

ETag vs Cache-Control (no-cache, no-store, ecc.)

 **Ottima domanda finale!**

Capire la **differenza tra ETag e no-cache** (e in generale la gestione della **cache HTTP**) è fondamentale per padroneggiare il comportamento del browser.

Vediamo le **differenze principali**, chiare e senza confusione.

1. ETag vs Cache-Control (no-cache, no-store, ecc.)

Concetto	Descrizione semplice
ETag	Identificatore univoco del contenuto (hash), usato per sapere se il contenuto è cambiato
Cache-Control: no-cache	Dice al browser: "Non usare direttamente la cache. Verifica prima col server se è cambiato"
Cache-Control: no-store	Dice al browser: "Non salvare nulla in cache, mai"
Expires / max-age	Specifica per quanto tempo il contenuto può restare nella cache

Differenza pratica

Con ETag:

- Il browser **può salvare la risposta**
- Alla prossima richiesta, invia If-None-Match: "etag-valore"
- Il server risponde:
 - 304 Not Modified se **non è cambiato**
 - 200 OK con nuovo contenuto se **è cambiato**

Vantaggio: risparmio di banda, verifica intelligente

Con Cache-Control: no-cache:

- Il browser **deve sempre** chiedere al server
- Ma può ancora **salvare una copia** della risposta
- Serve per forzare una verifica **prima di usare la cache**

 **Attenzione:** "no-cache" **non vuol dire** "non usare la cache", ma "non usarla **senza chiedere prima**".

Con Cache-Control: no-store:

- Il browser **non salva nulla**
- Ogni richiesta è **completamente nuova**

- Usato per contenuti **sensibili o dinamici**, come:
 - dati personali
 - login/logout
 - info bancarie

Esempio comparativo

Header HTTP	Cosa succede
ETag: "abc123"	Salva in cache, confronta la prossima volta
Cache-Control: no-cache	Salva in cache, ma chiede sempre al server
Cache-Control: no-store	Non salva niente , mai
Cache-Control: max-age=3600	Usa dalla cache per 1 ora (senza chiedere)

In sintesi

Meccanismo	Tipo	Scopo principale
ETag	Condizionale	Capire se il contenuto è cambiato
no-cache	Cache policy	Forza verifica prima di usare la cache
no-store	Cache policy	Impedisce del tutto il caching
max-age, expires	Cache policy	Stabilisce quanto dura la cache

Quando usare cosa?

Caso d'uso	Cosa usare?
File statici (JS, CSS, immagini)	ETag, Last-Modified, max-age
Dati dinamici (JSON, PHP, API)	ETag + Cache-Control: no-cache
Dati sensibili (autenticazione)	Cache-Control: no-store, private
Vuoi sempre bypassare la cache	Cache-Control: no-store

Se vuoi ti preparo una **tabella di esempio con header reali** oppure un **diagramma visivo** di come lavora la cache. Fammi sapere!