

Assignment No 2

Important Note:

- For all the questions, you must use Visual Studio and upload all files of the solution.
- Create separate solutions files for each question (i.e. one project per solution)
- Every Question should be named as “k<Year><StudentID>_Q<QuestionNumber>” for instance: k190123_Q1
- Your Assignment should have proper Objected Oriented Programming and Data Structures
- Last Date for Submission: **Nov 20, 2022 (11:59 PM)**
- There will be no extensions.
- All URLs, File Paths will be included in the app.config or web.config. No Hardcoding
- Your submission will be marked as 0 if:
 - Submitted late.
 - The submitted assignment does not open or the file is corrupted.
 - Assignments submitted via any medium except for Google Classroom.
 - Copied amongst students or from another source

Question No 1: Downloading a Web Page Periodically

Create a Windows Service which downloads a web page after every 5 minutes. Hint: You can use the “Process” class available in Dot Net to execute the code which you had created in Assignment 1 Question 1.

Question No 2: Parsing the Data

Create a Windows Service which parses the data from the previous question after every 10 minutes. You should use the parsing logic implemented in the previous assignment (A1Q2) by including the logic as a class library (dll). After every 10 minutes, a new file would be created and the previous one will remain in the same folder.

Note: Folder logic is the same as mentioned in Assignment 1.

Question No 3: Historical Data using Azure Function

Create a timer based Azure Function which is responsible to combine XML files and generate a JSON file for each of the script. You will have one JSON file output per script. The input for this service is the output directory of Question 2. The output of this question will be as following:

Information Processing Techniques

- Folder Structure:
 - <some folder>\<CategoryNameFolder>\<scriptId-Accumulated>.json

Example:
D:\Output\AutomobileAssembler\AtlasHondaLimited.json
- JSON File Structure:
 - {
 "scriptData":
 "lastUpdatedOn":"\Date(1617465735294)\",
 [
 {"Date":"\Date(1617465735295)\", "Price":10},
 {"Date":"\Date(1617466035295)\", "Price":12},
 {"Date":"\Date(1617466335295)\", "Price":11},
 {"Date":"\Date(1617466635295)\", "Price":14},
 {"Date":"\Date(1617466935295)\", "Price":10}
]
 }

The service will execute after every 20 minutes. If the JSON file does not exist, it will create a new file but if the file already exists, it will append to the same file. The field lastUpdatedOn will be changed whenever the file has been last modified. Once the service has read the XML file and generated a JSON file, it will delete the XML file.

Question No 4: Azure Function as an API

Create an Azure Function (Http Trigger) where the user will pass the script name as a query string parameter and the function will return the data from the json file created in Question 3.

Example: Browser URL: <http://localhost/myAzureFunction/AtlasHondaLimited>

Output:

```
{
  "scriptData":
    "lastUpdatedOn":"\Date(1617465735294)\",
    [
      {"Date":"\Date(1617465735295)\", "Price":10},
      {"Date":"\Date(1617466035295)\", "Price":12},
      {"Date":"\Date(1617466335295)\", "Price":11},
      {"Date":"\Date(1617466635295)\", "Price":14},
      {"Date":"\Date(1617466935295)\", "Price":10}
    ]
}
```