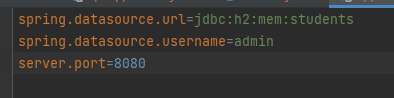
**Rapport : Spring Data JPA**

**Introduction**

La JPA est une API Java qui permettant aux développeurs de manipuler des objets Java sans avoir à écrire de code SQL pour effectuer des opérations de base de données tel que la création d’une table de base de données à partir d’une class modèle ou l’application des différentes transactions INSERT, UPDATE, DELETE grâce à des méthodes prédéfinis fournis par l’interface JpaRepository ou par des méthode qu’on va créer dans une interface qui héritera de l’interface JPA.

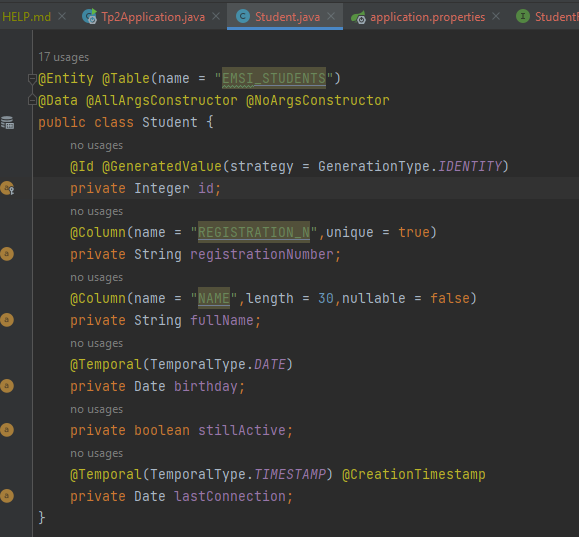
**Réalisation**

**1/ Propriété de l’application**

****

Cette page permet de configurer le port, le nom d’utilisateur ainsi que le lien de notre BD.

**2/ Class Student**

****

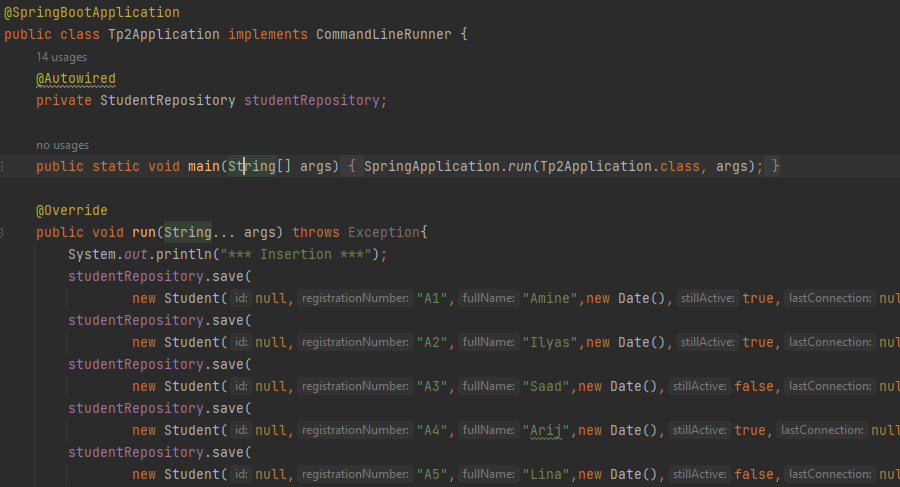
Cette classe permet de créer le modèle de notre table de base de données en spécifiant les colonnes ainsi que la clé primaire.

**3/ Interface studentRepository**

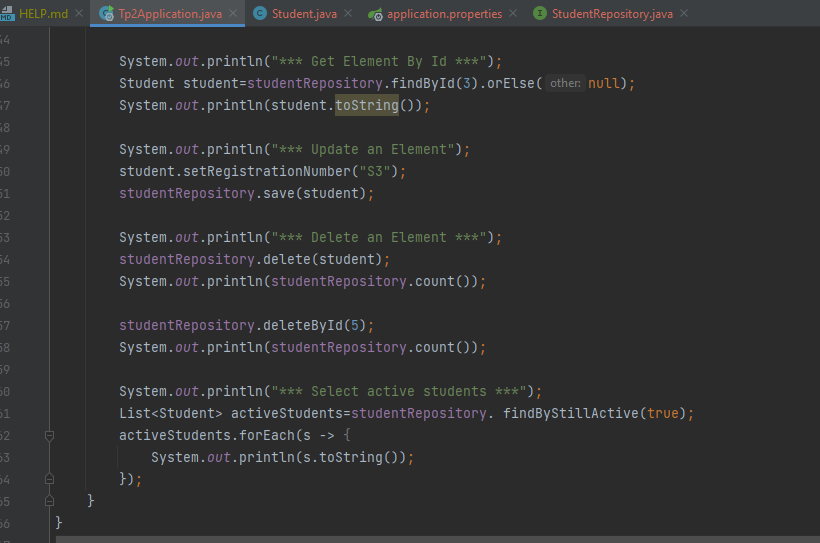
****

L’interface studentRepository hérite des méthodes de l’interface JpaRepository qui prend en considération la classe et sa clé primaire tout en ayant la possibilité d’ajouter d’autres méthodes.

**4/ Class Main**

****

La méthode run utilisé dans la class main s’exécute automatiquement après le lancement du programme, dans ce cas on utilisera la méthode save pour insérer une ligne dans notre table.

****

Voici d’autre méthodes qu’on peut utiliser citons parmi eux findById qui permet de récupérer un enregistrement à partir de son id indiqué comme paramètre, setNomColumn permet d’affecter une valeur a un enregistrement précis puis on utilise la méthode save pour appliquer les modifications, delete pour supprimer un élément en question ou deleteById pour specifier son identifiant.

**Conclusion**

En conclusion, la JPA est une spécification Java EE qui permet la gestion des objets persistants dans une base de données relationnelle. Elle fournit un environnement de programmation qui permet d'interagir avec la base de données sans avoir à écrire de code SQL.