Exercise - Create files and directories

4 minutes

100 XP

The `File` and `Directory` classes in .NET let you create new files and directories programmatically.

So far, you've created a robust command-line application in .NET that can read any folder structure to find files with a .json extension. Now, you need to read those files to summarize the data in them. Then, write the totals to a new file in a new directory called *salesTotals*.

In this exercise, you create the *salesTotalDir* directory and the *totals.txt* file where the sales totals are collated.

Create the SalesTotals directory

1. In the `Program.cs` file, remove the `foreach` loop that iterates and writes each filename returned from the `FindFiles` function to the *Console* output. This change results in the `salesFiles` variable going unused. However, we'll leave it in here for now because it will be used again in a future lesson.

2. In the `Program.cs` file, create a variable called `salesTotalDir`, which holds the path to the *salesTotalDir* directory:

   C# Copy

   ```
   var currentDirectory = Directory.GetCurrentDirectory();
   var storesDirectory = Path.Combine(currentDirectory, "stores");

   var salesTotalDir = Path.Combine(currentDirectory, "salesTotalDir");

   var salesFiles = FindFiles(storesDirectory);
   ```

3. In the `Program.cs` file, add code to create the directory:

   C# Copy

   ```
   var currentDirectory = Directory.GetCurrentDirectory();
   var storesDirectory = Path.Combine(currentDirectory, "stores");

   var salesTotalDir = Path.Combine(currentDirectory, "salesTotalDir");
   Directory.CreateDirectory(salesTotalDir);   // Add this line of code

   var salesFiles = FindFiles(storesDirectory);
   ```

Write the totals.txt file

1. In the `Program.cs` file, add the code to create an empty file called *totals.txt* inside the newly created *salesTotalDir* directory. Use an empty string for the file's contents for now:

   C# Copy

   ```
   var currentDirectory = Directory.GetCurrentDirectory();
   var storesDirectory = Path.Combine(currentDirectory, "stores");

   var salesTotalDir = Path.Combine(currentDirectory, "salesTotalDir");
   Directory.CreateDirectory(salesTotalDir);

   var salesFiles = FindFiles(storesDirectory);

   File.WriteAllText(Path.Combine(salesTotalDir, "totals.txt"), String.Empty);
   ```

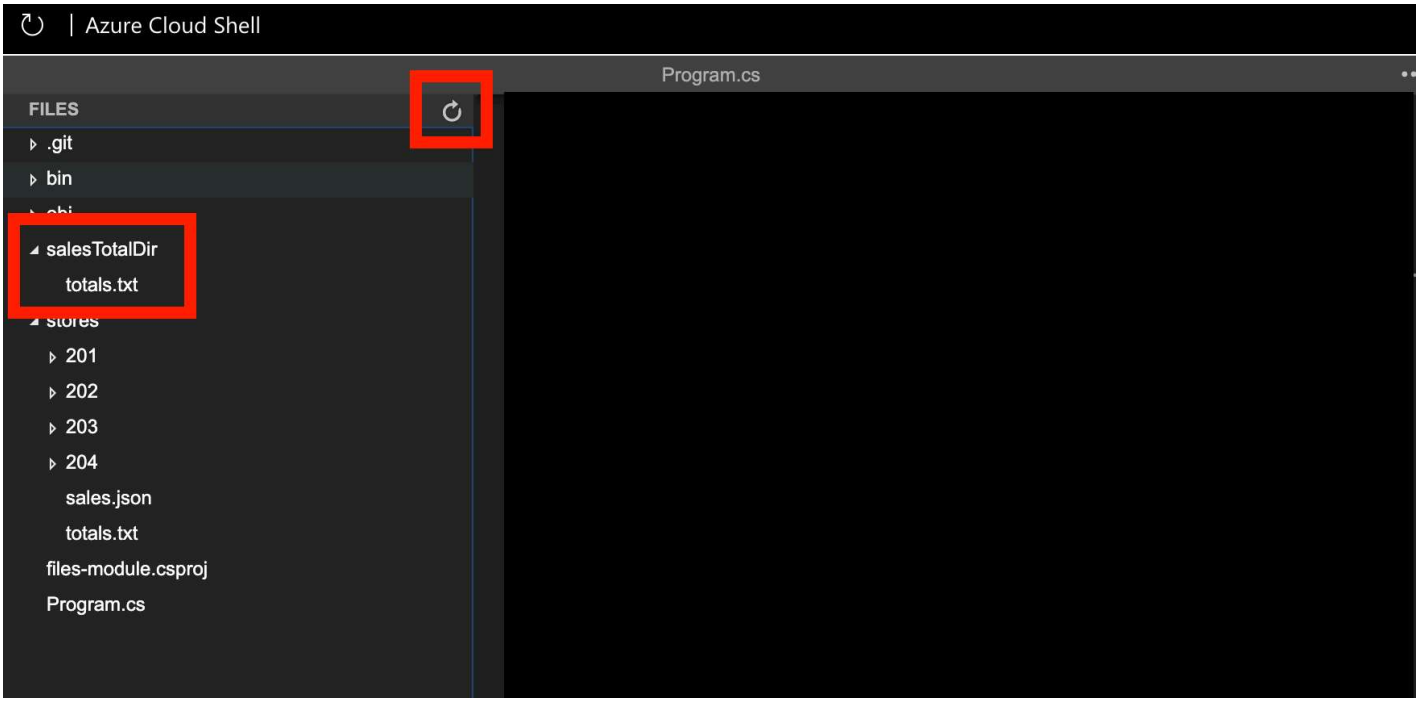
2. Press `Ctrl+S` / `Cmd+S` to save the file.

3. Run the program by running the following code from the terminal prompt:

   Bash Copy

   ```
   dotnet run
   ```

4. Select the **Refresh** icon in the **Files** explorer.



You're almost finished. The last step is to read the sales files, add up the totals, and write the grand total to the new *totals.txt* file. Next, you learn how to read and parse data inside files.

Got stuck?

If you got stuck during this exercise, here's the full code up to this point:

C# Copy

```
var currentDirectory = Directory.GetCurrentDirectory();
var storesDirectory = Path.Combine(currentDirectory, "stores");

var salesTotalDir = Path.Combine(currentDirectory, "salesTotalDir");
Directory.CreateDirectory(salesTotalDir);
var salesFiles = FindFiles(storesDirectory);

File.WriteAllText(Path.Combine(salesTotalDir, "totals.txt"), String.Empty);

IEnumerable<string> FindFiles(string folderName)
{
    List<string> salesFiles = new List<string>();

    var foundFiles = Directory.EnumerateFiles(folderName, "*", SearchOption.AllDirectories);

    foreach (var file in foundFiles)
    {
        var extension = Path.GetExtension(file);
        if (extension == ".json")
        {
            salesFiles.Add(file);
        }
    }

    return salesFiles;
}
```