

Come usare Elasticsearch in Laravel

Elasticsearch è un motore di ricerca distribuito e altamente performante, ideale per la ricerca full-text e l'analisi dei dati. Laravel non lo supporta nativamente, ma è possibile integrarlo facilmente usando una libreria come `elasticsearch/elasticsearch`.

1. Installazione della libreria Elasticsearch

Per iniziare, installa il client ufficiale di Elasticsearch tramite Composer:

```
composer require elasticsearch/elasticsearch
```

Assicurati anche di avere un'istanza Elasticsearch in esecuzione (localmente o in cloud).

2. Configurazione del client

Puoi creare un service provider personalizzato o registrare il client nel container di Laravel. Un esempio semplice è creare un binding nel `AppServiceProvider`:

```
use Elasticsearch\ClientBuilder;

public function register()
{
    $this->app->singleton(\Elasticsearch\Client::class, function ()
    {
```

```

        return ClientBuilder::create()
            ->setHosts(['localhost:9200'])
            ->build();
    });
}

```

3. Creazione di un indice

Puoi creare un indice direttamente dal controller o da un command personalizzato:

```

use Elasticsearch\Client;

public function createIndex(Client $client)
{
    $params = [
        'index' => 'articoli',
        'body' => [
            'mappings' => [
                'properties' => [
                    'titolo' => ['type' => 'text'],
                    'contenuto' => ['type' =>
'text'],
                    'data_pubblicazione' => ['type'
=> 'date'],
                ]
            ]
        ]
    ];

    $response = $client->indices()->create($params);
    return response()->json($response);
}

```

4. Indicizzare un documento

Per aggiungere un documento all'indice:

```
$params = [  
  'index' => 'articoli',  
  'id'     => 1,  
  'body'  => [  
    'titolo' => 'Introduzione a Elasticsearch',  
    'contenuto' => 'Elasticsearch è un motore di  
ricerca open-source...',  
    'data_pubblicazione' => '2025-07-22',  
  ]  
];  
  
$client->index($params);
```

5. Ricerca full-text

Per eseguire una ricerca sul contenuto:

```
$params = [  
  'index' => 'articoli',  
  'body'  => [  
    'query' => [  
      'match' => [  
        'contenuto' => 'motore di ricerca'  
      ]  
    ]  
  ]  
];
```

```
$response = $client->search($params);  
return response()->json($response);
```

6. Suggerimenti finali

- Gestisci le eccezioni con try/catch per evitare errori silenziosi.
- Usa Laravel Jobs per indicizzazioni asincrone se hai grandi volumi di dati.
- Per progetti complessi, considera l'uso di pacchetti wrapper come Laravel Scout Elasticsearch Driver.

Conclusione

Integrare Elasticsearch in Laravel ti consente di costruire funzionalità di ricerca avanzate e performanti. Con pochi passaggi puoi configurare il client, creare indici, indicizzare documenti e offrire una ricerca full-text agli utenti della tua applicazione.