
HarProfiler

Release 0.1.0

Sep 27, 2017

Contents

1	About	3
2	Table Of Contents	5
2.1	harprofiler	5
2.2	haruploader	6

Project Home <https://github.com/edx/harprofiler>

License Apache License, Version 2.0

Author Copyright (c) 2014, edX TestEng

CHAPTER 1

About

harprofiler is a python utility used for profiling web pageloads. It loads web pages and saves JSON files in HAR (HTTP Archive) format. By default, it will load a page once uncached, and then again with it cached in the browser. A HAR file for each pageload is saved locally, and optionally uploaded to a HARStorage server instance. HAR files contain a log of HTTP client/server conversation and can be used for analysis of page load performance.

harprofiler

Installation

Install via script (Ubuntu/Debian only)

Best method for getting up and running quickly. This will install the requirements and activate a virtualenv:

```
$ source dev-bootstrap.sh
```

Manual install

System Requirements

- Java JRE
- Firefox browser
- Python & Virtualenv
- [Browsermob proxy](#)

Python Requirements

create a virtualenv and install Python dependencies:

```
$ virtualenv env
$ source env/bin/activate
$ pip install -r requirements.txt
```

Configuration

harprofiler uses a yaml configuration file named *config.yaml*.

example config:

```
browsermob_dir: ./browsermob-proxy-2.0-beta-9
har_dir: ./hars
harstorage_url: http://localhost:5000
label_prefix: my-prefix-
run_cached: true
urls:
- https://www.edx.org
- https://www.edx.org/course-search
virtual_display: true
virtual_display_size_x: 1024
virtual_display_size_y: 768
```

Usage

run pageload profiler:

```
$ python harprofiler.py
```

- results are saved in timestamped .har files

haruploader

The *haruploader* module is used for sending har files to a *harstorage* instance.

harstorage references:

- [On google code](#)
- [Original Github repository](#)

Run haruploader as standalone script

- **Args:** Path to HAR file or directory containing har files to be uploaded.
- **Options:** `--url`: URL of *harstorage* instance (default: `'http://localhost:5000'`)
- **Example:** `python haruploader.py /path/to/HAR/file.har --url http://127.0.0.1:8000`
- **For help text:** `python haruploader.py -h`

Run uploader as part of harprofiler

Make sure that *harstorage_url* is set in the config file, and harprofiler will run the uploader after it creates the HARs. This will call the `upload_hars` method, using as args the `har_dir` and `harstorage_url` settings provided in the configuration file.

Error handling

- If the requests lib raises an exception, we will leave the file in the folder to be retried later. The error will still be logged though. These exceptions include the following:

```
requests.exceptions.ConnectionError
requests.exceptions.TooManyRedirects
requests.exceptions.Timeout
requests.exceptions.HTTPError
requests.exceptions.URLRequired
```

- If any other exception is raised while trying to upload the file, the file will be put in another folder, not to be retried. In this case, we assume the cause is a poorly formatted HAR file. The destination folder is titled `failed_uploads`, and will be automatically created as a subdirectory of the folder that the HAR file was originally located.
- If the file is successfully uploaded, it will be moved to a folder titled `completed_uploads`. Again, this will be automatically created as a subdirectory of the folder that the HAR file was originally located.