# Azure function -> Azure Cloud

\*\*Note: Deployed locally to check if its working completely:

## Requirement:

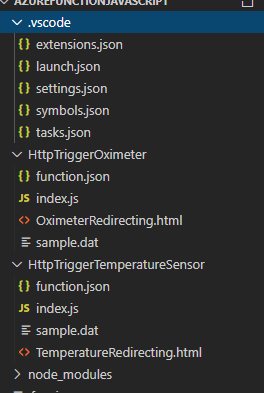
visual studio code for debugging and deployment

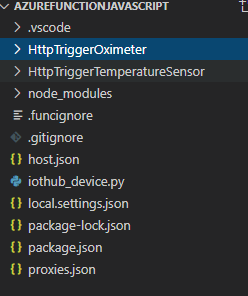
nodeJS version > 10

azure function extension

azure-functions-core-tools

## Folder Structure:





* 2 different trigger function has been created respectively for sensors Oximeter and temperature sensors, respectively.
* Package.Json have all the package dependencies which will install automatically and will create environment, launch.json (inside .vscode) have the debugging environment setup, it will directly install the azure core tools if debugger doesn’t have.
* Folder HttpTriggerOximeter have index.js as a javascript file which will invoke a method directly in the device for oximeter, similarly HttpTriggerTemperatureSensor have index.js as a javascript file which will invoke a method directly in the device for temperature sensor.
* OximeterRedirecting.html and TemperatureRedirecting.html is creating a UI for” timer” wait between execution of their sensor methods.

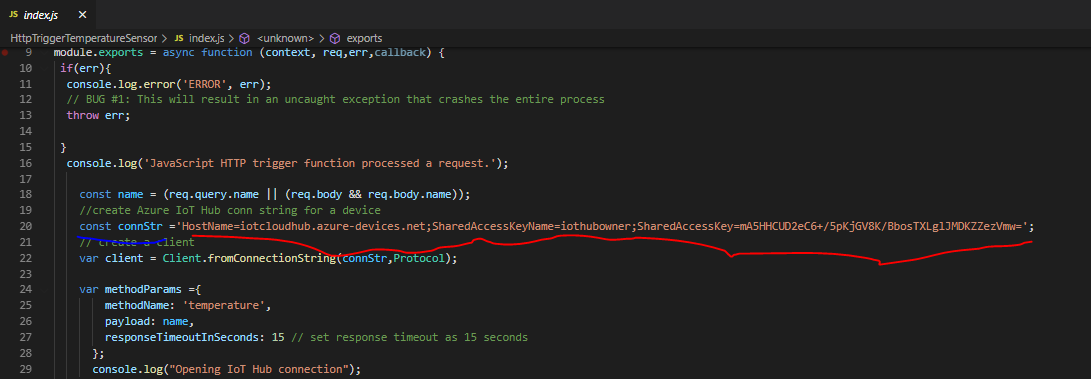
## Debugging:

Check before debugging:   
Get Iot hub connection string from iothub

E.g.,

HostName=<your iot cloud hub>.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=xjbIhUQxccyMOIpDv6bcOXBYJCFSqN3uHTQZ4YVz8nA=

And update it inside both index.js files:

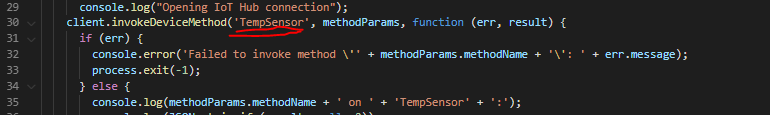


Iot Device name from IotHub ;

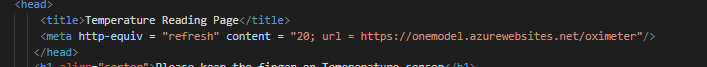
e.g.,

TempSensor

And update the same in both index.js files:



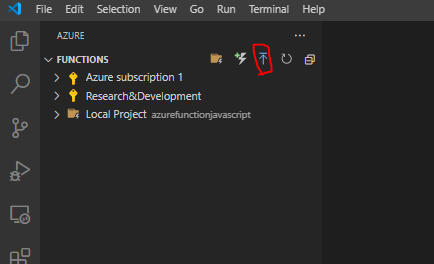
Always check if url : inside both html file is appropriate for redirecting before debugging



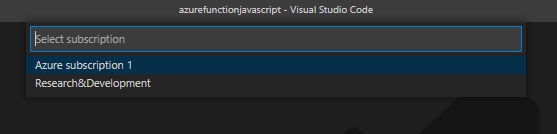
## Process to Deploy from local system (VSCODE) to azure:

Navigate to azure function extension.

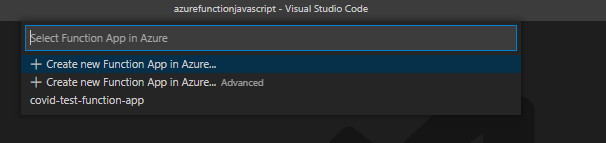
Click on deploy to function app



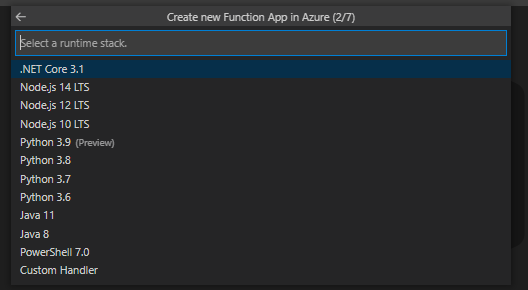
Select your subscription



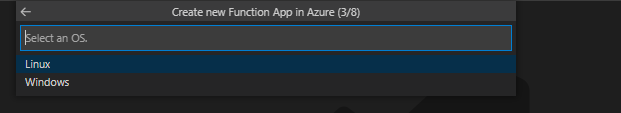
Create a new function app if you never deployed any:



Choose node js 14 lts



Then linux



If you are not sure go with app service plan

