

𝗦𝘆𝘀𝘁𝗲𝗺 𝗗𝗲𝘀𝗶𝗴𝗻 𝗕𝗮𝘀𝗶𝗰𝘀: 𝗪𝗵𝗮𝘁 𝗶𝘀 𝗖𝗮𝗰𝗵𝗲 𝗜𝗻𝘃𝗮𝗹𝗶𝗱𝗮𝘁𝗶𝗼𝗻?  
  
Cache invalidation is the process of removing or updating cached data when it becomes outdated or no longer accurate. "Purge," "refresh," and "ban" are commonly used cache invalidation methods that are frequently used in the application cache, content delivery networks (CDNs), and web proxies. Here's a brief description of a few famous cache invalidation methods:  
  
◇ 𝗣𝘂𝗿𝗴𝗲: The purge method removes cached content for a specific object, URL, or a set of URLs. It's typically used when there is an update or change to the content and the cached version is no longer valid. When a purge request is received, the cached content is immediately removed, and the next request for the content will be served directly from the origin server.  
  
◇ 𝗥𝗲𝗳𝗿𝗲𝘀𝗵: Fetches requested content from the origin server, even if cached content is available. When a refresh request is received, the cached content is updated with the latest version from the origin server, ensuring that the content is up-to-date. Unlike a purge, a refresh request doesn't remove the existing cached content; instead, it updates it with the latest version.  
  
◇ 𝗕𝗮𝗻: The ban method invalidates cached content based on specific criteria, such as a URL pattern or header. When a ban request is received, any cached content that matches the specified criteria is immediately removed, and subsequent requests for the content will be served directly from the origin server.  
  
◇ 𝗧𝗶𝗺𝗲-𝘁𝗼-𝗹𝗶𝘃𝗲 (𝗧𝗧𝗟) 𝗲𝘅𝗽𝗶𝗿𝗮𝘁𝗶𝗼𝗻: This method involves setting a time-to-live value for cached content, after which the content is considered stale and must be refreshed. When a request is received for the content, the cache checks the time-to-live value and serves the cached content only if the value hasn't expired. If the value has expired, the cache fetches the latest version of the content from the origin server and caches it.  
  
◇ 𝗦𝘁𝗮𝗹𝗲-𝘄𝗵𝗶𝗹𝗲-𝗿𝗲𝘃𝗮𝗹𝗶𝗱𝗮𝘁𝗲: This method is used in web browsers and CDNs to serve stale content from the cache while the content is being updated in the background. When a request is received for a piece of content, the cached version is immediately served to the user, and an asynchronous request is made to the origin server to fetch the latest version of the content. Once the latest version is available, the cached version is updated. This method ensures that the user is always served content quickly, even if the cached version is slightly outdated.