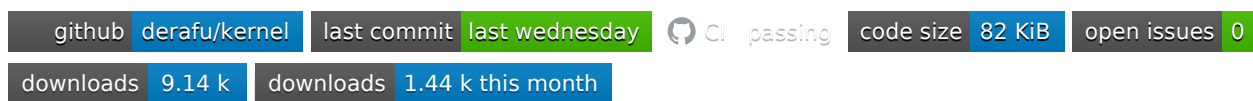


Introduction

Lightweight kernel with a dependency injection container, inspired by Symfony, with .env files and a container cache

Anonymous · 10/10/2026

Lightweight Kernel Implementation with DI Container



A lightweight kernel implementation with a dependency injection container, inspired by Symfony but with a minimal approach.

Overview

Derafu Kernel provides a clean, flexible foundation for PHP applications with minimal dependencies. It offers:

- A lightweight kernel implementation with DI container.
- Environment-aware configuration management.
- Support for PHP and YAML configuration files.
- Container caching for improved performance.
- Built-in support for multiple environments (dev, prod, test, etc.).

Installation

```
composer require derafu/kernel
```

Basic Usage

Create a Kernel

The simplest way to use the kernel is to create an instance with a specific environment:

```
use Derafu\Kernel\MicroKernel;
```

```
// Create a kernel with 'dev' environment and debug mode enabled.
$kernel = new MicroKernel('dev', true);

// Boot the kernel to initialize the container.
$kernel->boot();
```

Custom Kernel Implementation

You can extend the `MicroKernel` class to customize its behavior:

```
use Derafu\Kernel\MicroKernel;
use Symfony\Component\DependencyInjection\Loader\Configurator\ContainerConfigurator;

class AppKernel extends MicroKernel
{
    // Override the default configuration files.
    protected const CONFIG_FILES = [
        'services.php' => 'php',
        'routes.php' => 'routes',
        'parameters.yaml' => 'yaml',
    ];

    // Add additional container configuration.
    protected function configure(ContainerConfigurator $configurator): void
    {
        $services = $configurator->services();

        // Register application services.
        $services->set('app.service', MyService::class)
            ->public()
            ->args(['param1', 'param2']);
    }
}
```

Environment Configuration

The kernel uses an environment object to determine settings and directories:

```
use Derafu\Kernel\Environment;

// Create an environment with custom settings.
$environment = new Environment(
    'prod',                // Environment name.
    false,                 // Debug mode.
    ['custom' => 'value']  // Context variables.
);

// Create a kernel with the custom environment.
```

```
$kernel = new MicroKernel($environment);
```

Last updated on 10/10/2026