

python / cpython Public

<> Code Issues 5k+ Pull requests 2.2k Actions Projects Security and q

# Commit 89bfb8e



ambv authored last week · 83 / 86 · Verified

[3.13] [gh-143930](#): Tweak the exception message and increase test coverage ([GH-146476](#)) ([GH-148045](#))

(cherry picked from commit [3681d47](#))

3.13 (AcreationOS-Linux/python#2, #148045) · v3.13.13

1 parent [1a188aa](#) commit 89bfb8e

3 files changed +77 -7 lines changed

Top

- ✓ Lib
  - ✓ test
    - test\_webbrowser.py
    - webbrowser.py
- ✓ Misc/NEWS.d/next/Security
  - 2026-01-16-12-04-49.gh-issue-143930.zYC5x3.rst

3 files changed +77 -7 lines changed

```

Lib/test/test_webbrowser.py
... @@ -1,3 +1,4 @@
1 + import io
1 2 import os
2 3 import re
3 4 import shlex
... @@ -55,6 +56,14 @@ def _test(self, meth, *, args=[URL], kw={}, options,
arguments):

```

```

55 56         popen_args.pop(popen_args.index(option))
56 57         self.assertEqual(popen_args, arguments)
57 58
59 +     def test_reject_dash_prefixes(self):
60 +         browser = self.browser_class(name=CMD_NAME)
61 +         with self.assertRaisesRegex(
62 +             ValueError,
63 +             r"^Invalid URL \(\leading dash disallowed\): '--key=val http.*'$"
64 +         ):
65 +             browser.open(f"--key=val {URL}")
66 +
58 67
59 68     class GenericBrowserCommandTest(CommandTestMixin, unittest.TestCase):
60 69
@@ -65,11 +74,6 @@ def test_open(self):
65 74         options=[],
66 75         arguments=[URL])
67 76
68 -     def test_reject_dash_prefixes(self):
69 -         browser = self.browser_class(name=CMD_NAME)
70 -         with self.assertRaises(ValueError):
71 -             browser.open(f"--key=val {URL}")
72 -
73 77
74 78     class BackgroundBrowserCommandTest(CommandTestMixin, unittest.TestCase):
75 79
@@ -306,6 +310,72 @@ def test_open_new_tab(self):
306 310         self._test('open_new_tab')
307 311
308 312
313 + class MockPopenPipe:
314 +     def __init__(self, cmd, mode):
315 +         self.cmd = cmd
316 +         self.mode = mode
317 +         self.pipe = io.StringIO()
318 +         self._closed = False
319 +
320 +     def write(self, buf):
321 +         self.pipe.write(buf)

```

```
322 +
323 +     def close(self):
324 +         self._closed = True
325 +         return None
326 +
327 +
328 + @unittest.skipUnless(sys.platform == "darwin", "macOS specific test")
329 + @requires_subprocess()
330 + class MacOSXOSAScriptTest(unittest.TestCase):
331 +     def setUp(self):
332 +         # Ensure that 'BROWSER' is not set to 'open' or something else.
333 +         # See: https://github.com/python/cpython/issues/131254.
334 +         env = self.enterContext(os_helper.EnvironmentVarGuard())
335 +         env.unset("BROWSER")
336 +
337 +         support.patch(self, os, "popen", self.mock_popen)
338 +         self.browser = webbrowser.MacOSXOSAScript("default")
339 +
340 +     def mock_popen(self, cmd, mode):
341 +         self.popen_pipe = MockPopenPipe(cmd, mode)
342 +         return self.popen_pipe
343 +
344 +     def test_default(self):
345 +         browser = webbrowser.get()
346 +         assert isinstance(browser, webbrowser.MacOSXOSAScript)
347 +         self.assertEqual(browser.name, "default")
348 +
349 +     def test_default_open(self):
350 +         url = "https://python.org"
351 +         self.browser.open(url)
352 +         self.assertTrue(self.popen_pipe._closed)
353 +         self.assertEqual(self.popen_pipe.cmd, "osascript")
354 +         script = self.popen_pipe.pipe.getvalue()
355 +         self.assertEqual(script.strip(), f'open location "{url}"')
356 +
357 +     def test_url_quote(self):
358 +         self.browser.open('https://python.org/"quote"')
359 +         script = self.popen_pipe.pipe.getvalue()
360 +         self.assertEqual(
361 +             script.strip(), 'open location "https://python.org/%22quote%22"')
```

```

362 +     )
363 +
364 +     def test_explicit_browser(self):
365 +         browser = webbrowser.MacOSXOSAScript("safari")
366 +         browser.open("https://python.org")
367 +         script = self.popen_pipe.pipe.getvalue()
368 +         self.assertIn('tell application "safari"', script)
369 +         self.assertIn('open location "https://python.org"', script)
370 +
371 +     def test_reject_dash_prefixes(self):
372 +         with self.assertRaisesRegex(
373 +             ValueError,
374 +             r"^Invalid URL \(leading dash disallowed\): '--key=val http.*'$"
375 +         ):
376 +             self.browser.open(f"--key=val {URL}")
377 +
378 +
309 379     class BrowserRegistrationTest(unittest.TestCase):
310 380
311 381         def setUp(self):

```



Lib/webbrowser.py



```

@@ -168,7 +168,7 @@ def open_new_tab(self, url):
168 168     def _check_url(url):
169 169         """Ensures that the URL is safe to pass to subprocesses as a
        parameter"""
170 170         if url and url.lstrip().startswith("-"):
171 -             raise ValueError(f"Invalid URL: {url}")
171 +             raise ValueError(f"Invalid URL (leading dash disallowed): {url!r}")
172 172
173 173
174 174     class GenericBrowser(BaseBrowser):

```



...6-01-16-12-04-49.gh-issue-143930.zYC5x3.rst



```

@@ -1 +1 @@
1 - Reject leading dashes in URLs passed to :func:`webbrowser.open`
1 + Reject leading dashes in URLs passed to :func:`webbrowser.open`.

```

**Comments** 0