


Best Practices & "The marqably way to do things"

General

- Create ClickUp Tasks for `todo`s and `fixme`s: 
- Always use Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-2780>) for logging. Don't use it for debugging, there are better logging drivers for this purpose, but rather for production logging. This will help identify performance bottlenecks and bugs that made it to production. In a perfect world those things are fixed before pushing to prod, but sometimes it requires production workloads to notice issues, especially when it comes to performance. As a general rule of thumb it's better to log more than too little, but don't overdo it.
- Always use Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-14775>) for code quality analysis, insights and suggestions. Generating insights and suggestions for improving code quality and reducing complexity is a very good idea, in general. One can only benefit from it.
- Consider providing a [Makefile](#) with phony targets as a "task runner". This makes it very easy to get autocompletion in the terminal for commonly used tasks, e.g.

```
.PHONY: migrate-fresh
migrate-fresh:
    ./backend/vendor/bin/sail exec laravel php artisan qibri:migrate:fresh

.PHONY: attach
attach:
    ./backend/vendor/bin/sail exec laravel bash
```

This might be helpful for simple one-line tasks but can not replace the `mqbench` script. Using a makefile is very handy to shortcut long repetitive commands.

Laravel

- Use Laravel Sail or Laravel Herd for your local development environment. Older projects might use a different setup, but for new ones the preferred stack is **Laravel Sail** (<https://laravel.com/docs/master/sail>).
- Try to not over-engineer API abstraction using **DTOs**. Use them when necessary and meaningful (probably in large projects), and avoid them when introducing DTOs would be

overkill (small projects).

- Use packages created by Laravel for solving common problems, extending the packages with custom drivers as needed:
 - Websockets: Laravel Reverb
 - Payments: Laravel Cashier
 - Browser Testing: Laravel Dusk
 - Application Performance Monitoring (APM): Laravel Pulse
 - OAuth2 Authentication: Laravel Passport
 - Token-based authentication: Laravel Sanctum
 - Feature Flags: Laravel Pennant
 - Full-text search: Laravel Scout
 - Third-party authentication: Laravel Socialite
 - Development insights: Laravel Telescope
- Use the translation framework even if you currently don't have translations: ☐
- Use the **PSR-12** standard for formatting: ☐

Extension packages used at marqably

- Sitemap: ?
- SEO: ?
- Admin Panel builder: Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-3676>)
- Slugs: Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-897>)
- GraphQL: Lighthouse, Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-15755>)

Laravel Packages

- Always include a license in your Laravel packages: ☐

Code Quality & Formatting

- Use Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-14775>) and Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-15815>) for static analysis and code

quality improvements.

- Use Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-15795>) for automatically formatting code.
- Please check out Private (<https://app.clickup.com/2457460/docs/2azvm-512/2azvm-15475>) for further details.