ONLINE LAB: Create a Function that Calls Other Functions

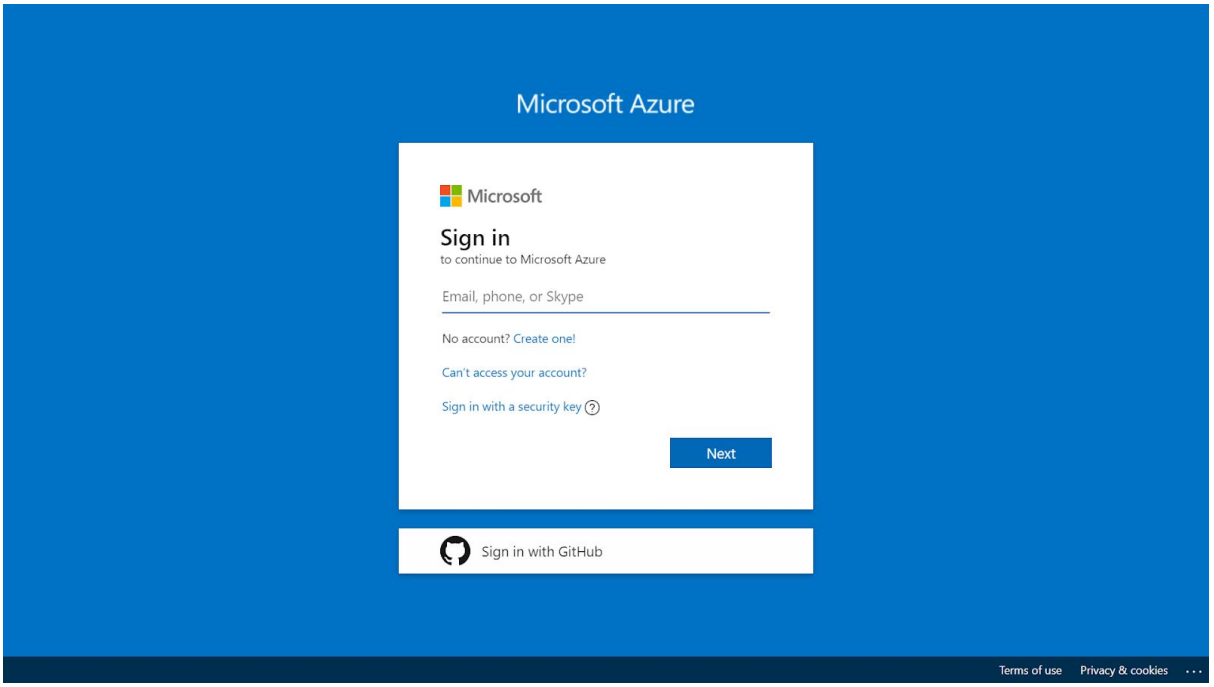
Your Challenge

- Create a function app
- Add a function to it
- Add a second function to it, that calls the first in a loop
- While running, observe the function performance
- Clean up all of your resources created after you're done

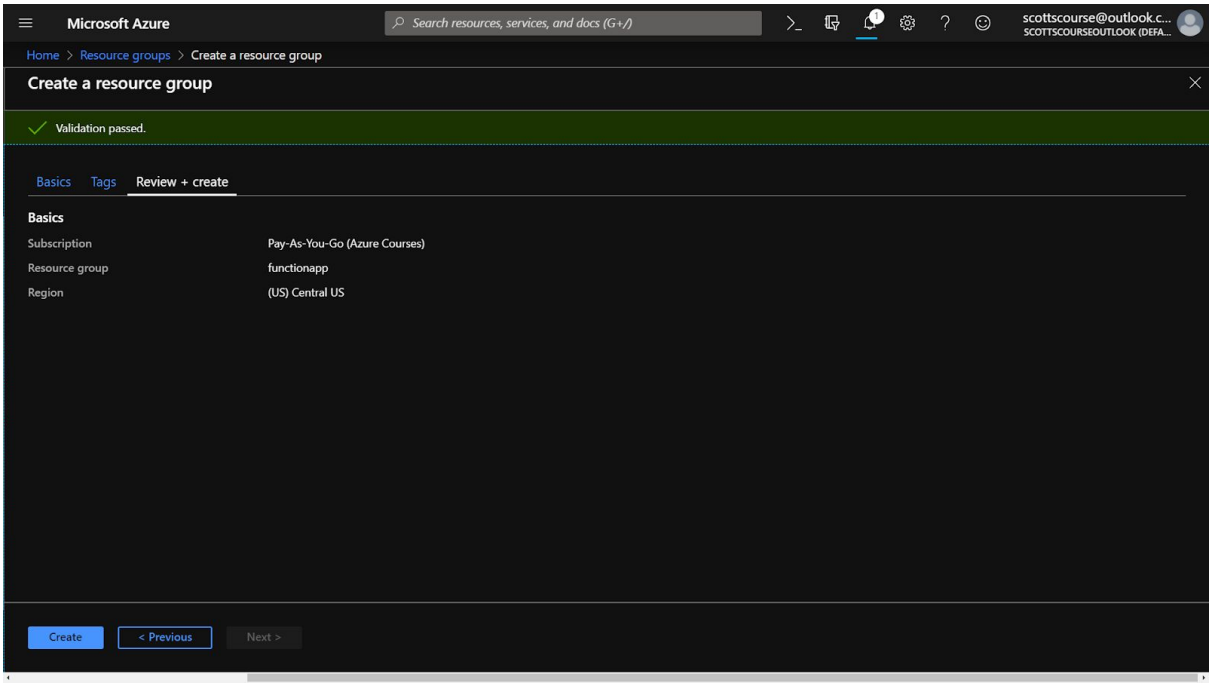
Solution

Step 1 Sign Into Azure

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

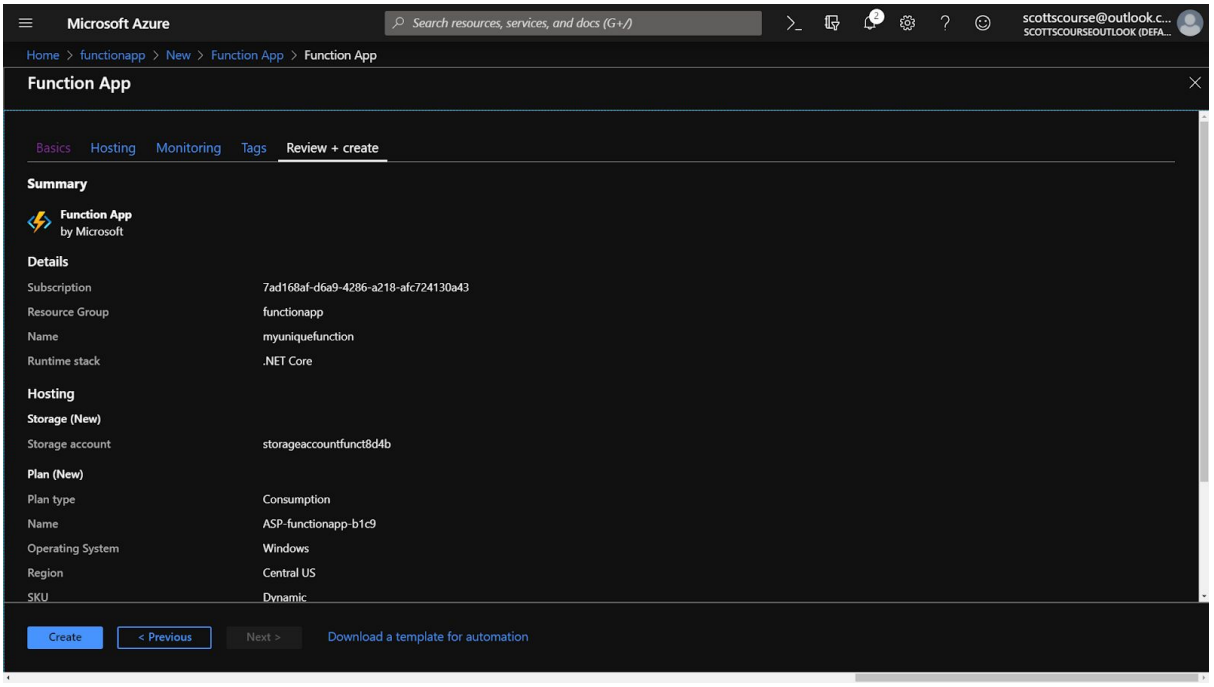


Step 2 Create Resource Group



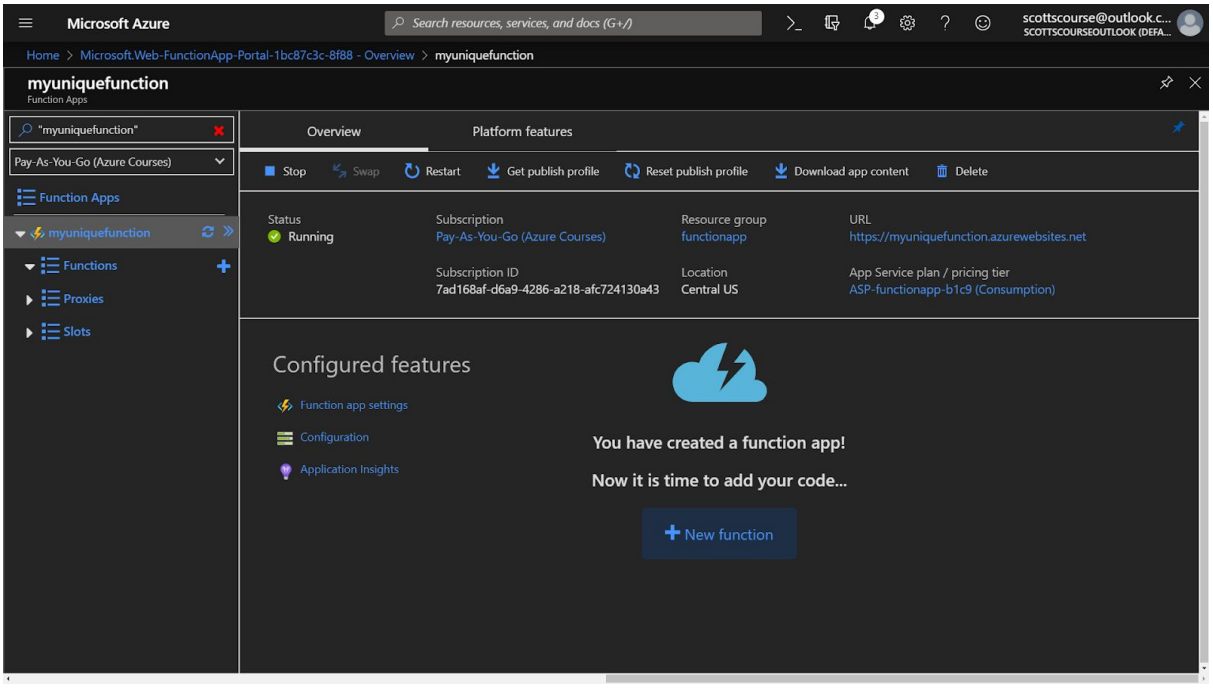
1. Create a new resource group named **functionapp**.

Step 3 Create a Function App

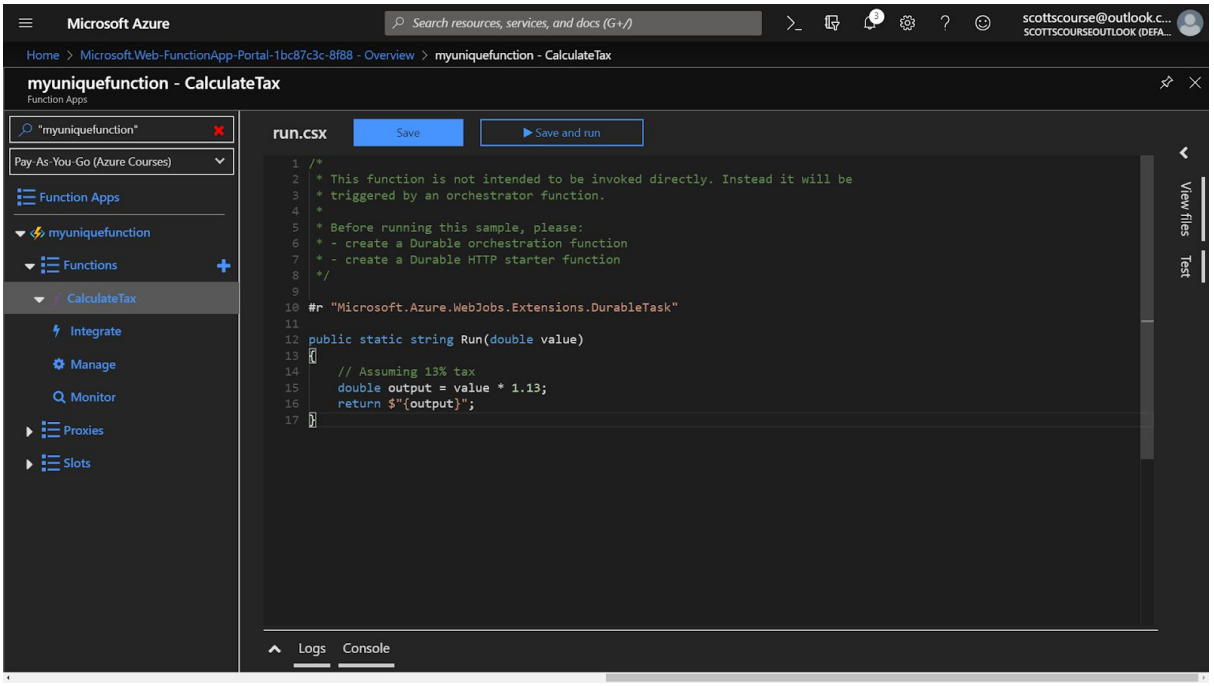


1. Navigate to the **functionapp** resource group
2. Add a resource to it
3. Find **Function App** from the list

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

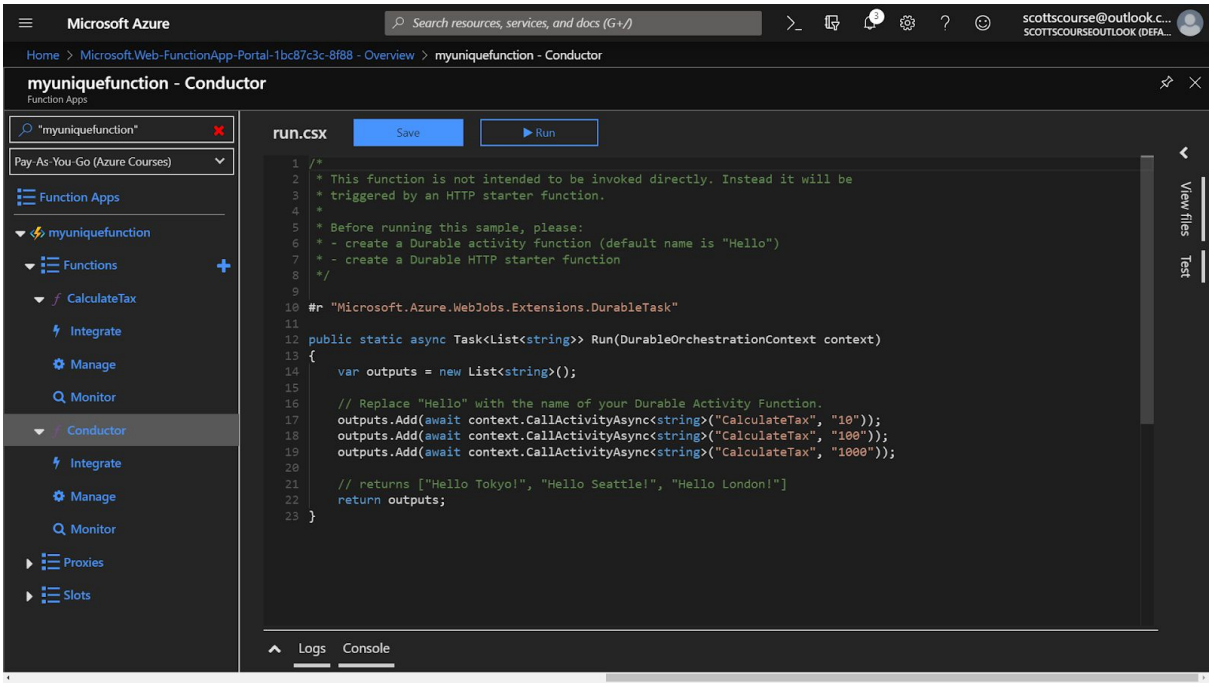


1. Navigate into the new function.
2. Click the "**+ New Function**" button in the overview screen.
3. Choose "**In Portal**" as the development environment and click **Continue**.
4. Choose "**More Templates**" as the trigger type and click **Find and Review Templates**.
5. Choose "**Durable Functions Activity**" as the template.
6. Install the extension when prompted. Wait for it to finish and click **Continue**.
7. Name the function "**CalculateTax**".
8. Replace the body of the code with the following:

```
public static string Run(string value)
{
    // Assuming 13% tax
    double output = double.Parse(value) * 1.13;
    return $"{output}";
}
```

9. Navigate to the "Integrate" menu on the left.
10. Change the "**Activity parameter name**" to value.
11. **Save** the function.

Step 5 Create A Second Function Inside the App

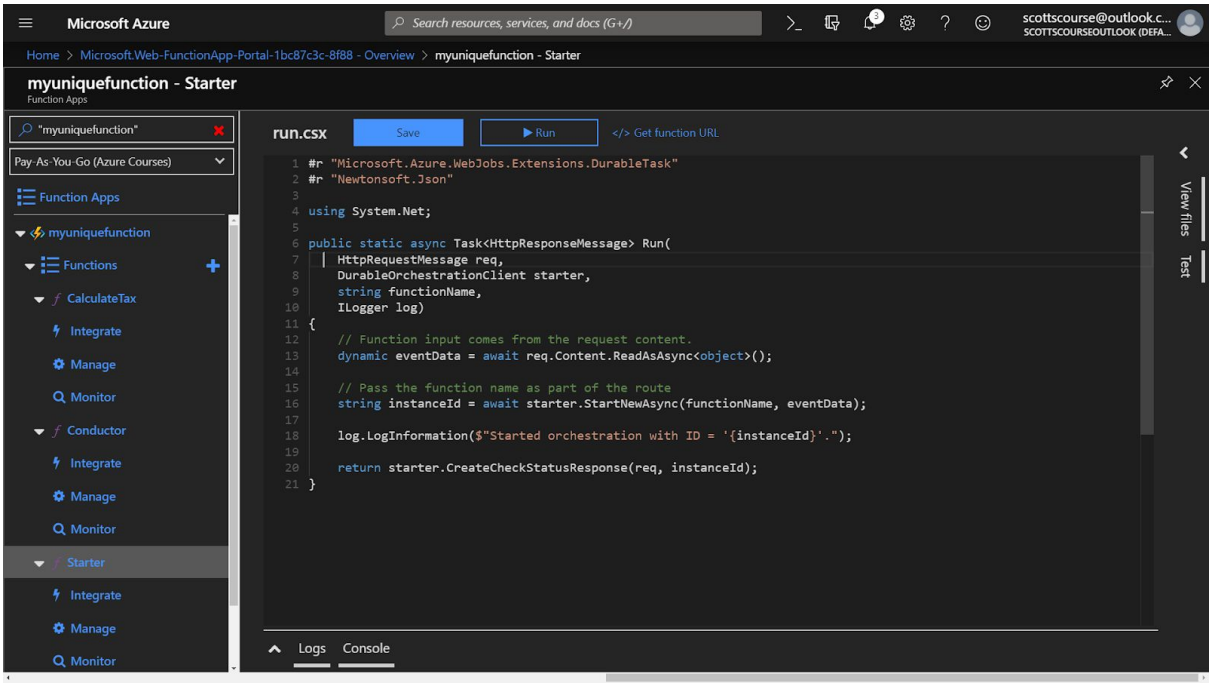


1. Navigate to the **Function App**
2. Clock on “**Functions**” on the left menu
3. Observe there is an existing function named “**CalculateTax**”, that is enabled.
4. Add a “**New Function**”.
5. Find the template called “**Durable Functions Orchestrator**”
6. Name the function “**Conductor**”.
7. Click **Create**.
8. Based on the template, replace the calls to function named “Hello” with the function “CalculateTax”.
9. Pass in values 10, 100 and 1000, like so:

```
outputs.Add(await context.CallActivityAsync<string>("CalculateTax", "10"));
outputs.Add(await context.CallActivityAsync<string>("CalculateTax", "100"));
outputs.Add(await context.CallActivityAsync<string>("CalculateTax", "1000"));
```

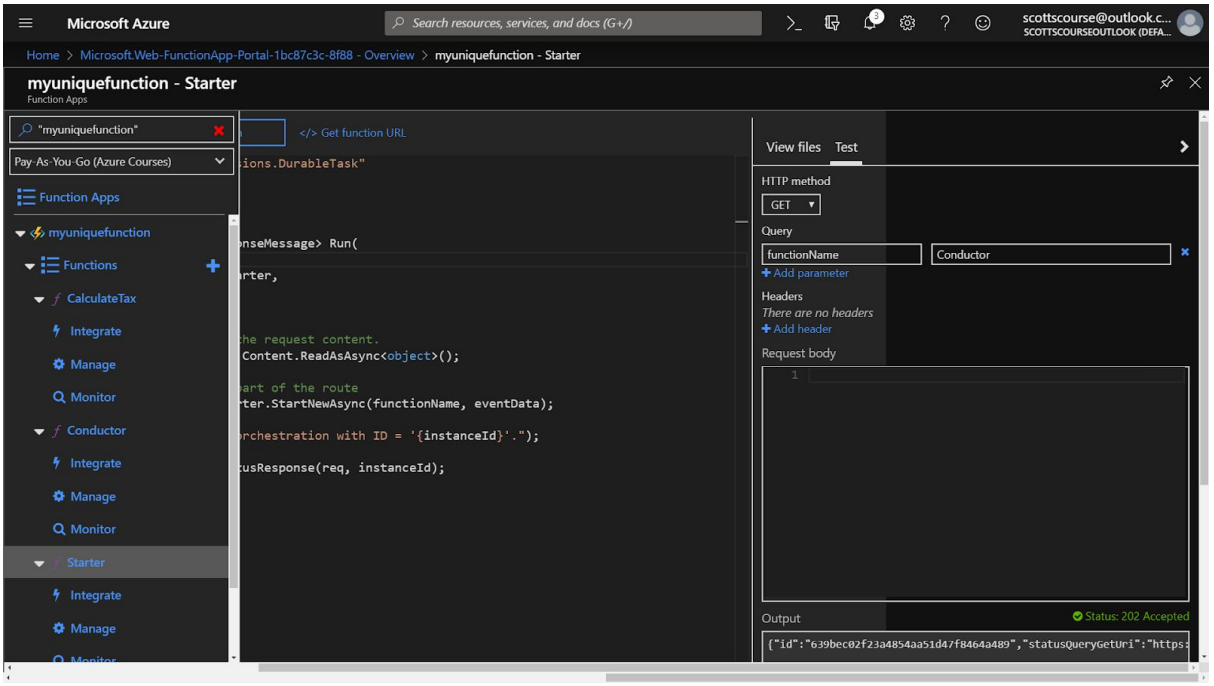
10. **Save** the function.

Step 6 Create A Third Function Inside the App

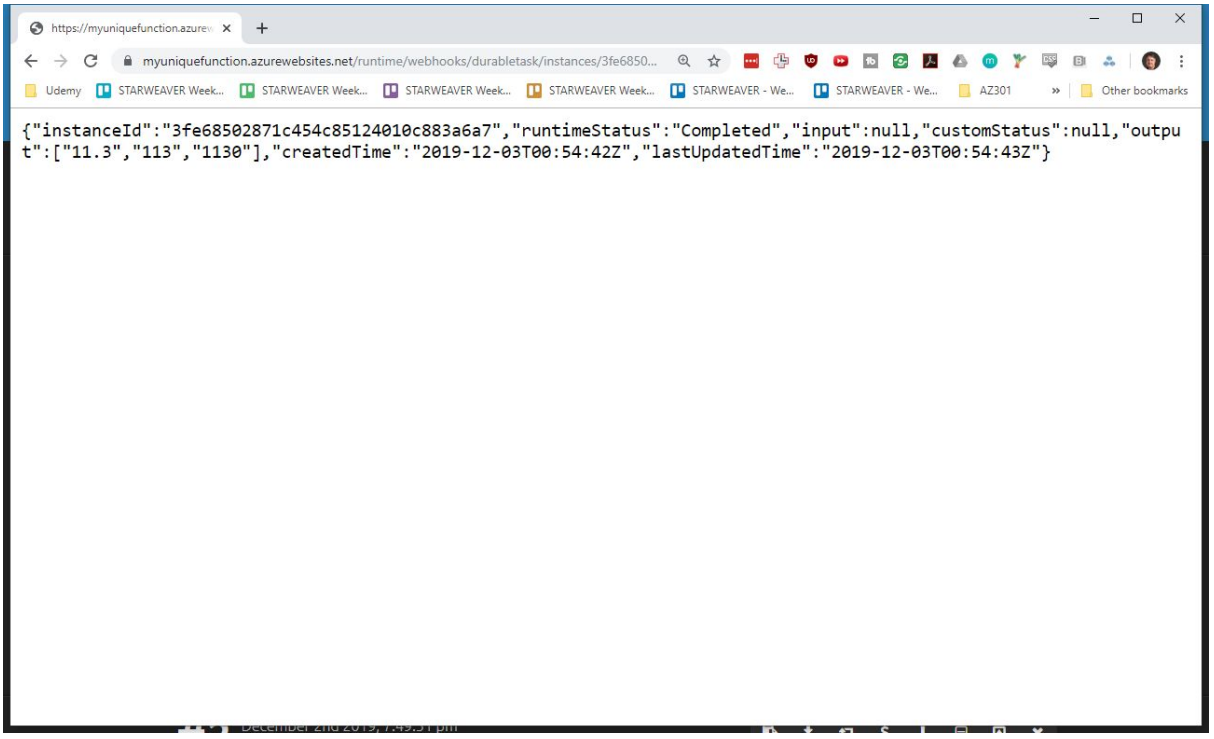


1. Navigate to the **Function App**
2. Clock on “**Functions**” on the left menu
3. Observe there is an existing function named “**CalculateTax**” and “**Conductor**”, that are both enabled.
4. Add a “**New Function**”.
5. Find the template called “**Durable Functions HTTP Starter**”
6. Name the function “**Starter**”.
7. Click **Create**.
8. **Save** the function.

Step 7 Test the Starter App



1. Open the Test blade.
2. Provide the functionName of **Conductor** as a parameter.
3. Click **Run**.
4. Copy the Output string.
5. Open the browser to a new tab and go to the JSON formatter at https://jsonformatter.curiousconcept.com/
6. Paste the JSON of the output string into the JSON formatter and click Process.
7. Get the value of the "**statusQueryGetUri**" parameter of the output.
8. Open the browser to a new tab and paste the URL from that parameter to the address bar.
    a. It might look like this: https://myuniquefunction.azurewebsites.net/runtime/webhooks/durabletask/instances/3fe68502871c454c85124010c883a6a7?taskHub=DurableFunctionsHub&connection=Storage&code=qu3dDY3vAaRXLhgSXsTScFsDaN6Gbg0plYCTqBdOagBBs530fWDK0g==
9. Examine the output.
10. If the output indicates and error, something happened along the way. Double check all the steps.



You can see above that the “output” parameter has the results of the three CalculateTax runs.

Step 8 Clean up

1. In the navigation list, click **Resource groups**.
2. Click **functionapp** to open the resource group.
3. Click **Delete resource group** to delete the resource group.
4. On the **Are you sure you want to delete** blade, type the resource group name: **functionapp**.
5. Click **Delete** to delete the resource group.