Slenium, Python and Chromium

Docker file

The following is the docker file I use for automated web testing using Selenium, Chromium and Python. It's the smallest I could get so far (565MB). However, if you know how to make it even smaller, please post in the comments, I would be very interested.

```
FROM python:3.8-slim-buster

COPY every5min.py /root/every5min.py
COPY requirements.txt /root/requirements.txt

# Ensure any piped commands exit on error
SHELL ["/bin/bash", "-o", "pipefail", "-c"]

ENV PATH /usr/local/bin:$PATH
ENV LANG C.UTF-8

# Install the latest versions of Google Chrome and Chromedriver
# Patches Chrome launch script to disable /dev/shm and sandbox for use in docker
RUN export DEBIAN_FRONTEND=noninteractive && apt-get update \
  && apt-get install --no-install-recommends --no-install-suggests --assume-yes \
    curl \
    unzip \
    gnupg \
    unzip \
    wget \
    locales \
    cron \
    vim \
    dos2unix \
  && CHROME_DOWNLOAD_URL='https://dl.google.com/linux' \
  && curl -sL "${CHROME_DOWNLOAD_URL}/linux_signing_key.pub" | apt-key add - \
  && curl -sL "${CHROME_DOWNLOAD_URL}/direct/google-chrome-stable_current_amd64.deb" > /tmp/chrome.deb \
  # TODO find packages that cause fail before this and install first
  && (dpkg -i /tmp/chrome.deb || apt-get install --no-install-recommends --no-install-suggests --assume-yes --fix-broken) \
  && CHROMIUM_OPTIONS_FILE=/opt/google/chrome/google-chrome \
  && echo "$(cat ${CHROMIUM_OPTIONS_FILE}) ${CHROMIUM_FLAGS}" > "${CHROMIUM_OPTIONS_FILE}"  \
  && BASE_URL='https://chromedriver.storage.googleapis.com' \
  && VERSION=$(curl -sL "${BASE_URL}/LATEST_RELEASE") \
  && curl -sL "${BASE_URL}/${VERSION}/chromedriver_linux64.zip" -o /tmp/driver.zip \
  && unzip /tmp/driver.zip \
  && chmod 0755 chromedriver \
  && mv chromedriver /root/chromedriver \
  && apt-get purge -y \
    curl \
    unzip \
    gnupg \
  && apt-get purge -y --auto-remove -o APT::AutoRemove::RecommendsImportant=false \
  && rm -rf /tmp/* /usr/share/doc/* /var/cache/* /var/lib/apt/lists/* /var/tmp/* \
  && mkdir /root/log \
  && chmod +x /root/unit-test-a.py \
  && pip install --user -r /root/requirements.txt \
  && dos2unix /root/every5min.py

RUN echo '*/5 * * * *    /root/every5min.py' > /etc/crontabs/root

CMD [ "cron", "-l", "2", "-f" ]
```

This will create an image that will run cron, and every 5 minutes execute the script `/root/every5min.py`.

Python arguments

Just as a reminder, you need to set up the following variables in your chromedriver to have this work without a window manager.

```
chrome_options.add_argument('--disable-dev-shm-usage')
chrome_options.add_argument('--headless')
chrome_options.add_argument('--disable-gpu')
```

The whole function that returns my browser object looks like this:

```
def open_browser():
    fake_user_agent = Faker()
    chrome_options = webdriver.ChromeOptions()
    resolution = random.choice(resolutions_list)
    # Disabling warning that browser is controlled by software.
    chrome_options.add_experimental_option("excludeSwitches", ['enable-automation'])
    chrome_options.add_argument('--disable-infobars')
    chrome_options.add_argument('--window-size=' + resolution)
    chrome_options.add_argument('--no-sandbox')
    chrome_options.add_argument('--disable-dev-shm-usage')
    # Uncomment following two to enable or disable headless mode
    chrome_options.add_argument('--headless')
    chrome_options.add_argument('--disable-gpu')
    chrome_options.add_argument('--disable-extensions')
    chrome_options.add_argument('--no-first-run')
    chrome_options.add_argument('--ignore-certificate-errors')
    chrome_options.add_argument('--disable-client-side-phishing-detection')
    # chrome_options.add_argument('--incognito')
    chrome_options.add_argument('--allow-running-insecure-content')
    chrome_options.add_argument('--disable-web-security')
    # chrome_options.add_argument('--start-maximized')
    chrome_options.add_argument('--lang=' + random.choice(language_list))
    chrome_options.add_argument('--user-agent=' + fake_user_agent.user_agent())
    chrome_browser = webdriver.Chrome(chrome_driver_loc, options=chrome_options)
    chrome_browser.set_page_load_timeout(120)
    resolution = resolution.split(',')
    chrome_browser.set_window_size(resolution[0], resolution[1])
    chrome_browser.set_window_position(0, 0)
    return chrome_browser
```

There are some variables taken from other places like resolution, user-agent and language... but that's not part of this article.

That's it for now, was it helpful to you? Now get some coffee and maybe get me one too 🙂

Liked it ? Buy me a drink :)

Last update: May 26, 2022

Comments