

#### **EADME.md**

# 'yppeteer

# 'yppeteer has moved to pyppeteer/pyppeteer

```
pi v1.0.2 python 3.7 | 3.8 | 3.9 | 3.10 docs latest build unknown 9 build passing 200 possing 200 poss
```

nofficial Python port of puppeteer JavaScript (headless) chrome/chromium browser automation library.

- Free software: MIT license (including the work distributed under the Apache 2.0 license)
- Documentation: https://miyakogi.github.io/pyppeteer

#### nstallation

rppeteer requires python 3.6+. (experimentally supports python 3.5) stall by pip from PyPI:

python3 -m pip install pyppeteer

r install latest version from github:

python3 -m pip install -U git+https://github.com/miyakogi/pyppeteer.git@dev

#### sage

**Note**: When you run pyppeteer first time, it downloads a recent version of Chromium (~100MB). If you don't prefer this behavior, run pyppeteer-install command before running scripts which uses pyppeteer.

**cample**: open web page and take a screenshot.

```
import asyncio
from pyppeteer import launch
async def main():
  browser = await launch()
  page = await browser.newPage()
  await page.goto('http://example.com')
  await page.screenshot({'path': 'example.png'})
  await browser.close()
asyncio.get_event_loop().run_until_complete(main())
cample: evaluate script on the page.
import asyncio
from pyppeteer import launch
async def main():
  browser = await launch()
  page = await browser.newPage()
  await page.goto('http://example.com')
  await page.screenshot({'path': 'example.png'})
  dimensions = await page.evaluate("'() => {
       width: document.documentElement.clientWidth,
       height: document.documentElement.clientHeight,
       deviceScaleFactor: window.devicePixelRatio,
    }
  }''')
  print(dimensions)
  # >>> {'width': 800, 'height': 600, 'deviceScaleFactor': 1}
  await browser.close()
asyncio.get_event_loop().run_until_complete(main())
```

ppeteer has almost same API as puppeteer. More APIs are listed in the document.

uppeteer's document and troubleshooting are also useful for pyppeteer users.

## ifferences between puppeteer and pyppeteer

rppeteer is to be as similar as puppeteer, but some differences between python and JavaScript make it difficult.

nese are differences between puppeteer and pyppeteer.

#### eyword arguments for options

appeteer uses object (dictionary in python) for passing options to functions/methods. Pyppeteer accepts both ctionary and keyword arguments for options.

ictionary style option (similar to puppeteer):

browser = await launch({'headless': True})

eyword argument style option (more pythonic, isn't it?):

browser = await launch(headless=True)

#### lement selector method name (\$ -> querySelector)

python, \$ is not usable for method name. So pyppeteer uses age.querySelector() / Page.querySelectorAll() / Page.xpath() instead of Page.\$() / Page.\$\$() / Page.\$x() . Pyppeteer also has northands for these methods, Page.J() , Page.JJ() , and Page.Jx() .

#### rguments of Page.evaluate() and Page.querySelectorEval()

Ippeteer's version of evaluate() takes JavaScript raw function or string of JavaScript expression, but pyppeteer kes string of JavaScript. JavaScript strings can be function or expression. Pyppeteer tries to automatically detect e string is function or expression, but sometimes it fails. If expression string is treated as function and error is ised, add force\_expr=True option, which force pyppeteer to treat the string as expression.

cample to get page content:

content = await page.evaluate('document.body.textContent', force\_expr=True)

cample to get element's inner text:

element = await page.querySelector('h1')
title = await page.evaluate('(element) => element.textContent', element)

#### uture Plan

- .. Catch up development of puppeteer
  - Not intend to add original API which puppeteer does not have

#### redits

nis package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

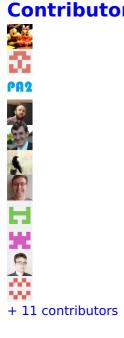
#### **Releases**

24 tags

### **Packages**

No packages published

## **Contributors 22**



## Languages

• Python 96.2% • HTML 2.8% • Other 1.0%

© 2023 GitHub, Inc.

Terms

Privacy

Security

**Status** 

Docs

Contact GitHub

Pricing

API

**Training** 

Blog

**About**