

IUT Nancy-Charlemagne – BUT Informatique

S5 – DWM

Développement Web Serveur avancé

TD1.2 : Injection de dépendances

L'objectif du TD est de mettre en place, configurer et utiliser un conteneur d'injection de dépendances dans le projet Toubilib. Vous utiliserez le conteneur d'injection de dépendances PHP-DI, qui est compatible PSR-11, en le configurant pour ne pas utiliser le mode `autowire`.

Préliminaires

Vérifiez que PHP-DI est bien installé dans votre projet. Si ce n'est pas le cas, installez-le avec Composer. Complétez votre bootstrap de l'application pour y ajouter la configuration du conteneur d'injection de dépendances et créer une application Slim utilisant ce conteneur.

Exercice 1: Injection de dépendances pour les composants et classes de la fonctionnalité 1

Programmer l'injection de dépendances pour les composants et classes de la fonctionnalité 1 du projet Toubilib, qui consiste à lister les praticiens. Pour cela, assurez-vous ou transformez votre code de façon adéquate pour que :

- le service métier `ServicePraticien` soit injecté dans l'action de l'application Slim qui liste les praticiens,
- le repository `PraticienRepository` soit injecté dans le service métier `ServicePraticien`,

Déclarez ensuite les dépendances dans le conteneur d'injection de dépendances. Organisez vos définitions dans trois fichiers distincts :

- un fichier `settings` contenant les définitions de variables de configuration,
- un fichier `services` contenant les définitions des services et les repositories,
- un fichier `api`, contenant les définitions des actions de l'application Slim.

Concrètement vous aurez :

- une factory pour instancier la classe `ListerPraticiensAction`,
- une factory pour instancier l'interface `ServicePraticienInterface`, qui injecte le repository `PraticienRepository` dans le constructeur du service,
- une factory pour instancier l'interface `PraticienRepositoryInterface`, qui injecte la connexion `PDO` à la base de données dans le constructeur du repository.
- une factory pour instancier la connexion `PDO`.

Exercice 2 : Fonctionnalités 2,3,4

Programmer les fonctionnalités 2, 3 et 4 du projet Toubilib, qui consistent à afficher le détail d'un praticien, lister les créneaux de rdvs déjà occupés d'un praticien sur une période donnée, et consulter un rendez-vous à partir de son identifiant.

Comme pour la fonctionnalité 1, vous devez :

- respecter scrupuleusement les principes de l'architecture hexagonale et de l'inversion de dépendances : vous devrez donc prévoir les entités métier, les services métier, les DTOs, les repositorys nécessaires,
- programmer l'injection de dépendances pour les composants et classes de ces fonctionnalités, en complétant les fichiers de configuration du conteneur d'injection de dépendances,
- vous conformer aux principes RESTful, notamment en utilisant les méthodes HTTP et les URLs adéquates, et en renvoyant des statuts de retour corrects.