C# Application uses the following libraries:

* RestSharp
* Newtonsoft.Json
* Newtonsoft.Json.Linq

Data comes from the NetGate device’s XML file [details.xml]

* URL for direct Ethernet Cable connection: <http://169.254.1.1/details.xml>
* URL for main page: <http://169.254.1.1/index.htm>

1. We used Postman to do a GET call to the detail.xml
2. We Used Postman’s code generation to automatically create RestSharp C# code for us.

|  |
| --- |
| string xmlURL = "http://169.254.1.1/details.xml"  var client = new RestClient(xmlURL);  var request = new RestRequest(Method.GET);  request.AddHeader("cache-control", "no-cache");  request.AddHeader("Connection", "keep-alive");  request.AddHeader("Accept-Encoding", "gzip, deflate");  request.AddHeader("Host", "169.254.1.1");  request.AddHeader("Cache-Control", "no-cache");  request.AddHeader("Accept", "\*/\*");  request.AddHeader("User-Agent", "PostmanRuntime/7.17.1");  request.AddHeader("Content-Type", "application/json");  IRestResponse response = client.Execute(request);  Console.WriteLine(response.Content); |

1. We ran the XML response through Newtonsoft’s XML to JSON converter

|  |
| --- |
| XmlDocument doc = new XmlDocument();  doc.LoadXml(xmlData);  string json = JsonConvert.SerializeXmlNode(doc);  Console.WriteLine(json); |

1. We then used <https://jsonlint.com/> to view it and just get a better look at the data.
2. Next we created a set of C# classes to represent the data using [http://json2csharp.com/#](http://json2csharp.com/)
3. We do this in a loop.
4. The starting state temperatures are saved.
5. Each time through the loop we:
   1. Update the current temperatures (getting the sensor data)
   2. Check for the Next sensor
      1. Remove any previously found sensors that are already in the “Hot List”
      2. Check if any sensor is above 2 degrees C from the Starting State
      3. If so, add it to our “Hot List”
   3. Check if we are done:
      1. If the # of sensors on the wire is equal to the number of sensors in our “Hot List” then we are done.
6. We generate a report and save it with the tech’s name, and high level information about the wire. Included in the output is the in-order list of SensorID’s.
7. The next step would be for someone to BURN those to the EEPROM.