# Enterprise Console Development Environment Setup

This document defines the authoritative workstation provisioning process for the Enterprise Console project.

## 1. Hardware & OS Preparation

1. Windows 10/11 64-bit updated.
2. Enable virtualization (for WSL2/Docker future use).
3. Admin-capable local user.

## 2. Core System Prerequisites

1. Install Microsoft Visual C++ Redistributables (latest x64 bundle).
2. Install Git for Windows (use credential manager, checkout CRLF/commit LF).
3. (Optional) Windows Terminal from Store.
4. Configure global git identity:

* `git config --global user.name "YOUR NAME"`
* `git config --global user.email you@example.com`
* `git config --global init.defaultBranch main`

## 3. PHP & Stack (Laragon)

1. Install Laragon (choose PHP 8.3 build aligning with composer platform).
2. Ensure extensions: openssl, mbstring, tokenizer, xml, ctype, json, fileinfo, curl, pdo\_sqlite (and pdo\_mysql if needed), opcache (optional).
3. Confirm `php -v` in new PowerShell session.

## 4. Node.js Toolchain

1. Install Node.js LTS (18.x or 20.x) from nodejs.org.
2. Verify `node -v`, `npm -v`.

## 5. Global CLI Tools

1. Composer (bundled with Laragon) `composer -V`.
2. Laravel Installer (optional): `composer global require laravel/installer`.
3. GitHub CLI: `winget install --id GitHub.cli`.
4. (Optional) ncu: `npm install -g npm-check-updates`.

## 6. VS Code Installation

1. Install VS Code 64-bit.
2. Enable Settings Sync (GitHub/Microsoft sign-in).
3. Key settings (User Settings JSON):

```

{

"editor.formatOnSave": true,

"files.eol": "\n",

"php.validate.executablePath": "C:\\laragon\\bin\\php\\php-8.3.x\\php.exe",

"terminal.integrated.defaultProfile.windows": "PowerShell"

}

```

1. (Optional) Set default formatter for JS/TS when adding ESLint/Prettier.

## 7. VS Code Extensions

1. PHP Intelephense (bmewburn.vscode-intelephense-client)
2. Laravel Artisan (ryannaddy.laravel-artisan)
3. Laravel Blade Formatter (shufo.vscode-blade-formatter)
4. Blade Snippets (onecentlin.laravel-blade)
5. Laravel Extra Intellisense (amiralizadeh9480.laravel-extra-intellisense)
6. Laravel Pint (open-sourcing.laravel-pint)
7. Tailwind CSS IntelliSense (bradlc.vscode-tailwindcss)
8. PostCSS Language Support (csstools.postcss)
9. GitLens (eamodio.gitlens)

10. GitHub PR (GitHub.vscode-pull-request-github)

11. DotENV (mikestead.dotenv)

12. Error Lens (usernamehw.errorlens)

13. Path Intellisense (christian-kohler.path-intellisense)

14. Markdown All in One (yzhang.markdown-all-in-one)

15. Optional: PHP Namespace Resolver, Code Spell Checker, Indent-Rainbow

## 8. Clone & Project Bootstrap

1. `cd C:\laragon\www`
2. `git clone <repo-url> enterprise-console`
3. `cd enterprise-console`
4. `copy .env.example .env`
5. `composer install`
6. `npm install`
7. `php artisan key:generate`
8. Ensure SQLite file: `if not exist database\database.sqlite type nul > database\database.sqlite`
9. Migrate (if needed): `php artisan migrate`

10. (Optional) `php artisan db:seed`

11. Symlink storage: `php artisan storage:link`

## 9. Development Scripts

1. Start concurrent dev: `composer run dev` (serve, queue:listen, pail, vite).
2. Quality full: `composer run check-all` (pint + phpstan + tests).
3. Static analysis: `composer run stan`.
4. Tests: `php artisan test` (Pest + PHPUnit).
5. Build assets: `npm run build`.

## 10. Frontend Pipeline

1. Tailwind config: `tailwind.config.js`.
2. Vite dev server: `npm run dev`.
3. Production build: `npm run build` -> outputs to `public/build`.
4. Alpine.js + axios loaded via `resources/js` entrypoints.

## 11. Auth & Security Components

1. Sanctum for SPA/token auth (set SANCTUM\_STATEFUL\_DOMAINS if SPA).
2. Socialite providers in `config/services.php` (add keys to `.env`).
3. Spatie Permission: publish/migrate if not yet: `php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"`.

## 12. Observability

1. Pail for real-time logs (already in dev script).
2. Telescope (local only): `php artisan telescope:install && php artisan migrate`.
3. Restrict Telescope via gate or environment.

## 13. Quality Automation

1. Code style: `vendor\bin\pint` or `composer run lint`.
2. Pre-commit hook example (`.git/hooks/pre-commit`):

```

#!/usr/bin/env bash

composer run check-all || exit 1

```

1. Incrementally raise phpstan level (currently 6).

## 14. Environment Variables (Core)

1. APP\_KEY (generated)
2. APP\_ENV=local, APP\_DEBUG=true
3. DB\_CONNECTION=sqlite (or mysql credentials)
4. QUEUE\_CONNECTION=database (or redis)
5. CACHE\_DRIVER=file (or redis)
6. MAIL\*\\*, SANCTUM\*\_, SESSION\\_\_

## 15. Optional: Sail / Docker

1. Enable Sail (if you want a containerized stack):

* `php artisan sail:install` (choose: mysql/redis/meilisearch as needed)
* `php artisan sail:publish` (optional to customize Dockerfiles)

1. Start environment (Linux/macOS): `./vendor/bin/sail up -d`
2. Windows (PowerShell): `bash ./vendor/bin/sail up -d` (from WSL) or use Git Bash.
3. Executing commands:

* `./vendor/bin/sail artisan migrate`
* `./vendor/bin/sail npm run dev`

1. Performance tips (Windows): prefer WSL2 for volume mounts; put project inside `\\wsl$` distro filesystem for speed.
2. If adopting Sail exclusively, stop Laragon services to avoid port conflicts (80, 3306, 6379).
3. Example override `docker-compose.override.yml` (create at project root):

```

services:

laravel.test:

environment:

PHP\_IDE\_CONFIG: "serverName=enterprise-console"

volumes:

* ./:/var/www/html:delegated

mysql:

command: --default-authentication-plugin=mysql\_native\_password --max-connections=250

```

1. Running quality inside Sail:

* `./vendor/bin/sail composer run check-all`
* `./vendor/bin/sail php artisan test`

1. Rebuild after PHP extension changes: `./vendor/bin/sail build --no-cache`

## 16. Verification Checklist

1. `php artisan about` OK
2. `composer run check-all` passes
3. `npm run build` succeeds
4. App loads at http://localhost:8000 or Laragon vhost
5. Queue processing works
6. Permissions tables present

## 17. Maintenance

1. Monthly: `composer outdated`, `npm outdated`.
2. Run pint before PR.
3. Add CI workflow (future) replicating dev scripts.

## 18. Regenerating Documentation

1. Update this file as needed.
2. Convert to DOCX using Python helper (see below).
3. CI Automation: A GitHub Action can auto-regenerate `environment-setup.docx` when this markdown changes. See workflow file `regenerate-env-setup-docx.yml` once added.

## 19. Matrix Summary

### 19.1 Platform & Core Tooling

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer | Component | Role | Install Source | Validation Command | Key Notes |
| OS | Windows 10/11 (x64) | Host OS | Standard installer | winver | Keep patched |
| Shell | Windows Terminal | Enhanced terminal | Microsoft Store | wt -v | Optional but handy |
| Version Control | Git | SCM & hooks | winget / git-scm.com | git --version | Use credential manager |
| Runtime | PHP 8.3 | App runtime | Laragon bundle | php -v | Match composer platform php=8.3 |
| Stack Manager | Laragon | Web/PHP services | laragon.org | (tray running) | Provides Apache/MySQL |
| JS Runtime | Node.js LTS (18/20) | Frontend build | nodejs.org | node -v / npm -v | Required for Vite |
| Dependency | Composer 2 | PHP deps | Bundled (Laragon) | composer -V | Global vendor/bin path |
| Dependency | npm | JS deps | With Node | npm -v | Lockfile: package-lock.json |
| Global CLI | GitHub CLI (gh) | PR & auth | winget GitHub.cli | gh --version | gh auth login |
| Optional | Laravel Installer | New project scaffolding | composer global require laravel/installer | laravel --version | Not required for existing repo |

### 19.2 PHP Application Dependencies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Package | Purpose | Installed Via | Verify | Notes |
| Core | laravel/framework ^12 | Framework | composer install | php artisan --version | Keep within constraint |
| Auth | laravel/sanctum | API / SPA tokens | composer install | php artisan about | Configure stateful domains if SPA |
| OAuth | laravel/socialite | Social login | composer install | php artisan about | Needs provider creds in services.php |
| RBAC | spatie/laravel-permission | Roles/Permissions | composer install | migrations present | Publish config & migrate |
| Dev | larastan/larastan | Static analysis | composer install | composer run stan | Level=6 in phpstan.neon |
| Dev | laravel/pint | Code style | composer install | vendor/bin/pint -V | Use --test in CI |
| Dev | pestphp/pest & phpunit | Testing | composer install | php artisan test | Pest wrapper on PHPUnit |
| Dev | laravel/telescope | Debug tooling | composer install | php artisan telescope:install | Gate for local only |
| Dev | laravel/pail | Live log stream | composer install | php artisan pail --help | Included in dev script |
| Optional | laravel/breeze | Auth scaffolding | composer require | n/a until used | Only if scaffolding needed |

### 19.3 Node / Frontend Dependencies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Package | Role | Install | Verify | Notes |
| Build Tool | vite | Bundler/Dev server | npm install | npx vite -v | Config via vite.config.js |
| CSS Framework | tailwindcss | Utility CSS | npm install | npx tailwindcss -v | `tailwind.config.js` present |
| Plugin | @tailwindcss/forms | Form styles | npm install | (build output) | Extend tailwind plugins |
| Processor | postcss | CSS transforms | npm install | npx postcss -v | Used by Vite pipeline |
| Plugin | autoprefixer | Vendor prefixes | npm install | (build output) | PostCSS plugin |
| JS Microframework | alpinejs | Lightweight interactivity | npm install | (runtime) | Loaded in entrypoint |
| HTTP Client | axios | API requests | npm install | (bundle) | Used for AJAX calls |

### 19.4 IDE & Extensions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Name | Install Command | Purpose | Notes |
| Editor | VS Code | Installer | Primary IDE | Sync settings enabled |
| PHP | Intelephense | code --install-extension bmewburn.vscode-intelephense-client | Language server | Consider license for pro |
| Laravel | Artisan | code --install-extension ryannaddy.laravel-artisan | Run artisan from UI | Convenience |
| Blade | Blade Formatter | code --install-extension shufo.vscode-blade-formatter | Format blade | Configure on save |
| Blade | Blade Snippets | code --install-extension onecentlin.laravel-blade | Snippets | Speeds template work |
| Laravel | Extra Intellisense | code --install-extension amiralizadeh9480.laravel-extra-intellisense | Helper discovery | Cache may need refresh |
| Style | Pint Extension | code --install-extension open-sourcing.laravel-pint | Run pint | Optional (CLI sufficient) |
| CSS | Tailwind IntelliSense | code --install-extension bradlc.vscode-tailwindcss | Class suggestions | Uses config scanning |
| CSS | PostCSS Support | code --install-extension csstools.postcss | Syntax highlight | Optional |
| Git | GitLens | code --install-extension eamodio.gitlens | Git insights | Blame/heatmap |
| GitHub | PR & Issues | code --install-extension GitHub.vscode-pull-request-github | PR management | Auth required |
| Env | DotENV | code --install-extension mikestead.dotenv | .env syntax |  |
| UI | Error Lens | code --install-extension usernamehw.errorlens | Inline diagnostics | Improves visibility |
| Paths | Path Intellisense | code --install-extension christian-kohler.path-intellisense | Auto-complete paths |  |
| Docs | Markdown All in One | code --install-extension yzhang.markdown-all-in-one | Markdown authoring | TOC / formatting |
| Optional | Namespace Resolver | code --install-extension MehediDracula.php-namespace-resolver | Insert imports | Optional |
| Optional | Spell Checker | code --install-extension streetsidesoftware.code-spell-checker | Spell checking | Optional |

### 19.5 Operational Scripts & Quality Gates

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aspect | Name/Command | Function | Success Criteria | Notes |
| Dev Runtime | composer run dev | Concurrent serve/queue/logs/vite | All processes healthy | Kills on error |
| Style | vendor/bin/pint --test | Style check | No diffs | Fail CI if diff |
| Static Analysis | composer run stan | PHPStan level 6 | 0 errors (or baseline) | Raise level later |
| Tests | php artisan test | Execute Pest suite | All tests pass | Coverage optional |
| Full Gate | composer run check-all | Style+Analyse+Tests | Single green pass | Mirrors local & CI |
| Assets | npm run build | Production assets | No errors; build folder | Tree-shakable |
| Sail Quality | ./vendor/bin/sail composer run check-all | Container QA | Same as host | For Docker parity |

### 19.6 Environment & Configuration Items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Location | Purpose | Validation | Notes |
| .env | project root | Runtime configuration | Key presence (APP\_KEY) | Copy from example then adjust |
| database/database.sqlite | database/ | Local DB | File exists & writable | Used in tests/local dev |
| phpstan.neon.dist | root | Static analysis config | Level=6 read | Adjust path filters |
| tailwind.config.js | root | Tailwind config | Classes generated | Add purging paths |
| vite.config.js | root | Asset bundling | Dev server runs | Customize alias if needed |
| Makefile | root | Task shortcuts | make help | Cross-env automation |
| scripts/dev-tasks.ps1 | scripts/ | Windows tasks | ./scripts/dev-tasks.ps1 help | Mirrors Makefile |
| .editorconfig | root | Consistent formatting | Editor respects | Helps teammate consistency |
| .vscode/settings.json | .vscode/ | Workspace settings | Loaded in VS Code | Avoid secrets here |
| .github/workflows/\\*.yml | CI | Automation | Successful runs | Update when adding steps |

### 19.7 Quick Onboarding Flow (Condensed)

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Action | Command/Ref | Expected Result |
| 1 | Clone | git clone ... | Repo local |
| 2 | Configure env | copy .env.example .env | .env present |
| 3 | Install deps | composer install & npm install | Vendor + node\_modules |
| 4 | Generate key | php artisan key:generate | APP\_KEY set |
| 5 | DB prepare | Ensure sqlite file | File exists |
| 6 | Migrate | php artisan migrate | Tables created |
| 7 | Dev run | composer run dev | Local app accessible |
| 8 | QA gate | composer run check-all | All pass |
| 9 | Build assets | npm run build | public/build populated |
| 10 | (Optional Sail) | ./vendor/bin/sail up -d | Containers healthy |

## 20. DOCX Generation Helper

A Python script `scripts/md\_to\_docx.py` exists but targets `docs/modules.md`. For this file either:

1. Temporarily copy to `docs/modules.md` then run script; or
2. Duplicate script to target this file.

A dedicated script will be added to generate a DOCX for this environment spec. 3) Current dedicated script: `python scripts/env\_setup\_md\_to\_docx.py` (already committed). 4) Planned automation: Add workflow `regenerate-env-setup-docx.yml` to watch `docs/environment-setup.md`.