ONLINE LAB: Create a Function that Writes to a Queue

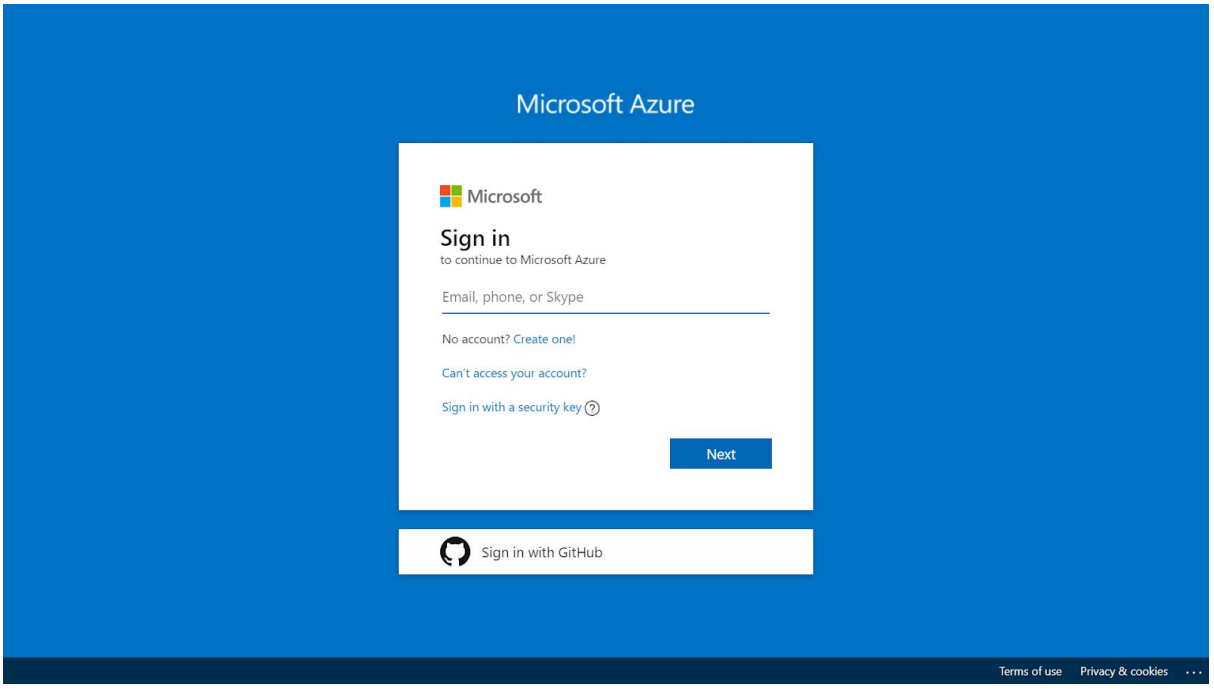
Your Challenge

- Create a storage account
- Create a storage queue
- Create a function app
- Add a function to it that writes to that queue

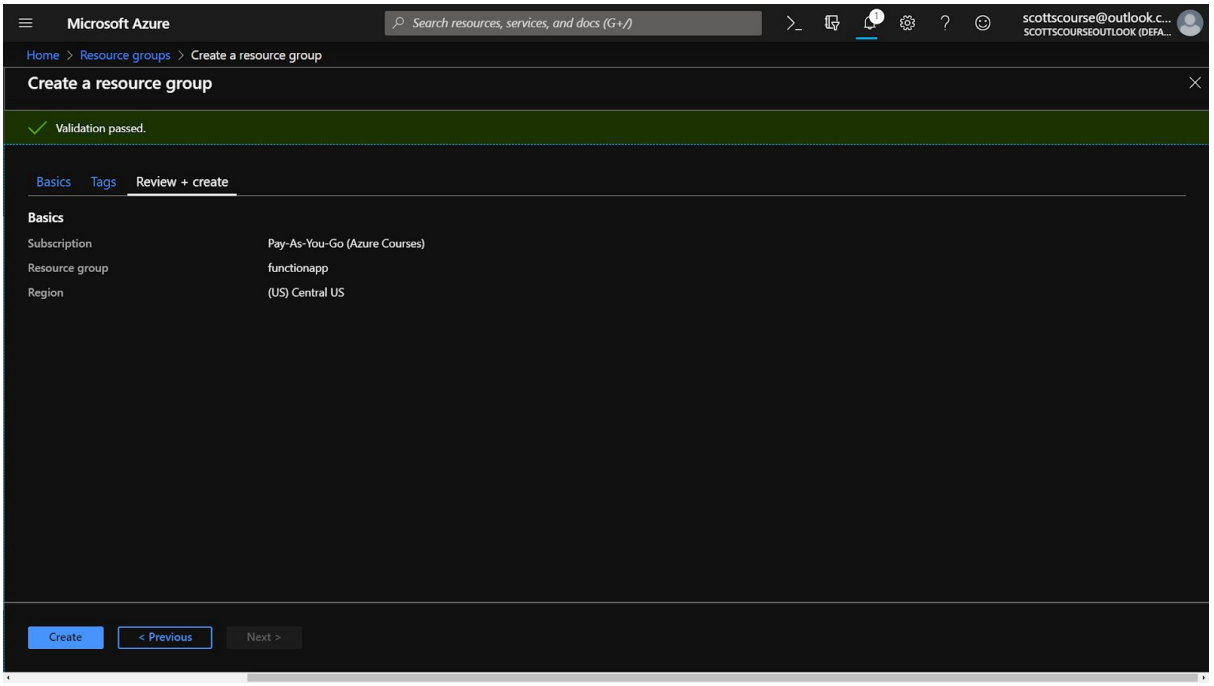
Solution

Step 1 Sign Into Azure

Sign into Azure at https://portal.azure.com/

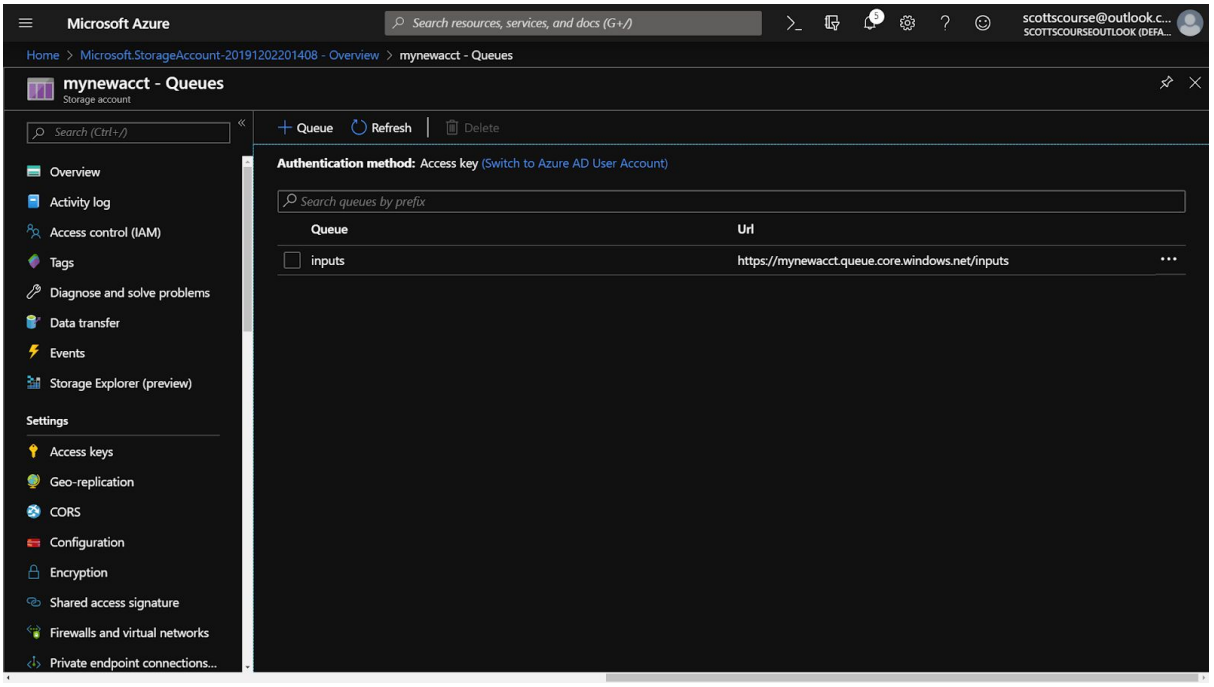


Step 2 Create Resource Group



1. Create a new resource group named **functionapp**.

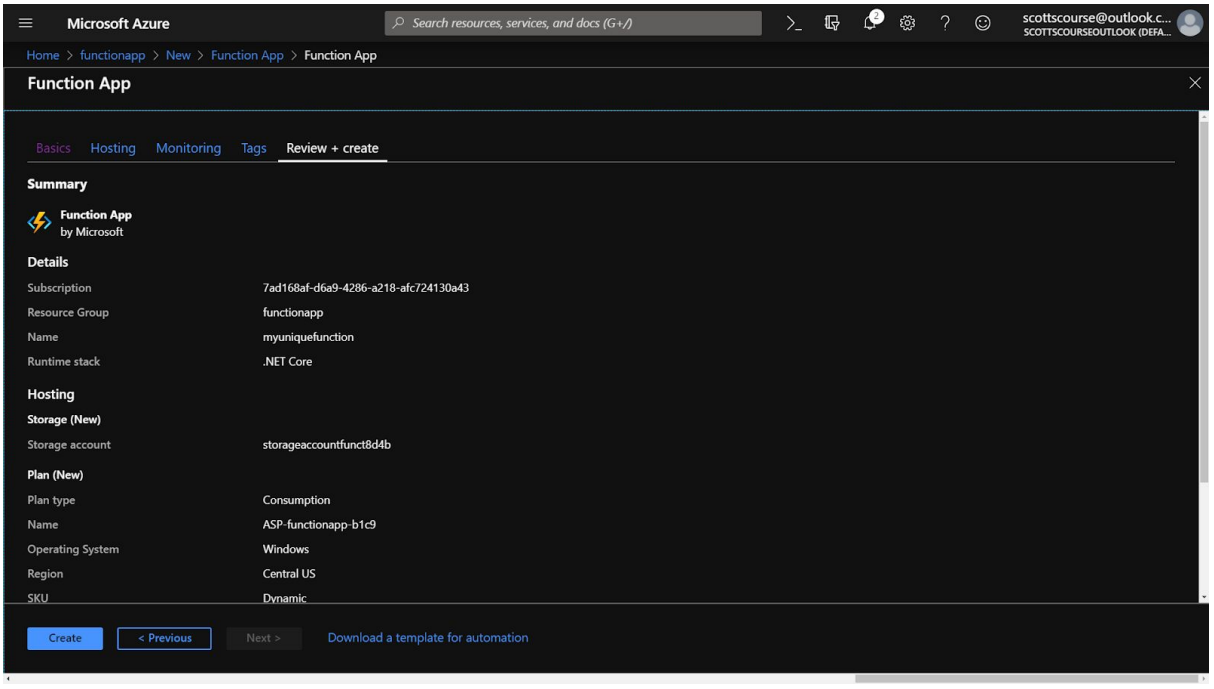
Step 3 Create a Storage Account + Queue



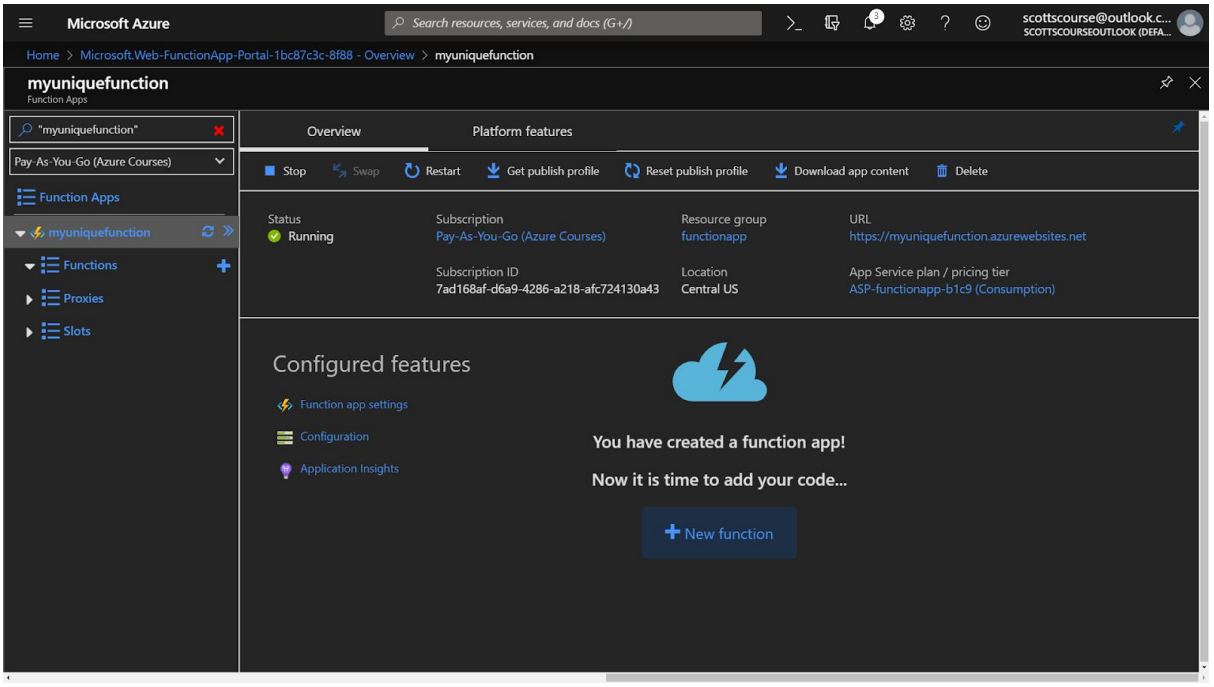
1. Navigate to the **functionapp** resource group

2. Add a resource to it
3. Find “**Storage Account**” from the list
4. Click **Create**
5. Give the storage account **a unique name**.
6. Ensure it’s the same region as the function app.
7. Choose LRS - locally-redundant storage.
8. Leave all the defaults and click “**Review + Create**”.
9. Click **Create**.
10. Wait for the account to be created.
11. Navigate to the storage account.
12. Click on “**Queues**” on the overview screen.
13. Add a new Queue named “**inputs**”.

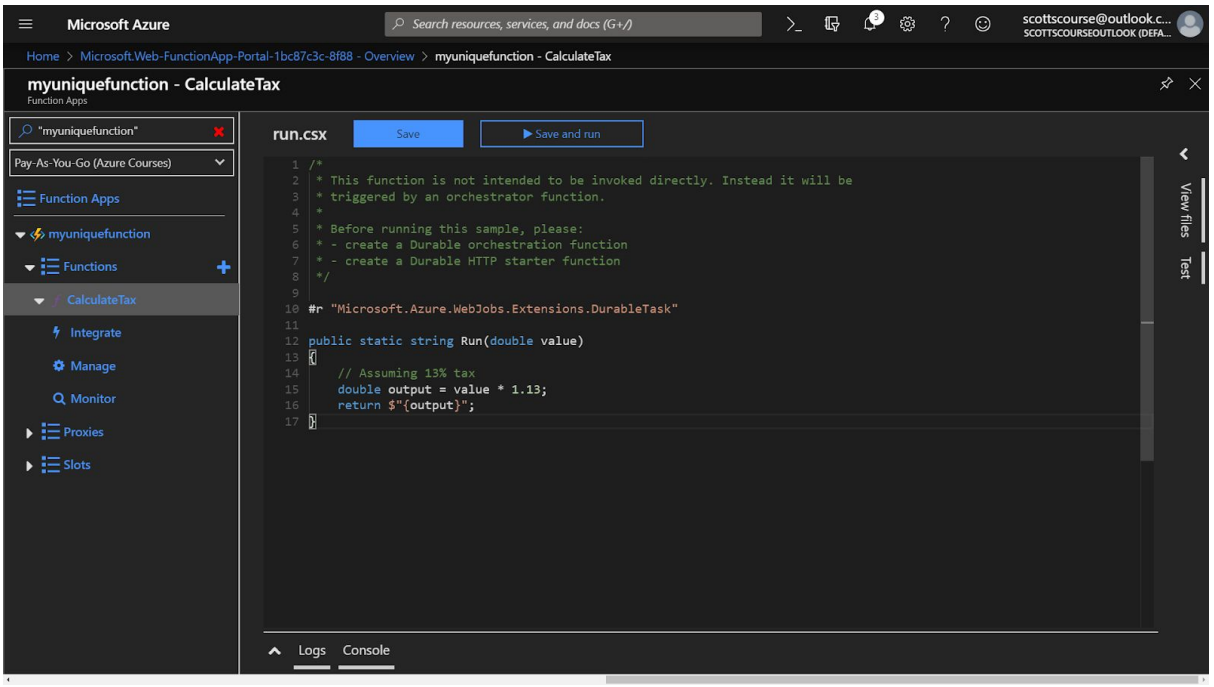
Step 3 Create a Function App



1. Navigate to the **functionapp** resource group
2. Add a resource to it
3. Find **Function App** from the list
4. Click **Create**
5. Give the function app **a unique name**.
6. Ensure that it’s a **code** function, using **.NET core** stack
7. Click “**Hosting**” to go to the next screen.
8. Ensure that it is running on **Windows** under the normal **Consumption** plan.
9. Click **Review + Create**.
10. Click **Create**.
11. Wait for the function app to be created.

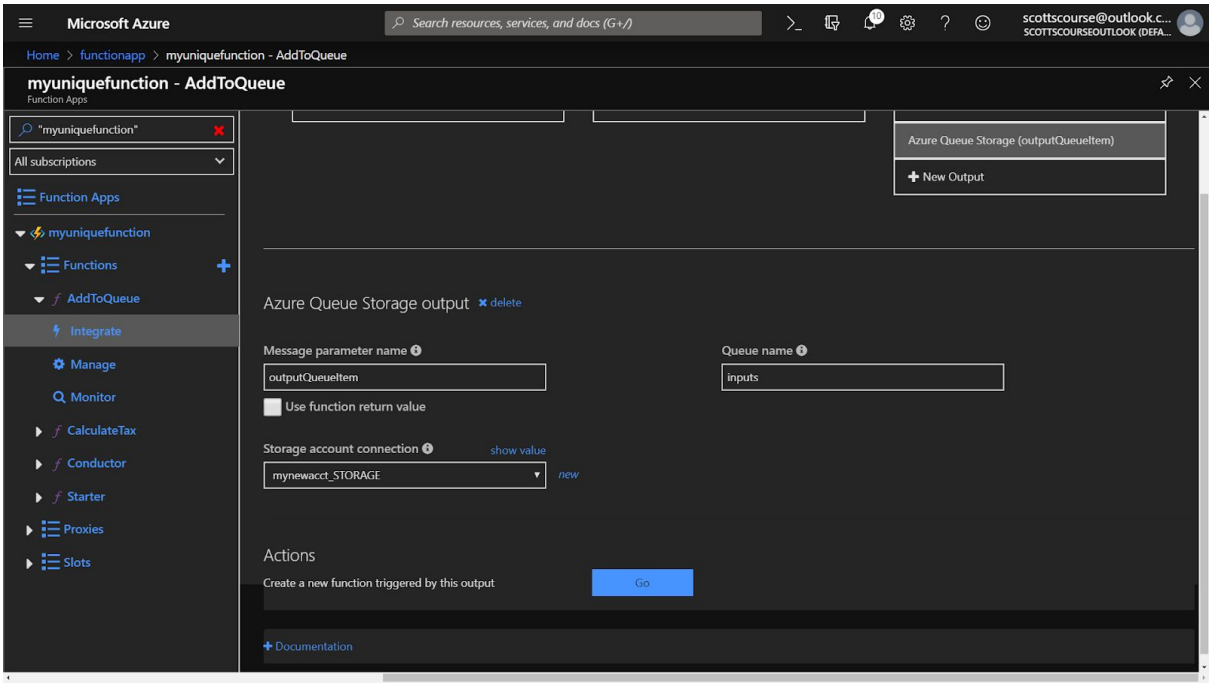


Step 4 Create a Function



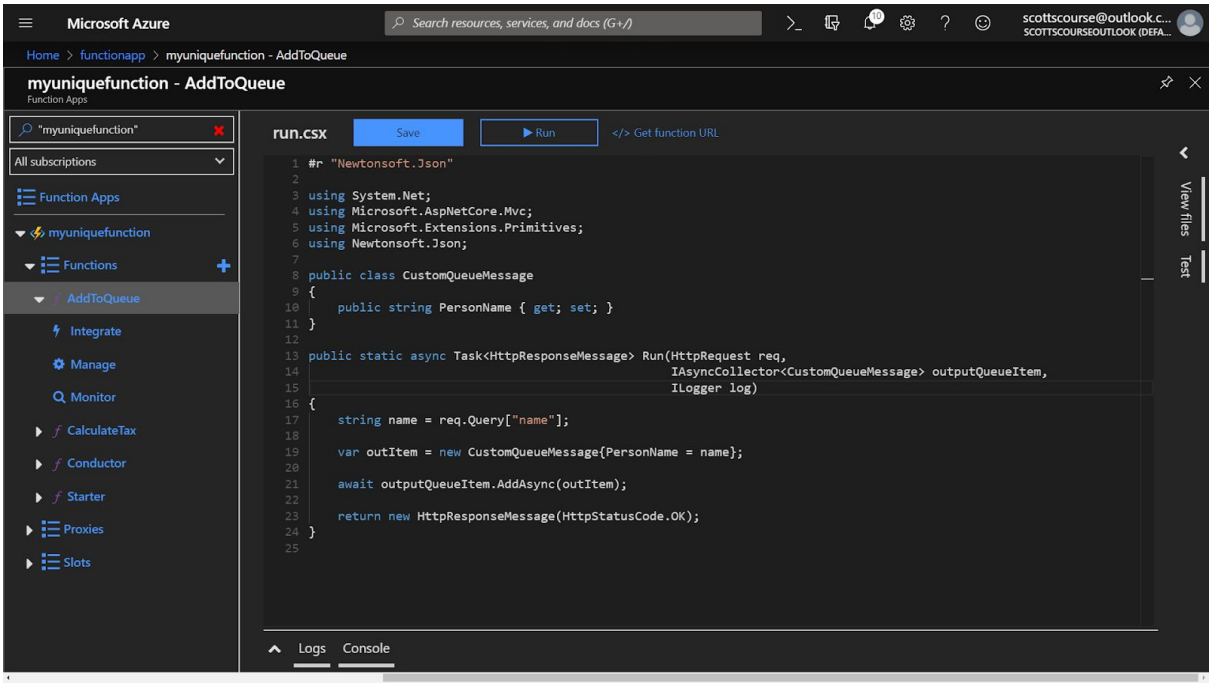
1. Navigate into the new function.
2. Click the "**+ New Function**" button in the overview screen.
3. Choose "**In Portal**" as the development environment and click **Continue**.
4. Choose "**More Templates**" as the trigger type and click **Find and Review Templates**.
5. Choose "**HTTP Trigger**" as the template.
6. Name the function "**AddToQueue**". Click **Create**.

Step 5 Connect the Function to Queue Storage



1. Click on **Integrate** on the left menu.
2. Under Outputs, choose "**+ New Output**".
3. Choose "**Azure Queue Storage**". Scroll to the bottom and hit **Select**.
4. Install the **Azure Storage Extensions**, as prompted. Wait for installation to complete.
5. Create a new connection string by clicking **new**.
6. Choose the storage account you created from the list.
7. Change queue name to **inputs**.
8. Click **Save**.

Step 6 Modify the Function Code



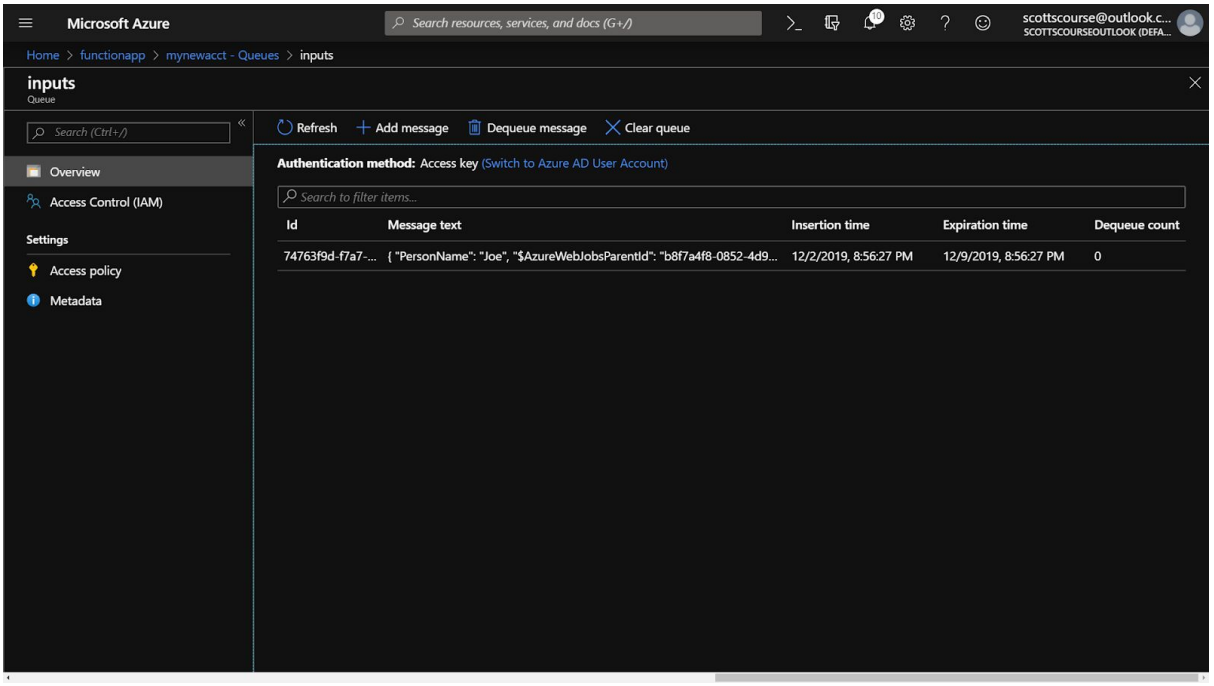
1. Return to the function code screen by clicking "**AddToQueue**" on the left menu.
2. Replace the body of the code with the following.

```
public class CustomQueueMessage
{
    public string PersonName { get; set; }
}

public static async Task<HttpResponseMessage> Run(HttpRequest req,

IAsyncCollector<CustomQueueMessage> outputQueueItem,
                                                    ILogger log)
{
    string name = req.Query["name"];

    var outItem = new CustomQueueMessage{PersonName = name};

    await outputQueueItem.AddAsync(outItem);

    return new HttpResponseMessage(HttpStatusCode.OK);
}
```

This creates a class that represents a message that we will insert into the queue.

3. **Save** the function.

Step 7 Test the Function



1. Click **Get Function URL** and copy the function URL.
2. Open a new browser tab and paste the URL to the address bar. Do not hit enter.
3. Append the URL with the name parameter "**&name=Joe**". Hit **enter**.
4. The screen should return blank.
5. Navigate to the storage account using the Azure Portal.
6. Open the Queues section.
7. Click on the inputs queue.
8. See the Queue entry with the value **Joe**.
9. Return to the browser tab, and add a few more entries with different names.

Do not delete this, as we will use it again shortly.