

PARTE TECNICA :

Si realizzi un'applicativo Java + Spring Boot + MySql + TS + ANGULAR

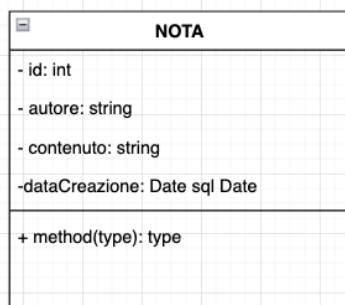
Parte di Backend:

Per la parte di back-end si vuole realizzare un sistema in grado di memorizzare delle note che vengono generate da parte di un utente.

Usare la gestione delle Exception in modo corretto.

Usare Rested per controller correttamente le operazioni di CRUD

La classe Poco che andrà nel database sarà la seguente :



NEI CONTROLLI USARE LA VALIDAZIONE IN JAVA
PER VALIDARE CORRETTAMENTE UN DATO

<https://www.baeldung.com/javax-validation>

I controlla che dovranno essere Validati con un Validator vedi :

Validazioni che dovranno essere controllate :

- 1- Autore diverso da null e diverso stringa vuota
- 2- Contenuto diverso da null e diverso da stringa vuota

La data di creazione viene sempre generata in base alla data odierna del sistema
(Attenzione usare sql.date come libreria per inserirla correttamente a database)

Possibile query da cercare su Google : current date in java.sql.date

La data quindi non dovrà comparire nel costruttore come neanche id.

Quindi nella classe Pojo avrò che il costruttore prende in ingresso un autore e un contenuto

DOCUMENTAZIONE :

<https://www.baeldung.com/spring-data-jpa-query>

Creare una query sql all'interno della classe repository

Che sia in grado di ritornare una lista di note ordinate
Dalla meno recente alla più recente.

> Creare una query sql che sia in grado di ritornare
Le note byAutore

La parte backEnd Api deve necessariamente esporre i seguenti servizi :

NOTA :

RICORDASI DI GESTIRE IL CROSSORIGIN

>GetAllNote() -> ritorna una lista json di Nota

>GetNotaById(int id) -> prende in ingresso un id e ritorna la nota sotto forma di json

>GetNotaByAutore(String autore) -> ritorna tutte le note di un determinato autore

>DeleteById(int id) -> cancella una nota per id se esiste , ritorna true se è stata cancellata
altrimenti ritorna false.

>DeleteAll() -> ritorna true se non si sono verificate exception e cancella tutte le note

>CreaNota(Nota) —> ritorna true se è stata creata una nuova nota altrimenti false.

>ModificaNota(int id , Nota nota) , prende in input un id e una nota , sovrascrive la nota
andandola a modificare , se l'id non esiste all'interno dalla repository ,

Il programma ritorna false, altrimenti se la modifica è andata a buon fine ritorna true

PARTE FRONT-END

In angular usare bootstrap 5 + TS + MODULI Http + Varie Librerie

Si implementi un applicativo angular che avrà il routing per la gestione tra più componenti.

Si dovrà implementare un service per la gestione corretta delle Api pack-end.

I componenti che dovranno essere implementati sono i seguenti :

1-Un componente per la visualizzazione di tutte le note. verrà quindi richiamata la parte di java per la gestione corretta.

2-Un componente per la gestione e la ricerca di Nota by Id e by Autore
ovvero avrà una form che è in grado di prendere un id e un pulsante per confermare
Stesso discorso per la ricerca by id.

3- Un componente per la Gestione delle operazioni di Delete.
Ovvero una form che mi faccia cancellare per id
e un button che mi permetta da cancellare tutto .ovvero richiami il deleteAll di Spring boot

4- Un componente per l'inserimento e la modifica delle note.
ovvero una form che mi prenda in input autore e contenuto

Form che prende in input un id e una nota ed eseguire la modifica

Nota : Sarebbe conveniente aver creato Interface-o classe che possa modellare la NOTA:
(Vedi ultimo progetto)

NOTA :

Il sito dovrà essere presentabile esteticamente , magari usando nel modo più opportuno tutte le classi e css di bootstrap.

NOTA :

Per dare un riscontro visivo all'utente potreste utilizzare

La libreria di PrimeNg

<https://www.primefaces.org/primeng/toast>

Che ovviamente dovrà essere importata e installata correttamente a inizio progetto vedi Documentazione :

<https://www.primefaces.org/primeng/setup>