

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Jijel
Faculté des Sciences exactes et de l'informatique
Département d'informatique



– Module –

Environnements et Programmation Dédiés

Master 1 : IA

Enseignant du module : Dr. Hemza FICEL

Contact: hemza.ficel@univ-jijel.dz

TP 2

Objectifs

L'objectif de ce TP est de concevoir, développer et exécuter **une application web monolithique de type CRUD** (Create, Read, Update, Delete) destinée à la gestion de données (ex. la gestion de produits, d'étudiants, de clients ou d'employés).

Cette application, développée avec **le framework Spring Boot**, doit permettre d'effectuer les quatre opérations fondamentales sur les données : ajout, consultation, modification et suppression.

Elle s'appuiera sur une base de données **MySQL pour la persistance des informations**, et la **couche de présentation sera réalisée avec Thymeleaf et Bootstrap**, afin d'offrir une interface web moderne, claire et responsive.

Architecture du projet

L'architecture suivra un modèle **monolithique en couches** :

Couche de présentation (Web)

Contrôleurs Spring MVC, Pages HTML avec Thymeleaf + Bootstrap

Couche Métier (Service)

Logique applicative et règles de gestion

Couche Accès aux données (Repository / DAO)

Accès à la base MySQL via Spring Data JPA

Base de données

MySQL

Technologies et outils

- ✚ JAVA 17+
- ✚ Spring Boot 3.x
- ✚ Spring Web (MVC)
- ✚ Spring Data JPA / ORM Hibernate
- ✚ MySQL
- ✚ Thymeleaf
- ✚ Bootstrap 5.x
- ✚ Spring Tools for Eclipse

Structure générale du projet


```
└─ src/ main/ java
    |   |   └─ com/example/tpmonolithique/
    |   |   |   └─ controller/      # Couche Web (Contrôleurs MVC)
    |   |   |   |   └─ ProduitController.java
    |   |   |   └─ entity/        # Modèles / Entités JPA
    |   |   |   |   └─ Produit.java
    |   |   |   └─ repository/    # Accès aux données (DAO)
    |   |   |   |   └─ ProduitRepository.java
    |   |   |   └─ service/      # Logique métier
    |   |   |   |   └─ ProduitService.java
    |   |   |   └─ config/       # Configuration de l'application
    |   |   |   |   └─ WebConfig.java
    |   |   |   └─ TpMonolithiqueApplication.java # Classe principale Spring Boot
```

```
└─ src/ main/ java
    |   └─ resources/
    |       └─ static/          # Fichiers statiques (CSS, JS, images, Bootstrap)
    |           └─ css/
    |           └─ js/
    |           └─ images/
    |       └─ templates/      # Vues Thymeleaf
    |           └─ fragments/   # Fragments réutilisables (header, footer, navbar)
    |               └─ header.html
    |           └─ produits.html # Liste
    |           └─ form.html    # Formulaire
    |       └─ application.properties # Configuration Spring Boot
    └─ pom.xml                 # Dépendances Maven
```

Résultat attendu

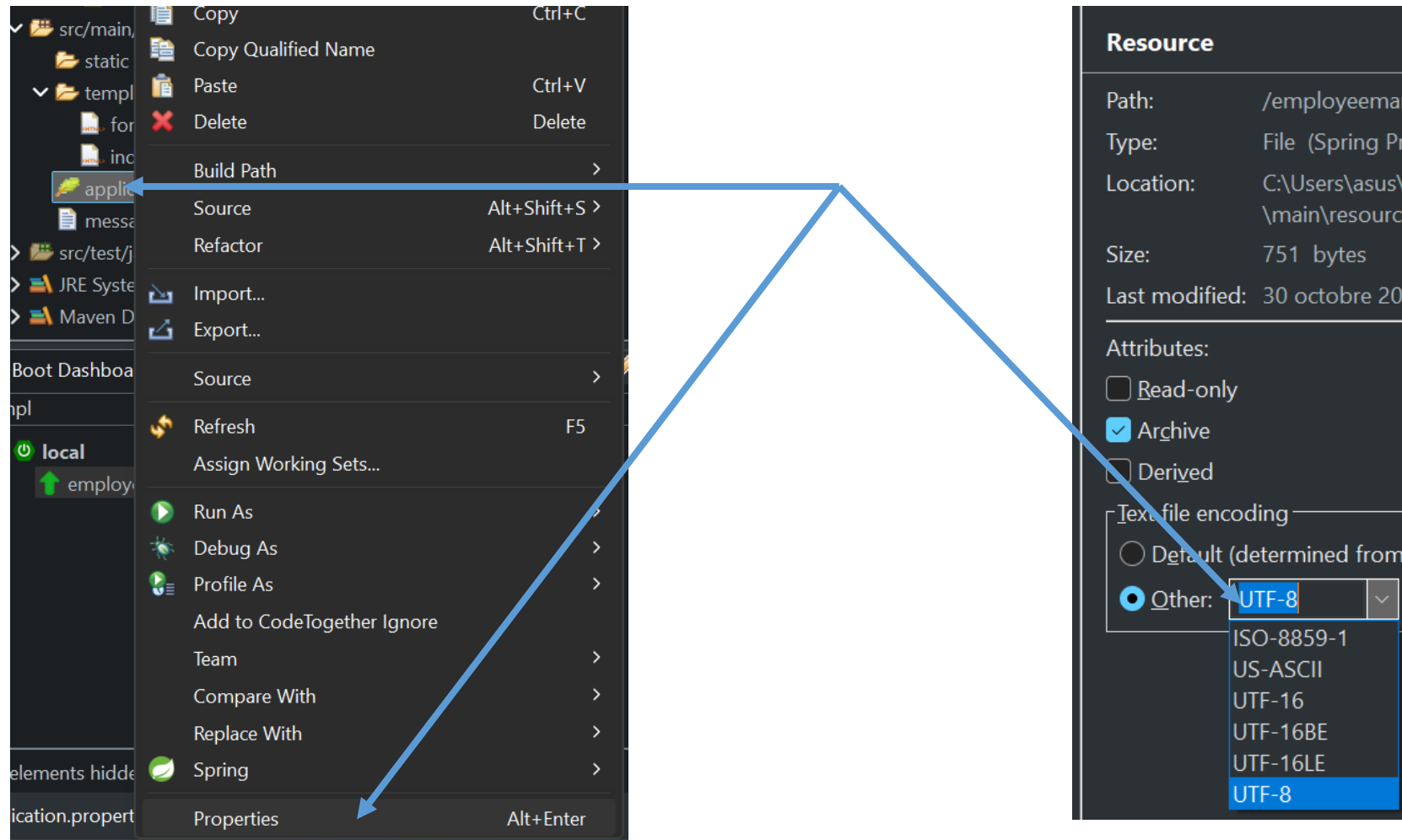
L'application devra :

1. Assurer une gestion complète des entités à travers les opérations CRUD
2. Générer des rendues côté serveur avec Thymeleaf.
3. Persister les données dans MySQL (Spring Data JPA)
4. Respecter la séparation des couches (Spring MVC + Monolithique)
5. Adopter une organisation structurée des packages, conforme aux conventions et bonnes pratiques de développement Spring Boot
6. Afficher une UI ergonomique et responsive (Bootstrap)

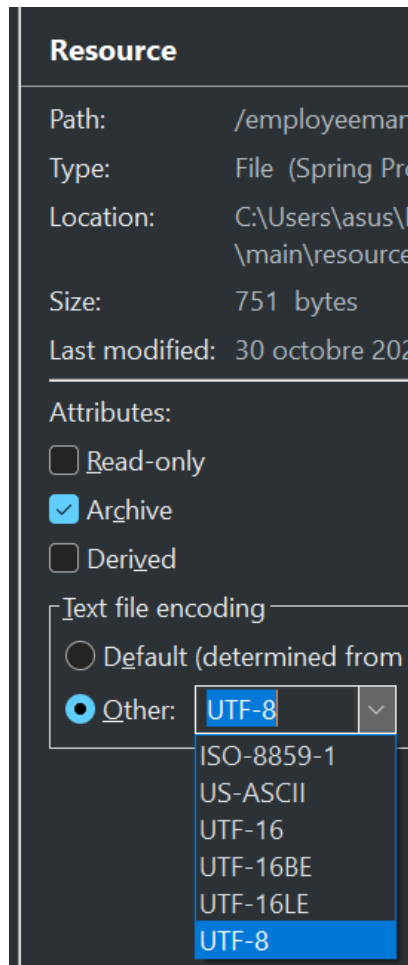
Options avancées

1. Appliquer des contraintes de validation sur les champs d'entités avec Hibernate Validator (starter validation);
2. Gérer les messages d'erreurs dans les formulaires Thymeleaf ;
3. Mettre en place un mécanisme **de pagination et de tri** des données qui permet de diviser un grand ensemble de données en plusieurs pages pour améliorer les performances et l'expérience utilisateur (avec Spring Data JPA);

Astuces



Pensez à changer l'encodage de votre fichier application.properties en UTF-8 si vous utilisez des commentaires contenant des caractères accentués, afin d'éviter les problèmes d'affichage ou d'interprétation par Spring Boot.



Exemple

Configuration de la base de données pour l'application

Assurez-vous que le nom d'utilisateur et le mot de passe sont corrects

`spring.datasource.url=jdbc:mysql://localhost:3306/employeemanager`

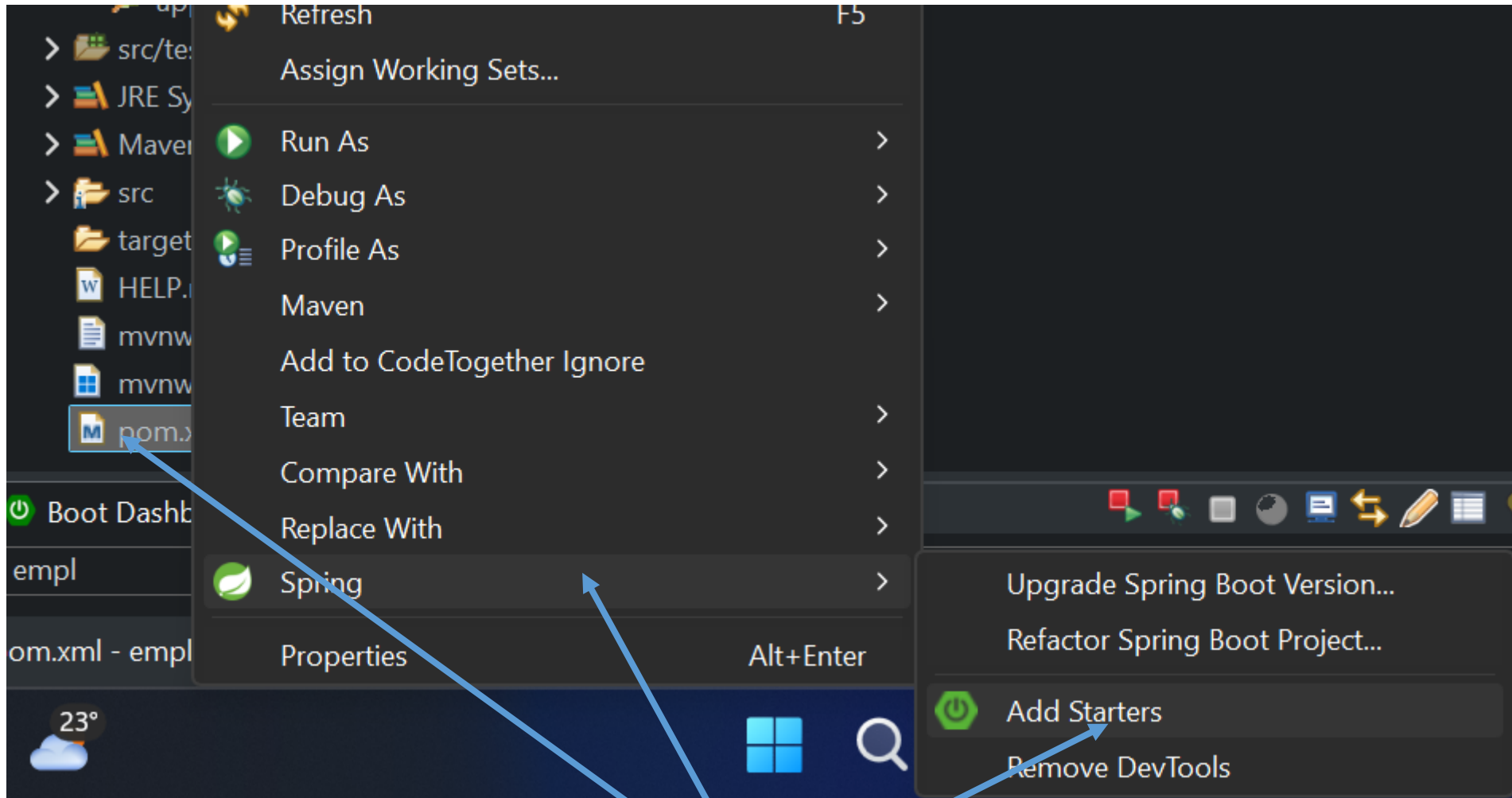
`spring.datasource.username=root`

`spring.datasource.password=root`

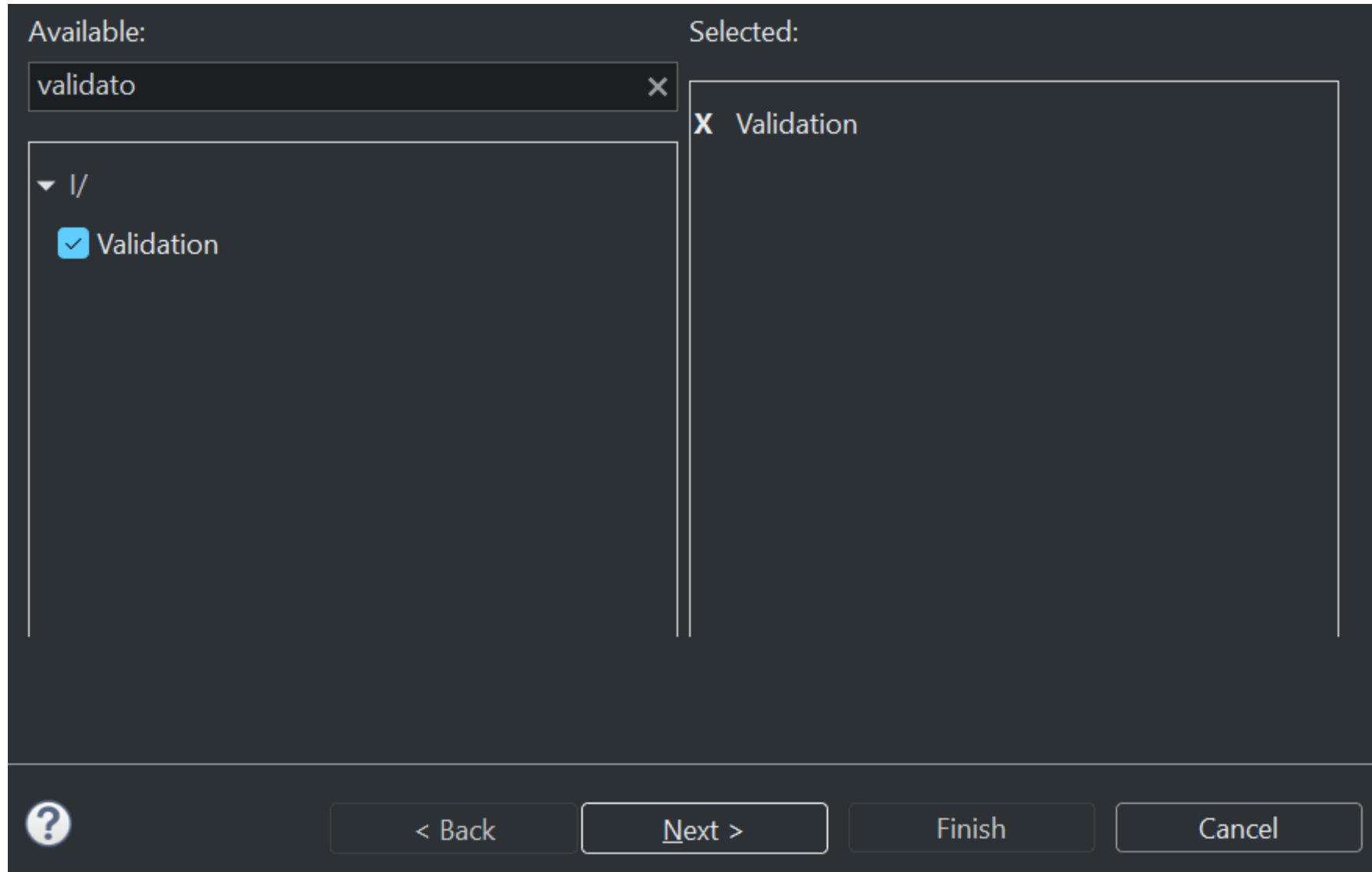
Ce commentaire contient des accents : vérifiez l'encodage UTF-8

Exemple : Échec possible si le fichier n'est pas en UTF-8

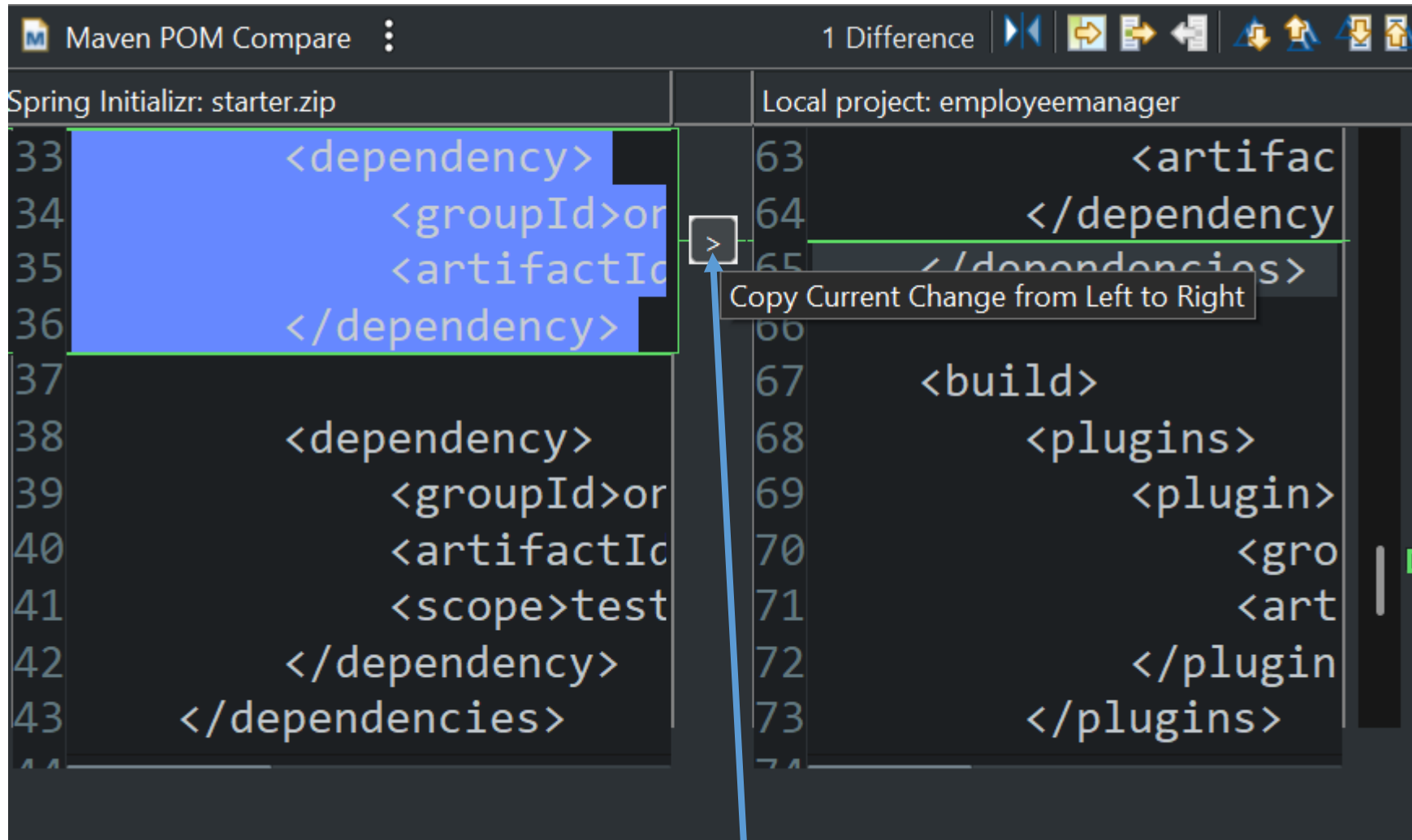
Pensez à changer l'encodage de votre fichier `application.properties` en UTF-8 si vous utilisez des commentaires contenant des caractères accentués, afin d'éviter les problèmes d'affichage ou d'interprétation par Spring Boot.



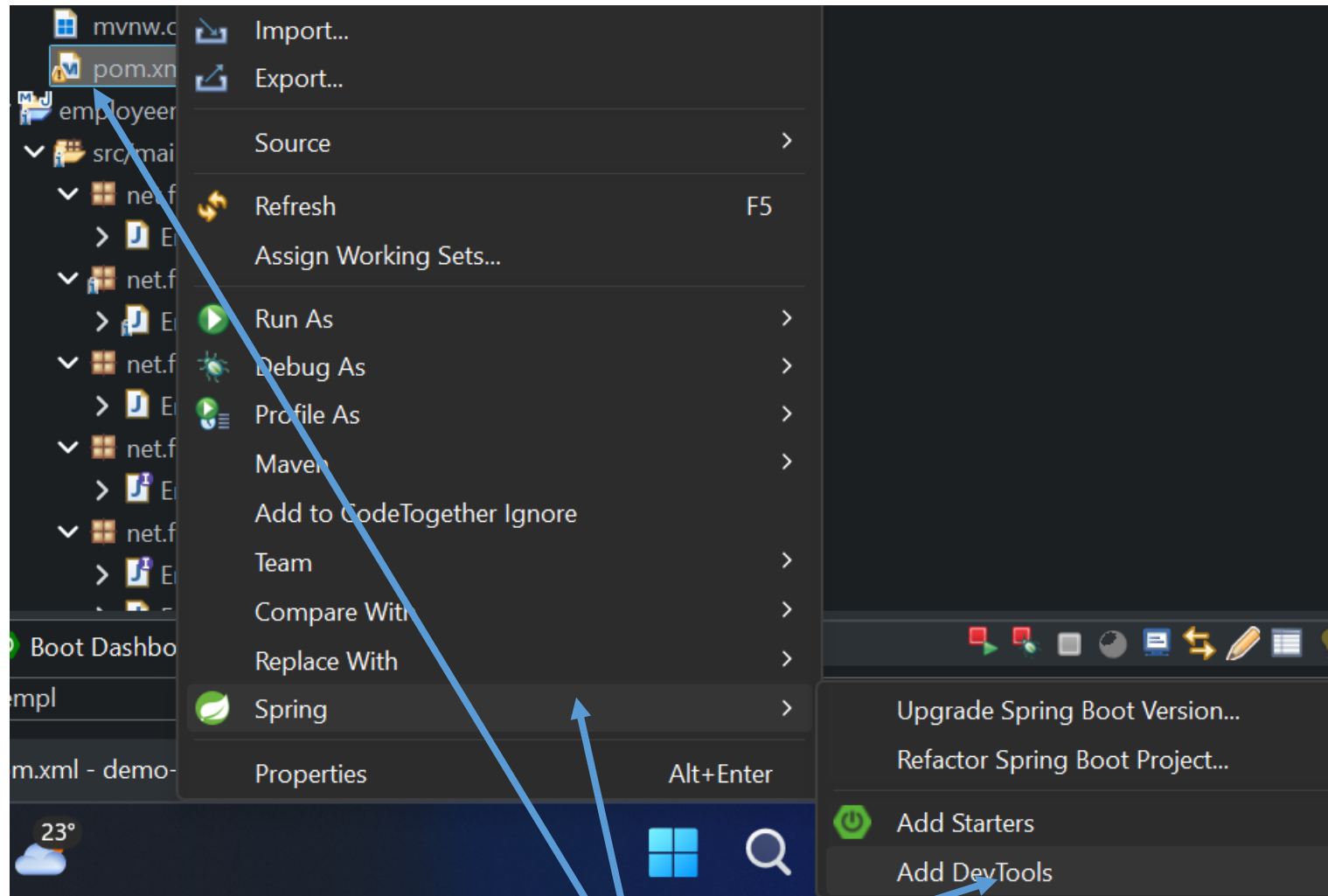
Comment ajouter un starter Spring Boot à un projet existant ?



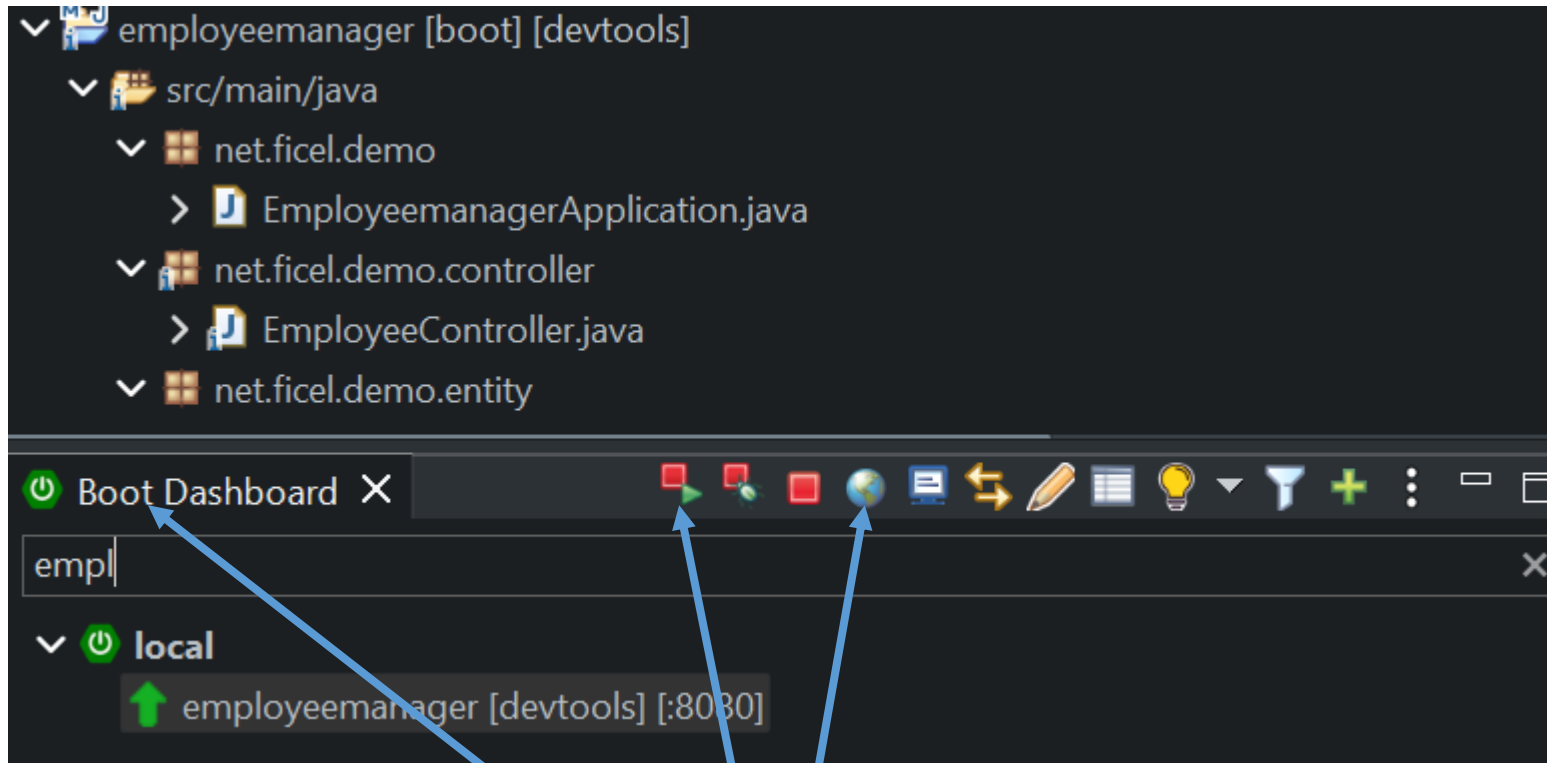
Comment ajouter un starter Spring Boot à un projet existant ?



Comment ajouter un starter Spring Boot à un projet existant ?



Comment activer le redéploiement automatique d'une application Spring Boot lors de modifications du code source (en ajoutant le starter spring-boot-devtools) ?



Comment utiliser Boot Dashboard dans Spring Tool Suite pour lancer directement un projet Spring Boot ?