Creating Automated Tests for File Access Code

Jason Roberts

.NET Developer

@robertsjason   dontcodetired.com

Overview

The problems with testing file access code

An overview of System.IO.Abstractions

Refactor processors for testability

Add a TextFileProcessor test

Add a BinaryFileProcessor test

Add a CsvFileProcessor test

Asserting mock file output with ApprovalTests

The Problems with Testing File Access Code

Physical path(s)

Test execution speed

Setup complexity

Additional cleanup code

Read/write?

Security?

Every PC?

CI server?

IO overhead

Slow to run

Later feedback

Run as often?

Files/dirs exist?

Existing files?

Overwrite?

Test files

Test directories

An Overview of System.IO.Abstractions

Abstraction layer for the file system

Provides testable file IO access

IFileSystem interface

FileSystem class

An Overview of System.IO.Abstractions

Aims to have same API as:
System.IO.File
System.IO.Directory
Etc.

System.IO.File.ReadAllText(...)

IFileSystem.File.ReadAllText(...)

An Overview of System.IO.Abstractions

Production

Test

FileSystem instance

Mock IFileSystem
System.IO.Abstractions.TestingHelpers
MockFileSystem
MockFileData

Summary

The problems with testing file access code

- Test execution speed

- Setup/clean-up code

An overview of System.IO.Abstractions

Refactored processors for testability

- IFileSystem constructor parameter

- new FileSystem()

Added tests

- new MockFileData(...)

- new MockFileSystem()

- mockFileSystem.AddFile(...)

- mockFileSystem.GetFile(...)

Approvals.Verify(...)

Resources and Further Learning

Pluralsight courses:

- Testing .NET Code with xUnit.net 2: Getting Started [https://bit.ly/psxunitcore]

- Dynamic C# Fundamentals [https://bit.ly/psdynamic]

- Approval Tests for .NET [https://bit.ly/approvaltests]

Documentation:

- System.IO [https://bit.ly/pssystemio6]

- CsvHelper [https://bit.ly/pscsvhelper]

Code:

- System.IO.Abstractions [https://bit.ly/pssysio]

- CsvHelper [https://bit.ly/csvhelpercode]