Exercise - Work with paths

6 minutes

100 XP

The .NET `Path` class and `Directory.GetCurrentDirectory` are two ways to define and compose file-system paths.

In the previous exercise, you wrote a program that iterates through a folder to find any of the *sales.json* files in it or any subfolders.

In this exercise, you use the `Path` class and `Directory.GetCurrentDirectory` to improve the program so it finds *any* file with a .json extension.

Use the current directory and combine paths

In the current Program.cs code, you're passing the static location of the *stores* folder. Now, we change that code to use the `Directory.GetCurrentDirectory` value instead of passing a static folder name.

1. In the editor, insert the following code above the first line of `Program.cs` file. This code uses the `Directory.GetCurrentDirectory` method to obtain the path for the current directory and store it in a new variable `currentDirectory`:

   C#

   Copy

   ```
   var currentDirectory = Directory.GetCurrentDirectory();
   ```

2. Insert the following code after the one that you just added. This code uses the `Path.Combine` method to create the full path to the *stores* directory and store it in a new variable `storesDirectory`:

   C#

   Copy

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

3. Replace the string `stores` in the `FindFiles` function call with the new variable `storesDirectory`:

   C#

   Copy

   ```
   var salesFiles = FindFiles(storesDirectory);
   ```

   The top of your file should now look similar to the following snippet:

   C#

   Copy

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

   foreach (var file in salesFiles)
   {
       Console.WriteLine(file);
   }
   ```

4. Press `Ctrl+S` (or `Cmd+S` macOS) to save the file.

5. Run the program from the command line:

   Bash

   Copy

   ```
   dotnet run
   ```

6. The program should show the following output:

   Output

   Copy

   ```
   /home/username/dotnet-files/stores/sales.json
   /home/username/dotnet-files/stores/201/sales.json
   /home/username/dotnet-files/stores/202/sales.json
   /home/username/dotnet-files/stores/203/sales.json
   /home/username/dotnet-files/stores/204/sales.json
   ```

   Notice that the file names returned include the full system path. This path is included because `Directory.GetCurrentDirectory` method returns the full path to the current location.

Find all .json files

Instead of looking for only *sales.json* files, the program needs to search for any file with a .json extension. To do that, you can use the `Path.GetExtension` method to check the extension for each file.

1. In the `foreach` loop that iterates through `foundFiles`, insert the following line of code above the `if` statement to define a new variable `extension`. This code uses the `Path.GetExtension` method to get the extension of each file.

   C#

   Copy

   ```
       var extension = Path.GetExtension(file);
   ```

2. Change the `if` statement to look like the following line of code. This statement checks whether the file's extension is equal to .json.

   C#

   Copy

   ```
           if (extension == ".json")
   ```

   The `foreach` loop should look similar to the following code:

   C#

   Copy

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

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

4. Run the program from the command line:

   Bash

   Copy

   ```
   dotnet run
   ```

   The output now shows all .json files in each of the store ID directories:

   Output

   Copy

   ```
   /home/username/dotnet-files/stores/sales.json
   /home/username/dotnet-files/stores/201/sales.json
   /home/username/dotnet-files/stores/201/salestotals.json
   /home/username/dotnet-files/stores/202/sales.json
   /home/username/dotnet-files/stores/202/salestotals.json
   /home/username/dotnet-files/stores/203/sales.json
   /home/username/dotnet-files/stores/203/salestotals.json
   /home/username/dotnet-files/stores/204/sales.json
   /home/username/dotnet-files/stores/204/salestotals.json
   ```

Great job! You've used the `Path` class and the `Directory.GetCurrentDirectory` method to make the program much more robust. In the next unit, you'll learn how to create directories and move files between locations.

Got stuck?

If you got stuck at any point in this exercise, here's the completed code. Remove everything in *Program.cs* and replace it with this solution:

C#

Copy

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

var salesFiles = FindFiles(storesDirectory);

foreach (var file in salesFiles)
{
```

```
    Console.WriteLine(file);
}

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;
}
```