

# RoboBrowser: Your friendly neighborhood web scraper

pypi package 0.5.3
build passing
coverage 92%

Fork me on GitHub

Homepage: <http://robobrowser.readthedocs.org/>

RoboBrowser is a simple, Pythonic library for browsing the web without a standalone web browser. RoboBrowser can fetch a page, click on links and buttons, and fill out and submit forms. If you need to interact with web services that don't have APIs, RoboBrowser can help.

```
import re
from robobrowser import RoboBrowser

# Browse to Rap Genius
browser = RoboBrowser(history=True)
browser.open('http://rapgenius.com/')

# Search for Queen
form = browser.get_form(action='/search')
form # <RoboForm q=>
form['q'].value = 'queen'
browser.submit_form(form)

# Look up the first song
songs = browser.select('.song_name')
browser.follow_link(songs[0])
lyrics = browser.select('.lyrics')
lyrics[0].text # \n[Intro]\nIs this the real life...

# Back to results page
browser.back()

# Look up my favorite song
browser.follow_link('death on two legs')

# Can also search HTML using regex patterns
lyrics = browser.find(class_=re.compile(r'\blyrics\b'))
lyrics.text # \n[Verse 1]\nYou suck my blood like a leech...
```

RoboBrowser combines the best of two excellent Python libraries: [Requests](#) and [BeautifulSoup](#). RoboBrowser represents browser sessions using Requests and HTML responses using BeautifulSoup, transparently exposing methods of both libraries:

```
import re
from robobrowser import RoboBrowser

browser = RoboBrowser(user_agent='a python robot')
browser.open('https://github.com/')

# Inspect the browser session
browser.session.cookies['_gh_sess'] # BAh7Bzo...
browser.session.headers['User-Agent'] # a python robot

# Search the parsed HTML
browser.select('div.teaser-icon') # [<div class="teaser-icon">
# <span class="mega-octicon octicon-checklist"></span>
# </div>,
# ...
browser.find(class_=re.compile(r'column', re.I)) # <div class="one-third column">
# <div class="teaser-icon">
# <span class="mega-octicon octicon-checklist">
# ...
```

RoboBrowser also includes tools for working with forms, inspired by [WebTest](#) and [Mechanize](#).

 v: latest ▾

```
from robobrowser import RoboBrowser

browser = RoboBrowser()
browser.open('http://twitter.com')

# Get the signup form
signup_form = browser.get_form(class_='signup')
signup_form      # <RoboForm user[name]=, user[email]=, ...

# Inspect its values
signup_form['authenticity_token'].value      # 6d03597 ...

# Fill it out
signup_form['user[name]'].value = 'python-robot'
signup_form['user[user_password]'].value = 'secret'

# Serialize it to JSON
signup_form.serialize()      # {'data': {'authenticity_token': '6d03597...',
#   'context': '',
#   'user[email]': '',
#   'user[name]': 'python-robot',
#   'user[user_password]': ''}}
```

```
# And submit
browser.submit_form(signup_form)
```

## Requirements

- Python >= 2.6 or >= 3.3

## License

MIT licensed. See the bundled [LICENSE](#) file for more details.