Best practices for cookie notices

- Performance
  - Best practices
    - Load cookie notices scripts asynchronously
    - Load cookie notice scripts directly
    - Establish an early connection with the cookie notice origin
    - Preload cookie notices as appropriate
    - Be aware of performance tradeoffs when styling cookie notices
    - Avoid layout shifts
    - Advanced loading optimizations
- Performance measurement
  - Real User Monitoring (RUM)
  - Synthetic monitoring
  - Testing cookie notices with WebPageTest
    - Scripting
    - Set cookies
    - Change the test location
  - Testing cookie notices with Lighthouse
    - DevTools
    - CLI
    - PageSpeed Insights
- User experience
  - Placement
    - Header, footer, and inline cookie notices
    - Modals
  - Configurability
    - No configurability

- Some configurability
- Full configurability

This article discusses how cookie notices can affect performance, performance measurement, and user experience.

Performance

Cookie notices can have a significant impact on page performance due to the fact that they are typically loaded early in the page load process, are shown to all users, and can potentially influence the loading of ads and other page content.

Here's how cookie notices can impact Web Vitals metrics:

- **Largest Contentful Paint (LCP):** Most cookie consent notices are fairly small and therefore typically don't contain a page's LCP element. However, this can happen—particularly on mobile devices. On mobile devices, a cookie notice typically takes up a larger portion of the screen. This usually occurs when a cookie notice contains a large block of text (text blocks can be LCP elements too).
- **First Input Delay (FID):** Generally speaking, your cookie consent solution in and of itself should have a minimal impact on FID—cookie consent requires little JavaScript execution. However, the technologies that these cookies enable—namely advertising and tracking scripts—may have a significant impact on page interactivity. Delaying these scripts until cookie acceptance can serve as a technique to decrease First Input Delay (FID).
- **Cumulative Layout Shift (CLS):** Cookie consent notices are a very common source of layout shifts.

Generally speaking, you can expect a cookie notice from third-party providers to have a greater impact on performance than a cookie notice that you build yourself. This is not a problem unique to cookie notices—but rather the nature of third-party scripts in general.

Best practices

The best practices in this section focus on third-party cookie notices. Some, but not all, of these best practices will also be applicable to first-party cookie notices.

Load cookie notices scripts asynchronously

Cookie notice scripts should be loaded asynchronously. To do this, add the `async` attribute to the script tag.

```
<script src="https://cookie-notice.com/script.js" async>
```

Scripts that are not asynchronous block the browser parser. This delays page load and LCP. For more information, see Efficiently load third-party JavaScript.

If you must use synchronous scripts (for example, some cookie notices rely on synchronous scripts to implement cookie blocking) you should make sure that this request loads as quickly as possible. One way to do this is to use resource hints.

Load cookie notice scripts directly

Cookie notice scripts should be loaded "directly" by placing the script tag in the HTML of the main document—rather than loaded by a tag manager or other script. Using a tag manager or secondary script to inject the cookie notice script delays the loading of the cookie notice script: it obscures the script from the browser's lookahead parser and it prevents the script from loading before JavaScript execution.

Establish an early connection with the cookie notice origin

All sites that load their cookie notice scripts from a third-party location should use either the `dns-prefetch` or `preconnect` resource hints to help establish an early connection with the origin that hosts cookie notice resources. For more information, see Establish network connections early to improve perceived page speed.

```
<link rel="preconnect" href="https://cdn.cookie-notice.com/">
```

It is common for cookie notices to load resources from multiple origins— for example, loading resources from both `www.cookie-notice.com` and `cdn.cookie-notice.com`. Separate origins require separate connections and therefore separate resource hints.

Preload cookie notices as appropriate

Some sites would benefit from using the `preload` resource hint to load their cookie notice script. The `preload` resource hint informs the browser to initiate an early request for the specified resource.

```
<link rel="preload" href="https://www.cookie-notice.com/cookie-script.js">
```

`preload` is most powerful when its usage is limited to fetching a couple key resources per page. Thus, the usefulness of preloading the cookie notice script will vary depending on the situation.

Be aware of performance tradeoffs when styling cookie notices

Customizing the look and feel of a third-party cookie notice may incur additional performance costs. For example, third-party cookie notices aren't always able to reuse the same resources (for example, web fonts) that are used elsewhere on the page. In addition, third-party cookie notices tend to load styling at the end of long request chains. To avoid any surprises, be aware of how your cookie notice loads and applies styling and related resources.

Avoid layout shifts

These are some of the most common layout shift issues associated with cookie notices:

- **Top-of-screen cookie notices:** Top-of-screen cookie notices are a very common source of layout shift. If a cookie notice is inserted into the DOM after the surrounding page has already rendered, it will push the page elements below it further down the page. This type of layout shift can be eliminated by reserving space in the DOM for the consent notice. If this is not a feasible solution—for example, if the dimensions of your cookie notice vary by geography, consider using a sticky footer or modal to display the cookie notice. Because both of these alternative approaches display the cookie notice as an "overlay" on top of the rest of the page, the cookie notice should not cause content shifts when it loads.
- **Animations**: Many cookie notices use animations—for example, "sliding in" a cookie notice is a common design pattern. Depending on how these effects are implemented, they can cause layout shifts. For more information, see Debugging layout shifts.
- **Fonts**: Late-loading fonts can block render and or cause layout shifts. This phenomena is more apparent on slow connections.

Advanced loading optimizations

These techniques take more work to implement but can further optimize the loading of cookie notice scripts:

- Caching and serving third-party cookie notice scripts from your own servers can improve the delivery speed of these resources.
- Using service workers can allow you more control over the fetching and caching of third-party scripts such as cookie notice scripts.

Performance measurement

Cookie notices can impact performance measurements. This section discusses some of these implications and techniques for mitigating them.

Real User Monitoring (RUM)

Some analytics and RUM tools use cookies to collect performance data. In the event that a user declines usage of cookies these tools cannot capture performance data.

Sites should be aware of this phenomenon; it is also worthwhile to understand the mechanisms that your RUM tooling uses to collect its data. However, for the typical site this discrepancy probably isn't a cause for alarm given the direction and magnitude of the data skew. Cookie usage is not a technical requirement for performance measurement. The web-vitals JavaScript library is an example of a library that does not use cookies.

Depending on how your site uses cookies to collect performance data (that is, whether the cookies contain personal information), as well as the legislation in question, the use of cookies for performance measurement might not be subject to the same legislative requirements as some of the cookies used on your site for other purposes—for example, advertising cookies. Some sites choose to break out performance cookies as a separate category of cookies when asking for user consent.

Synthetic monitoring

Without custom configuration, most synthetic tools (such as Lighthouse and WebPageTest) will only measure the experience of a first-time user who has not responded to a cookie consent notice. However, not only do variations in cache state (for example, an initial visit versus a repeat visit) need to be considered when collecting performance data, but also variations in cookie acceptance state—accepted, rejected, or unresponded.

The following sections discuss WebPageTest and Lighthouse settings that can be helpful for incorporating cookie notices into performance measurement workflows. However, cookies and cookie notices are just one of many factors that can be difficult to perfectly simulate in lab environments. For this reason, it is important to make RUM data the cornerstone of your performance benchmarking, rather than synthetic tooling.

Testing cookie notices with WebPageTest

Scripting

You can use scripting to have a WebPageTest "click" the cookie consent banner while collecting a trace.

Add a script by going to the **Script** tab. The script below navigates to the URL to be tested and then clicks the DOM element with the id `cookieButton`.

WebPageTest scripts are tab-delimited.

```
combineSteps
navigate    %URL%
clickAndWait    id=cookieButton
```

When using this script be aware that:

- `combineSteps` tells WebPageTest to "combine" the results of the scripting steps that follow into a single set of traces and measurements. Running this script without `combineSteps` can also be useful—separate traces make it easy to see whether resources were loaded before or after cookie acceptance.
- `%URL%` is a WebPageTest convention that refers to the URL that is being tested.
- `clickAndWait` tells WebPageTest to click on the element indicated by `attribute=value` and wait for the subsequent browser activity to complete. It follows the format `clickAndWait attribute=Value`.

If you've configured this script correctly, the screenshot taken by WebPageTest should not show a cookie notice (the cookie notice has been accepted).

For more information on WebPageTest scripting, check out WebPageTest documentation.

Set cookies

To run WebPageTest with a cookie set, go to the **Advanced** tab and add the cookie header to the **Custom headers** field:

Screeshot showing the 'Custom headers' field in WebPageTest

Change the test location

To change the test location used by WebPageTest, click the **Test Location** dropdown located on the **Advanced Testing** tab.

Screenshot of the 'Test Location' dropdown in WebPageTest

Testing cookie notices with Lighthouse

Setting cookies on a Lighthouse run can serve as a mechanism for getting a page into a particular state for testing by Lighthouse. Lighthouse's cookie behavior varies slightly by context (DevTools, CLI, or PageSpeed Insights).

DevTools

Cookies are not cleared when Lighthouse is run from DevTools. However, other types of storage are cleared by default. This behavior can be changed by using the **Clear Storage** option in the

Lighthouse settings panel.

Screenshot higlighting the Lighthouse 'Clear Storage' option

CLI

Running Lighthouse from the CLI uses a fresh Chrome instance, so no cookies are set by default. To run Lighthouse from the CLI with a particular cookie set, use the following command:

```
lighthouse <url> --extra-headers "{\"Cookie\":\"cookie1=abc; cookie2=def; \_id=foo\"}"
```

For more information on setting custom request headers in Lighthouse CLI, see Running Lighthouse on Authenticated Pages.

PageSpeed Insights

Running Lighthouse from PageSpeed Insights uses a fresh Chrome instance and does not set any cookies. PageSeed Insights cannot be configured to set particular cookies.

User experience

The user experience (UX) of different cookie consent notices is primarily the result of two decisions: the location of the cookie notice within the page and the extent to which the user can customize a site's use of cookies. This section discusses potential approaches to these two decisions.

Cookie notice UX is often heavily influenced by legislation which can vary widely by geography. Thus, some of the design patterns discussed in this section may not be relevant to your particular situation. This article should not be considered a substitute for legal advice.

When considering potential designs for your cookie notice, here are some things to think about:

- UX: Is this a good user experience? How will this particular design affect existing page elements and user flows?
- Business: What is your site's cookie strategy? What are your goals for the cookie notice?
- Legal: Does this comply with legal requirements?
- Engineering: How much work would this be to implement and maintain? How difficult would it be to change?

Placement

Cookie notices can be displayed as a header, inline element, or footer. They can also be displayed on top of page content using a modal or served as an interstitial.

Diagram showing examples of different placement options for cookie notices

Header, footer, and inline cookie notices

Cookie notices are commonly placed in the header or footer. Of these two options, the footer placement is generally preferable because it is unobtrusive, does not compete for attention with banner ads or notifications, and typically does not cause CLS. In addition, it is a common place for placing privacy policies and terms of use.

Although inline cookie notices are an option, they can be difficult to integrate into existing user interfaces, and therefore are uncommon.

Modals

Modals are cookie consent notices that are displayed on top of page content. Modals can look and perform quite differently depending on their size.

Smaller, partial-screen modals can be a good alternative for sites that are struggling to implement cookie notices in a way that doesn't cause layout shifts.

On the other hand, large modals that obscure the majority of page content should be used carefully. In particular, smaller sites may find that users bounce rather than accept the cookie notice of an unfamiliar site with obscured content. Although they are not necessarily synonymous concepts, if you are considering using a full-screen cookie consent modal, you should be aware of legislation regarding cookie walls.

Large modals can be considered a type of interstitial. Google Search does not penalize the usage of interstitials when they are used to comply with legal regulations such as in the case of cookie banners. However, their usage of interstitials in other contexts—particularly if they are intrusive or create a poor user experience—may be penalized.

Configurability

Cookie notice interfaces give users varying levels of control over which cookies they accept.

No configurability

These notice-style cookie banners do not present users with direct UX controls for opting out of cookies. Instead, they typically include a link to the site's cookie policy which may provide users with information about managing cookies using their web browser. These notices typically include either a "dismiss" and/or "Accept" button.

Diagram showing examples of cookie notices with no cookie configurability

Some configurability

These cookie notices give the user the option of declining cookies but do not support more granular controls. This approach to cookie notices is less common.

Diagram showing examples of cookie notices with some cookie configurability

Full configurability

These cookie notices provide users with more fine-grained controls for configuring the cookie usage that they accept.

Diagram showing examples of chookie notices with full cookie configurability

- **UX:** Controls for configuring cookie usage are most commonly displayed using a separate modal that is launched when the user responds to the initial cookie consent notice.
  However, if space permits, some sites will display these controls inline within the initial cookie consent notice.
- **Granularity:** The most common approach to cookie configurability is to allow users to opt-in to cookies by cookie "category". Examples of common cookie categories include functional, targeting, and social media cookies.
  However, some sites will go a step further and allow users to opt-in on a per-cookie basis. Alternatively, another way of providing users with more specific controls is to break down cookie categories like "advertising" into specific use cases—for example, allowing users to separately opt-in to "basic ads" and "personalized ads".

Diagram showing examples of cookie notices with full cookie configurability

---

□□□ #3
□ □□□ □□□ 2026-10-31 22:34:54 UTC
□ cocobook-sync □□□ 2026-11-04 17:38:47 UTC