



Ecole Marocaine  
des Sciences de l'Ingénieur

## Compte Rendu JEE :

### Spring Data JPA

- **Réalisé par : BENNOUNA LORIDY MOHAMED**
- **GROUPE : 4IIR G2/ EMSI CENTRE**

Année universitaire 2023-2024

## Introduction :

Un bon projet informatique doit respecter des normes diverses (métier) à savoir les fonctionnalités de l'application techniques ( sécurité, journalisation, gestion de transactions, gestion des exceptions et accès au base de données... ) et financières (respecter le cout dédié ).

Parmi ces exigences techniques on note la performance consistant à éviter le problème de montée en charge (le nombre d'utilisateur élevé bloque l'accès aux fonctionnalités de l'app comme Masar ) le serveur effectue beaucoup de traitement en même temps, il gère beaucoup de requêtes ce qui signifie cpu utilisé avec un taux trop élevé.

Alors pour régler ce problème, il faut aller vers **la scalabilité verticale** ou **horizontale**.

- Scalabilité verticale : c'est ajouter capacité disque, mémoire dans le serveur.

- Scalabilité horizontale : c'est démarrer l'application dans plusieurs machines serveurs.

L'application doit être facile à maintenir :

Fermé à la modification et ouverte à l'extension.

Récupérer une application pour appliquer une modification au niveau du code ça ne se fait pas au marché. La question qui se pose comment implémenter ça dans un projet ?

On note deux principe :

- couplage faible

- couplage fort

Le couplage fort : Les classes dépendent des classes . Ce qui s'avère désavantageux lors d'une modification

Le couplage faible : Les classes dépendent que des interfaces, puisque l'interface peut être implémentée par plusieurs classes, on a le droit d'ajouter plusieurs extensions. Quand

on décide d'effectuer une modification on crée de nouvelles classes (extensions d'interfaces) implementant les interfaces sans avoir besoin de toucher le code ce qui s'

D'un côté je peux bénéficier de plusieurs classes (fonctionnalités) et de ce côté je ne peux pas toucher le code, ceci résume ne pas toucher le code c'est avoir la flexibilité au niveau de l'application.

## L'inversion de contrôle :

- **L'inversion de contrôle** : consiste à séparer l'aspect technique et l'aspect fonctionnel (métier) . Elle permet au développeur de se concentrer sur l'aspect métier càd le code métier de l'application alors que c'est le Framework qui va s'occuper du code technique, l'inversion de contrôle est basé sur la programmation orienté aspect AOP .

Pour s'assurer que le principe de l'inversion de contrôle fonctionne concrètement sur un projet, on s'occupe du code métier et le Framework s'occupe du code technique par **l'injection de dépendances**. (les dépendances sont comme des bibliothèques utilisés pour bénéficier de services et fonctionnalités bien précis)