

Introduction

Base for Derafu's Projects

Anonymous · 09/09/2026

Base for Derafu's Projects

last commit **march**  CI **passing** code size **17.4 KiB** open issues **0** downloads **2.63 k**
downloads **196 this month**

Overview

Derafu Foundation is a standardized base for creating consistent PHP projects. It provides a pre-configured environment with best practices for code organization, documentation, testing, and code quality assurance. This foundation is designed to streamline the creation of new projects while ensuring they follow consistent patterns and standards.

Key Features

- **Standardized Structure:** Consistent file and directory organization across all projects.
- **Code Quality Tools:** Pre-configured PHP CS Fixer, PHPStan, and PHPUnit setups.
- **Asset Management:** Built-in Vite configuration for CSS, JavaScript, and image optimization.
- **Documentation:** Automatic website generation from README.md with GitHub Pages deployment.
- **CI/CD Integration:** GitHub Actions workflows for continuous integration and deployment.
- **Middleware-based HTTP Layer:** Ready-to-use PSR-compliant HTTP handling.
- **Routing System:** Flexible routing with support for static routes and filesystem-based routes.
- **Templating:** Twig and Markdown rendering support.

Quick Start

Installation

Create a new project based on this foundation:

```
composer create-project derafu/project project_dir --stability=dev
```

Asset Compilation

If your project uses CSS, JavaScript, or images, add them to the `assets` directory and compile them:

```
npm install
npm run build
```

Development Server

Start the built-in PHP development server:

```
php -S localhost:9000 public/index.php
```

Then visit <http://localhost:9000> to see your project.

Project Structure

```
project_dir/
├── .github/           # GitHub configurations and workflows.
├── app/               # Application bootstrap files.
├── assets/            # Frontend assets (CSS, JS, images).
├── config/            # Configuration files.
│   ├── routes.yaml    # Route definitions.
│   └── services.yaml   # Service container configuration.
├── public/            # Web server root directory.
│   ├── index.php       # Application entry point.
│   └── static/          # Compiled assets (generated).
├── src/               # Source code.
├── templates/         # Template files.
│   ├── components/     # Reusable template components.
│   ├── layouts/        # Layout templates.
│   └── pages/          # Page templates.
├── tests/             # Test files.
│   ├── fixtures/       # Test fixtures.
│   └── src/            # Source code tests.
├── vendor/            # Composer dependencies.
├── var/               # Temporary files and caches.
├── .gitignore         # Git ignore configuration.
├── LICENSE            # MIT License.
├── package.json       # NPM configuration.
├── php-cs-fixer.php    # PHP CS Fixer configuration.
├── phpstan.neon       # PHPStan configuration.
├── phpunit.xml        # PHPUnit configuration.
├── README.md          # Project documentation.
└── vite.config.js     # Vite build configuration.
```

Customization

After creating a project based on this foundation, you can:

1. Modify `README.md` with your project-specific details.
2. Update `package.json` with your project information.
3. Add your routes in `config/routes.yaml`.
4. Customize services in `config/services.yaml`.
5. Create controllers and other classes in the `src/` directory.
6. Add templates to the `templates/` directory.
7. Write tests in the `tests/` directory.

Last updated on 09/09/2026