

Acceptance criteria for the files Subplot library

The Subplot project

2021-05-09 05:54

Contents

1	Introduction	1
2	Create on-disk files from embedded files	1
3	File metadata	2
4	File modification time	2
5	File contents	2
6	Directories	2

1 Introduction

The Subplot¹ library `files` provides scenario steps and their implementations for managing files on the file system during tests. The library consists of a bindings file `lib/files.yaml` and implementations in Python in `lib/files.py`.

This document explains the acceptance criteria for the library and how they're verified. It uses the steps and functions from the `files` library.

2 Create on-disk files from embedded files

Subplot allows the source document to embed test files, and the `files` library provides steps to create real, on-disk files from the embedded files.

```
given file hello.txt  
then file hello.txt exists  
and file hello.txt contains "hello, world"  
and file other.txt does not exist  
given file other.txt from hello.txt
```

¹<https://subplot.liw.fi/>

then file **other.txt** exists
and files **hello.txt** and **other.txt** match
and only files **hello.txt**, **other.txt** exist

File: **hello.txt**

1 hello, world

3 File metadata

These steps create files and manage their metadata.

given file **hello.txt**
when I remember metadata for file **hello.txt**
then file **hello.txt** has same metadata as before
when I write "yo" to file **hello.txt**
then file **hello.txt** has different metadata from before

4 File modification time

These steps manipulate and test file modification times.

given file **foo.dat** has modification time **1970-01-02 03:04:05**
then file **foo.dat** has a very old modification time
when I touch file **foo.dat**
then file **foo.dat** has a very recent modification time

5 File contents

These steps verify contents of files.

given file **hello.txt**
then file **hello.txt** contains "hello, world"
and file **hello.txt** matches regex "hello, .*"
and file **hello.txt** matches regex /hello, .*/

6 Directories

There are also a large number of directory based steps and some directory based behaviour available in creating files which are available in the files library.

given a directory **first**
then directory **first** exists
and directory **first** is empty
and directory **second** does not exist
when I remove directory **first**

then directory **first** does not exist
when I create directory **second**
then directory **second** exists
and directory **second** is empty
given file **second/third/hello.txt** from **hello.txt**
then directory **second** is not empty
and directory **second/third** exists
and directory **second/third** is not empty
when I remove directory **second**
then directory **second** does not exist