

Coding Challenge fashionette

JSON-API consuming 3rd party service TVMaze

The application returns information on TV shows, if found by search criteria, submitted to the service API TVMaze.

Only this route to JSON API is valid:

`/?q={SEARCH_VALUE}`

All other http request will be responded with an JSONError message and corresponding HTTP status 404 (file not found)

Cache

Each response will be stored in a cache (TTL is configured as 3600 seconds by default - see: `config/main.php`) to minimize external HTTP requests. If found in cache, the data for current search criterion will be read from cache instead of calling external API. Currently the cache is implemented as simple file system based key-value- store. Serialized data is indexed by (lower cased) search criterion as key.

Filtering

The search results are filtered follwing these criteria:

- non-typo-tolerant
- case-insensitive

There are two filter modes (to be set in `App\Models\Show`):

- CONTAINS (default) - the original name must contain search criterion ()
- EQUALS - the original name must be equl to search criterion

System requirements

- **PHP 7.3+**,
- **composer** and
- **git**

All other dependencies are managed by composer

Current environment:

- PHP 7.4.8 ^(PHP 7.3+ required)
- PHPUnit 8.5.8
- Lumen (7.2.1) (Laravel Components ^7.0)
- git (required) version 2.27.0
- Composer (required) version 1.10.9
- PhpStorm 2020.1 (IDE)

The application source code, building information and unit test is available as GitHub repository:

- <https://github.com/svenschrodt/tvm-json-api>

CI environment for repository can be found at Travis CI:

- <https://travis-ci.org/svenschrodt/tvm-json-api>

Install instructions

```
> git clone https://github.com/svenschrodt/tvm-json-api
> cd tvm-json-api
> composer install
> php artisan config:cache
```

Ideas on evolving API

- Implementing other cache mechanism, e.g: memcached for performance improvement
- Using translation services
- Mapping data structure for individual needs
- Using different Content-Types for service additional to JSON (SOAP, HTML, prop. XML, CSV etc.)
- Implementing automated frontend-testing (Selenium, Cypress ...)
- Extending CI Workflow (Code coverage testing, calculating metrics etc.)