

python / cpython Public

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

# Commit 96fc504



miss-islington and ambv authored last week · ✖ 60 / 74 · Partially verified

[3.11] [gh-143930](#): Tweak the exception message and increase test coverage ([GH-146476](#)) ([GH-148045](#)) ([GH-148051](#)) ([GH-148052](#))

(cherry picked from commit [cc02351](#))  
(cherry picked from commit [89bfb8e](#))  
(cherry picked from commit [3681d47](#))

Co-authored-by: Łukasz Langa <lukasz@langa.pl>

🔑 3.11 (#98846, #148052)

1 parent [a78cbcd](#) commit 96fc504

3 files changed +77 -8 lines changed

↑ Top ⚙️

Filter files...

- 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 -8 lines changed

Search within code ⚙️

```

Lib/test/test_webbrowser.py
... @@ -1,6 +1,7 @@
1 + import io
2 + import os
1 3 import webbrowser

```

```

2     4     import unittest
3     - import os
4     5     import sys
5     6     import subprocess
6     7     from unittest import mock
@@ -49,6 +50,14 @@ def _test(self, meth, *, args=[URL], kw={}, options,
arguments):
49    50         popen_args.pop(popen_args.index(option))
50    51         self.assertEqual(popen_args, arguments)
51    52
53    +     def test_reject_dash_prefixes(self):
54    +         browser = self.browser_class(name=CMD_NAME)
55    +         with self.assertRaisesRegex(
56    +             ValueError,
57    +             r"^Invalid URL \(leading dash disallowed\): '--key=val http.*'$"):
58    +             browser.open(f"--key=val {URL}")
59    +
60    +
52    61
53    62     class GenericBrowserCommandTest(CommandTestMixin, unittest.TestCase):
54    63
@@ -59,11 +68,6 @@ def test_open(self):
59    68         options=[],
60    69         arguments=[URL])
61    70
62    -     def test_reject_dash_prefixes(self):
63    -         browser = self.browser_class(name=CMD_NAME)
64    -         with self.assertRaises(ValueError):
65    -             browser.open(f"--key=val {URL}")
66    -
67    71
68    72     class BackgroundBrowserCommandTest(CommandTestMixin, unittest.TestCase):
69    73
@@ -224,6 +228,71 @@ def test_open_new_tab(self):
224   228         arguments=['openURL({}, new-tab)'.format(URL)]
225   229
226   230
231  +     class MockPopenPipe:
232  +         def __init__(self, cmd, mode):

```

```
233 +         self.cmd = cmd
234 +         self.mode = mode
235 +         self.pipe = io.StringIO()
236 +         self._closed = False
237 +
238 +     def write(self, buf):
239 +         self.pipe.write(buf)
240 +
241 +     def close(self):
242 +         self._closed = True
243 +         return None
244 +
245 +
246 + @unittest.skipUnless(sys.platform == "darwin", "macOS specific test")
247 + class MacOSXOSAScriptTest(unittest.TestCase):
248 +     def setUp(self):
249 +         # Ensure that 'BROWSER' is not set to 'open' or something else.
250 +         # See: https://github.com/python/cpython/issues/131254.
251 +         env = self.enterContext(os_helper.EnvironmentVarGuard())
252 +         env.unset("BROWSER")
253 +
254 +         support.patch(self, os, "popen", self.mock_popen)
255 +         self.browser = webbrowser.MacOSXOSAScript("default")
256 +
257 +     def mock_popen(self, cmd, mode):
258 +         self.popen_pipe = MockPopenPipe(cmd, mode)
259 +         return self.popen_pipe
260 +
261 +     def test_default(self):
262 +         browser = webbrowser.get()
263 +         assert isinstance(browser, webbrowser.MacOSXOSAScript)
264 +         self.assertEqual(browser.name, "default")
265 +
266 +     def test_default_open(self):
267 +         url = "https://python.org"
268 +         self.browser.open(url)
269 +         self.assertTrue(self.popen_pipe._closed)
270 +         self.assertEqual(self.popen_pipe.cmd, "osascript")
271 +         script = self.popen_pipe.pipe.getvalue()
272 +         self.assertEqual(script.strip(), f'open location "{url}"')
```

```

273 +
274 +     def test_url_quote(self):
275 +         self.browser.open('https://python.org/"quote"')
276 +         script = self.popen_pipe.pipe.getvalue()
277 +         self.assertEqual(
278 +             script.strip(), 'open location "https://python.org/%22quote%22"'
279 +         )
280 +
281 +     def test_explicit_browser(self):
282 +         browser = webbrowser.MacOSXOSAScript("safari")
283 +         browser.open("https://python.org")
284 +         script = self.popen_pipe.pipe.getvalue()
285 +         self.assertIn('tell application "safari"', script)
286 +         self.assertIn('open location "https://python.org"', script)
287 +
288 +     def test_reject_dash_prefixes(self):
289 +         with self.assertRaisesRegex(
290 +             ValueError,
291 +             r"^Invalid URL \(leading dash disallowed\): '--key=val http.*'$":
292 +         ):
293 +             self.browser.open(f"--key=val {URL}")
294 +
295 +

```

```

227 296     class BrowserRegistrationTest(unittest.TestCase):
228 297
229 298         def setUp(self):

```



Lib/webbrowser.py



```
@@ -159,7 +159,7 @@ def open_new_tab(self, url):
```

```

159 159     def _check_url(url):
160 160         """Ensures that the URL is safe to pass to subprocesses as a
161 161         parameter"""
162 161         if url and url.lstrip().startswith("-"):
162 -             raise ValueError(f"Invalid URL: {url}")
162 +             raise ValueError(f"Invalid URL (leading dash disallowed): {url!r}")
163 163
164 164
165 165     class GenericBrowser(BaseBrowser):

```



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

|     |  |
|-----|--|
| ... | @@ -1 +1 @@  |
| 1   | - Reject leading dashes in URLs passed to <code>:func:`webbrowser.open`</code>   |
| 1   | + Reject leading dashes in URLs passed to <code>:func:`webbrowser.open`</code> . |

Comments 0