

Critical Request Chains

Overview


The Critical Request Chain is a concept from the Critical Rendering Path (CRP) optimization strategy. CRP enables the browser to load a page as quickly as possible by prioritizing which resources get loaded and the order in which they load.

Check out the [Critical Rendering Path](#) docs to learn more.

Recommendations

This audit is currently not structured as something to "pass" or "fail". The information that this audit provides gives you an opportunity to improve the page load performance of your app.

In the Chrome Extension version of Lighthouse, your report generates a diagram like the following:

Initial navigation 

```
|---lighthouse/ (developers.google.com)
  |---/css (fonts.googleapis.com) - 1058.34ms, 72.80KB
  |---css/devsite-googler-buttons.css (developers.google.com) - 1147.25ms, 70.7
  |---jsi18n/ (developers.google.com) - 1155.12ms, 71.20KB
  |---css/devsite-google-blue.css (developers.google.com) - 2034.57ms, 85.83KB
  |---2.2.0/jquery.min.js (ajax.googleapis.com) - 2699.55ms, 99.92KB
  |---contributors/kaycebasques.jpg (developers.google.com) - 2841.54ms, 84.74K
  |---MC30SXJEl4/photo.jpg (lh3.googleusercontent.com) - 3200.39ms, 73.59KB
```

This diagram represents the page's critical request chains. The path from `lighthouse/` to `/css` is one chain. The path from `lighthouse/` to `css/devsite-googler-buttons.css` is another chain. And so on. The top-most score of the audit represents this number of chains. For example, the diagram above would have a "score" of seven.

The diagram also breaks down how much time was spent downloading each resource, and the number of bytes that was required to download each resource.

You can use this diagram to improve your CRP by:

- Minimizing the number of critical resources: eliminating them, deferring their download, marking them as async, and so on.
- Optimizing the number of critical bytes to reduce the download time (number of roundtrips).
- Optimizing the order in which the remaining critical resources are loaded: downloading all critical assets as early as possible to shorten the critical path length.

Optimizing any of these factors results in a faster page load.

More information

Lighthouse uses network priority as a proxy for identifying render-blocking critical resources. See [Chrome Resource Priorities and Scheduling](#) for more information on how Chrome defines these priorities.

Data on critical request chains, resource sizes, and time spent downloading resources is extracted from the Chrome Debugger Protocol.

Feedback

Was this page helpful?

YES

NO

Great! Thank you for the feedback.

Sorry to hear that. Please [open an issue](#) and tell us how we can improve.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 24, 2018.