

What is Time to Live (TTL) | TTL Best Practices | CDN Guide | Imperva

4-5 minutes

What is Time to Live

Time to live (TTL) is the time that an object is stored in a caching system before it's deleted or refreshed. In the context of CDNs, TTL typically refers to [content caching](#), which is the process of storing a copy of your website resources (e.g., images, prices, text) on CDN [proxies](#) to improve page load speed and reduce origin server bandwidth consumption.

In this context, TTL governs the refresh rate of these copies, ideally ensuring that “stale” versions of your content aren't served to your website visitors.

Stale Content / Expired TTL



Fresh Content



Best Practices for Setting Your TTL Policy

TTL is set via HTTP headers, such as the [Cache-Control header](#), and is measured in seconds. For example, the value "Cache-Control: max-age=30" is a directive for a given resource to be refreshed every 30 seconds before time to live is exceeded. Conversely, the directive "max-age=0" indicates that the resource shouldn't be [cached](#) at all.

Time to live directly impacts your site's [page load time](#) (i.e., cached data loads faster), as well as content freshness (i.e., data cached for too long can become stale).

To ensure that your visitors only see the latest version of your website, TTLs should be set according to:

Resource type

There are two resource types to consider when setting your cache TTLs: static and dynamic.

Static files (e.g., image files, PDFs, etc.) are rarely updated, and therefore typically have a longer TTL. For example, an ecommerce site's pool of product images represents static content. Because they're rarely refreshed, it's safe to cache them for an extended period (e.g., days or weeks). This makes setting their TTL predictable and easy to maintain.

Conversely, dynamic content (e.g. HTML files) is constantly updated, complicating the setting of accurate TTLs. For example, the comments section under a product is considered dynamic, as it changes frequently. Because you're not in full control of this content and cannot predict its refresh time, especially if there is an option for users to modify existing posts, caching is not advisable.

Resource use

A resource's cache time to live should also be adjusted based on how often it's used. For example, an ecommerce site may choose to update an HTML file displaying their product prices every day. Therefore, pages including pricing should be set with a TTL that expires at the end of the day, ensuring customers only see the most updated information.

How CDNs Simplify Time to Live Configuration

The efficiency of your time to live configuration is a function of your ability to accurately predict the rate at which your different website resources are updated.

Clearly, the more moving pieces you have to deal with, the harder this task become. This is why most websites that have dozens, or even hundreds of elements per page are rarely able to maintain a granular TTL policy.

Using a [CDN](#) simplifies the process by providing a user-friendly method of updating TTL settings from a control dashboard. This enables you to change settings for individual resources or entire file groups, without having to manually tinker with their HTTP headers.

Certain [CDNs](#) also employ machine learning techniques to manage your entire time to live policy, effectively automating the process. Significantly, this allows for the caching of some of your website dynamic content.

By monitoring user interactions within your website, a CDN is able to track the differences between the versions of content served to different users. Based on these differences, the service is able to determine whether a certain object is truly dynamic or simply built using a dynamic language (e.g., PHP), with an identical copy served to all visitors.

Expanding your cache policy to include dynamic objects can have a dramatic impact on your website performance, often boosting

cache utilization by 20% or more. Furthermore, this ensures that users receive accurate and up-to-date information.