

Supercharged Laravel Debugging

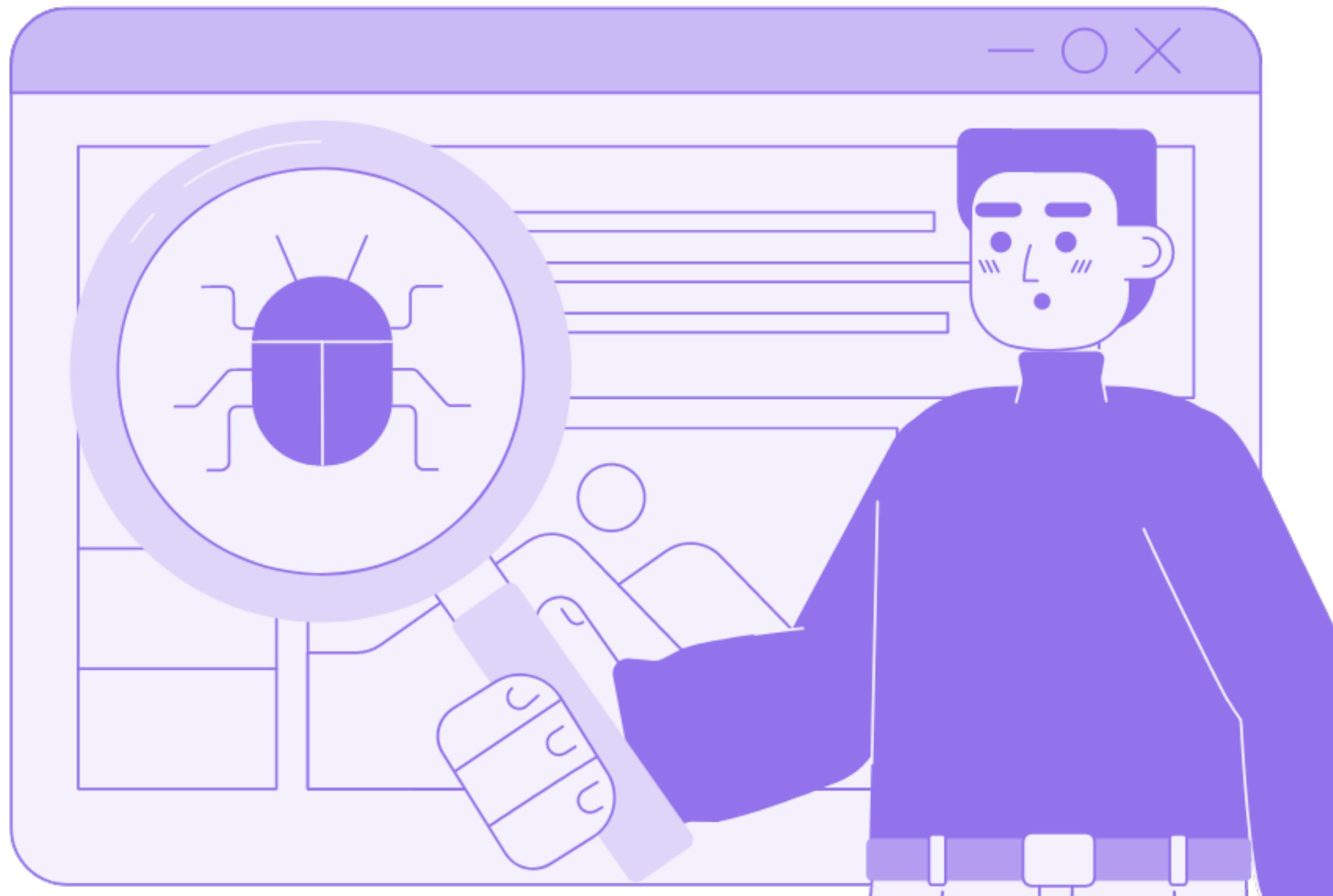
The challenges of every software development story!

PROFESSIONAL LARAVEL DEVELOPMENT

Debugging mindset

The fundamentals of the battle

- Computer is not doing the mistake.
- With proper effort and patience, it can be solved.
- Don't hesitate to seeking help / handover.



Battle Plan



Meilisearch\Exceptions\ApiException

PHP 8.2.16 10.48.4

Attribute `created_at` is not sortable. This index does not have configured sortable attributes.

Expand vendor frames

8 vendor frames ▾

App\Livewire\AllCourseView:67
render

9 vendor frames ▾

resources/views/livewire/all-course.blade.php:23
include

12 vendor frames ▾

resources/views/home/all_courses.blade.php:5
require

15 vendor frames ▾

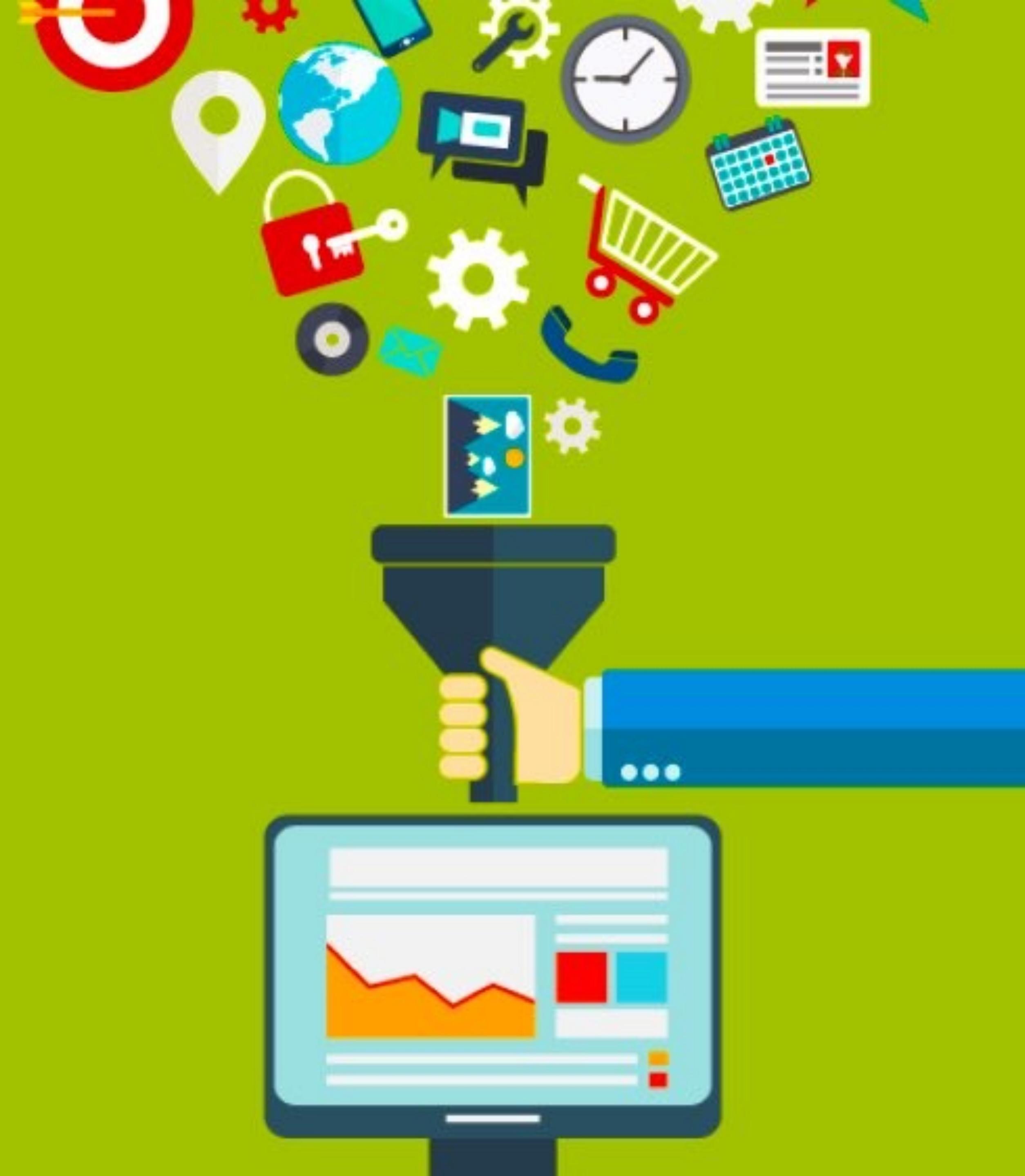
App\Http\Middleware\Localization:26
handle

app/Livewire/AllCourseView.php:67

```
52         return $this->query = Course::search($this->search);
53     }
54
55     $this->query = Course::query()->with(['categories', 'enrollments', 'teachers'])->published(
56
57     }
58
59     public function render(): View
60     {
61         $items = $this->query->get();
62
63         if ($items->count() == 0) {
64             return view('livewire.no-course-found');
65         }
66
67         $this->courses = $this->query->latest()->get();
68
69         if ($this->view == ViewStyle::GRID->value) {
```

Collect information and observe carefully

- Reading error messages
- Get screenshot, video, logs, whatever
- Watch application Logs
- Watch fpm and web server log



Understand the Context and presume the origin

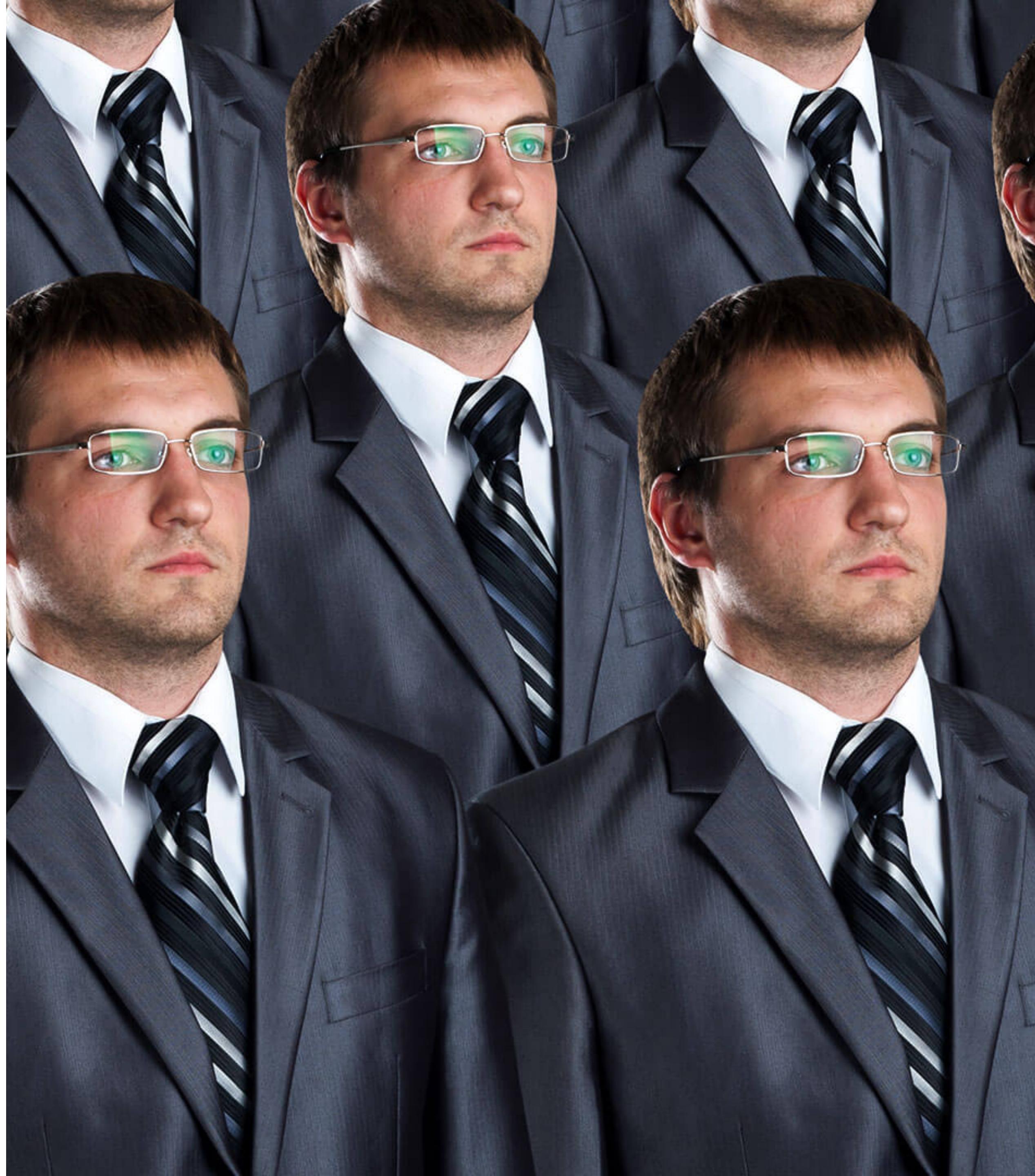
- Who (someone / a few / everyone)
- When (always / sometimes)
- Reproducibility (regular / unpredictable)



Reproduce

It's 50% of the resolution

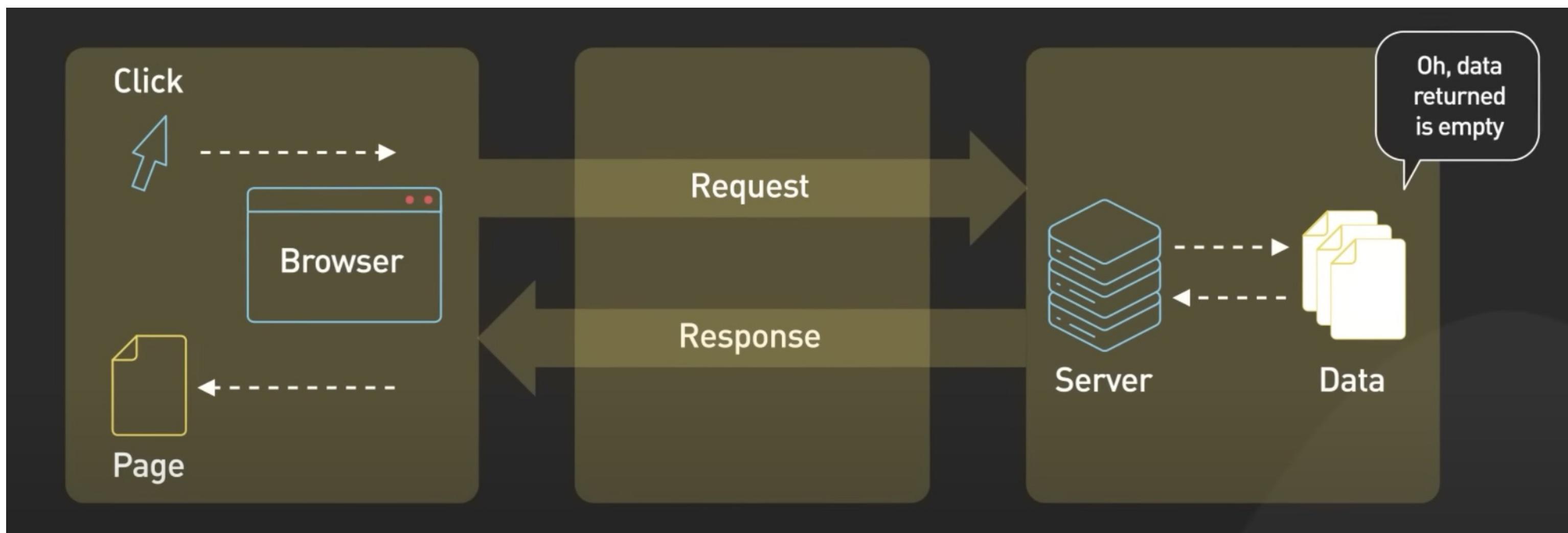
- With appropriate prerequisite
- Write failing test with expectation



Isolation

Identify the stages of execution flow

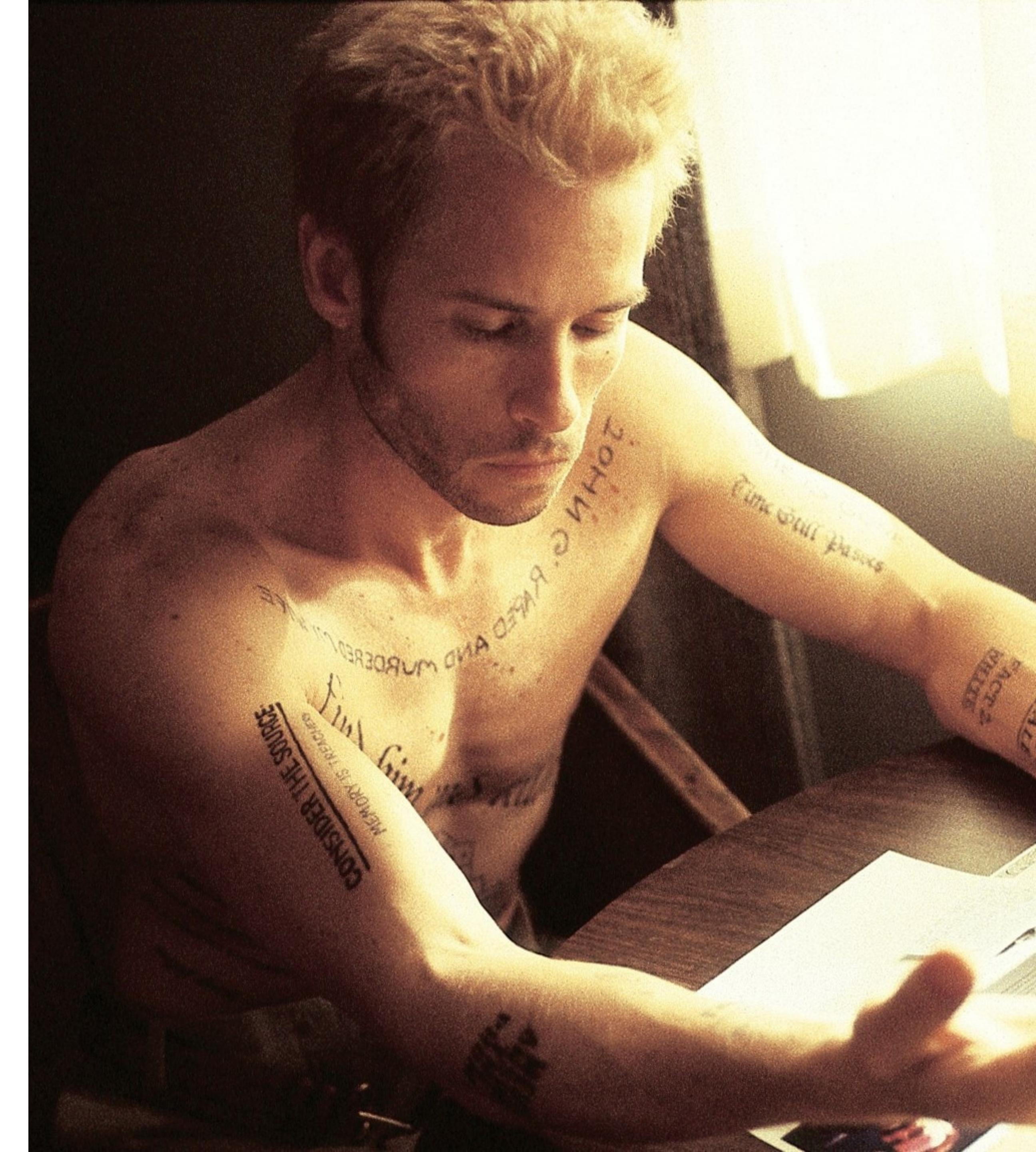
- Inspect the transition between stages
- Inspect inside the stages



Memento

Take note of findings

- Quickly reach to the flagged spot
- Faster to onboard team member
- Easy to Handover
- Easy to seek for external help



Rubber Duck Debugging



Communicate...

(It's helpful most of the times)





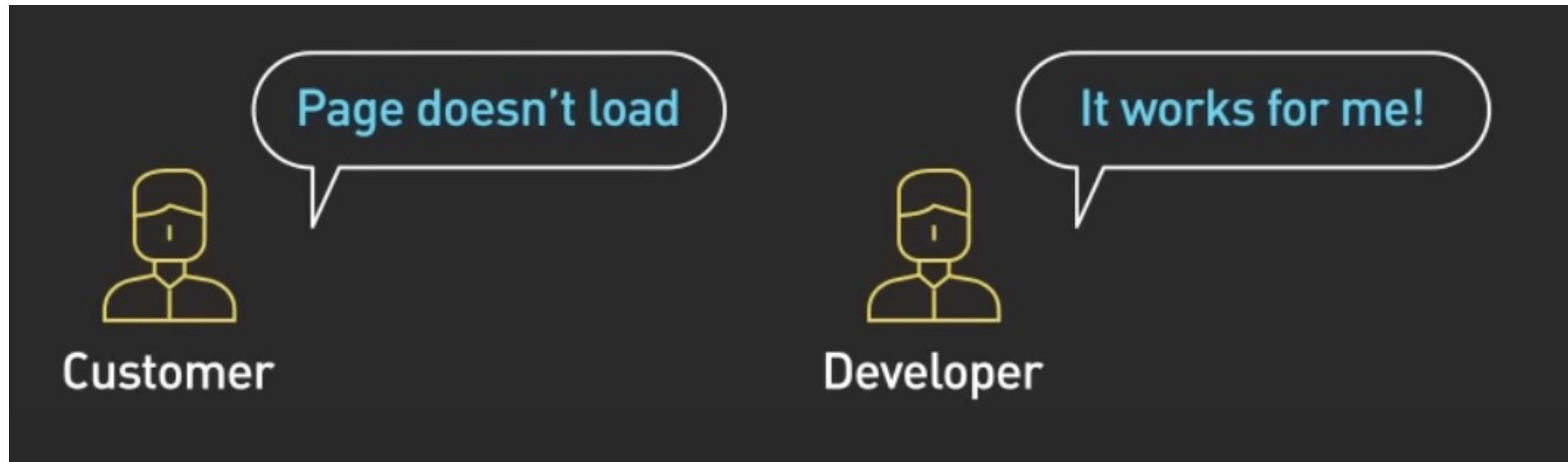
ME

Backing out the changes

- Just get me out of here..



Hard to trace issues



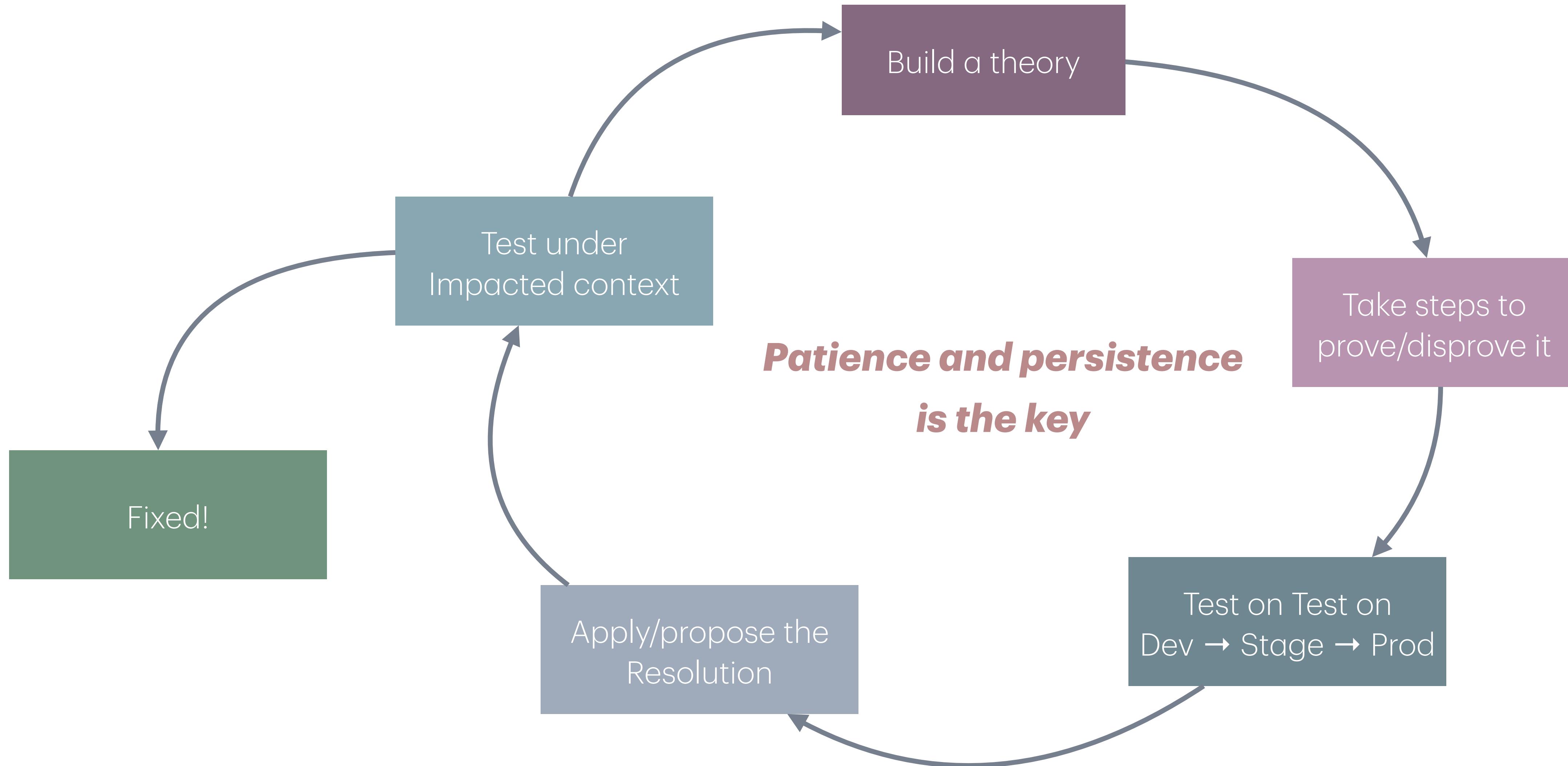
Hard to trace issues

Issue produces for...

- Specific environment
- Specific person
- Specific time slot (day/month/year)
- At certain workload
- (Maybe) in a Race condition

Hard to trace issues

How to deal?



The field of Laravel



- ▶ Identifying the journey
- ▶ Things that are not always in front of eyesight
 - ▶ Middleware
 - ▶ Event Listeners
 - ▶ Model observers
- ▶ Logging: Utilizing Laravel's logging features - Log::info(), Log::debug(), with context.
- ▶ Debugging using dump(), dd(),

Dealing with SQL / Eloquent

- ▶ Model::query() -> dd()
- ▶ Model::query() ->toSql()
- ▶ Debugging Queries
 - ▶ DB::enableQueryLog()
 - ▶ DB::getQueryLog()
- ▶ Debugging using dump(), dd(),

```
// Add in boot function
DB::listen(function($query)
    Log::info(
        $query->sql,
        $query->bindings,
        $query->time
    );
);
```

Utilising Logs for inspection

Laravel Log Levels

```
Log::shareContext(['request-id' => $requestId]);  
Log::emergency($message);
```

```
Log::alert($message);
```

```
Log::critical($message);
```

```
Log::error($message);
```

```
Log::warning($message);
```

```
Log::notice($message);
```

```
Log::info($message);
```

```
Log::debug($message);
```

Log Context

```
Log::shareContext(['request-id' => $requestId]);
```

Write to specific Channel

```
Log::channel('slack')->info('Something happened!');
```

Write to on demand Channel

```
Log::build([  
    'driver' => 'single',  
    'path' => storage_path('logs/custom.log'),  
)->info('Something happened!');
```

Logging without throwing

```
catch (Throwable $e) {  
    report($e);  
    return false;  
}
```

Tools

Laravel Pail

```
php artisan pail  
→ php artisan pail  
  
INFO Tailing application logs. Press Ctrl+C to exit  
Use -v|-vv to show more details  
  
13:31:54 INFO —————  
This is some useful information.  
————— GET: / · Auth ID: guest  
  
13:31:54 Exception ————— routes/web.php:19  
Error Processing Request  
————— GET: / · Auth ID: guest  
  
13:31:54 WARNING —————  
Something could be going wrong.  
————— GET: / · Auth ID: guest
```

Tools

Laravel Top

HTTP		Database		Cache	
Requests/sec	141.20	Queries/sec	353.80	Hit/sec	108.00
Avg. Memory(MB)	30.86	Avg. Exec. Time(ms)	8.19	Miss/sec	9.60
Avg. Response Time(ms)	44.67			Write/sec	71.40
Method	URI	Avg. Memory(MB)	Avg. Resp. Time(ms)	Req./sec	
GET	/user/profile	29.56	55.12	35.60	
POST	/login	34.78	34.97	29.00	
GET	/products/{id}	35.16	44.26	18.40	
GET	/notifications	22.41	35.13	17.20	
PATCH	/categories	30.71	45.48	9.60	
GET	/categories/{category}/products/{product}	28.86	35.07	8.40	
GET	/settings	30.76	44.55	8.40	
POST	/user/profile	39.66	62.78	8.20	
DELETE	/messages/{id}	23.00	45.21	6.60	

Tools

Laravel Debugbar

The screenshot shows the Laravel Debugbar interface. At the top, there's a navigation bar with tabs: Messages (4), Timeline, Exceptions, Views (1), Route, Queries (3), Models (0), Mails, Gate, Session, Request. To the right of the tabs, it displays: GET /, 1MB, 22.48ms, 8.2.16, a dropdown, and a close button. Below the navigation bar, there's a list of messages:

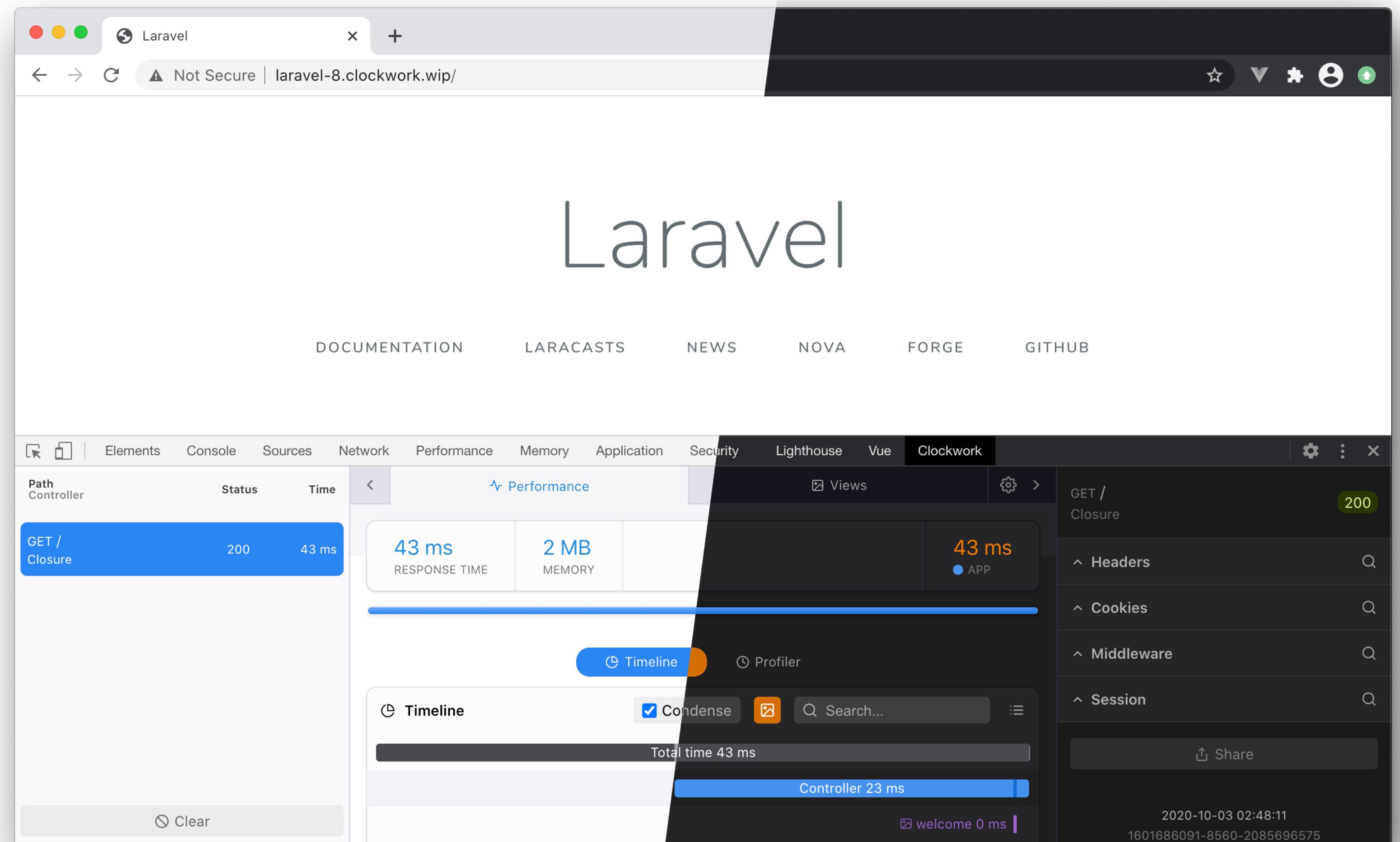
- INFO**: array:2 [▼ "name" => "Barry vd. Heuvel", "email" => "barry@fruitcake.nl"] (web.php#8)
- ERROR**: Error! (web.php#12)
- WARNING**: Watch out... (web.php#13)
- MYLABEL**: Another message (web.php#14)

At the bottom, there are buttons for MYLABEL (gray), WARNING (orange), ERROR (red), and INFO (blue). A search bar with a magnifying glass icon and the placeholder "Search" is at the very bottom.

<https://github.com/barryvdh/laravel-debugbar?tab=readme-ov-file>

Tools

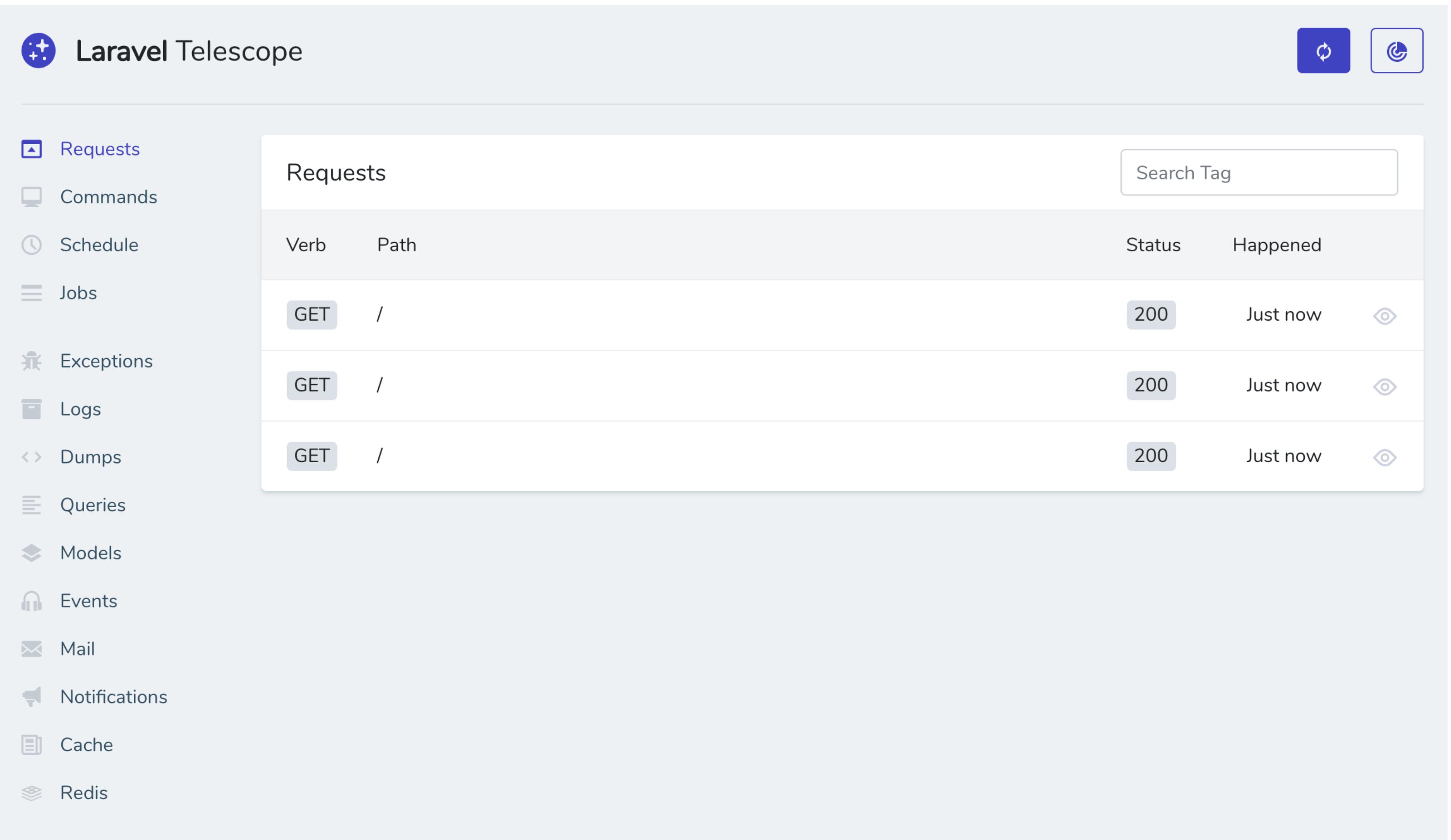
Clockwork



<https://github.com/itsgoingd/clockwork>

Tools

Laravel Telescope



The screenshot shows the Laravel Telescope dashboard. On the left, a sidebar lists various monitoring categories: Requests, Commands, Schedule, Jobs, Exceptions, Logs, Dumps, Queries, Models, Events, Mail, Notifications, Cache, and Redis. The 'Requests' category is selected and expanded, showing a table titled 'Requests'. The table has columns for Verb, Path, Status, and Happened. Three recent requests are listed:

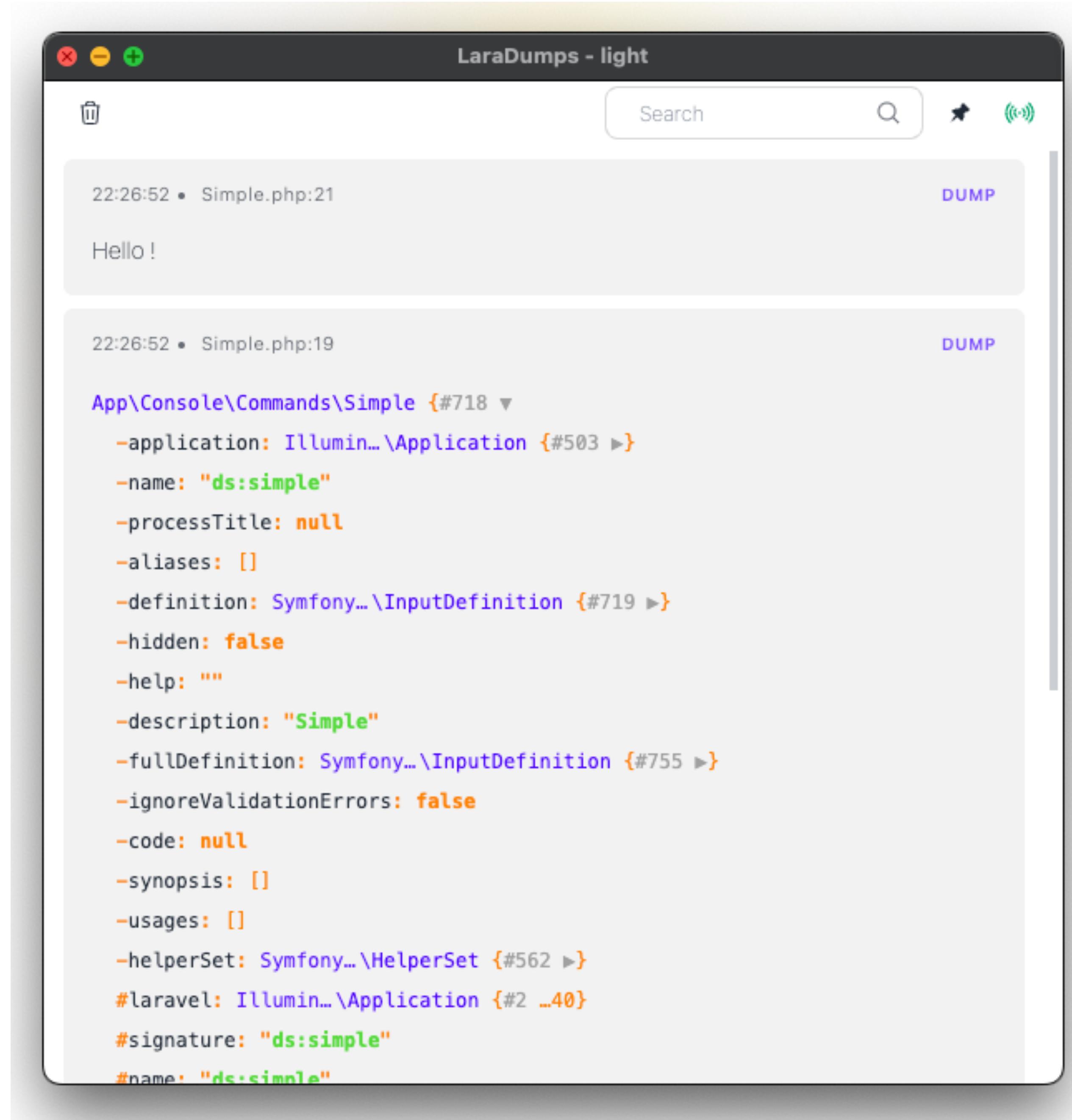
Verb	Path	Status	Happened
GET	/	200	Just now
GET	/	200	Just now
GET	/	200	Just now

On the right side of the dashboard, there are two small blue buttons: a refresh icon and a search icon. A search bar labeled 'Search Tag' is also present.

<https://github.com/laravel/telescope>

Tools

LaraDumps



<https://laradumps.dev/>

External Services

Sentry

The screenshot shows the Sentry Issues interface. On the left is a dark sidebar with navigation links: Issues (selected), Projects, Performance, Profiling, Replays, Cron, Alerts, Discover, Dashboards, Releases, User Feedback, Stats, Settings, Help, and What's new.

The main area is titled "Issues" and has tabs for Unresolved, For Review, Regressed, Escalating, and Archived. It includes filters for react, All Envs, 24H, Custom Search (is:unresolved), and sorting by Last Seen.

The list of issues includes:

- SyntaxError** json([native code])
The string did not match the expected pattern.
New REACT-3TF Unhandled 25min ago | 6d old > 50+
components/Checkout.js in call at line 58:17
53 },
54 }).catch((err) => {
55 return { ok: false, status: 500 };
56 });
57 if (!response.ok) {
58 throw new Error(
59 [response.status, response.statusText || 'Internal Server Error'].join(
60 ' - '
61)
62);
500 - Internal Server Error
New REACT-3CJ 26min ago | 4mo old DTP-76
- Error** call(components/ProductsJoin)
app unable to load products
New REACT-3CJ 26min ago | 4mo old DTP-76

A tooltip is shown over the stack trace line 58, providing options to "Open this line in GitHub" and "Open in Codecov".

Below the list are metrics: GRAPH, 24h, EVENTS, USERS, and ASSIGNEE, with values 240, 3.4k, 1.4k, and user icons respectively.

<https://github.com/getsentry/sentry-laravel>

External Services

Flare

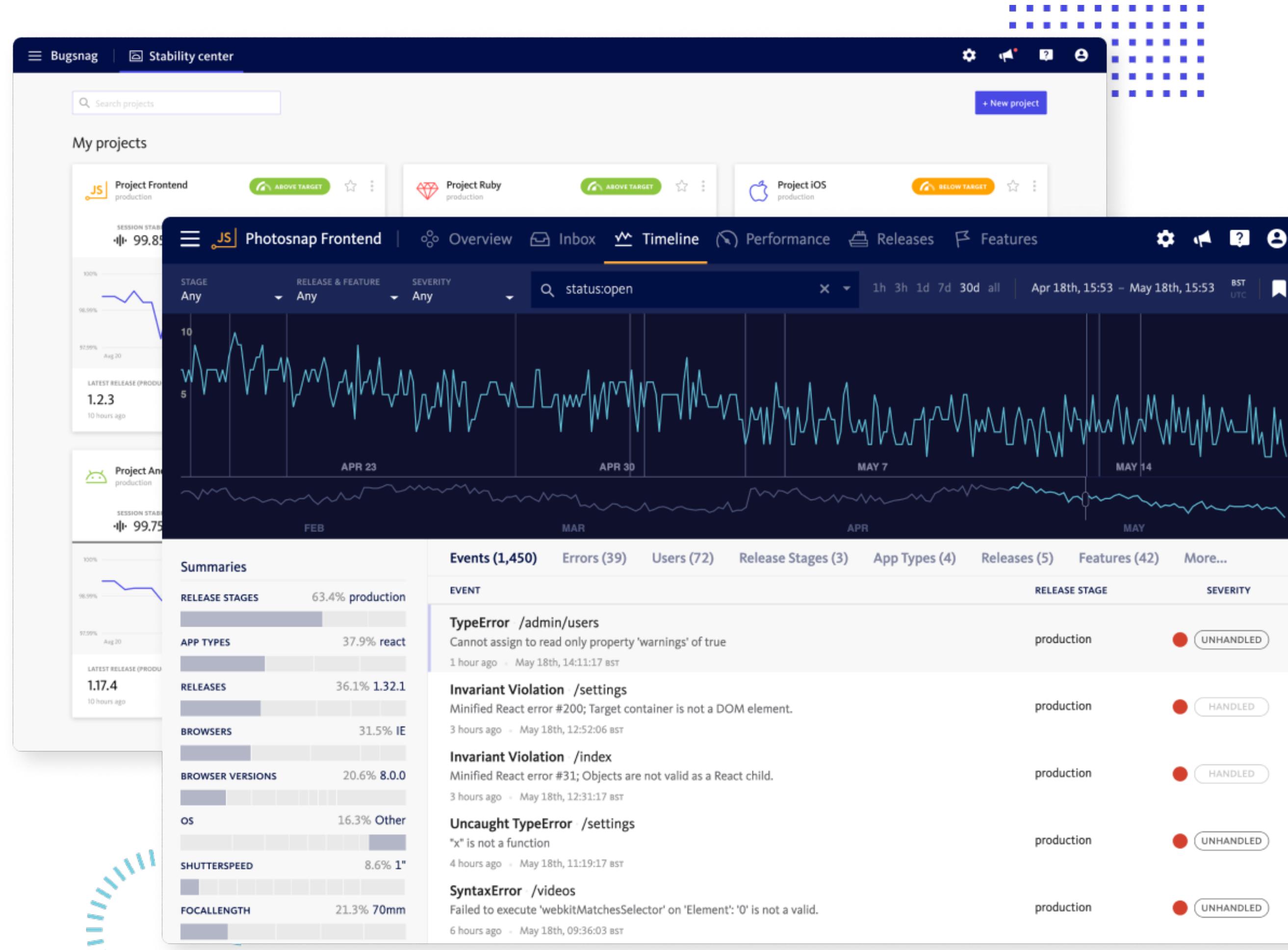
The screenshot shows the Flare application interface for the project "Titanic.app API PROD". The main view displays an error entry for a "QueryException" from the "UserRepository". The error message is: "Illuminate\Database\QueryException UserRepository SQLSTATE[HY000] [2002] Connection refused select * from `posts` where `published` = 1 order by `publish_date` desc, `id` desc, `publish_date` desc limit 20 App\Domains\Housekeeping\Jobs\SendSosJob". To the right of the error details is a bar chart showing the occurrence of the error over time, with 11 occurrences from February 5, 2023, to May 11, 2023. Below the chart are filters for "Occurrences" (11), "First occurred" (2023-02-05), "Last occurred" (2023-05-11), "Users" (None), "Type" (Job), and "Environment" (production). At the bottom left, there are insights: "Jobs" (Occurred in 7 jobs), "Versions" (4 affected), "Users" (2 affected), and "Type" (Received from 2 sources). At the bottom center, there is a detailed view of the error occurrence #11, which happened 2 hours ago. This view includes the error stack trace, PHP version (8.2.0), MySQL version (10.9.0), and an AI-generated solution: "To connect to MySQL using Laravel Sail, use port 33066 instead of the default MySQL port. To do this, set the DB_PORT environment variable to 33066." A note states: "This solution was generated by AI and might not be 100% accurate." At the bottom right, the code for the UserRepository.php file is shown:

```
public function getUsers(string $email = null): Collection
{
    return $this->getQueryBuilder()
        ->active()
```

<https://flareapp.io/>

External Services

Bugsnag



<https://www.bugsnag.com/>

Questions?

