

Laravel Development Guides

- [Bootstrapping Laravel Project](#)
- [Laravel Tooling Guide](#)
- [Authentication Support](#)
- [Immutable Date Setup](#)
- [Security Hardening Setup](#)
- [Health Check Setup](#)

Bootstrapping Laravel Project

[laravel.com documentation](https://laravel.com/docs/10.x) [laravel/laravel](https://laravel.com)

Create a New Laravel Application

```
laravel new --pest "[PROJECT_NAME]"
```

Create a New Laravel Package

```
laravel package "[PACKAGE_NAME]"
```

Laravel Tooling Guide

PHP Tooling Guide

Laravel Package Justfile

```
# SPDX-License-Identifier: 0BSD
# SPDX-FileCopyrightText: none
#
# Manage the PHP and Laravel toolchain of this package.

# Run the recipes at the project root, not at the `.just` directory.
set working-directory := '..'

HOST := "127.0.0.1"
PORT := "8000"

# Install the Composer dependencies.
install:
    composer install

# Serve the workbench application on the PHP development server.
dev host=HOST port=PORT:
    @php vendor/bin/testbench workbench:build --ansi
    @php vendor/bin/testbench serve --ansi --host="{{ host }}" --port="{{ port
}}"

# Run the Pest test suite.
test:
    composer test:unit

# Manage the PHP and Laravel toolchain of this package.
[group('laravel')]
mod laravel '.just/php-laravel.just'
```

Authentication Support

Installation

```
composer require laravel/sanctum  
composer require laravel/fortify
```

```
php artisan install:api  
php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"  
php artisan config:publish cors
```

```
php artisan fortify:install  
php artisan vendor:publish --provider="Laravel\Fortify\FortifyServiceProvider"
```

Configuration

Immutable Date Setup

Configuration

```
// in app/Service/ImmutableDateService.php

use Carbon\CarbonImmutable;
use Illuminate\Support\Facades\Date;
use Illuminate\Support\ServiceProvider;

class ImmutableDateService extends ServiceProvider
{
    /**
     * Bootstrap any application services.
     */
    public function boot(): void
    {
        Date::use(CarbonImmutable::class);
    }
}
```

Security Hardening Setup

Configuration

```
// in app/Service/SecurityService.php

use Illuminate\Support\Facades\DB;
use Illuminate\Support\ServiceProvider;

class SecurityService extends ServiceProvider
{
    /**
     * Bootstrap any application services.
     */
    public function boot(): void
    {
        DB::prohibitDestructiveCommands($this->app->isProduction());
    }
}
```

Health Check Setup

[documentation spatie/laravel-health](#)

Installation

```
composer require spatie/laravel-health
```

```
php artisan vendor:publish --tag="health-config"  
php artisan vendor:publish --tag="health-migrations"
```

Configuration

```
// in app/Service/HealthCheckService.php  
  
use Illuminate\Support\ServiceProvider;  
use Spatie\Health\Checks\Checks\DatabaseCheck;  
use Spatie\Health\Checks\Checks\DebugModeCheck;  
use Spatie\Health\Checks\Checks\EnvironmentCheck;  
use Spatie\Health\Checks\Checks\UsedDiskSpaceCheck;  
use Spatie\Health\Facades\Health;  
  
class HealthCheckService extends ServiceProvider  
{  
    /**  
     * Bootstrap any application services.  
     */  
    public function boot(): void  
    {  
        Health::checks([  
            // https://spatie.be/docs/laravel-health/v1/available-  
checks/environment  
            EnvironmentCheck::new()->unless($this->app->environment('local')),  
            // https://spatie.be/docs/laravel-health/v1/available-checks/debug-  
mode  
            DebugModeCheck::new()->unless($this->app->environment('local')),  
            // https://spatie.be/docs/laravel-health/v1/available-checks/db-  
connection  
        ]);  
    }  
}
```

```

        DatabaseCheck::new(),
        // https://spatie.be/docs/laravel-health/v1/available-checks/used-
disk-space
        UsedDiskSpaceCheck::new(),
        // https://spatie.be/docs/laravel-health/v1/available-checks/queue
        // https://spatie.be/docs/laravel-health/v1/available-
checks/schedule
    ]);
}
}

```

```

// in routes/web.php

use Spatie\Health\Http\Controllers\HealthCheckResultsController;

Route::get('/up', HealthCheckResultsController::class);

```

```

// in routes/console.php
use Illuminate\Support\Facades\Schedule;

Schedule::command(\Spatie\Health\Commands\RunHealthChecksCommand::class)-
>everyMinute();

```

Custom Health Check Model

```

// in app/Model/HealthCheckResult.php

use Illuminate\Database\Eloquent\Attributes\Table;
use Spatie\Health\Models\HealthCheckResultHistoryItem;

#[Table(name: 'HEALTH_CHECK')]
class HealthCheckResult extends HealthCheckResultHistoryItem
{
    //
}

```

```

// in config/health.php

return [
    'result_stores' => [
        EloquentHealthResultStore::class => [
            'model' => HealthCheckResult::class,
        ],
    ],
];

```