**StudenteService.java**

**🧾 Metodo createStudente (StudenteDto studenteDto)**

Questo metodo si occupa della creazione di uno studente che abbia i seguenti parametri:

* Nome
* Cognome
* Email
* Matricola
* Corso di laurea
* Dipartimento

**🟨 Problema riscontrato al momento di inserimento del valore di Corso di laurea.** Tutte le altre variabili presenti nell’entità studente sono già validate grazie ai tag poiché non hanno bisogno di altre tipologie di validazione. **Il corso di laurea è specifico e va stilata una lista per evitare errori di inserimento.**

🟦 **Sarebbe stato meglio inserire il corso di laurea all’interno dell’entità “Dipartimento” ma possiamo apporre rimedio inserendo un Enum collegato all’entità “Studente”.**

**La soluzione senza “pasticciare” il progetto è quella di inserire un enum che converte il dato in entrata**, in questo modo il dato predefinito che vedremo in tabella sarà DATA\_SCIENCE, ma se per errore inseriremo Scienze e tecnologie Informatiche automaticamente avremo la conversione diretta in “DATA\_SCIENCE”.

***DATA\_SCIENCE*("Scienze e Tecnologie Informatiche"),  
*MATHEMATICAL\_SCIENCES*("Scienze Matematiche");**

**Nel contesto di “enum” si può utilizzare un for-each con (X x : values()) in questo modo tutti gli elementi della lista enum verranno esaminati.**

🟦 Quando si crea **un’eccezione personalizzata** se il suo utilizzo è vincolato solamente ad un’operazione anziché definirla generica quindi senza implementazione a livello di **stringa in output è possibile definirla dentro la stessa eccezione in modo da evitare la ripetizione.**

**🟨 Problema riscontrato anche durante l’inserimento del nome e cognome**, al momento dell’input si potrebbe **inserire per errore un numero o un carattere**. E’ consigliabile inserire un **@Pattern per evitare in fase di input questi errori.**