Run Third-Party Scripts From A Web Worker

ON THIS PAGE

- Overview
- Goals
- Web Workers
- Negative Impacts from Third-Party Scripts
- Use-Cases
- “Ready to Party” Plugins / Libraries
- Articles

Partytown is a lazy-loaded library to help relocate resource intensive scripts into a web worker, and off of the main thread. Its goal is to help speed up sites by dedicating the main thread to your code, and offloading third-party scripts to a web worker.

Even with a fast and highly tuned website following all of today’s best practices, it’s all too common for your performance wins to be erased the moment third-party scripts are added. By third-party scripts we mean code that is embedded within your site, but not directly under your control. A few examples include: analytics, metrics, ads, A/B testing, trackers, etc.

Partytown is maintained by Builder.io and is currently in Beta.

Smashing Magazine: How Partytown Eliminates Website Bloat From Third-Party Scripts

Goals

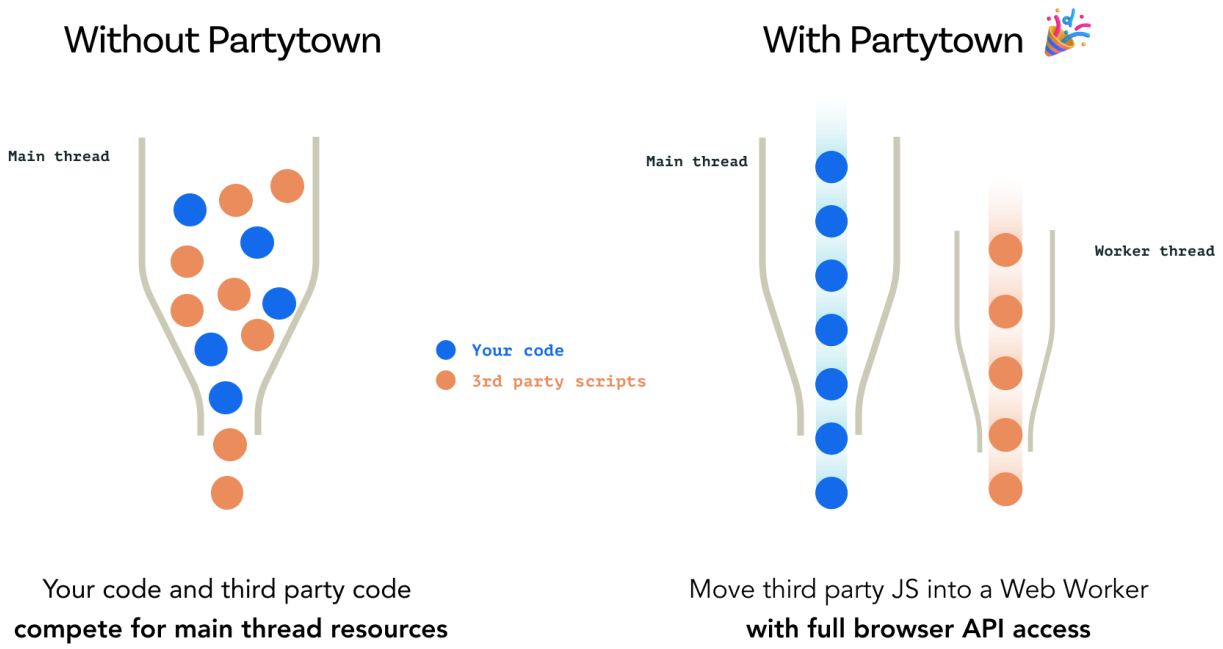
We set out to solve this situation, so that apps of all sizes will be able to continue to use third-party scripts without the performance hit. Some of Partytown’s goals include:

- Free up main thread resources to be used only for the primary web app execution.
- Sandbox third-party scripts and allow or deny their access to main thread APIs.
- Isolate long-running tasks within the web worker thread.

- Reduce layout thrashing coming from third-party scripts by batching DOM setters/getter into group updates.
- Throttle third-party scripts' access to the main thread.
- Allow third-party scripts to run exactly how they're coded and without any alterations.
- Read and write main thread DOM operations *synchronously* from within a web worker, allowing scripts running from the web worker to execute as expected.

Web Workers

Partytown's philosophy is that the main thread should be dedicated to your code, and any scripts that are not required to be in the critical path should be moved to a web worker. Main thread performance is, without question, more important than web worker thread performance. Please see the test pages for some live demos.



Negative Impacts from Third-Party Scripts

Below is a summary of potential issues after add third-party scripts, referenced from Loading Third-Party JavaScript:

- Firing too many network requests to multiple servers. The more requests a site has to make, the longer it can take to load.

- Sending too much JavaScript which keeps the main thread busy. Too much JavaScript can block DOM construction, delaying how quickly pages can render.
- CPU-intensive script parsing and execution can delay user interaction and cause battery drain.
- Third-party scripts that were loaded without care can be a single-point of failure (SPOF).
- Insufficient HTTP caching, forcing resources to be fetched from the network often.
- The use of legacy APIs (e.g `document.write()`), which are known to be harmful to the user experience.
- Excessive DOM elements or expensive CSS selectors.
- Including multiple third-party embeds that can lead to multiple frameworks and libraries being pulled in several times, which exacerbates the performance issues.
- Third-party scripts also often use embed techniques that can block `window.onload`, even if the embed is using async or defer.

Use-Cases

While full webapps “can” run from within Partytown, it’s actually best intended for code that doesn’t need to be in the critical rendering path. Most third-party scripts fall into this category, as they’re not required for the first-paint. Additionally, the asynchronous nature of most third-party script works well with Partytown, as they can lazily receive user events and post data to their respective services.

Below are just a few examples of third-party scripts that might be a good candidate to run from within a web worker. The goal is to continue validating commonly used services to ensure Partytown has the correct API, but Partytown itself does not hardcode to any specific services. Help us test and join the conversation in Partytown’s Discord!

- Google Tag Manager (GTM)
- Google Analytics (GA)
- Facebook Pixel
- Mixpanel
- Hubspot
- Segment
- Amplitude

“Ready to Party” Plugins / Libraries

We try and keep a list of all the plugins and libraries that we know of that works out-of-the-box in Partytown, but we would love your help as plugin & library authors and contributors to keep this list growing.

We have some documentation on how you could create and check-in an integration test that shows how your library / plugin works on Partytown. And if it works, then we would love to have the configuration needed (if any) on our Common Services page

Articles

- Smashing Magazine: How Partytown Eliminates Website Bloat From Third-Party Scripts
- Introducing Partytown: Run Third-Party Scripts From a Web Worker
- How Partytown's Sync Communication Works
- How we cut 99% of our JavaScript with Qwik + Partytown
- Partytown is now in Beta
- Patterns: Optimize loading third-parties