

April 23, 2025

TEsting new subheadings

Introduction to Drupal Caching

Drupal uses a powerful caching system to improve performance and scalability. It includes render caching, dynamic page caching, and internal page caching. These systems work together to store parts of the page or full page responses so future visits are much faster.

What Are Cache Tags?

Cache tags in Drupal provide a smart way to group cache entries. Instead of clearing the entire cache, you can invalidate just the pieces associated with a specific tag—like a single node or config item—ensuring that only necessary content is refreshed.

Invalidating Cache Tags

Developers can invalidate cache tags using the ``Cache::invalidateTags()`` method in custom modules. This enables fine-grained control, allowing dynamic content to stay fresh without taking a performance hit from clearing global caches.

Using Cache Tags in Blocks

When building custom blocks, it's best practice to define relevant cache tags in the render array. This allows your block to be automatically cleared from cache when related content—such as nodes or config—changes.

Best Practices for Managing Cache

Use specific and meaningful cache tags. Avoid invalidating broad tags like ``rendered`` unless necessary. Consider leveraging Drupal's built-in cache contexts and max-age settings for even more precise control over what gets cached and when.