ScrapeHero a data company

Tutorial: Web Scraping Hotel Prices using Selenium and Python

Everyone would like to pay the least amount of money for the best hotel room – simple isn't it?

In this tutorial we will show you how to make your own little tracking web scraper for scraping Hotels.com so that you can snag the room you want at the lowest rate. All you need to do is change the City, the Check In and Check Out date and run it on a schedule.

Feel free to copy and modify it to your needs – that is the best way to learn ! You can download the code directly from here .

Pre-Requisites

Below are the frameworks used:

1. **Selenium Web Driver** – a framework that is widely using for automating routines in Web Browsers for scraping and testing purposes. Selenium receives commands such as – load a page, click a location or button etc from the scraper. We can also read what is being rendered in the browser. Learn how to install Selenium here – http://www.seleniumhq.org/download and install the Python Bindings for Selenium here – http://selenium-python.readthedocs.io/installation.html
2. **LXML** for extracting data from the page source HTML. LXML lets you parse HTML / XML tree structure

Web Scraping. Learn how to install that here – http://lxml.de/installation.html

3. **Python 2.7** available here (https://www.python.org/downloads/)

The Code

```
#!/usr/bin/env python
from re import findall,sub
from lxml import html
from time import sleep
from selenium import webdriver
from pprint import pprint
from xvfbwrapper import Xvfb

def parse(url):
    searchKey = "Las Vegas" # Change this to your
    checkInDate = '27/08/2016' #Format %d/%m/%Y
    checkOutDate = '29/08/2016' #Format %d/%m/%Y
    response = webdriver.Firefox()
    response.get(url)
    searchKeyElement = response.find_elements_by_
    checkInElement = response.find_elements_by_xp
    checkOutElement = response.find_elements_by_x
    submitButton = response.find_elements_by_xpat
    if searchKeyElement and checkInElement and ch
        searchKeyElement[0].send_keys(searchKey)
        checkInElement[0].clear()
        checkInElement[0].send_keys(checkInDate)
        checkOutElement[0].clear()
        checkOutElement[0].send_keys(checkOutDate
        randomClick = response.find_elements_by_x
        if randomClick:
            randomClick[0].click()
        submitButton[0].click()
        sleep(15)
```

```
            dropDownButton[0].click()
            priceLowtoHigh = response.find_elemen
            if priceLowtoHigh:
                priceLowtoHigh[0].click()
                sleep(10)

    parser = html.fromstring(response.page_source
    hotels = parser.xpath('//div[@class="hotel-wr
    for hotel in hotels[:5]: #Replace 5 with 1 to
        hotelName = hotel.xpath('.//h3/a')
        hotelName = hotelName[0].text_content() i
        price = hotel.xpath('.//div[@class="price
        price = price[0].text_content().replace('
        if price==None:
            price = hotel.xpath('.//div[@class="p
            price = price[0].text_content().repla
        price = findall('([\d\.]+)',price) if pri
        price = price[0] if price else None
        rating = hotel.xpath('.//div[@class="star
        rating = rating[0] if rating else None
        address = hotel.xpath('.//span[contains(@
        address = "".join([x.text_content() for x
        locality = hotel.xpath('.//span[contains(
        locality = locality[0].text_content().rep
        region = hotel.xpath('.//span[contains(@c
        region = region[0].text_content().replace
        postalCode = hotel.xpath('.//span[contain
        postalCode = postalCode[0].text_content()
        countryName = hotel.xpath('.//span[contai
        countryName = countryName[0].text_content

        item = {
                    "hotelName":hotelName,
                    "price":price,
                    "rating":rating,
                    "address":address,
                    "locality":locality,
                    "region":region,
                    "postalCode":postalCode,
                    "countryName":countryName,
        }
        pprint(item)
if __name__ == '__main__':
    parse('http://www.hotels.com')
    vdisplay.stop()
```

hotels_scraper.py hosted with ♡ by GitHub view raw

Open your favorite text editor and modify the line below with – City Name, Check In Date, Check Out Date and you'll get the top 5 cheapest hotels to stay.

```
def parse(url):
    searchKey = "Las Vegas" # Change
    checkInDate = '27/08/2016' #Forr
    checkOutDate = '29/08/2016' #For
    response = webdriver.Firefox()
```

And run this from the command prompt like this (if you name the file hotels_scraper.py)

```
python hotels_scraper.py
```

This should print the results in the command prompt as a python dictionary.

For Las Vegas the output looks like this

```
[
  {
    "hotelName": "Antonio Hotel",
    "address": "229 N Soto Street",
    "postalCode": "90033",
    "price": "$241",
    "locality": "Los Angeles",
    "region": "CA"
  },
  {
    "countryName": "United States",
    "rating": 2.0,
    "hotelName": "Comet Motel",
    "address": "10808 Avalon Bouleva
    "postalCode": "90061",
    "price": "$124",
    "locality": "Los Angeles",
    "region": "CA"
  },
  {
    "countryName": "United States",
    "rating": 2.5,
    "hotelName": "Arthur Emery",
    "address": "907 W 17th Street",
    "postalCode": "90015",
    "price": "$298",
    "locality": "Los Angeles",
    "region": "CA"
  },
  {
    "countryName": "United States",
    "rating": 2.5,
    "hotelName": "LA Ramona Motel",
    "address": "3211 W. Jefferson Bl
    "postalCode": "90018",
    "price": "$125",
    "locality": "Los Angeles",
    "region": "CA"
  },
  {
      "hotelName": "Central Inn Motel'
      "address": "954 E 88th St",
      "postalCode": "90002",
      "price": "$185",
      "locality": "Los Angeles",
      "region": "CA"
    }
  ]
```

You can modify this code a bit and connect it to chatbots in Slack, Facebook or email etc to find the cheapest room rates.

The code above is good for small-scale scraping for fun. If you want to scrape some hotel pricing details from thousands of pages you should read Scalable do-it-yourself scraping – How to build and run scrapers on a large scale

If you need a faster option you can use Puppeteer, a Node.js library that controls headless Chrome or Chromium.

Web Scraping Tutorial using a Headless Browser: How to Build a Web Scraper using Puppeteer and Node.js

ScrapeHero a data company

Need some help with scraping data?

Turn the Internet into meaningful, structured and usable data

Your Name
email@company.com
Get in touch with us

Please DO NOT contact us for any help with our Tutorials and Code using this form or by calling us, instead please add a comment to the bottom of the tutorial page for help

Contact Sales

Disclaimer: *Any code provided in our tutorials is for illustration and learning purposes only. We are not responsible for how it is used and assume no liability for* *does not imply that we encourage scraping or scrape the websites referenced in the code and accompanying tutorial. The tutorials only help illustrate the technique of programming web scrapers for popular internet websites. We are not obligated to provide any support for the code, however, if you add your questions in the comments section, we may periodically address them.*

Continue Reading ..

How to scrape TripAdvisor.com for Hotel Data, Pricing and Reviews using Python

Tripadvisor.com has tons of information regarding hotels from all over the world, which can be used for monitoring prices of hotels in a locality, competitive pricing, analyzing how the price changes with each season, understand ratings...

How to scrape Tripadvisor.com Hotel Details using Python and LXML

such as hotel name, address, ranking and more from Tripadvisor using Python and LXML.

How to scrape Amazon Reviews using Python

This tutorial is a follow-up to Tutorial: How To Scrape Amazon Product Details and Pricing using Python, by extending the Amazon price data to also cover product reviews. The scope of this tutorial is limited...

Posted in: Travel Data Gathering Tutorials,Web Scraping Tutorials
Published On: July 28, 2016

Responses

Stan L August 16, 2018

Hi SHero, thank you for the tutorial blog! very clear and concise instructions for scraping data across various types of websites. One challenge I am facing is scraping data from a website such as Forbes. " https://www.forbes.com/top-wealth-managers " It looks like some scripts get actioned upon the first attempt to the website and pops-up a Forbe's Quote Window. I am very curious to know how we can bypass this window without using Selenium to action the "Continue to Site page or send words in a text box on the webpage, while staying with the convenience method of lxml (that is you don't need a browser or headless browser to run the scraping script to do the task mentioned above). Thanks!

Reply

ScrapeHero August 16, 2018

Thanks for the comment.
You really have to poke through the whole request and responses and cookies as you navigate the site.
Something in that exchange signals the site to show or not show the page.
Set that value most likely in the cookie and that might help.

Reply

Comments or Questions?

Enter your comment here...

ScrapeHero a data company

Automation
Sales Intelligence
Research and Journalism

Training Data for Machine Learning
Distribution Channel Monitoring
Human Capital Management
Real Estate and Housing Data

Sales Leads
Social Media Data
Web Scraping Tutorials
Insights