
Web Profiler Documentation

Release 1.0

David Naylor

January 15, 2017

1	Loaders	1
2	Results	7
3	Indices and tables	11

Loaders

```
class webloader.loader.Loader(outdir='', num_trials=1, http2=False, timeout=30, disable_local_cache=True, disable_network_cache=False, full_page=True, user_agent=None, headless=True, restart_on_fail=False, restart_each_time=False, proxy=None, save_har=False, save_screenshot=False, save_content='never', retries_per_trial=0, stdout_filename=None, check_protocol_availability=True, save_packet_capture=False, disable_quic=False, disable_spdy=False, log_ssl_keys=False, ignore_certificate_errors=False, delay_after_onload=0, delay_first_trial_only=False, primer_load_first=False, configs=[{'tag': 'default', 'settings': {}}])
```

Superclass for URL loader. Subclasses implement actual page load functionality (e.g., using Chrome, PhantomJS, etc.).

Parameters

- **outdir** – directory for HAR files, screenshots, etc.
- **num_trials** – number of times to load each URL
- **http2** – use HTTP 2 (not all subclasses support this)
- **timeout** – timeout in seconds
- **disable_local_cache** – disable the local browser cache (RAM and disk)
- **disable_network_cache** – send “Cache-Control: max-age=0” header
- **full_page** – load page’s subresources and render; if False, only the object is fetched
- **user_agent** – use custom user agent; if None, use browser’s default
- **headless** – don’t use GUI (if there normally is one – e.g., browsers)
- **restart_on_fail** – if a load fails, set up the loader again (e.g., reboot chrome)
- **restart_each_time** – tear down and set up the loader before each page load (e.g., reboot chrome to close open connections)
- **save_har** – save a HAR file to the output directory
- **save_screenshot** – save a screenshot to the output directory
- **save_content** – save HTTP message bodies (options: ‘always’, ‘first’, ‘never’)
- **retries_per_trial** – if a trial fails, retry this many times (beyond first)

- **stdout_filename** – if the loader launches other procs (e.g., browser), send their stdout and stderr to this file. If None, use parent proc’s stdout and stderr.
- **check_protocol_availability** – before loading the page, check to see if the specified protocol (HTTP or HTTPS) is supported. (otherwise, the loader might silently fall back to a different protocol.)
- **save_packet_capture** – save a pcap trace for each load (separate files)
- **disable_quic** – disable use of the QUIC transport protocol
- **disable_spdy** – disable use of SPDY/HTTP2
- **log_ssl_keys** – instruct browser to save SSL session keys (by setting SSLKEYLOG-FILE environment variable)
- **ignore_certificate_errors** – continue loading page even if certificate check fails
- **delay_after_onload** – continue recording objects after onLoad fires (ms)
- **delay_first_trial_only** – if fetching a URL multiple times, only delay after on-Load on the first trial. (The delay is useful to count how many objects are loaded after onLoad, and this is less likely to change from trial to trial than load time.)
- **primer_load_first** – load the page once before beginning normal trials (e.g., to prime DNS caches)
- **configs** – TODO: document

load_pages (*urls*)

Load each URL in *urls* *num_trials* times and collect stats.

Parameters *urls* – list of URLs to load

load_results

A dict mapping URLs to a list of LoadResult.

num_restarts

Number of times the loader was restarted (e.g., rebooted browser process) due to failures if restart_on_fail is True.

page_results

A dict mapping URLs to a PageResult.

urls

A cumulative list of the URLs this instance has loaded in the order they were loaded. Each trial is listed separately.

class `webloader.phantomjs_loader.PantomJSLoader` (**kwargs)

Subclass of `Loader` that loads pages using PhantomJS.

Note: The `PantomJSLoader` currently does not support HTTP2.

Note: The `PantomJSLoader` currently does not support local caching.

Note: The `PantomJSLoader` currently does not support disabling network caching.

Note: The `PhantomJSLoader` currently does not support single-object loading (i.e., it always loads the full page).

Note: The `PhantomJSLoader` currently does not support saving content.

class `webloader.chrome_loader.ChromeLoader` (***kwargs*)
Subclass of `Loader` that loads pages using Chrome.

Note: The `ChromeLoader` currently does not time page load.

Note: The `ChromeLoader` currently does not save screenshots.

Note: The `ChromeLoader` currently does not support single-object loading (i.e., it always loads the full page).

Note: The `ChromeLoader` currently does not support saving screenshots.

class `webloader.firefox_loader.FirefoxLoader` (*selenium=True, **kwargs*)
Subclass of `Loader` that loads pages using Firefox.

Note: The `FirefoxLoader` currently does not extract HARs.

Note: The `FirefoxLoader` currently does not save screenshots.

Note: The `FirefoxLoader` currently does not support single-object loading (i.e., it always loads the full page).

Note: The `FirefoxLoader` currently does not support disabling network caches.

Note: The `FirefoxLoader` currently does not support saving screenshots.

Note: The `FirefoxLoader` currently does not support saving content.

class `webloader.pythonrequests_loader.PythonRequestsLoader` (***kwargs*)
Subclass of `Loader` that loads pages using Python requests.

Note: The `PythonRequestsLoader` currently does not support HTTP2.

Note: The `PythonRequestsLoader` currently does not support local caching.

Note: The `PythonRequestsLoader` currently does not support disabling network caching.

Note: The `PythonRequestsLoader` currently does not support full page loading (i.e., fetching a page's subresources).

Note: The `PythonRequestsLoader` currently does not support saving HARs.

Note: The `PythonRequestsLoader` currently does not support saving screenshots.

Note: The `PythonRequestsLoader` currently does not support saving content.

class `webloader.curl_loader.CurlLoader` (***kwargs*)
Subclass of `Loader` that loads pages using curl.

Note: The `CurlLoader` currently does not support HTTP2.

Note: The `CurlLoader` currently does not support caching.

Note: The `CurlLoader` currently does not support full page loading (i.e., fetching a page's subresources).

Note: The `CurlLoader` currently does not support saving HARs.

Note: The `CurlLoader` currently does not support saving screenshots.

Note: The `CurlLoader` currently does not support saving content.

class `webloader.nodejs_loader.NodeJsLoader` (***kwargs*)
Subclass of `Loader` that loads pages using NODE.JS.

Note: The `NodeJsLoader` currently does not support caching.

Note: The `NodeJsLoader` currently does not support full page loading (i.e., fetching a page's subresources).

Note: The `NodeJsLoader` currently does not support disabling network caches.

Note: The `NodeJsLoader` currently does not support saving HARs.

Note: The `NodeJsLoader` currently does not support saving screenshots.

Note: The `NodeJsLoader` currently does not support saving content.

class `webloader.tcp_loader.TCPLoader` (***kwargs*)

Subclass of `Loader` that loads pages using custom executable so we can change TCP settings.

Note: The `TCPLoader` currently does not support HTTP2.

Note: The `TCPLoader` currently does not support local caching.

Note: The `TCPLoader` currently does not support disabling network caching.

Note: The `TCPLoader` currently does not support single-object loading (i.e., it always loads the full page).

Note: The `TCPLoader` currently does not support saving content.

Results

```
class webloader.loader.LoadResult (status,      url,      final_url=None,      time=None,
                                   size=None,    har=None,    img=None,      raw=None,
                                   server=None,    tcp_fast_open_supported=False,
                                   tls_false_start_supported=False,
                                   tls_session_resumption_supported=False)
```

Status and stats for a single URL load (i.e., one trial).

Parameters

- **status** – The status of the page load.
- **url** – The original URL.
- **final_url** – The final URL (maybe be different if we were redirected).
- **time** – The page load time (in seconds).
- **size** – Size of object if loading a single object; total size if loading a full page.
- **har** – Path to the HAR file.
- **img** – Path to a screenshot of the loaded page.
- **tcp_fast_open_supported** – True if TCP fast open was used successfully; False otherwise or unknown

FAILURE_NO_200 = 'FAILURE_NO_200'

HTTP status code was not 200

FAILURE_TIMEOUT = 'FAILURE_TIMEOUT'

Page load timed out

FAILURE_UNKNOWN = 'FAILURE_UNKNOWN'

Unkown failure occurred

FAILURE_UNSET = 'FAILURE_UNSET'

Status has not been set

SUCCESS = 'SUCCESS'

Page load was successful

final_url

The final URL (could be different if we were redirected).

har_path

Path to the HAR captured during this page load.

image_path

Path to a screenshot of the loaded page.

raw

Raw output from the underlying command.

server

Web server software name.

size

???

status

The status of this page load.

tcp_fast_open_supported

Bool indicating whether or not TCP fast open succeeded for this connection.

time

The page load time in seconds.

tls_false_start_supported

Bool indicating whether or not TLS false start succeeded for this connection.

tls_session_resumption_supported

Bool indicating whether or not TLS session resumption succeeded for this connection.

url

The original URL requested.

class `webloader.loader.PageResult` (*url, status=None, load_results=None*)

Status and stats for one URL (all trials).

Parameters

- **url** – The original URL.
- **status** – The overall status of all trials.
- **load_results** – List of individual `LoadResult` objects

FAILURE_NOT_ACCESSIBLE = 'FAILURE_NOT_ACCESSIBLE'

The page could not be loaded with the specified protocol

FAILURE_UNKNOWN = 'FAILURE_UNKNOWN'

An unknown failure occurred

FAILURE_UNSET = 'FAILURE_UNSET'

Status has not been set

PARTIAL_SUCCESS = 'PARTIAL_SUCCESS'

some trials were successful

SUCCESS = 'SUCCESS'

All trials were successful

load_statuses

A list of statuses from individual trials.

mean_time

Mean load time across all trials.

median_time

Median load time across all trials.

server

Web server software name.

sizes

A list of the page sizes from individual trials.

status

The overall status across all trials.

stddev_time

Standard deviation of load time across all trials.

tcp_fast_open_support_statuses

A list of bools indicating whether or not TCP fast open succeeded for each load.

times

A list of the load times from individual trials.

tls_false_start_support_statuses

A list of bools indicating whether or not TLS false start succeeded for each load.

tls_session_resumption_support_statuses

A list of bools indicating whether or not TLS session resumption succeeded for each load.

url

The URL.

Indices and tables

- `genindex`
- `modindex`
- `search`

C

ChromeLoader (class in webloader.chrome_loader), 3

CurlLoader (class in webloader.curl_loader), 4

F

FAILURE_NO_200 (webloader.loader.LoadResult attribute), 7

FAILURE_NOT_ACCESSIBLE (webloader.loader.PageResult attribute), 8

FAILURE_TIMEOUT (webloader.loader.LoadResult attribute), 7

FAILURE_UNKNOWN (webloader.loader.LoadResult attribute), 7

FAILURE_UNKNOWN (webloader.loader.PageResult attribute), 8

FAILURE_UNSET (webloader.loader.LoadResult attribute), 7

FAILURE_UNSET (webloader.loader.PageResult attribute), 8

final_url (webloader.loader.LoadResult attribute), 7

FirefoxLoader (class in webloader.firefox_loader), 3

H

har_path (webloader.loader.LoadResult attribute), 7

I

image_path (webloader.loader.LoadResult attribute), 7

L

load_pages() (webloader.loader.Loader method), 2

load_results (webloader.loader.Loader attribute), 2

load_statuses (webloader.loader.PageResult attribute), 8

Loader (class in webloader.loader), 1

LoadResult (class in webloader.loader), 7

M

mean_time (webloader.loader.PageResult attribute), 8

median_time (webloader.loader.PageResult attribute), 8

N

NodeJsLoader (class in webloader.nodejs_loader), 4

num_restarts (webloader.loader.Loader attribute), 2

P

page_results (webloader.loader.Loader attribute), 2

PageResult (class in webloader.loader), 8

PARTIAL_SUCCESS (webloader.loader.PageResult attribute), 8

PhantomJSLoader (class in webloader.phantomjs_loader), 2

PythonRequestsLoader (class in webloader.pythonrequests_loader), 3

R

raw (webloader.loader.LoadResult attribute), 8

S

server (webloader.loader.LoadResult attribute), 8

server (webloader.loader.PageResult attribute), 8

size (webloader.loader.LoadResult attribute), 8

sizes (webloader.loader.PageResult attribute), 9

status (webloader.loader.LoadResult attribute), 8

status (webloader.loader.PageResult attribute), 9

stddev_time (webloader.loader.PageResult attribute), 9

SUCCESS (webloader.loader.LoadResult attribute), 7

SUCCESS (webloader.loader.PageResult attribute), 8

T

tcp_fast_open_support_statuses (webloader.loader.PageResult attribute), 9

tcp_fast_open_supported (webloader.loader.LoadResult attribute), 8

TCPLoader (class in webloader.tcp_loader), 5

time (webloader.loader.LoadResult attribute), 8

times (webloader.loader.PageResult attribute), 9

tls_false_start_support_statuses (webloader.loader.PageResult attribute), 9

tls_false_start_supported (webloader.loader.LoadResult attribute), 8

tls_session_resumption_support_statuses (webloader.loader.PageResult attribute), 9

tls_session_resumption_supported (webloader.loader.LoadResult attribute), 8

U

url (webloader.loader.LoadResult attribute), 8

url (webloader.loader.PageResult attribute), 9

urls (webloader.loader.Loader attribute), 2