CSV Stream-Parser

Thursday, May 19, 2022 9:15 AM

Using any object-oriented language that you are already familiar, write a program that takes a CSV file as argument (it can be a either local and optionally remote* file) and parse it on a streaming-fashion*.

After each successfully parsed CSV row, the parser should trigger a specific event by returning all the CSV cell values and headers in a Dictionary structure*.

Think about the solution on a more generic way. Your interfaces may be used in the future to implement many other file object definition structures as JSON, XML, etc.

To keep in mind

----

1. Remote files can be usually accessed through the HTTP protocol. But this is an advanced feature.
2. Streaming-fashion: We don't want the entire file to be loaded in memory. Instead use a Stream object.
3. In .NET6 a Dictionary structure is any instance of an IDictionary object.
4. Do not forget to write unit tests that covers some of the most important edge-cases, I.e. empty rows.
5. The preferable framework to use is .NET6.

Program.cs Example

```
using System.Collections;

Stream csvFileStream = new FileStream("C://persons.csv", FileMode.Open, FileAccess.Read);


CSVStreamParser csvStreamParser = new CSVStreamParser(csvFileStream); // Class to be implemented...
csvStreamParser.CallbackClass(new OutputHandler()); // Register the class where the parser will trigger the RowParsedEvent(IDictionary row) event...
csvStreamParser.StartParsing();


internal class OutputHandler : IStreamEvents // Interface to be implemented...
{
    /*
     * Event method triggered by the CSVStreamParser class...
     */
    public void RowParsedEvent(IDictionary row)
    {
        foreach (DictionaryEntry item in row)
        {
            Console.WriteLine("{0} : {1}", (string)item.Key, item.Value);
        }
    }
}
```