





Cache

Sidrit Trandafli

 @sidis405 @strandafli

- Sviluppatore PHP 15+ anni
- **Laravel** da 4 anni
- Back-end
 - php
 - java
 - node
 - python
- Front-end
 - VueJS
 - Vanilla JS
 - TailwindCSS
 - Sass
- Devops + Sysops da 13 anni





Cache

“Laravel ci dà la possibilità di gestire la Cache tramite varie tecnologie (file, redis, memcached)”

Il driver di default di Laravel è 'file'. Questo vuol dire che la nostra cache verrà salvata su file.

I driver di cache supportati sono "apc", "array", "database", "file", "memcached" e "redis".

La cache può essere salvata sul database, oppure può essere salvata in coppie chiave-valore

È possibile configurare per quanto tempo i dati rimarranno in cache.

È possibile svuotare la cache se necessario.

L'implementazione della cache garantisce un miglioramento in performance.

Implementando un metodo efficace di caching vengono impiegate meno risorse nel reperire i dati, meno chiamate al database, più utenti ricevono i dati più velocemente e l'infrastruttura si stressa di meno.

Cache

Per poter usare la cache con Laravel basta usare la facade Cache

(Illuminate\Support\Facades\Cache) oppure la funzione helper cache()

Salvare dati in cache è facilissimo. Per questo serve la funzione **put()**

put() - accetta 3 parametri

- la chiave con la quale vogliamo salvare i dati
- i dati da salvare
- la durata in minuti (questo parametro è opzionale)

Inoltre è possibile specificare programmaticamente la cache sulla quale salvare i dati

```
Cache::put('avenger_1', 'iron man', 10);  
Cache::store('redis')->put('avenger_1', 'hulk', 10);  
Cache::store('file')->put('avenger_3', 'thor', 10);
```



Cache

Reperire i dati dalla cache è altrettanto semplice . Per questo serve la funzione **get()**

get() - accetta 2 parametri

- la chiave dei dati che vogliamo
- un valore opzionale di default nel caso quella chiave non avesse un valore

Come secondo parametro possiamo passare anche una funzione anonima

```
$value = Cache::get('key');  
  
$value = Cache::get('key', 'default');  
  
$value = Cache::get('key', function () {  
    return DB::table(...)->get();  
});
```

Per controllare se una certa chiave è presente o meno in cache, usiamo **has()**

```
if (Cache::has('key')) {  
    //  
}
```

Cache

In uno scenario più complesso per esempio caching di query, è possibile interrogare la cache, e nel caso in cui i dati non fossero presenti è possibile eseguire le query e mettere il risultato nella cache allo stesso tempo

```
$value = Cache::remember('posts', $minutes, function () {  
    return Posts::all();  
});  
  
$value = Cache::rememberForever('posts', function() {  
    return Posts::all();  
});
```

La cancellazione dei dati dalla cache avviene tramite due metodi: **pull()** e **forget()**

Tutti e due rimuovono una chiave specifica dalla cache ma con la differenza che **pull()** restituisce il valore rimosso dalla cache, mentre **forget()** lo rimuove e non dà valori.