

# OUTILS DE DÉVELOPPEMENT ET DÉPLOIEMENT

## 1. Environnement de Développement (IDE & SDK)

### Language & Frameworks

- **Java JDK 21:** Version LTS utilisée pour garantir la compatibilité avec les fonctionnalités modernes (Records, Switch expressions) et Jakarta EE.
- **Jakarta EE 10 :** API standard pour les Servlets et la persistance (remplaçant l'ancien `javax.*`).
- **Spring Boot 3.3.0 :** Utilisé pour la seconde partie du projet (Refonte).

### Environnement de Développement Intégré (IDE)

- **IntelliJ IDEA Ultimate / Eclipse Enterprise :**
  - Utilisé pour l'édition du code, la compilation à la volée et le débogage.
  - Intégration native des serveurs d'application pour les tests locaux.
  - Outils d'analyse de code (Linter) pour respecter les conventions Java.

## 2. Gestion de Projet et Versionning

### Git & GitHub

L'ensemble du code source est versionné sous Git.

- **Workflow :** Utilisation de branches par fonctionnalité (ex: `feature/payslips`, `fix/login`) avant fusion sur la branche principale `main`.
- **Collaboration :** Le dépôt GitHub sert de point de synchronisation pour le groupe.

### Build & Dépendances : Maven

Le projet est structuré via **Apache Maven**. Le fichier `pom.xml` gère :

- Les dépendances (Drivers MySQL, API Servlet, JSTL, Hibernate, Spring Boot Starter).
- Le cycle de vie du build (`clean`, `compile`, `package`).
- La génération de l'artefact final.

## 3. Base de Données

### SGBD : MySQL 8.0

- Base de données relationnelle robuste choisie pour sa compatibilité avec Hibernate et JPA.
- **Script d'initialisation** : Un script SQL (`script.sql`) a été créé pour générer automatiquement la structure des tables (`Employees`, `Departements`, etc.) et insérer les données de test (Compte Admin par défaut).

### Outils d'administration BDD

- **MySQL Workbench / DBeaver** : Utilisés pour concevoir le modèle physique, exécuter des requêtes de test et vérifier l'intégrité des données insérées par l'application.

## 4. Déploiement et Serveur

### Version J2EE (Standard)

- **Serveur d'Application** : Apache Tomcat 10.1.
- **Procédure** :
  1. Lancement via la commande : `mvn install` puis `mvn`.
  2. Génération d'un fichier **WAR** (Web Application Archive).
  3. Déploiement du WAR dans le dossier `/webapps` de Tomcat.

### Version Spring Boot

- **Serveur Embarqué** : Tomcat est directement intégré ("embedded") dans l'archive finale.
- **Procédure** :
  1. Lancement via la commande : `mvn install` puis `mvn`.
  2. L'application est accessible directement sur `http://localhost:8080` sans installation de serveur externe.

## 5. Frontend & Design

- **JSP & JSTL** : Moteur de template côté serveur.
- **CSS 3** : Design responsive (Flexbox/Grid) réalisé "à la main" sans framework lourd (comme Bootstrap) pour optimiser le chargement et démontrer la maîtrise des standards Web.