Webpack

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.

> {% Img src="image/T4FyVKpzu4WKF1kBNvXepbi08t52/14nTVvq5PEzMdpksfenJ.png", alt="Webpack logo.", width="800", height="311" %}

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.

> {% Img src="image/T4FyVKpzu4WKF1kBNvXepbi08t52/9IVFhBrXUo9CU1Hwl0FK.png", alt="Before and after applying JavaScript optimizations. Time-to-Interactive is improved", width="800", height="575" %}
>
> Inspired by Code-splitting in Bundle Buddy by Susie Lu

{% Aside %} 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` {% endAside %}

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

□□□ #2
□ □□□ □□□ 2026-10-31 22:28:56 UTC
□ □□□ □□□ 2026-10-31 22:28:58 UTC