Working test

IoT-Backend

C#, Azure, and IoT

# 1 Assignment

The purpose of the assignment is partly to test your knowledge of C# and Azure, and your ability to work with somewhat vague instructions. We will also, with the help of this assignment, open a discussion with you on the solution as well as the architecture and components in Azure.

The assignment must be implemented in C# and any version of .Net (Framework or Core) that you are comfortable with. For your reference, you have any version of Visual Studio and the documentation that is available on the Internet. The solution must be able to run in Microsoft Azure in the simplest possible way.

## Background/Scenario

A new customer has developed their own IoT solution which is able to collect weather information from a weather station. The weather station sends information over a Microsoft Azure IoT hub in near-real-time. Today there are two Azure services that handle the information.

* Data Receiver: Saves the information that arrives in CSV files to the Azure Blob Storage. The files are written continuously, one file per day, and contain the weather station units and sensor types (such as temperature, humidity, and rainfall).
* Data Compressor: Compresses the CSV files regularly and moves them into a common compressed ZIP file for each unit and sensor type.

The temporary files are saved in the format: /{deviceId}/{sensorType}/{date}.csv

The compressed file is stored according to the format: /{deviceId}/{sensorType}/historical.zip

## Problem

The solution now needs to be complemented with a new service/program that exposes the information via a REST API to a second system. The new service must have the ability to run in Azure. The ideal goal for the API methods that the customer wishes to use as a minimum requirement to begin are listed below. Otherwise, the API is unrestricted.

There is no requirements for security in the API endpoints, because the customer wants to be able to offer the information as an open API. It is also desirable that the model and documentation are as detailed and clear as possible for the end users. The customer also wishes to be able to add on to the functionality of the API in the future.

## Specification

The API will initially have two methods.

**Collect all of the measurements for one day, one sensor type, and one unit.**

Examples of how a call could look:

* **/api/v1/devices/testdevice/data/2018-09-18/temperature**
* **getdata?deviceId=testdevice&date=2018-09-18&sensorType=temperature**

**Collect all data points for one unit and one day.**

Examples of how a call could look:

* **/api/v1/devices/testdevice/data/2018-09-18**
* **getdatafordevice?deviceId=testdevice&date=2018-09-18**

Returns temperature, humidity, and rainfall for the day.

# Scoring

The score is calculated as soon as the program is built and structured, and we will examine the code and the solution using the following criteria:

* Transparency: Am I able to understand what the program does by just looking at the code?
* Reasonable: Is the decision on choice of technique well thought-out and grounded?
* Usable: Can the functionality be reused without being re-written, in the event that the customer wishes to add more methods?
* Performance: Does the code effectively solve problems related to latency, memory usage, cpu, network, etc.?
* Simplicity: How simple is the api to use and test, for an end user?
* Testability: Can unit tests be implemented on the code?
* Exemplary: Is this code a good example? We mean, if the code is searched, will we be referred to a solution that shows how similar problems could be solved?

After we make an assessment, we would like to discuss the solution with you. We will discuss the code for the solution as well as how the service collaborates with a complete solution. What improvements could be made to the solution? What additional services might a user wish to have, and how would they fit in with the current solution?

# delivery

We expect the code to be sent to us via email or a Git Repository that we are able to access.

# Aid

**Azure Blob Storage Connectionstring:** BlobEndpoint=https://sigmaiotexercisetest.blob.core.windows.net/;QueueEndpoint=https://sigmaiotexercisetest.queue.core.windows.net/;FileEndpoint=https://sigmaiotexercisetest.file.core.windows.net/;TableEndpoint=https://sigmaiotexercisetest.table.core.windows.net/;SharedAccessSignature=sv=2017-11-09&ss=bfqt&srt=sco&sp=rl&se=2028-09-27T16:27:24Z&st=2018-09-27T08:27:24Z&spr=https&sig=eYVbQneRuiGn103jUuZvNa6RleEeoCFx1IftVin6wuA%3D