

Carpetas principales

1. config/:

Configura servicios, rutas, bases de datos, y paquetes adicionales.

- Subcarpetas comunes:
 - packages/: Configuración específica de cada paquete.
 - routes/: Define rutas principales y API (ej.: annotations.yaml).
 - services.yaml: Configuración de dependencias y servicios personalizados.

2. public/:

- index.php: Punto de entrada para todas las solicitudes HTTP.
- Carpeta donde se colocan los archivos estáticos generados, como:
 - build/: Salida de Webpack (CSS/JS compilados).
 - Imágenes o favicons directamente accesibles.

3. src/:

Contiene todo el código fuente del backend:

- Controller/: Lógica de negocio y manejo de rutas.
- Entity/: Clases que representan tablas en la base de datos.
- Repository/: Consultas personalizadas relacionadas con las entidades.
- Form/: Clases para gestionar formularios (inputs, validaciones, etc.).
- Security/: Clases relacionadas con la autenticación/autorización.

4. templates/:

- Contiene las vistas con Twig, donde se combinan HTML y datos dinámicos del backend.

5. assets/:

Código fuente del frontend antes de compilar:

- css/: Estilos personalizados.
- js/: Scripts JavaScript (lógica del cliente o frameworks como React).
- images/: Imágenes sin optimizar.

6. var/:

Symfony la usa para almacenar datos temporales:

- cache/: Archivos de caché generados para acelerar la aplicación.
- log/: Registros de errores y eventos.

7. vendor/:

- Aquí se almacenan las dependencias de PHP instaladas por Composer (Symfony y otros paquetes).

8. tests/:

- Clases de prueba automatizada para garantizar que el código funcione correctamente.

Archivos clave

- .env y .env.local:
- Variables de entorno para configurar bases de datos, API, entorno de ejecución (dev, prod).
- composer.json y composer.lock:
- Gestionan las dependencias de PHP del proyecto.
- package.json:
- Configura las dependencias y tareas del frontend.
- webpack.config.js:
- Define cómo se compilan los archivos CSS/JS desde la carpeta assets/.
- Dockerfile y docker-compose.yaml:
- Configuración para ejecutar el proyecto en contenedores Docker (PHP, base de datos, servidor web, Node.js).
- README.md:
- Documentación básica del proyecto (cómo instalar, ejecutar, dependencias).