

Filter

Migrate to a service worker

Update your code

Replace blocking web request listeners

Improve extension security

Deal with remote hosted code violations

Publish your extension

Security and privacy ▼

How To ▼

Support and feedback ▼



- [inject remote hosted stylesheets](#) into a web page using `insertCSS`

- For extensions using `chrome.devtools`: `inspectWindow.eval` allows executing JavaScript in the context of the inspected page.
- Debugger extensions can use `chrome.debugger.sendCommand` to execute JavaScript in a debug target.

## Remove remotely hosted code

In Manifest V3, all of your extension's logic must be part of the extension package. You can no longer load and execute remotely hosted files according to [Chrome Web Store policy](#). Examples include:

- JavaScript files pulled from the developer's server.
- Any library hosted on a [CDN](#).
- Bundled third-party libraries that dynamically fetch remote hosted code.

Alternative approaches are available, depending on your use case and the reason for remote hosting. This section describes approaches to consider. If you are having issues with dealing with remote hosted code, we have [guidance available](#).

### Configuration-driven features and logic

Your extension loads and caches a remote configuration (for example a JSON file) at runtime. The cached configuration determines which features are enabled.

### Externalized logic with a remote service

### On this page

[Remove execution of arbitrary strings](#)[Remove remotely hosted code](#)[Configuration-driven features and logic](#)[Externalized logic with a remote service](#)[Embed remotely hosted code in a sandboxed iframe](#)[Bundle third-party libraries](#)[Use external libraries in tab-injected scripts](#)[Inject a function](#)[Look for other workarounds](#)