**Hands-on .NetCore 3.0**

**Reading Files**

Table of Contents

[Scaffolding Console App Basics 2](#_Toc23624076)

[ReadingFiles Console 2](#_Toc23624077)

[Creating the console app 2](#_Toc23624078)

[Reading CSV File 2](#_Toc23624079)

[Opening File and Counting Lines 3](#_Toc23624080)

[Handling exception 3](#_Toc23624081)

[Displaying the file content 4](#_Toc23624082)

# Scaffolding Console App Basics

We have already run the commands multiple times, here is the info

* dotnet new console
* dotnet restore [pulls in the dependencies needed by the application]
* dotnet run [compiles and run the application]
* dotnet build [compiles the application]
* dotnet publish [packages up the files for reuse]

Take a look at this document for more details:

<https://itplate.blogspot.com/2019/11/scaffolding-applications-with-net-cli.html>

# ReadingFiles Console

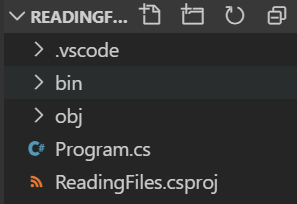
## Creating the console app

Run command **dotnet new console -o ReadingFiles**

It has done the restore for us as well

Open the app with VS Code by

1. **cd ReadingFiles**
2. and then typing **code .** [code space dot]



## Reading CSV File

We’ll be reading a CSV file so will need to create it first. Open notepad, put the following info in it and then save it in the same working folder as .csv extension. You can download the file from <https://support.staffbase.com/hc/en-us/articles/360007108391-CSV-File-Examples>

Username,Identifier,First name,Last name

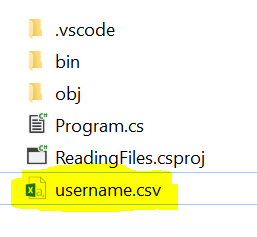
booker12,9012,Rachel,Booker

grey07,2070,Laura,Grey

johnson81,4081,Craig,Johnson

jenkins46,9346,Mary,Jenkins

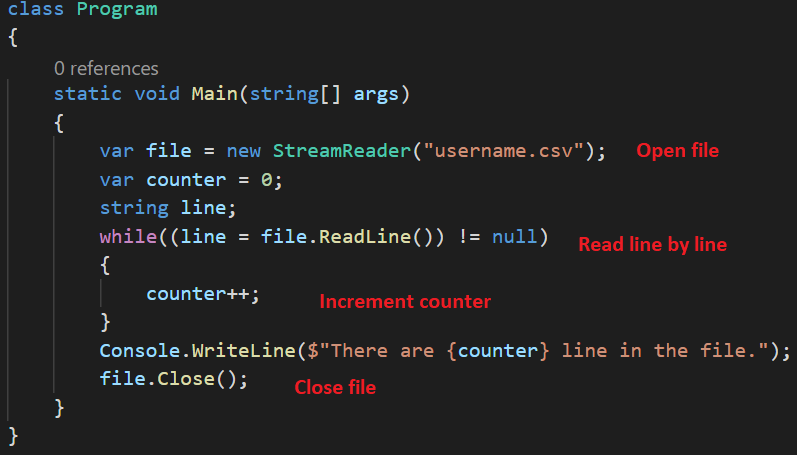
smith79,5079,Jamie,Smith



## Opening File and Counting Lines

Make sure to add a reference to **using System.IO;**

We’ll be using the StreamReader to read the file line by line, increment the counter and then will close the file once done.

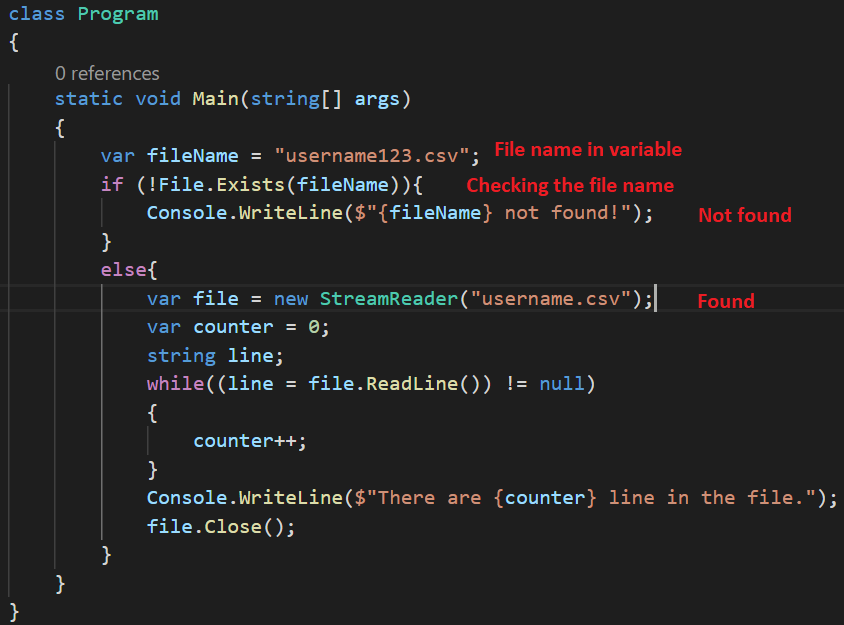


And then do **dotnet run**



## Handling exception

If we change the file name and rerun the app, we’ll get an exception. Lets check the file first and then perform the read.



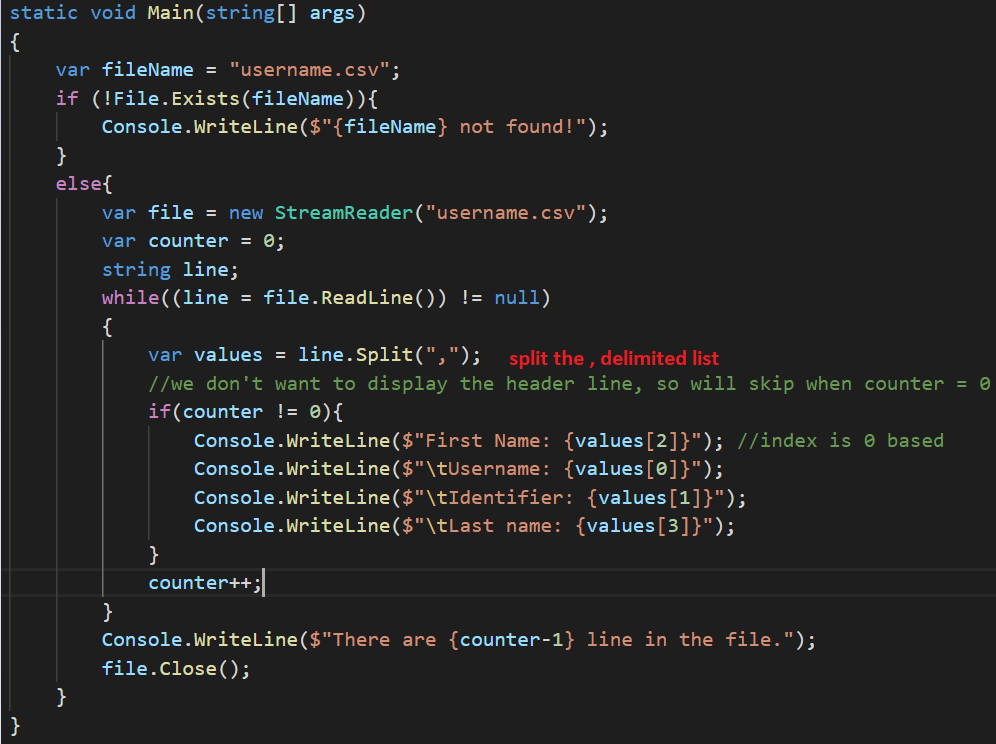
And then the result



Change the filename back to good file name and test again.

## Displaying the file content

We’ll work with the string array to split the line content to display.



Run the app using **dotnet run**

