

File Logger Kata

Source: <https://github.com/ardalis/kata-catalog>

Background

This kata is designed to help develop skill with using mock objects appropriately. The initial steps can be completed using direct infrastructure code (that is, actually creating files and getting times from the system clock directly), but later steps become increasingly difficult to test without the ability to mock out certain dependencies.

Try not to read ahead or guess what the next requirement might be - just complete one numbered step at a time.

Instructions

1. Write a class 'FileLogger' with one method, `Log(string message)`.
2. When this method is called, it should append the message to the end of a file, "log.txt", located in the same folder as the running application (or tests).
3. Messages should default to having a prefix of "YYYY-MM-DD HH:MM:SS " in front of the message. For example, `Log("test")` would yield "2020-07-14 10:25:23 test". Time can be in local timezone and should use 24-hour clock.
4. If the file doesn't exist, create it. If it does exist, use it and append to it.
5. Now update the method so that it writes to a file called `logYYYYMMDD.txt`, where `YYYYMMDD` corresponds to the current date. **Note:** It's not unusual for the logger instance to run for a long period of time, and the log messages should go into the appropriate file based on when they are logged, not when the logger was created.
6. Verify that a new file is created if it doesn't exist on each new day.
7. The IT manager doesn't want to have to open multiple files on Mondays. Any time logging is occurring on a Saturday or Sunday, have it log to a file called "weekend.txt". If it already exists, it can just append to it.

8. Actually, the manager just gave us new requirements. The first time you log to a file on a new weekend, make sure you start with a fresh "weekend.txt" file. Rename the old one based on the date of the Saturday of that weekend, e.g. weekend-YYYYMMDD.txt.

Example of final weekend behavior

Let's say the month starts on Saturday the 1st (e.g. 1 Feb 2020) and currently there are no log files present. Logging on Saturday the 1st goes to a file `weekend.txt`. Logging on Sunday the 2nd continues to go to `weekend.txt`. Throughout the week, new files are created for each date `log20200203.txt`, `log20200204.txt`, etc. Then Saturday 8 Feb 2020 rolls around and the last requirements comes into play. The existing `weekend.txt` file has metadata indicating it was created or last modified on Feb 1st/2nd (respectively). Thus, the logger renames the file to `weekend-20200201.txt` corresponding to the Saturday of that weekend log file. It then creates a new `weekend.txt` which is used for the rest of the 8th. And again on the 9th. That file will be renamed on 15 Feb to `weekend-20200208.txt`.

Note: An existing `weekend.txt` file isn't necessarily from the previous weekend (6 days ago). It could be that nothing was logged for several weeks. It's necessary to inspect the file to see when it was created/lastmodified to know what it should be renamed to.

Resources

- For .NET, [NSubstitute](#) or [via nuget](#)