# Sample Problem

MCC would like to gain additional visibility into interference on managed wifi services we offer customers as an additional charge. Our routers log metrics – unfortunately, the logs are in XML and not easy to query in a database. Please code up a solution that parses the log files in the attached ZIP file and puts the aggregate data in a database table with the following three columns:

* Serial (string)
* Timestamp (datetime)
* Status (string)

This data can be gathered by looking at the files in the folder that end with WL5GInterference.xml – the serial number can be retrieved from the path (C8D12A6EA4B5 for example) – the timestamp can be constructed from the folder under that (for the YYYYMMDD part), and the first part of the filename for the HHMMSSfff part – where the fff part is milliseconds and the Status can be gathered by looking inside the XML file. Not all devices support this parameter, so on some of them the XML will say invalid parameter name – no need to insert those into the database.

# Bonus Round One

After you get the data imported, make a simple web api for it that returns the data as JSON.

# Bonus Round Two

Make a simple HTML based user interface that uses the web api from Bonus Round One to display the data using Angular, Ionic Framework, React, jQuery, or another HTML framework of your choice – it can be as simple as showing the data in a table or list view.

# Getting Help

Don’t get discouraged if you can’t do all or any of the test – it’s designed to be a challenge. You are free to use whatever tools/technologies you are most comfortable with to get a working solution. After you are done, ZIP up your solution and email it back to me or alternatively post it on github and let me know when it is ready to take a look. You are welcome to use the internet or other resources as much as needed, but please don’t ask for outside help.

If you have questions or need clarification about the problem, contact me via e-mail or phone at 5038291119.

# Hints If We Were Doing It

1. Create a new “Console Application” in Visual Studio 2019 Community edition for C#/.NET core 5 and use the System.IO and System.Xml.Linq classes to process the XML files we are interested in and stuffs the records into the database.
2. Use Entity Framework Core 5 to create/manage your database and table. Put the DbContext and the POCO in a library so you can share it between the Console App and the Web API. You can just use the “localdb” native MSSQL driver that gets installed with Visual Studio – or if you prefer you can use MySQL or PostgreSQL which both have .NET core support via nuget.
3. Create a new “ASP.NET Core Web API” project for your web api in Visual Studio. Test your API using Postman or something similar if you like. Returning the data as a simple GET request is sufficient – no need to implement full CRUD actions although you can if you want.
4. Install a recent version of node.js if you don’t already have it installed, and follow the guide at <https://ionicframework.com/docs/intro/cli> to create a blank ionic framework app then open that folder in a good HTML/Javascript/Typescript editor like Visual Studio Code. Google to find examples how to connect your web API and display the information in a listview or table in the ionic app.