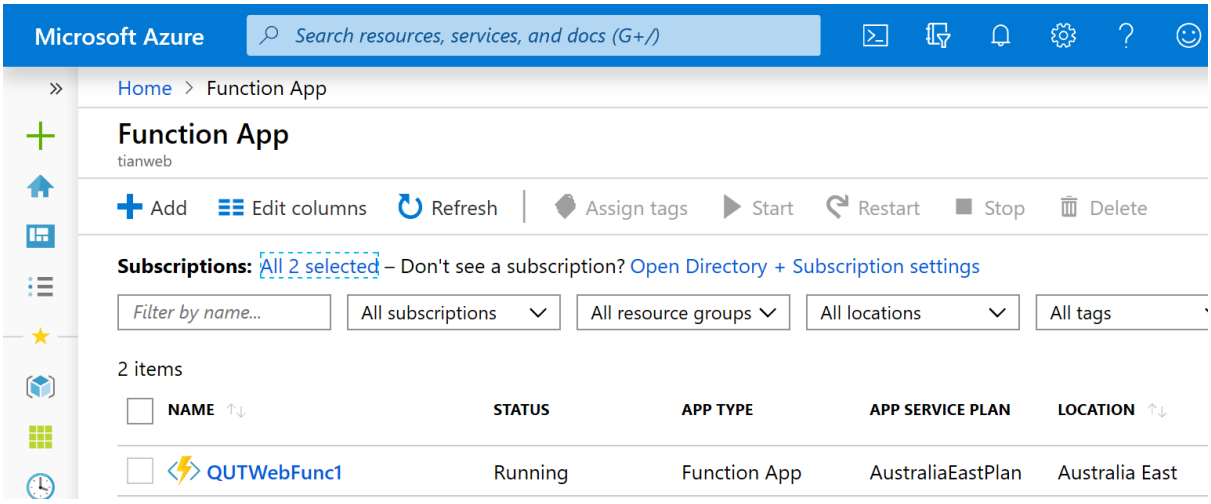
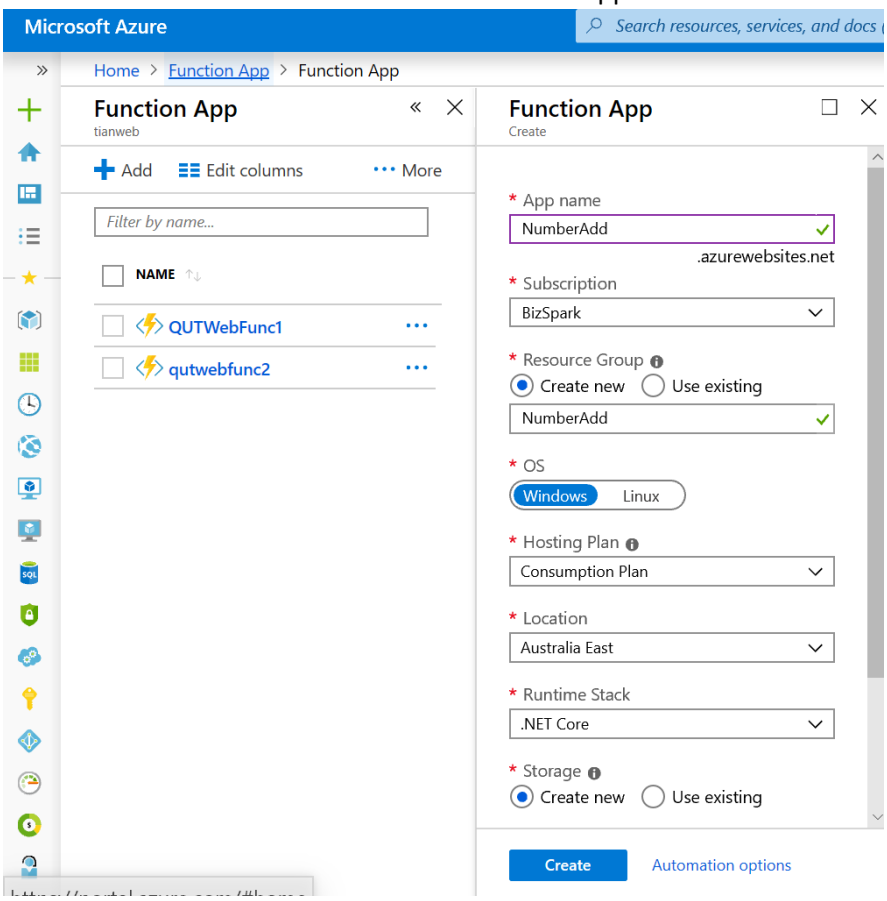
Function App

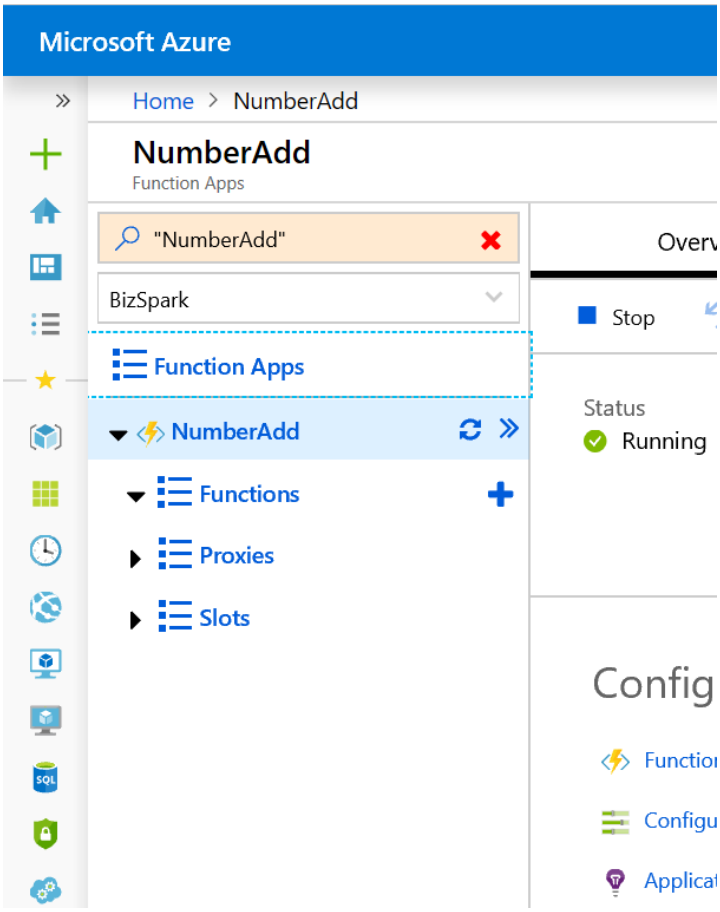
1. In the search field in the Azure portal , type Function App. This will bring up the Function Add command



2. Click on the + Add and fill the form to create the app

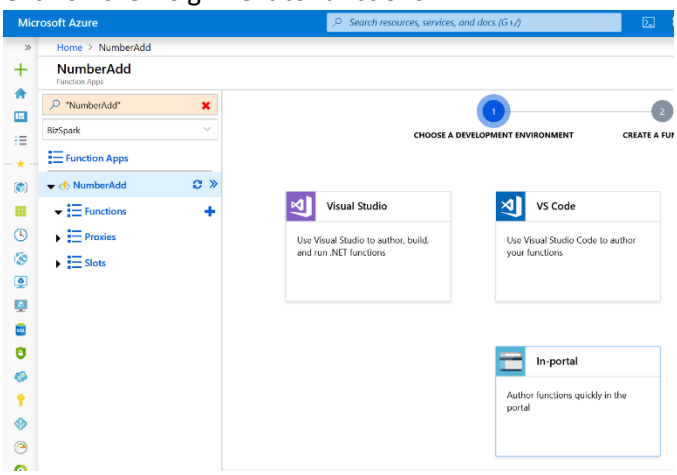


3. It will take a few minutes to be created. Once done it will provide a message to navigate to the App interface.



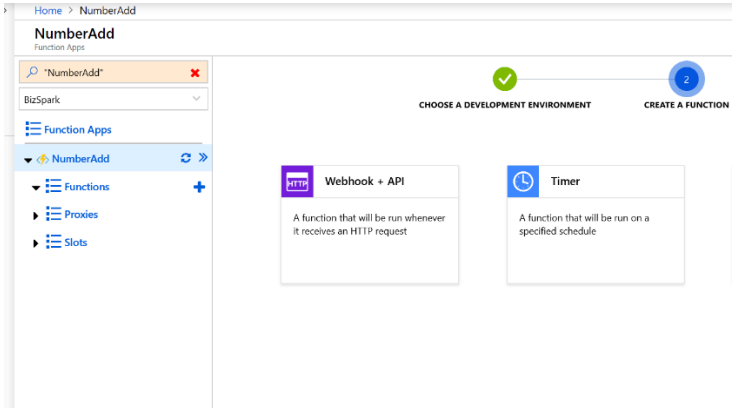
4. Two functions need to be created.
   1. DataReceiver :An http triggered function
   2. QueueProcessor : A queue triggered function.
5. DataReceiver

   Click on the + sign next to functions

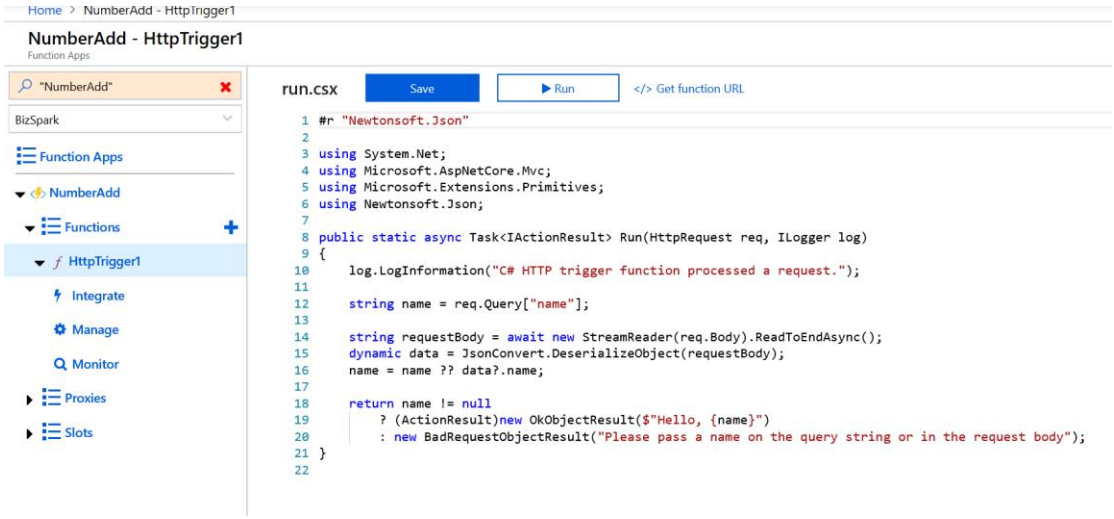


It will present different way to create the function. Select In Portal

And then web hook



The function will be created with simple code that will respond with hello. To try it out copy the url by clicking on get function url



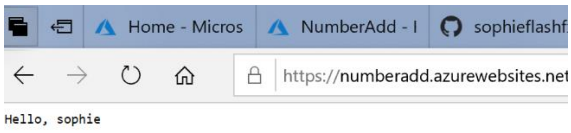
The url will be lime the following with a unique value at the end as shown below

https://numberadd.azurewebsites.net/api/HttpTrigger1?code=353ssddf32.....

Add to it the query param as follows

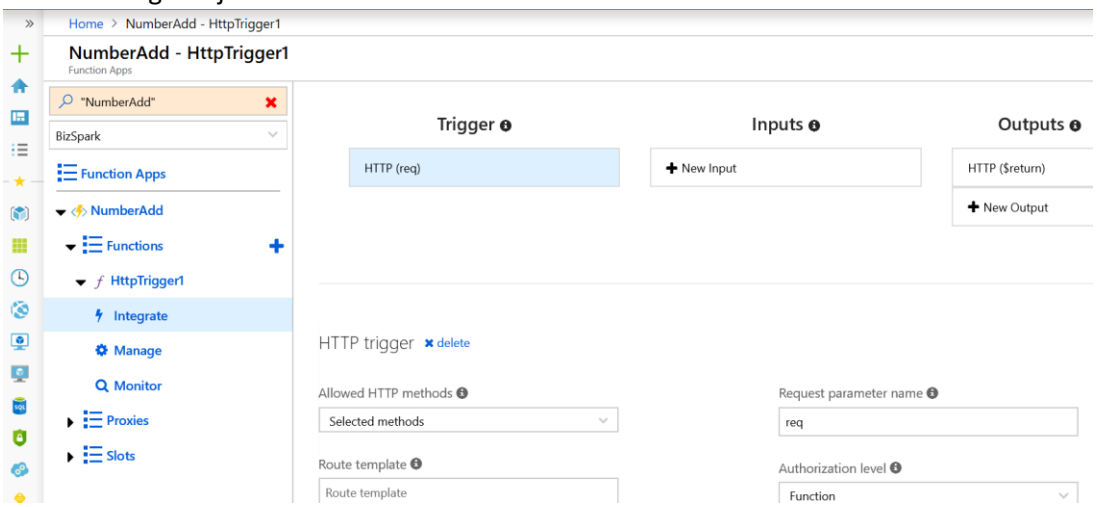
https://numberadd.azurewebsites.net/api/HttpTrigger1?code=353ssddf32.....&name=sophie

past the url into a browser and call the function. The function will respond with Hello, sophie

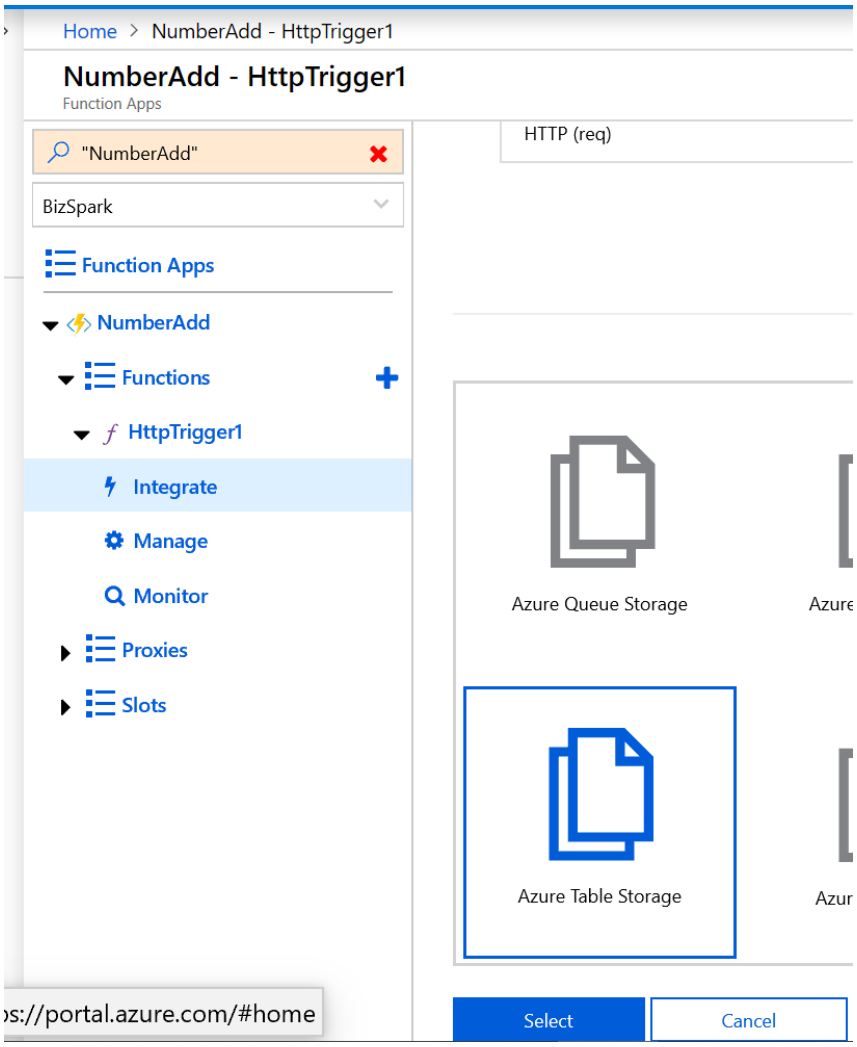


Next a Azure storage account needs to be created and integrated with the function. Do this as follows.

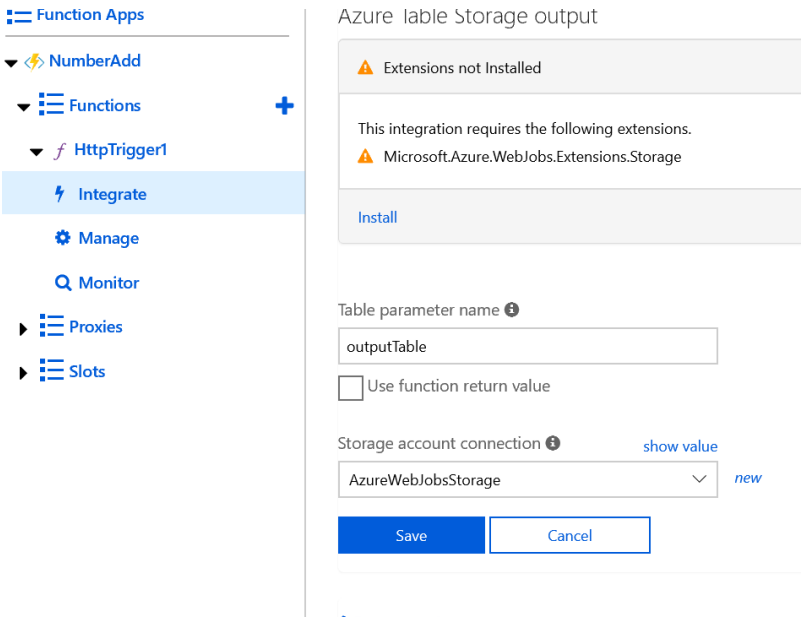
6. Click on integrate just below the function name



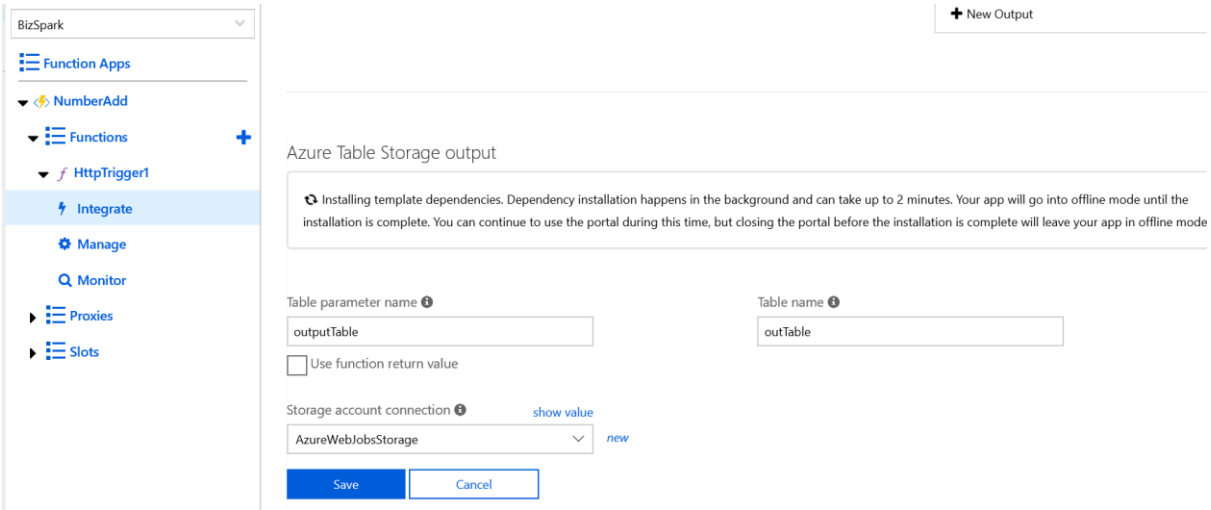
7 Click on +New Outputs and then Azure Table storage.



It will give a warning extension not installed. Click on install



The following screen results



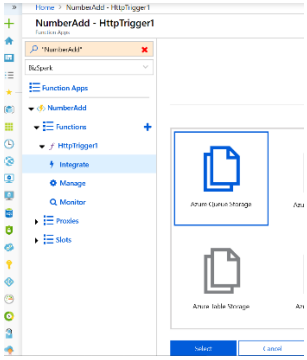
Once its done click save. A Azure table called outTable will be created and integrated with the app.

Next input integration to the same table is needed. Click integrate and this time choose + New Input

Select Azure table as before. For the table name it will have inTable. However we need to use the same table that was created for output. As such replace the inTable with outTable and click save.

Finally, a queue needs to be created.

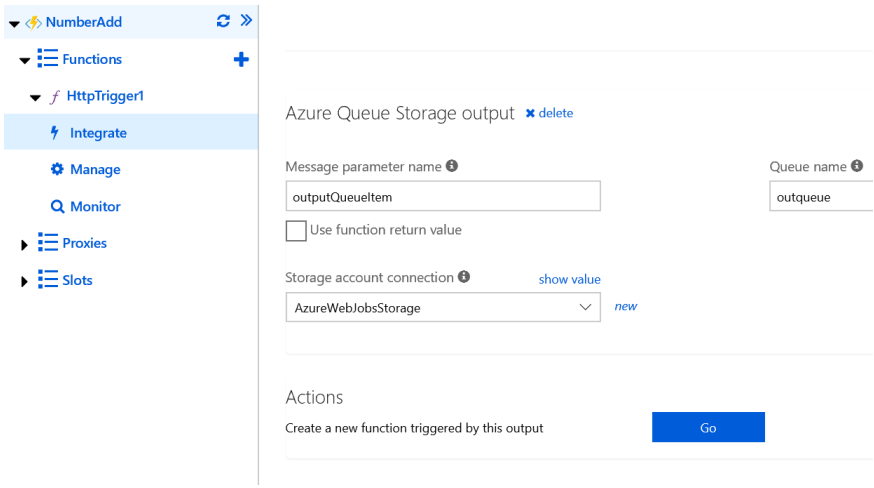
Click integrate and then + New output and select Azure Queue Storage



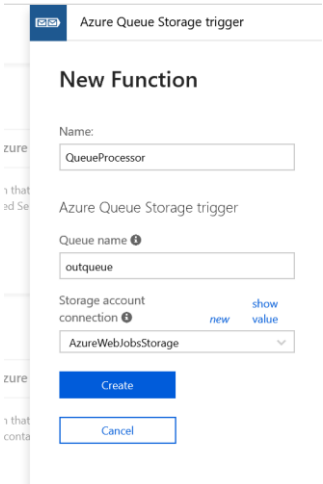
Keep the default setting and click save.

Now all the integrations with storage has been completed.

When an entry is made in the Queue we want it to trigger the QueueProcessor function. The interface Action provides the ability to create this function at this stage. Click on Go



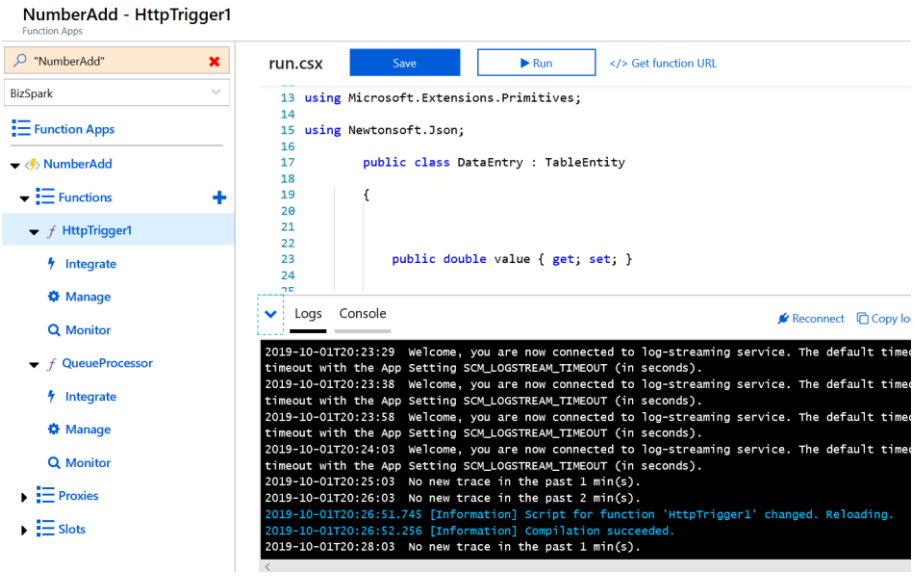
And then select Azure Queue Storage trigger. Set the name of the function to be QueueProcessor and the name of the queue to be outqueue



Open the code in the http trigger function and replace the code with the one on this page

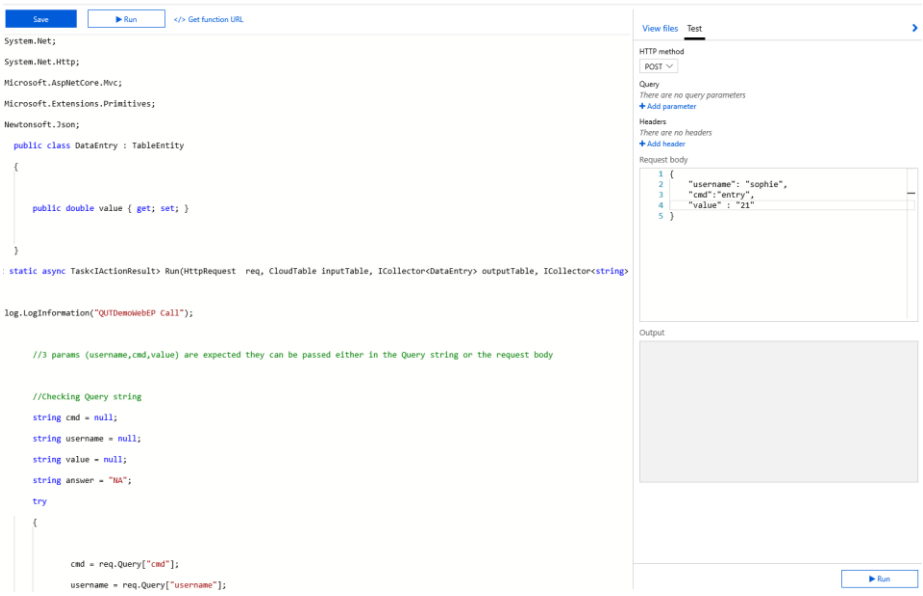
https://github.com/sophieflashfx/azureserverless/blob/master/datareceiver.csx

Pressing save will compile and save the function. Clicking on log will show if there are any compilation errors



The function can be tested by using the test values. Replace the default body with the following

```
{
   "username": "sophie",
   "cmd":"entry",
   "value" : "21"
}
```



If all is ok you will get the status 200 ok and the response

```
“Request Recieved username:sophie,cmd:entry,value:21 output answer =NA”
```

View files Test

HTTP method

POST

Query

There are no query parameters

+ Add parameter

Headers

There are no headers

+ Add header

Request body

```
1 {
2     "username": "sophie",
3     "cmd":"entry",
4     "value" : "21"
5 }
```

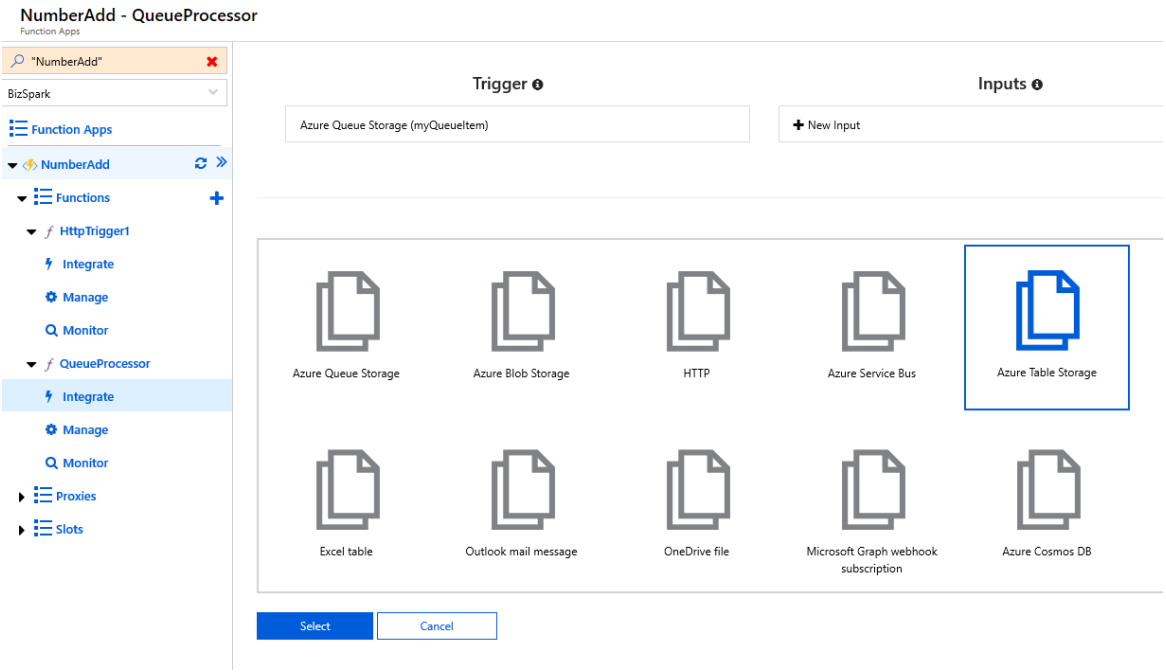
Output

Status: 200 OK

```
Request Recieved username:sophie,cmd:entry,value:21 output answer =NA
```

QueueProcessor

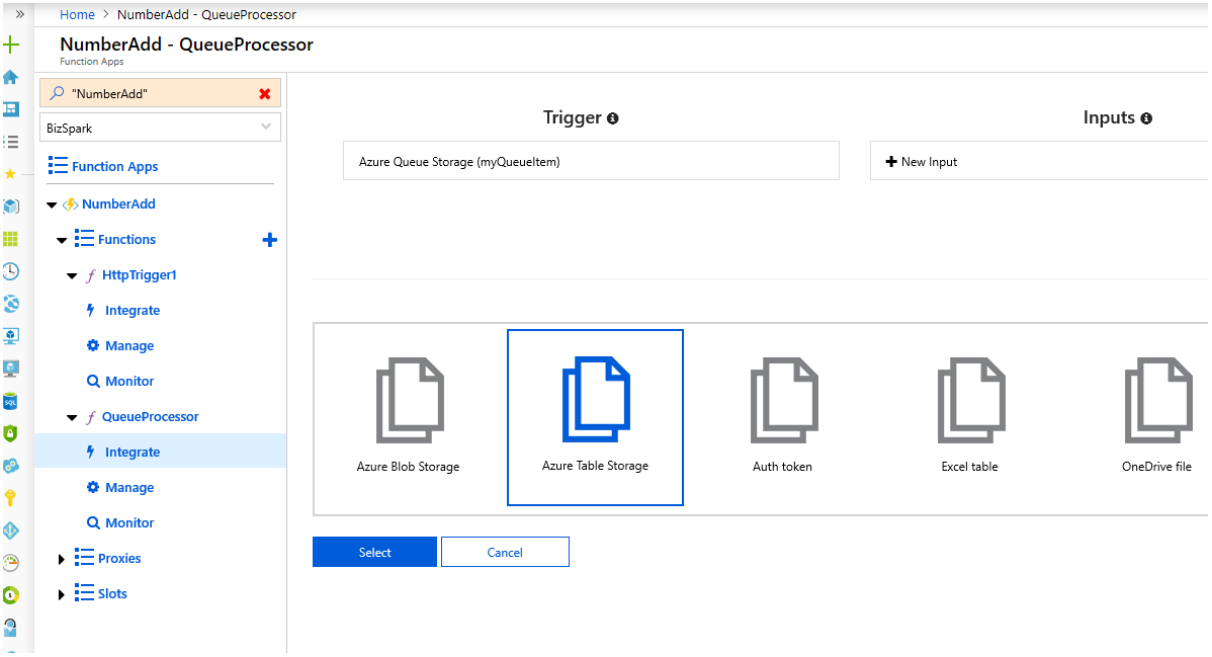
1.Next we need to integrate the QueueProccesor with the Azure Table as well. To do this click on integrate and then + New Output and select Azure Table storage.



Make sure the table name is outTable and click save.

3. The function needs to be also integrated for input from the table.

To do this click integrate and +New Input then Azure Table / select and change the table name to outTable and save.



Finally replace the code of the function with the code from this page

https://github.com/sophieflashfx/azureserverless/blob/master/queueprocessor.csx

The user Interface

The static html page for the app will be saved in an azue blob and served from it. To do this follow

The instruction on this page

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-static-website.

The code for the index.html page is given on this link

https://github.com/sophieflashfx/azureserverless/blob/master/index.html