

# prova tecnica

---

di seguito il testo dell'esercizio.

ti chiedo di analizzare il problema, anche alla luce del tuo livello attuale di conoscenza di php e laravel, dandoci una stima orientativa della consegna.

se qualcosa non fosse chiara, per qualsiasi dubbio o se dovessi riscontrare ambiguità nel testo, non esitare a contattarmi. qualsiasi miglioria o suggerimento, rispetto a quanto richiesto, è ben accetta.

il codice dovrà essere pubblicato su un repository privato (su github o gitlab), così da poter visionare eventuali progressi.

suggerimento: il codice dovrebbe rispettare i dettami dello standard PSR-12.

## obiettivo

progettare un webservice che permetta la ricerca di gif animate da diversi provider di immagini, oltre alla gestione dei provider disponibili, nonché di analizzare le ricerche effettuate.

il sistema deve tenere conto di:

- numero totale di interrogazioni effettive al provider di immagini
- numero totale di ricerche, per ciascuna keyword, raggruppate per provider

il sistema deve essere estendibile, ossia dovrà essere possibile implementare altri providers.

al momento è richiesto lo sviluppo della ricerca su giphy e tenor.

## models principali

### GifProvider

è l'elenco dei providers disponibili. ciascun provider deve avere:

- un identificativo (slug) composto da lettere minuscole ed underscores
- una descrizione testuale
- un contatore di chiamate
- una colonna json contenenti le credenziali utilizzate dal provider (es. secret + public oppure username + password)

### Keyword

è la lista delle keywords finora ricercate. ogni record ha:

- il valore testuale della keyword

**IMPORTANTE:** va creata una relazione molti a molti tra **Keyword** e **GifProvider** in cui vanno aggiunti i seguenti attributi:

- un contatore di ricerche effettuate per la keyword nel provider

in questo modo sarà possibile conoscere la quantità di ricerche, per ciascuna keyword, separata per provider

## Configuration

questo model è un semplice contenitore key-value globale per i settings dell'applicazione. eventualmente, puoi scegliere anche di non implementarlo e di usare un'altra strategia (non è prevista la configurazione per utenti).

endpoint pubblici

## GET /providers

restituisce la lista di providers di ricerca disponibili, nella seguente forma:

```
{
  "providers": [
    {
      "identifier": "provider_name",
      "description": "qui metto il nome del provider...",
      "calls": 1234,
    },
    ...
  ]
}
```

**calls** indica il numero totale di interrogazioni **effettive** al provider, dalla prima installazione dell'applicazione.

## GET /provider/{identifier}/stats

deve restituire le chiamate totali **effettive** al provider e la lista delle keywords ricercate, comprensive di totale ricerche effettuate mediante il provider in oggetto.

la forma della response deve essere la seguente:

```
{
  "calls": 1234,
  "keywords": [
    {
      "keyword": "nome keyword",
      "calls": 2345,
    },
    ...
  ]
}
```

la response deve avere status 404 nel caso di provider inesistente.

## GET /gifs/{keyword}

la keyword deve essere ESCLUSIVAMENTE composta da caratteri alfabetici ed underscores (che saranno, ai fini della ricerca, convertiti in spazi). la ricerca deve essere case insensitive e deve essere effettuato il trim degli spazi iniziali, finali ed intermedi superflui, per non compromettere i risultati della ricerca.

qualsiasi carattere non alfanumerico o diverso da underscore deve essere convertito in underscore.

es.: `Pippo_Baudo` sarà trasformato esattamente in `pippo baudo`; `__ ciao__il_Mondo _`  
`_è_rotondo` deve diventare esattamente `ciao mondo rotondo` e così via.

questo permetterà di indicizzare (e conteggiare) in maniera univoca le keywords ricercate anche nel caso di errori di battitura.

una volta effettuata la prima ricerca per una specifica keyword, le successive chiamate dovranno attingere ad una cache che, **per le successive 6 ore**, deve restituire gli stessi risultati, a meno che:

- sia arrivata una richiesta esplicita di flush della cache delle keywords (vedi più avanti)
- sia stato effettuato il cambio di provider di ricerca (in tal caso, tutte le cache devono essere azzerate)

ogni chiamata *effettiva* al provider di gif scelto dovrà incrementare il suo valore di `calls`; ogni interrogazione di una qualsiasi keyword - a prescindere dall'esistenza o meno di risultati e/o dall'effettiva interrogazione del provider - deve incrementare il contatore `calls` per la keyword.

i risultati devono essere presentati **esclusivamente** come array di stringhe:

```
{
  "results": [
    "http://example.com/animation1.gif",
    "http://example.com/animation2.gif",
    ...
  ]
}
```

il numero di risultati deve essere pari a 5 o uguale al valore settato mediante l'apposito comando di configurazione indicato più avanti.

## GET /gifs/{keyword}/stats

restituisce, per la keyword, il numero di chiamate raggruppate per provider nella seguente forma:

```
{
  "stats": {
    "provider-1": 123,
    "provider-2": 34543,
    ...
  }
}
```

se la keyword non è presente, deve essere restituita una response con status 404.

### **POST /provider/{identifier}**

setta il provider di ricerca da utilizzare. al cambio di provider, deve essere svuotata la cache dei risultati. la response deve essere senza body ed indicare status 204 nel caso di cambio effettuato con successo, altrimenti 404 nel caso di un provider non esistente.

### **comandi cli**

devono essere implementati i seguenti comandi per artisan (via riga di comando):

- set max results: indica un valore diverso da 5 per il totale di record da presentare
- flush cache dei risultati di ricerca
- set provider (con conseguente azzeramento della cache dei risultati)
- azzeramento dei conteggi di una keyword per tutti i providers

### **extra**

se riesci, crea anche un endpoint POST per effettuare il cambio di credenziali di un provider.