Laravel rescue() helper

rescue()

rescue() es un helper de Laravel que ejecuta una función closure (función anónima en PHP) y detecta cualquier excepción durante su ejecución. Las excepciones se enviarán a su controlador de excepciones, pero la solicitud continuará siendo procesada.

Requisitos

- Laravel (el helper existe desde versiones antiguas; cualquier 8.x/9.x+ lo trae).
- Un caso donde una operación puede fallar sin que deba tumbar la petición (en el ejemplo, comprobar si un fichero existe en un CDN que puede estar caído).

Ejemplo

```
// Viejo método - try / catch ignorando la excepción. Un poco feo
private static function existsOnCDN(string $path): bool
{
  $cdn = $false;

  try {
    $cdn = Storage::disk('cdn')->exists($path);
  } catch (\Exception $e) {
    // CDN no esta disponible por problemas de red.
  }
  return $cdn;
}

// Mas claro, rescue() ignora la excepcion y permite al código continuar. Opcionalmente
podemos pasar un valor de retorno
```

```
private static function existsOnCDN(string $path): bool
{
  return rescue(fn () => Storage::disk('cdn')->exists($path), false);
}
```

Cómo funciona: `rescue(fn () => ..., false)` ejecuta la closure y, si lanza, reporta la excepción al handler y devuelve el segundo argumento (`false` aquí) en lugar de romper la ejecución.

Agradecimientos

A @shawnlindstrom por su tuit

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Verificación

- Un test con el disco `cdn` falseado para lanzar: `rescue(..., false)` debe devolver `false` sin propagar la excepción, y el test debe terminar en verde.
- El segundo argumento es opcional: sin él, el valor de retorno en fallo es `null`; comprobar cuál de los dos comportamientos espera el llamante antes de decidir.

Revision #3
Created 2022-02-09 18:39:26 UTC by Abkrim
Updated 2026-11-13 17:11:15 UTC by Abkrim