with database performance in this module. To learn more about Azure Cosmos DB partition key strategies, explore the Microsoft Learn Azure Cosmos DB modules.

Accept the defaults for all the other settings.

4. Scroll to the bottom of the pane and select OK. Allow a few minutes for the database and container to be built.

When complete, the Data Explorer displays func-io-learn-db in DATA under SQL API.

5. Select func-io-learn-db to expand it. Notice that your func-io-learn-db database contains several child members, including Scale and Bookmarks.

## 12. Model4 – Unit5

翻譯詞要統一。

1. In the command bar, select Get Function Url. The Get Function Url dialog box appears.

2. Select default (function key) from the dropdown list, then select the Copy to clipboard icon, and select OK.

3. Paste the function URL you copied into the address bar of a new browser tab. Append the query string value &name=<your name> to the end of the URL, replacing <your name> with your name, and then press Enter. The Azure function should return a personalized response in the browser.

Now that we have our skeletal function working, let's turn our attention to reading data from your Azure Cosmos DB, or in our scenario, from your Bookmarks container.

Add an Azure Cosmos DB input binding

## 13. Model4 – Unit5

有部分內容未被翻譯。

```
body : "No bookmarks found",
headers: {
    'Content-Type': 'application/json'
}
}

context.done();
```

3. In the command bar, select Save. The Logs pane appears, showing you have Connected!

Let's examine what this code is doing.

- An incoming HTTP request triggers the function, and an `id` query parameter is passed to the Azure Cosmos DB input binding.
- If the database finds a document that matches this ID, the `bookmark` parameter will be set to the located document.

In this example, the code constructs a response that contains the URL value that is found in the corresponding document of the database.

- If no document is found matching this key, the request would respond with a payload and status code that tells the user the bad news.

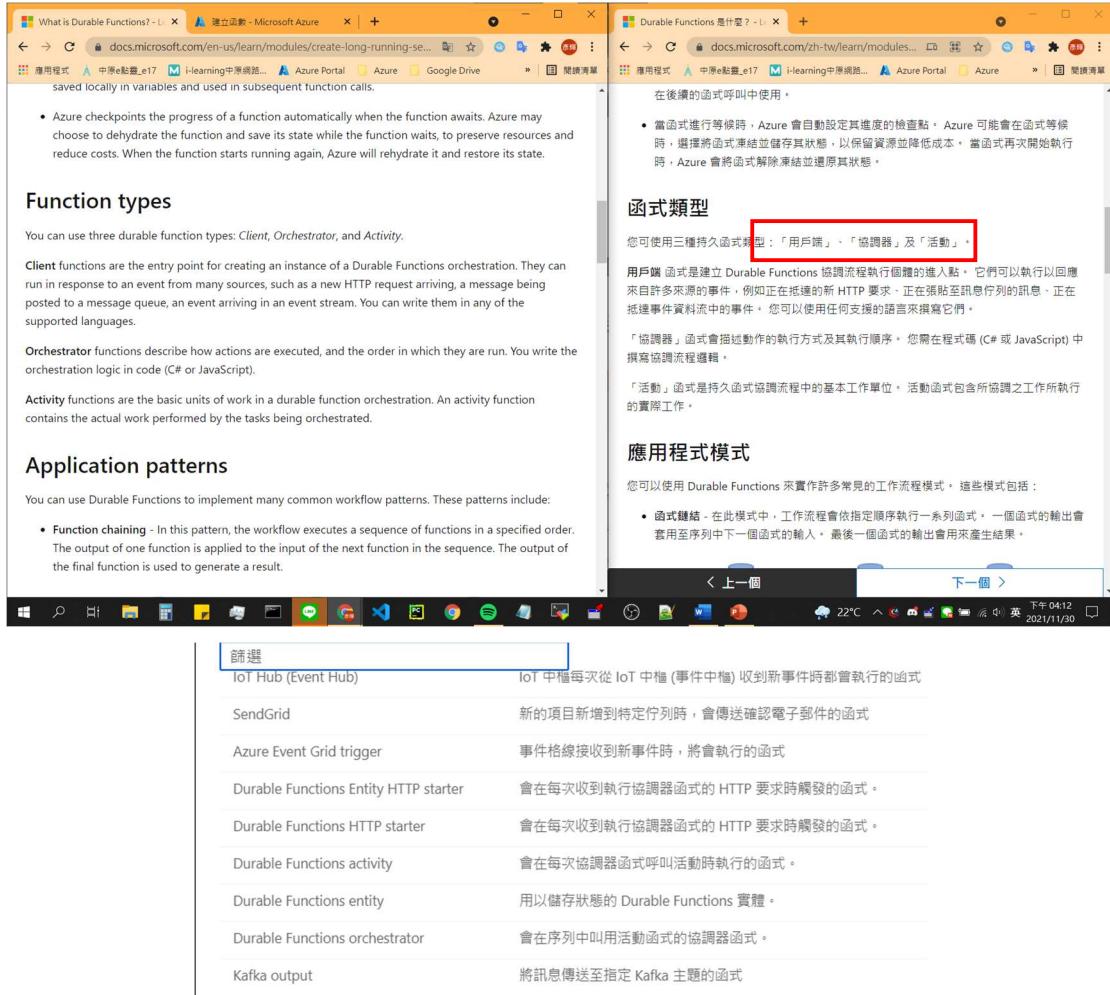
```
else {
    context.res = {
        status: 404,
        body : "No bookmarks found",
        headers: {
            'Content-Type': 'application/json'
        }
    }
    context.done();
}
```

讓我們來檢驗此程式碼的作用。

- 傳入 HTTP 要求隨即觸發函式，而且系統會將 `id` 查詢參數傳遞至 Azure Cosmos DB 輸入繫結。
- 如果資料庫找到符合此識別碼的文件，則會將 `bookmark` 參數設為所找到的文件。
- 在此範例中，程式碼會建立回應，其中包含在資料庫的對應文件中找到的 URL 值。
- 如果找不到符合索引鍵的文件，您會以承載和狀態碼回應，告知使用者壞消息。

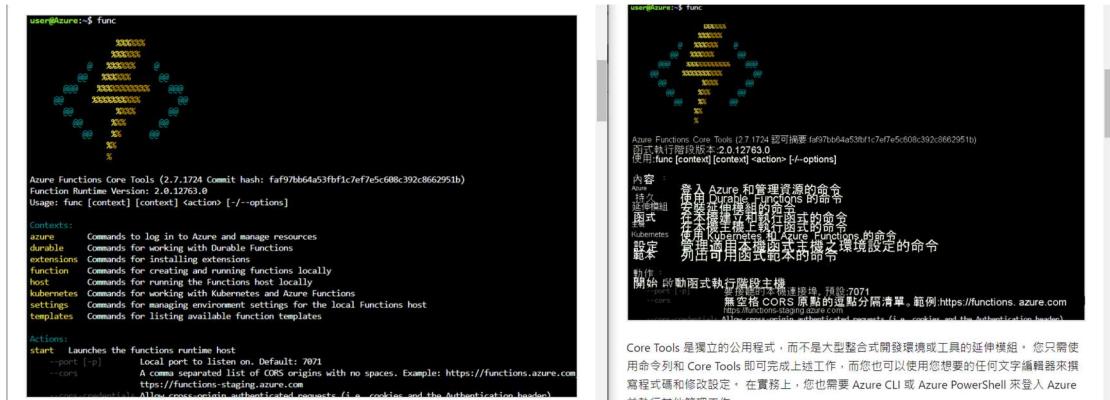
## 14. Model5 – Unit2

服務名稱要統一英文版本，在 Azure Portal 上創建名稱時，服務為英文版本。



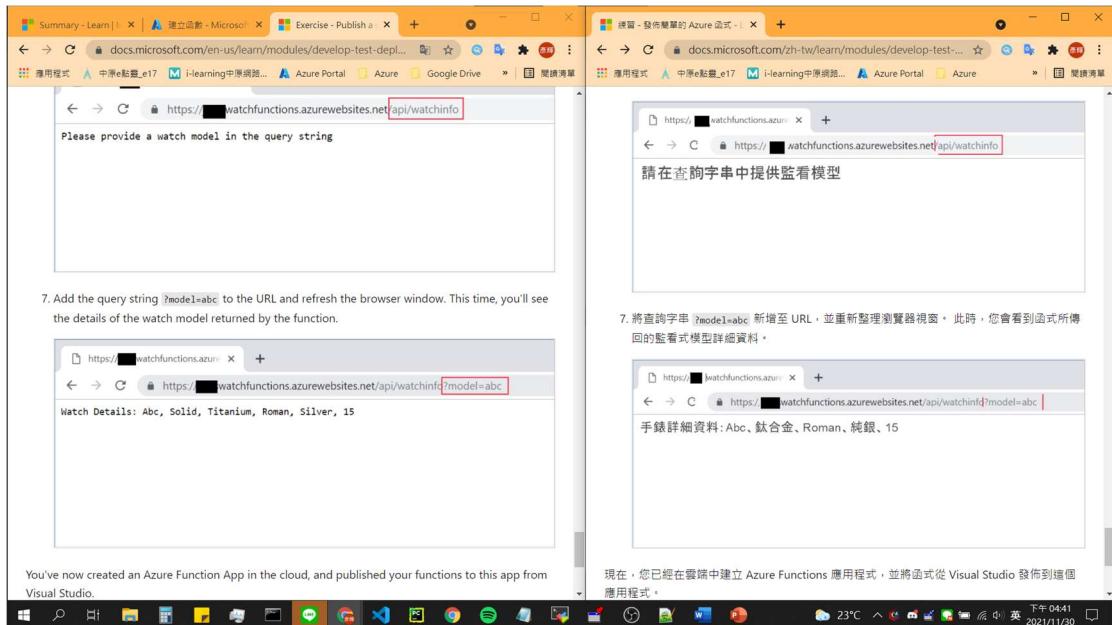
## 15. Model6 – Unit2

log 可以不必翻譯



## 16. Model7 – Unit5

過度翻譯。



## 17. Model10 – Unit3

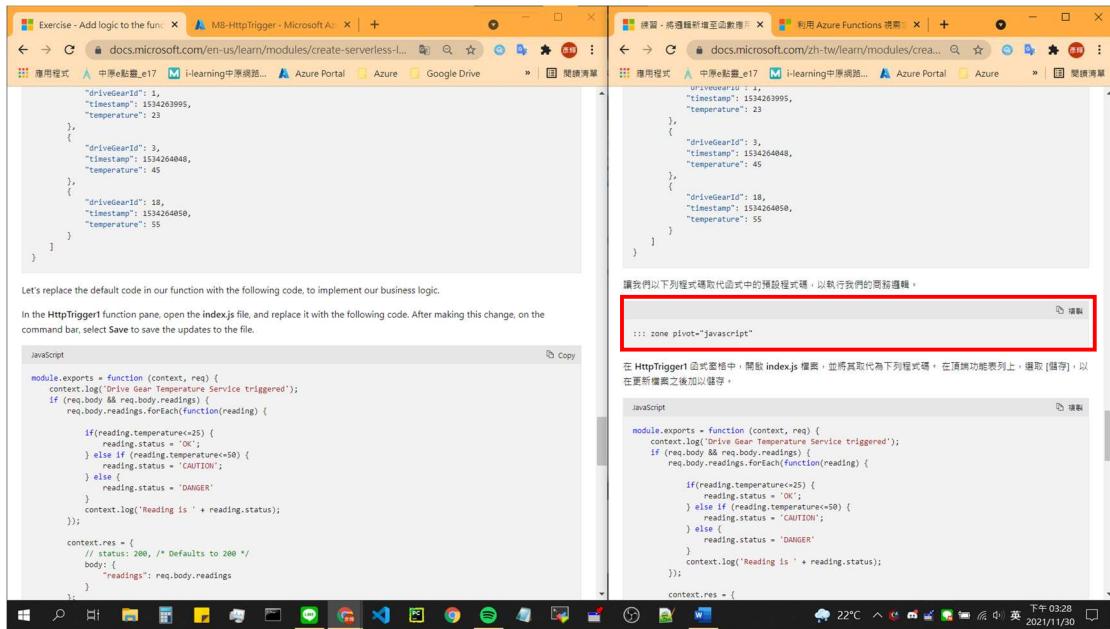
變數名稱不應被翻譯。

## 18. Model10 – Unit3

變數名稱不應被翻譯。

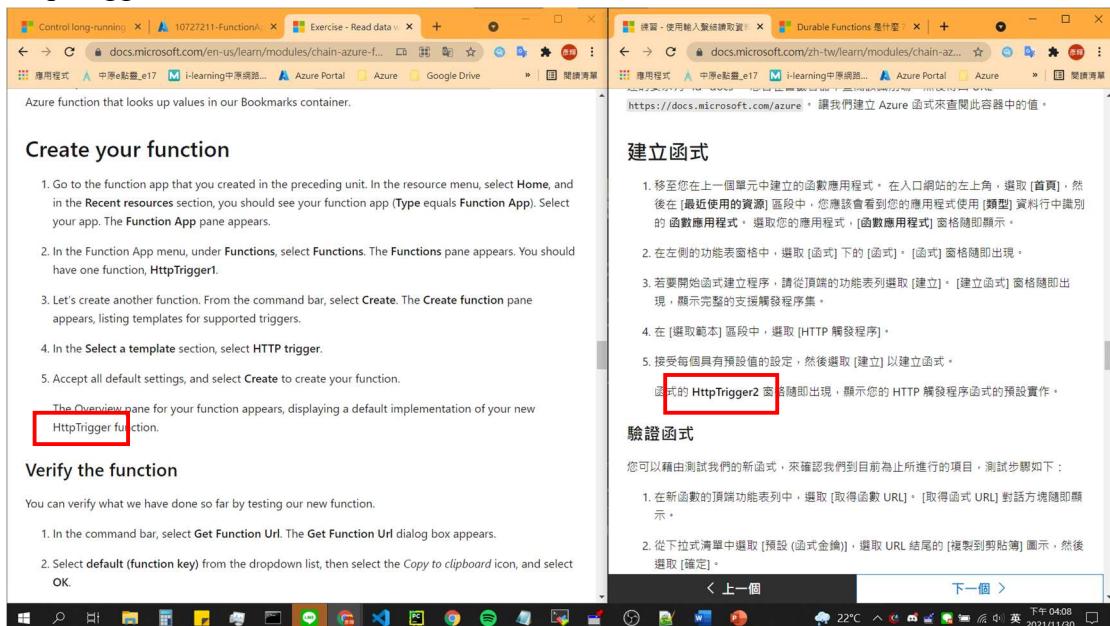
# 網頁 Bug

## 1. Model2 – Unit5



## 2. Model4 – Unit5

在 Unit3 創造的 HttpTrigger 並沒有接續著使用，在 Unit4 又創造一個 HttpTrigger & 翻譯錯誤



## 3. Model7 – Unit5

### 格式錯誤



## 4. Model9 – Unit7

應更正為以下格式

The screenshot displays two Microsoft Learn pages in a split-screen view. The left page is titled "Exercise - Use a storage account" and the right page is titled "練習 - 使用儲存體帳戶來裝載新". Both pages have similar navigation bars at the top, including Microsoft, Docs, Documentation, Learn, Q&A, Code Samples, Shows, Events, and Learn products like Products, Roles, Educator Center, Learn TV, Certifications, and FAQ & Help. Below the navigation, both pages show a breadcrumb trail: Docs / Learn / Browse / Create serverless applications / Enable automatic updates in a web application using Azure Functions. The main content area for both pages is titled "Deploy the function app". The left page provides instructions for opening the Visual Studio Code command palette (F1) and selecting the "Azure Functions: Deploy to Function App" command. The right page also provides similar instructions. Both pages include a table for providing deployment information, with columns for Name and Value. The right page's table includes fields for Function app name, OS, Plan, Language, Resource group, and Storage account. The right page also includes a note about machine translation and a list of steps: 1. 搜尋並選取 [Azure Functions:部署至函數應用程式] 命令。 2. 遵循提示來提供下列資訊。

Azure Functions: Deploy to Function App	
Template	Create new Function App in Azure... Advanced
Name	10727211-deployfunctionapp
Running Stack	Node.js 8 LTS
Resource Group	110-1-CS456L
Location for new resource	東南亞 (for CS456L)
Hosting Plan	Consumption
storage	10727211model9storage
Application Insights resource	Skip for now