

Introduction



By Addy Osmani

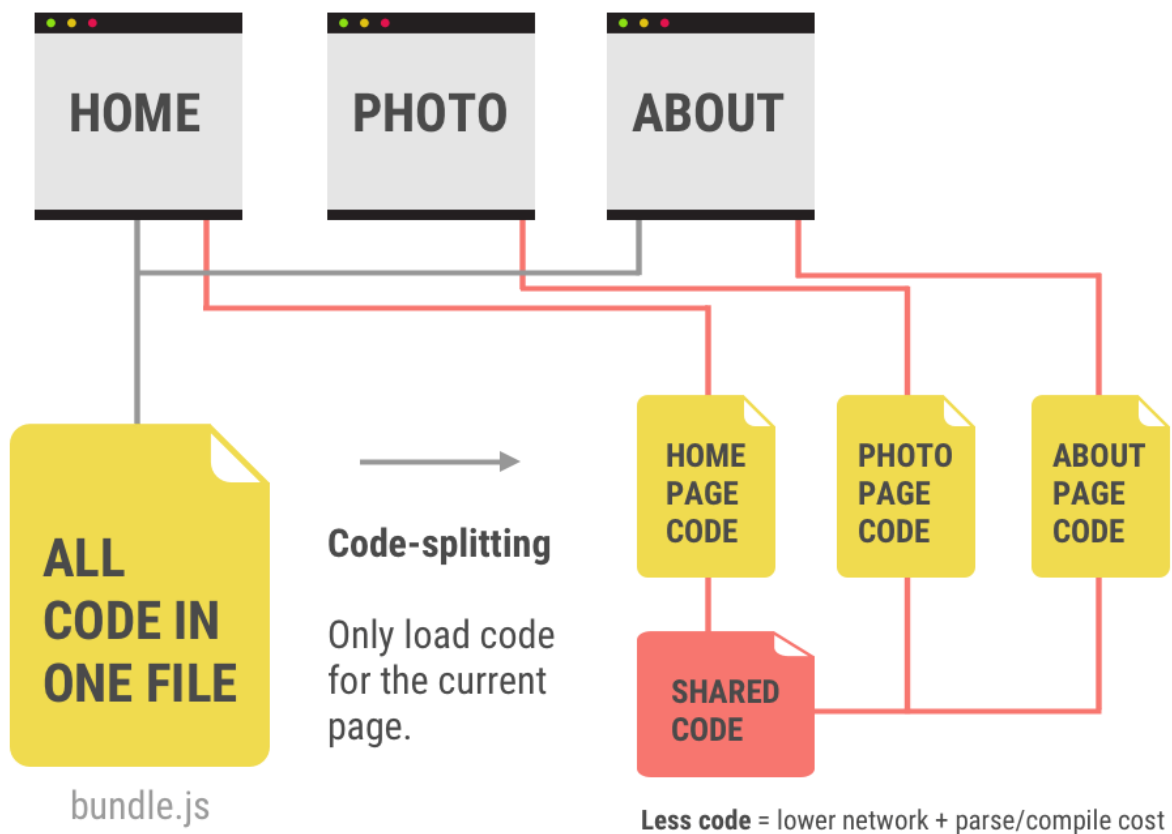
Eng Manager, Web Developer Relations

Modern web applications often use a **bundling tool** to create a production "bundle" of files (scripts, stylesheets, etc.) that is optimized, minified and can be downloaded in less time by your users. In **Web Performance Optimization with webpack**, we will walk through how to effectively optimize site resources using webpack. This can help users load and interact with your sites more quickly.



webpack

webpack is one of the most popular bundling tools in use today. Taking advantage of its features for optimizing modern code, code-splitting scripts into critical and non-critical pieces and stripping out unused code (to name but a few optimizations) can ensure your app has a minimal network and processing cost.



Inspired by [Code-splitting in Bundle Buddy](#) by Susie Lu

Note: We created a training app to play with optimizations described in this article. Try squeezing the most out of it to practice the tips: [webpack-training-project](#)

Let's get started by looking at optimizing one of the costliest resources in a modern app – JavaScript.

- [Decrease Front-end Size](#)
- [Make Use of Long-term Caching](#)
- [Monitor and analyze the app](#)
- [Conclusions](#)

[Next](#)
[Decrease Front-end Size](#) →

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 2, 2018.