Reflektion Dagboken av Jonatan Hellqvist

Jag började med att skapa några klasser för dagboken, då applikationen egentligen bara har en enda uppgift ”att skapa/hantera inlägg” så blev det väldigt sparsamt med klasser då jag tyckte det var vettigast så.

Till själva diary klassen lade jag förutom id, title, text och datetime till en deleted variable för att enkelt kunna sätta en post som deleted med en 1: a. Provade lite olika värden till datetime både i java och på serversidan men stannade vid LocalDateTime för jag tyckte det gjorde jobbet bra och hade lite problem med formateringen av TimeStamp som jag först tänkte använda mig av.

Funderade mycket på om jag skulle dela upp kontroller klassen och repositoryn i mindre delklasser men då de ändå inte blev särskilt stora så kände jag ändå att det var ganska enkelt att följa dess mönster.

Valde att göra en soft delete query för delete mest för att jag ville prova hur det fungerade och tycker det fungerar bra, inte gjort något mer än så med det på serversidan utan det var mest för att prova bara.

Annars så gjorde jag en ”main query”, selectByDatetime som laddas med indexmappingen som visar alla inlägg från dåtid till nutid, sen separata för att söka efter inlägg, visa alla inlägg (som inte är deletade och visa dagens inlägg. Alla dessa fungerar på ungefär samma sätt.

Sedan också en ”UPDATE” för att edita posts. Lade också till en ORDER BY där det behövdes för att sortera inläggen efter datum.

I mina controllers så har jag såklart en för varje query men /get-todays-post så gjorde jag 2 variabler en med startOfToday som satte localdatetime min med (LocalTime.MIN) för att sätta denna till midnatt och sen startOfTomorrow med plusDays (1) för att få nästa dags datum.

/search-posts letar efter 2 ingående parametrar i en lista och uppdaterar sedan modellen med de posts som hittas i intervallet.

/new-post sätter värden på datetime, title och text och sparar sedan som en ny Diary i diaryRepository idt är inte med här för det sätts med Auto increment på serversidan.

/edit-post uppdaterar helt enkelt värdena på en post med getters och sedan UPDATE med queryn. Resten av mina Controllers är ganska simpla och behöver ingen närmre förklaring.

Det blev ganska mycket pill med thymeleaf för att koppla allt korrekt, hade först tänkt ha edit funktionen i samma formulär som man använder för att skapa en ny post men fick det inte att fungera på ett vettigt sätt så det fick bli ett nytt förifyllt formulär som popup i stället. Fick det till slut att fungera med några riktiga trollformler och lite javascript.

Annars blev det en hel del sittande med css och grids hit och dit men tyckte resultatet blev ganska kul med att new post formet ligger i en dagbok.

Känner att det var mycket med thymeleaf som klickade under denna uppgift och det känns skojj att också ha server biten att leka lite med framöver! Känns som det blir lite mer ”riktiga” applikationer då.