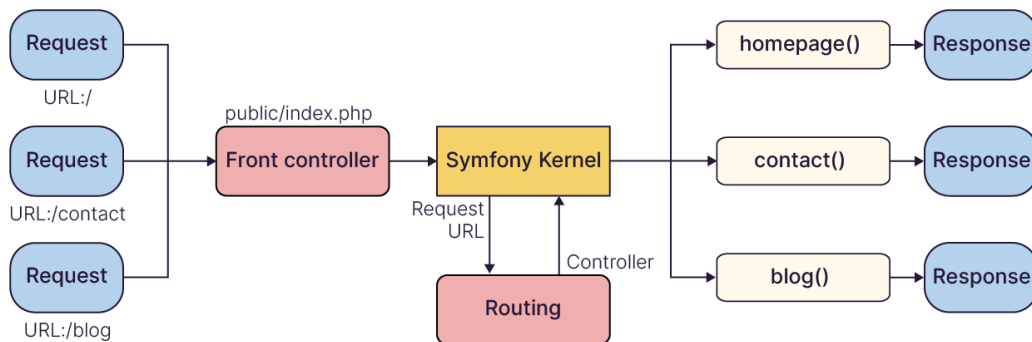


Initialisation du Projet Symfony "Biblios"

1. Architecture et Fonctionnement (Théorie)

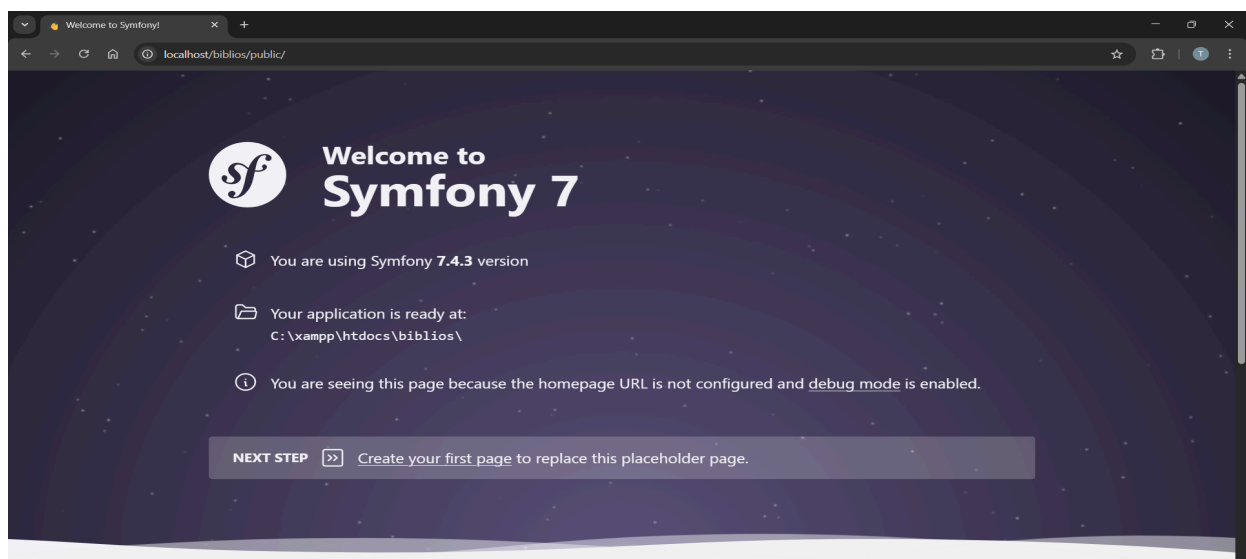
Avant de coder, il est essentiel de visualiser le cycle de vie d'une requête HTTP dans Symfony. Le schéma suivant illustre le parcours :

1. Le navigateur envoie une requête (**Request**).
2. Le **Front Controller** reçoit la demande.
3. Le **Kernel** (noyau) interroge le **Routing** pour savoir quel contrôleur appeler.
4. Le **Controller** traite la demande et renvoie une réponse (**Response**).



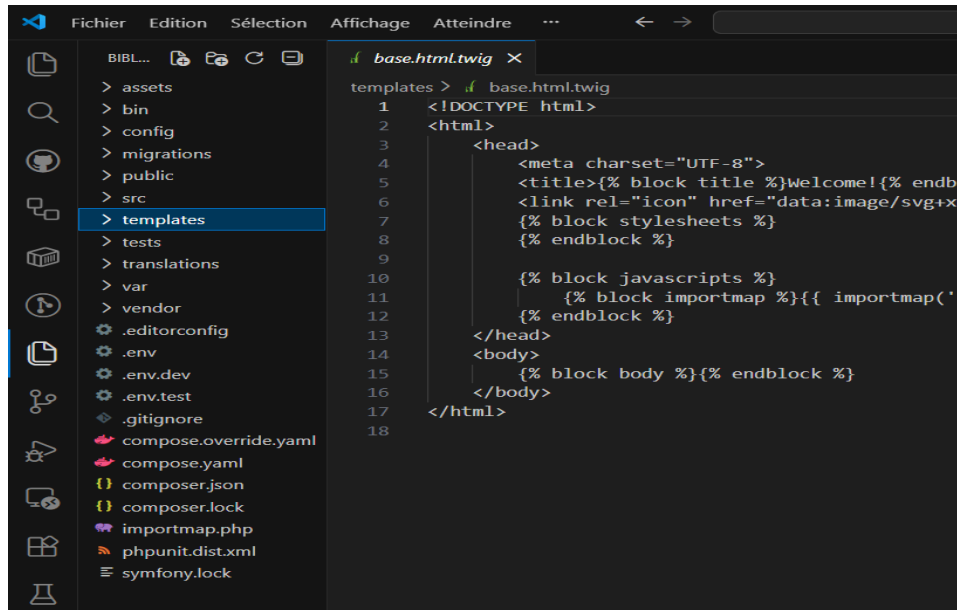
2. Installation et Vérification de l'environnement

Le projet "biblios" a été initialisé avec Symfony 7.4.3. Le serveur local fonctionne correctement, comme le prouve la page d'accueil par défaut de Symfony qui confirme que le système est prêt à l'emploi.



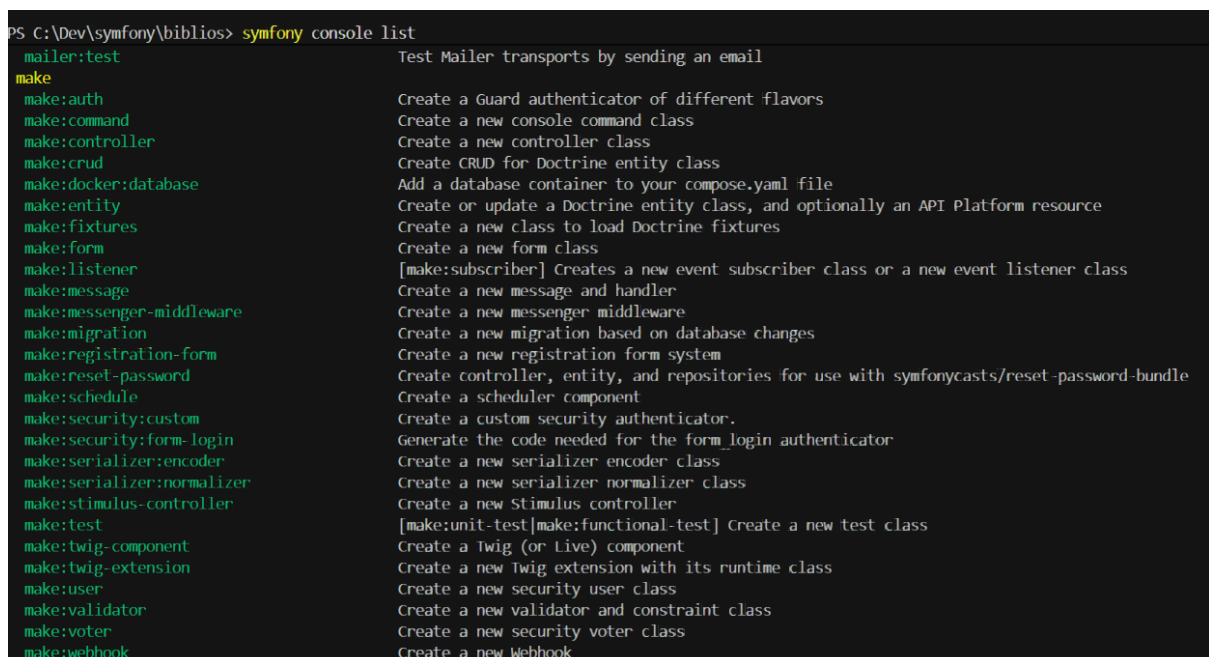
3. Structure des dossiers et Templating

L'architecture des dossiers suit le standard Symfony. Nous avons notamment exploré le dossier `templates`. Le fichier `base.html.twig` sert de squelette HTML global pour l'application, utilisant le système de blocs de Twig pour l'héritage de mise en page.



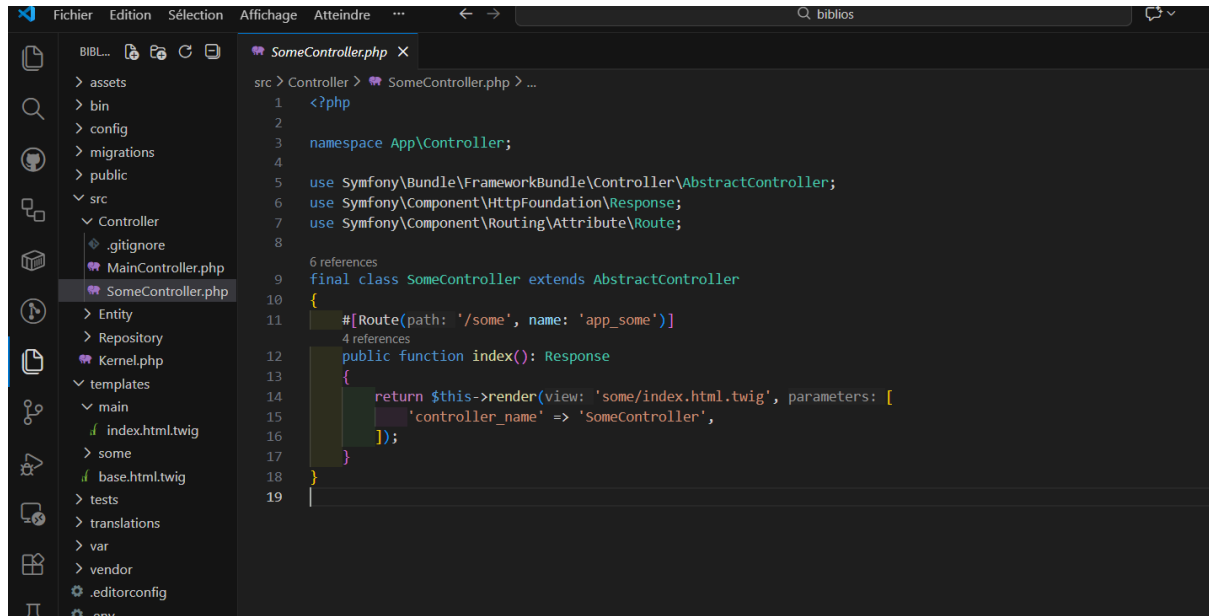
4. Utilisation du Maker Bundle

Pour accélérer le développement et éviter les erreurs de syntaxe, nous utilisons le composant `symfony/maker-bundle`. La commande `symfony console list make` nous permet de voir toutes les générations de code disponibles (Contrôleurs, Entités, Formulaires, etc.).



5. Création d'un Contrôleur (Code)

Nous avons généré un contrôleur (ex: `SomeController` ou `MainController`). Dans le code ci-dessous, on remarque l'utilisation des **Attributs PHP 8** (`#[Route]`) pour définir l'URL d'accès (`/some`). La méthode `render()` est utilisée pour appeler la vue Twig associée.



6. Résultat Final

Une fois le contrôleur et la route créés, l'accès à l'URL via le navigateur déclenche bien le code. La page s'affiche avec la barre de débogage (Profiler) de Symfony, confirmant que le lien entre la Route, le Contrôleur et la Vue fonctionne parfaitement.

