Spatie/

Laravel-error-solutions

Display solutions on the Laravel error page

Introduction, How to use and Curiosities!

About me

Jhonatan Morais

Converting Ideas in Software
for more than one decade.
Passionate about creating solutions
and connecting with people.

- PHP, GoLang and Node/js
- Brazilian
- Gamer and Animes Fan

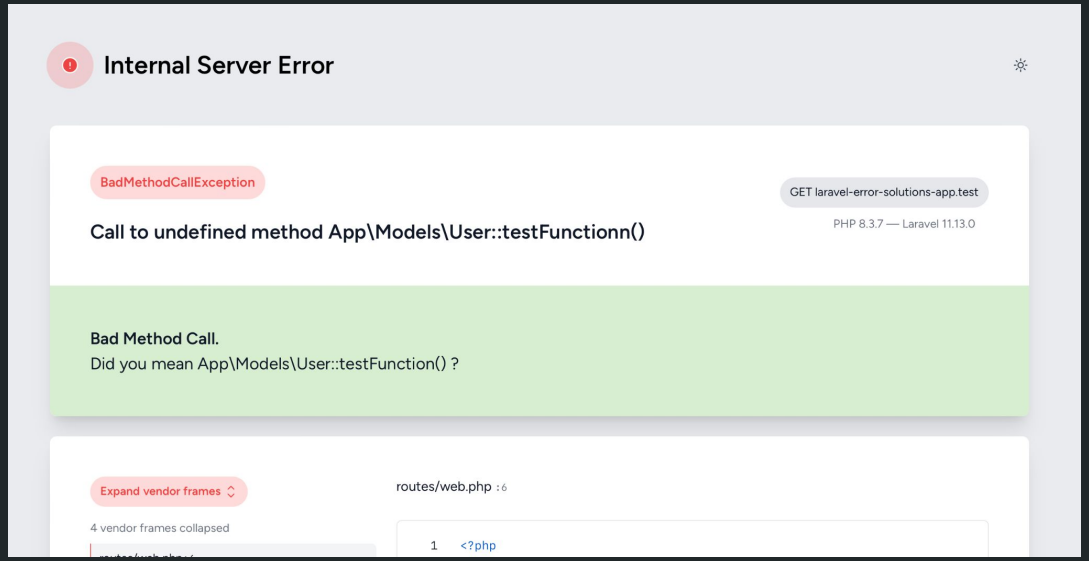
More about me:
jpmorais.com.br

Core topics!

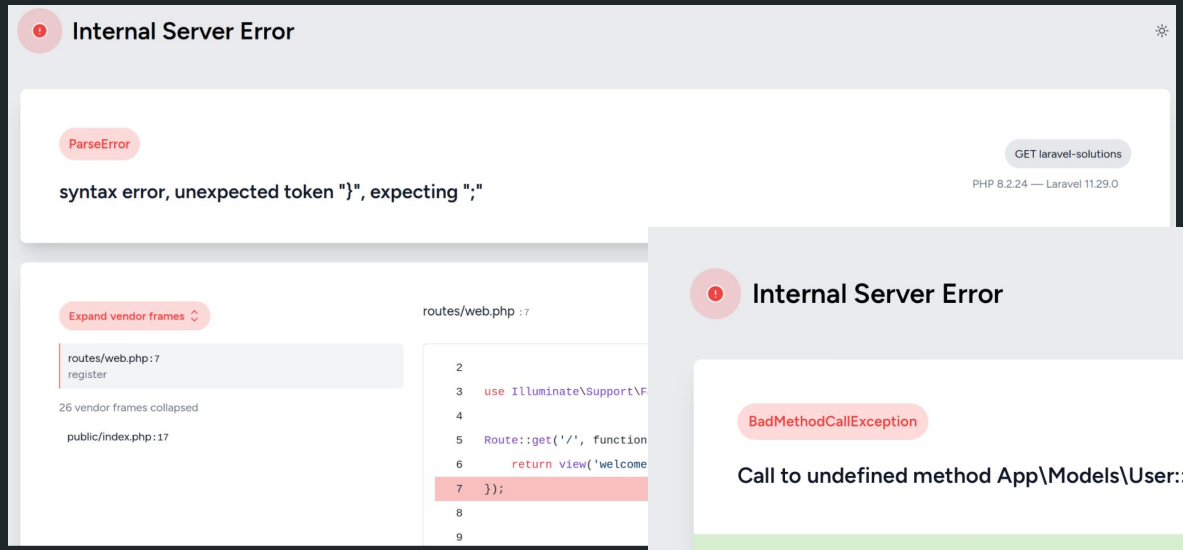
- Introduction to Laravel-solution-extension
- Features setup and configuration options
- How to extend the solutions
- Code Insights

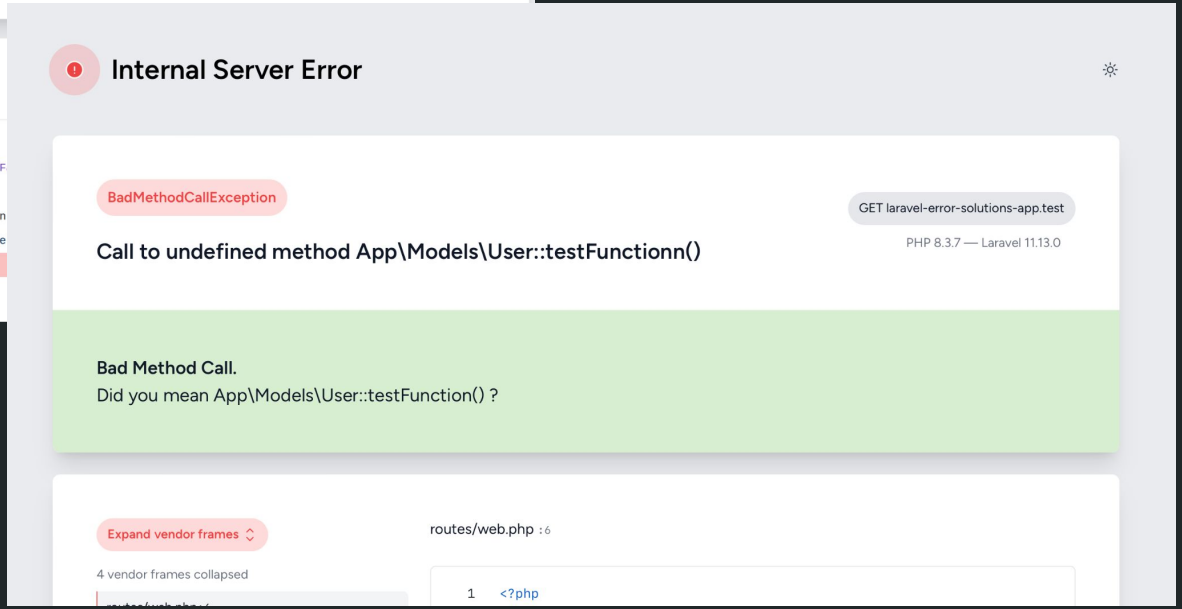
What is: Laravel-error-Solution

- Developed by Spatie
- Laravel 11 (php8.2)
- Aim to improve dev-exp
- 4 usage modes:
  - exception extension
  - custom solutions
  - command actions
  - AI support



Disable / Enable





App\Config\error-solution.php

Custom Solution

http://laravel-solutions/custom

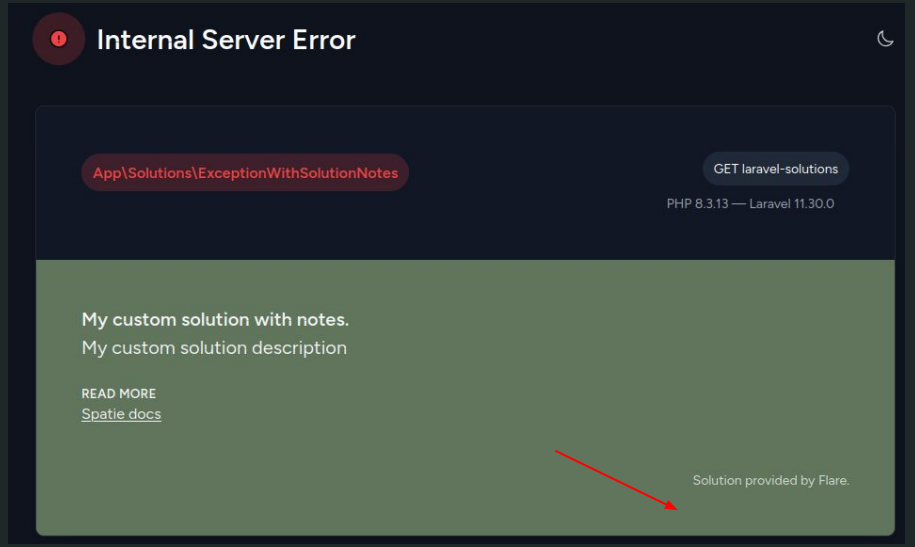
```php
use Spatie\ErrorSolutions\Contracts\Solution;

3 usages  Jhonatan Morais
class MySolution implements Solution
{
    no usages  Jhonatan Morais
    public function getSolutionTitle(): string
    {
        return 'My regular custom solution';
    }

    no usages  Jhonatan Morais
    public function getSolutionDescription(): string
    {
        return 'Description from a regular custom solution';
    }

    no usages  Jhonatan Morais
    public function getDocumentationLinks(): array
    {
        return [
            'Spatie docs' => 'https://spatie.be/docs',
        ];
    }

}
```

```php
<?php

namespace App\Solutions;

use Exception;
use Spatie\ErrorSolutions\Contracts\Solution;
use Spatie\ErrorSolutions\Contracts\ProvidesSolution;

 Jhonatan Morais
class ExceptionWithSolution extends Exception implements ProvidesSolution
{
    Jhonatan Morais
    public function __construct(string $message = '')
    {
        parent::__construct( message: $message ?? 'An custom exception');
    }

    no usages  Jhonatan Morais
    public function getSolution(): Solution
    {
        return new MySolution();
    }
}
```

Custom Solution + source

http://laravel-solutions/custom-with-notes

```php
class MySolutionWithNotes implements Solution
{
    public function getDocumentationLinks(): array
    {
        return [
            'Spatie docs' => 'https://spatie.be/docs',
        ];
    }


    no usages  Jhonatan Morais
    public function solutionProvidedByName(): string
    {
        return 'Flare';
    }

    no usages  Jhonatan Morais
    public function solutionProvidedByLink(): string
    {
        return 'https://flareapp.io';
    }
}
```

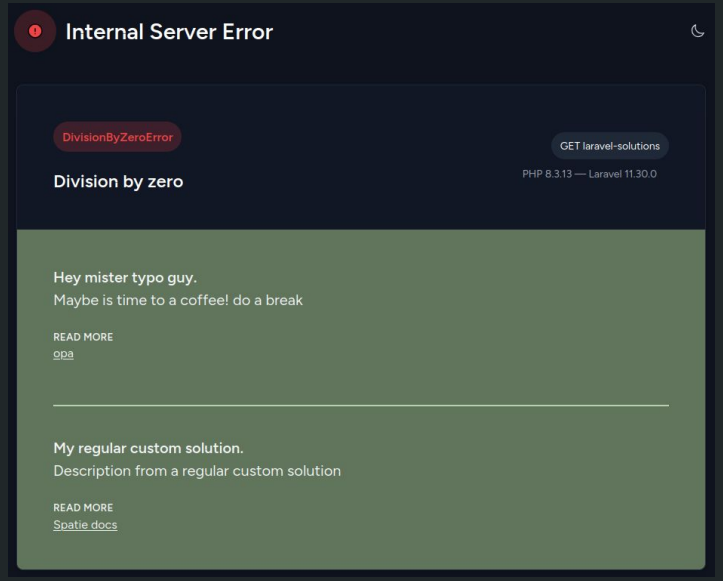


Solution Provider

http://laravel-solutions/custom-solution-provider

```php
1 usage  Jhonatan Morais
class CustomSolutionProvider implements HasSolutionsForThrowable
{
    no usages  Jhonatan Morais
    public function canSolve(Throwable $throwable): bool
    {
        // return true if you can provide a solution for this exception
        return $throwable->getMessage() === "Division by zero";
    }

    /**
     * @param \Throwable $throwable
     *
     * @return array<int, Solution>
     */
    no usages  Jhonatan Morais
    public function getSolutions(Throwable $throwable): array
    {
        // return an array of solutions
        return [
            new MySyntaxSolution(),
            new MySolution() // You can offer multiple solutions...
        ];
    }
}
```

\App\Providers



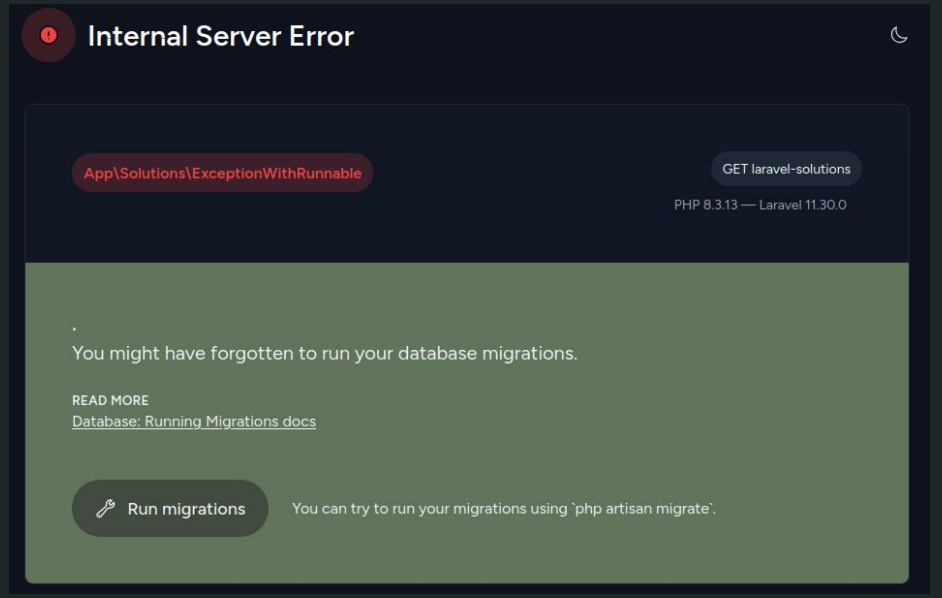
config/error-solutions.php

```php
    */
    'solution_providers' => [
        'php',
        'laravel',
        // your solution provider classes
        \App\Providers\CustomSolutionProvider::class,
    ],
```

Runnable Solution

http://laravel-solutions/runnable-solution

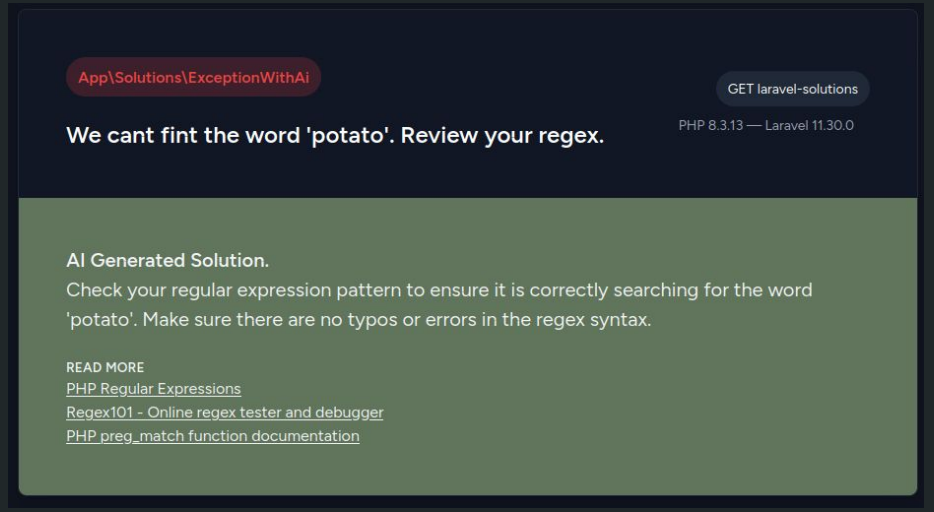
```php
class MyRunnableSolution implements RunnableSolution
{
    2 usages
    protected string $customTitle;
    Jhonatan Morais
    public function __construct(string $customTitle = ''){...}
    no usages  Jhonatan Morais
    public function getSolutionTitle(): string{...}
    no usages  Jhonatan Morais
    public function getSolutionDescription(): string{...}
    no usages  Jhonatan Morais
    public function getDocumentationLinks(): array{...}
    no usages  Jhonatan Morais
    public function getRunParameters(): array
    {
        return [];
    }
    2 usages  Jhonatan Morais
    public function getSolutionActionDescription(): string
    {
        return 'You can try to run your migrations using `php artisan migrate`.';
    }
    2 usages  Jhonatan Morais
    public function getRunButtonText(): string{...}
    Jhonatan Morais
    public function run(array $parameters = []): void
    {
        Artisan::call('migrate');
    }

}
```



AI generated Solution

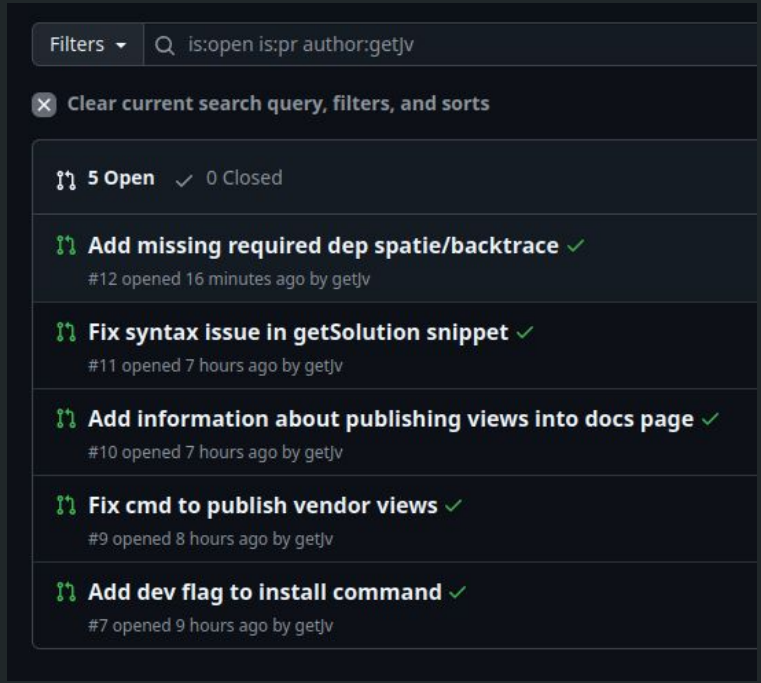
http://laravel-solutions/ai-solution

```php
class MySolutionWithAi implements Solution
{
    public function getDocumentationLinks(): array{...}

    1 usage  Jhonatan Morais *
    public function getAiSolution(): ?OpenAiSolutionResponse
    {
        $solution = $this->cache->get($this->getCacheKey());
        if ($solution) {...}

        $solutionText = OpenAI::client($this->openAiKey)
            ->chat()
            ->create([
                'model' => $this->getModel(),
                'messages' => [['role' => 'user', 'content' => $this->prompt]],
                'max_tokens' => 1000,
                'temperature' => 0,
            ])->choices[0]->message->content;

        $this->cache->set($this->getCacheKey(), $solutionText, $this->cacheTtlInSeconds);
        return new OpenAiSolutionResponse($solutionText);
    }

    1 usage  new *
    protected function generatePrompt(): string
    {
        $viewPath = __DIR__.'/../../resources/views/aiPrompt.php';

        $viewModel = new OpenAiPromptViewModel(
            file: $this->throwable->getFile(),
            exceptionMessage: $this->throwable->getMessage(),
            exceptionClass: get_class($this->throwable),
            snippet: "fake-snippet", //$this->getApplicationFrame($this->throwable)->getSnippetAsStri
            line: $this->throwable->getLine(),
            applicationType: $this->applicationType,
        );

        return (new AiPromptRenderer())->renderAsString(
            ['viewModel' => $viewModel],
            $viewPath,
        );
    }
```



Code Insights

- Community PRs :D
- Importance of dependencies
  spatie/ray -> backtrace
- php-dependencies ->
  Laravel-dependencies
- https://github.com/spatie/backtrace
- https://github.com/spatie/laravel-ray



Thank you!

Follow me!

https://www.qrcode-monkey.com/