

python / cpython Public

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

Commit 591ed89



miss-islington and ambv authored last week · ✖ 59 / 66 · Partially verified

[3.10] [gh-143930](#): Tweak the exception message and increase test coverage ([GH-146476](#)) ([GH-148045](#)) ([GH-148051](#)) ([GH-148053](#))

(cherry picked from commit [cc02351](#))
(cherry picked from commit [89bf8e](#))
(cherry picked from commit [3681d47](#))

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

3.10 (#148053)

1 parent [0ee938d](#) commit 591ed89

3 files changed +79 -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 +79 -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
@@ -47,6 +48,14 @@ def _test(self, meth, *, args=[URL], kw={}, options,
arguments):
47    48                popen_args.pop(popen_args.index(option))
48    49                self.assertEqual(popen_args, arguments)
49    50
51    + def test_reject_dash_prefixes(self):
52    +     browser = self.browser_class(name=CMD_NAME)
53    +     with self.assertRaisesRegex(
54    +         ValueError,
55    +         r"^Invalid URL \(leading dash disallowed\): '--key=val http.*'$"
56    +     ):
57    +         browser.open(f"--key=val {URL}")
58    +
50    59
51    60     class GenericBrowserCommandTest(CommandTestMixin, unittest.TestCase):
52    61
@@ -57,11 +66,6 @@ def test_open(self):
57    66                options=[],
58    67                arguments=[URL])
59    68
60    - def test_reject_dash_prefixes(self):
61    -     browser = self.browser_class(name=CMD_NAME)
62    -     with self.assertRaises(ValueError):
63    -         browser.open(f"--key=val {URL}")
64    -
65    69
66    70     class BackgroundBrowserCommandTest(CommandTestMixin, unittest.TestCase):
67    71
@@ -222,6 +226,73 @@ def test_open_new_tab(self):
arguments=['openURL({}, new-tab)'.format(URL)]
222   226
223   227
224   228
229   + class MockPopenPipe:
230   +     def __init__(self, cmd, mode):

```

```
231 +         self.cmd = cmd
232 +         self.mode = mode
233 +         self.pipe = io.StringIO()
234 +         self._closed = False
235 +
236 +     def write(self, buf):
237 +         self.pipe.write(buf)
238 +
239 +     def close(self):
240 +         self._closed = True
241 +         return None
242 +
243 +
244 + @unittest.skipUnless(sys.platform == "darwin", "macOS specific test")
245 + class MacOSXOSAScriptTest(unittest.TestCase):
246 +     def setUp(self):
247 +         # Ensure that 'BROWSER' is not set to 'open' or something else.
248 +         # See: https://github.com/python/cpython/issues/131254.
249 +         ctx = os_helper.EnvironmentVarGuard()
250 +         env = ctx.__enter__()
251 +         self.addCleanup(ctx.__exit__)
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 +

```

```

225 296     class BrowserRegistrationTest(unittest.TestCase):
226 297
227 298         def setUp(self):

```



Lib/webbrowser.py



```

@@ -158,7 +158,7 @@ def open_new_tab(self, url):
158 158     def _check_url(url):
159 159         """Ensures that the URL is safe to pass to subprocesses as a
        parameter"""
160 160         if url and url.lstrip().startswith("-"):
161 -             raise ValueError(f"Invalid URL: {url}")
161 +             raise ValueError(f"Invalid URL (leading dash disallowed): {url!r}")
162 162
163 163

```

164 164 `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