

Proton

Proton allows users to download a Wikipedia article as PDF. It supports both desktop and mobile-friendly prints.

Wikimedia Foundation project

Proton

Generate PDF version of any Wikipedia Article

Group: Readers
Start: December 2018

Contents

Technical details

- QueueSystem
- Renderer
- Additional features
- Known hacks

Development

- Running the tests

Technical documents

- Links for Proton developers

See also

Contact

Technical details

Proton is a simple service that generates the PDF using Chromium (<https://www.chromium.org/Home>) driven by Puppeteer (<https://github.com/GoogleChrome/puppeteer>) library. It consists of two components:

- Queue system that queues all requests (as PDF generation is both resource and time-intensive job)
- Renderer code which instructs Puppeteer to print requested page as PDF.

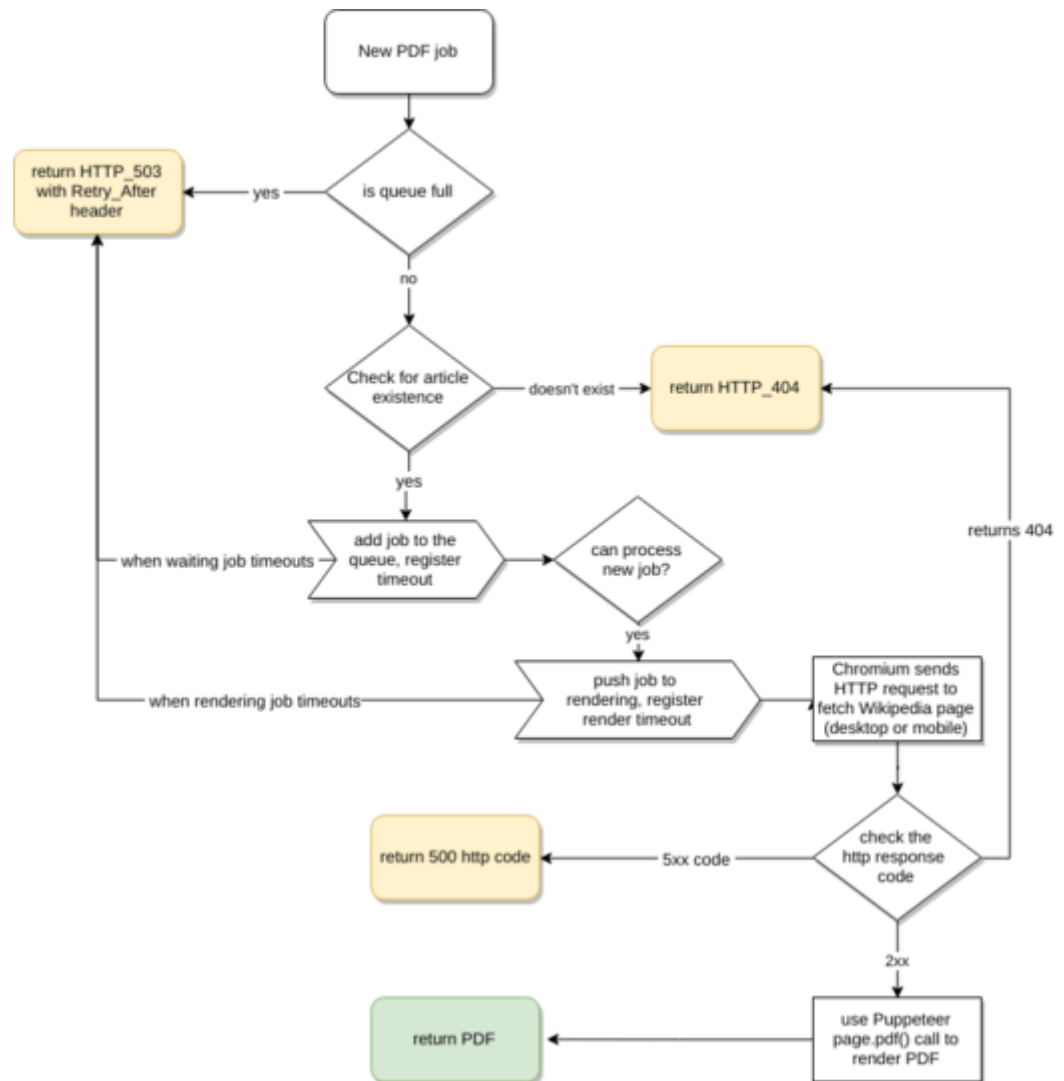
Proton is structured as a web service and is written in JavaScript, making use of Node.js. It is intended to provide beautiful and clean PDFs. On Wikimedia wikis, Proton will be proxied behind RESTBase. It uses the puppeteer-core (<https://github.com/GoogleChrome/puppeteer/blob/master/docs/api.md#puppeteer-vs-puppeteer-core>) library, chromium browser is not bundled with puppeteer-core and it has to be downloaded separately. The `PUPPETEER_EXECUTABLE_PATH` environment variable is used to point to chromium executable.

The best way to generate the Article PDF is to use browser built-in to PDF functionality. That method provides the best results and additionally allows us to reuse the existing print styles available for both Desktop and Mobile versions of Wikipedia. The system doesn't post-process the requested HTML. Articles are printed the same way as they appear in print preview in the user browser. The generated PDFs are very similar (if not identical) to what anyone can achieve by using **Print to PDF** on their Chrome browser. To get best results, Proton disables the JavaScript. It is done to disable all dynamic content transformations, like lazy-loaded images on Mobile pages.

Note: for some users, the PDF they get from browser print and the one they get from Proton service might differ a bit as fonts configuration on user system can have specific settings related to fonts hinting/kerning.

QueueSystem

The Queue system is the heart of Proton renderer. It handles the flow of each job through



waiting/processing/timeout logic. Each job in the queue can have two states - waiting and processing. The queue system not only allows a specific amount of jobs to run at the same time but it also handles job timeouts and job cancellation. Because of the queue complexity, we had to implement the solution that allows us to:

- limit the number of waiting jobs
- after a defined amount of seconds reject the waiting job
- limit the number of rendering jobs (as PDF rendering requires lots of resources)
- a safety net to reject rendering jobs that takes too much time
- to save resources, when the request is aborted queue will try to cancel the job, doesn't matter which state the aborted job is (processing/rendering).

The queue system is based on [Bluebird](http://bluebirdjs.com) (<http://bluebirdjs.com>) promises, and utilizes the cancellation (<http://bluebirdjs.com/docs/api/cancellation.html>) feature (about which see [#Known hacks](#) below).

Renderer

The Renderer is a simple facade to access `page.pdf()` method from puppeteer library. Renderer is responsible for setting proper chromium environment and browser viewport, requesting the Wikipedia page, calling the `page.pdf()` function. Plus it keeps an eye on the browser process. Each render starts new Chromium instance, and after successful render, the chromium process exits. To save resources, and keep

our system in good state Renderer asks Chromium to shut down and if because of any reason browser still keeps processing the request it will send the SIGKILL to browser process to make sure it doesn't use any more CPU nor the memory.

Additional features

When a job fails because the queue is full or job timeouts in any state the Proton service will return 503 Service Unavailable response with **Retry-After** header. The **Retry-After** header instruments load balancer to depool given Proton node so it can finish processing current jobs. System sets **Retry-After** header to `app.config.render_queue_timeout` configuration value. After that time all processing jobs should finish, and the system should be able to pick up new jobs.

Known hacks

Proton utilizes the BBPromise cancellation feature. Cancellation feature is disabled by default, to enable promise cancellation `BBPromise.config()` (<http://bluebirdjs.com/docs/api/promise.config.html>) has to be called with `cancellation:true` flag. The trick is that the BBPromise config has to be set before any promise is created. But because Proton uses the Service-runner, and Service-runner uses BBPromises for everything, even reading configuration files this wasn't easy to implement. The `cancellation` flag cannot be set in the Proton application, because the Proton code is executed after Service-runner initialization. It also couldn't be defined in config, as Service-runner uses promises when reading the config. In version 2.6.6 (<https://github.com/wikimedia/service-runner/pull/191>) of Service-runner introduces use of the `APP_ENABLE_CANCELABLE_PROMISES` environment variable, which has to be set to `truthy` value. If the environment variable is not set, Proton initialization will fail with error (<https://github.com/wikimedia/mediawiki-services-chromium-render/blob/6ba6069eb46e803f9249226dbca0222c34b3cac3/app.js#L134>).

Development

Development happens in the Proton service Git repository (<https://phabricator.wikimedia.org/r/project/mediawiki/services/chromium-render/>). Code review happens in Gerrit. See Gerrit/Getting started to set up an account for yourself. Service uses the ServiceTemplateNode project template and follows all Service development rules.

Running the tests

To run all swagger tests and mocha tests:

```
npm test
```

To run all coverage test:

```
npm run coverage
```

Technical documents

- [README.MD \(https://github.com/wikimedia/mediawiki-services-chromium-render/blob/master/README.md\)](https://github.com/wikimedia/mediawiki-services-chromium-render/blob/master/README.md) has the documentation about Proton internals and configuration variables.
- [ServiceTemplateNode/Deployment](#) How to deploy Proton service
- [Reading/Web/Projects/Print_Styles](#) Print styles and limitations are identical to Browser print styles.

Links for Proton developers


- [Proton GitHub repository \(https://github.com/wikimedia/mediawiki-services-chromium-render\)](https://github.com/wikimedia/mediawiki-services-chromium-render)
- [Puppeteer documentation \(https://github.com/GoogleChrome/puppeteer\)](https://github.com/GoogleChrome/puppeteer)
- [BBPromise documentation \(http://bluebirdjs.com/docs/api-reference.html\)](http://bluebirdjs.com/docs/api-reference.html)

See also

- [PDF export](#)
- [REStBase](#): a caching / storing API proxy for PDFs generated by Proton
- [wikitech:Proton](#): details on monitoring, deployment, and data flow

Contact

If you need help or have questions/feedback, you can contact us in [#wikimedia-infrastructure \(irc://chat.freenode.net:6667/wikimedia-infrastructure\)](#) [connect \(https://webchat.freenode.net/?channels=wikimedia-infrastructure\)](https://webchat.freenode.net/?channels=wikimedia-infrastructure) or [the wikitech-l \(https://lists.wikimedia.org/mailman/listinfo/wikitech-l\)](https://lists.wikimedia.org/mailman/listinfo/wikitech-l) [mailing list](#).



Proton is maintained by [Product Infrastructure](#).

Get help:

- [#wikimedia-infrastructure \(irc://chat.freenode.net:6667/wikimedia-infrastructure\)](#) [connect \(https://webchat.freenode.net/?channels=wikimedia-infrastructure\)](https://webchat.freenode.net/?channels=wikimedia-infrastructure) ← Live chat (IRC)
- [wikitech-l \(https://lists.wikimedia.org/mailman/listinfo/wikitech-l\)](https://lists.wikimedia.org/mailman/listinfo/wikitech-l) ← Mailing list
- [Talk:Proton](#) ← Talk page
- [proton \(add task \(https://phabricator.wikimedia.org/maniphest/task/edit/form/1/?tags=proton\)\)](https://phabricator.wikimedia.org/maniphest/task/edit/form/1/?tags=proton) ← [Phabricator](#) project page

Retrieved from "<https://www.mediawiki.org/w/index.php?title=Proton&oldid=3965278>"

This page was last edited on 15 July 2020, at 11:04.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.