



Varför DAO / Service lager



Vanligvis kommer man få arbeta med underhåll av befintlig databas. Genom att dela upp sin Restful service i olika lager kan man stärka dess kraft utan att riskera fördärva databasen.

Genom att ha ett väl genomtänkt transaktionslager behöver man inte ha för många olika öppningar gentemot databasen som kan skada dess prestanda. Dessutom, ju fler joins och komplexa frågor, desto längre laddtid i databasen.

SCENARIO #1



Säkerheten ska förbättras.

Tokens ska implementeras på användare. Entiteten ser ut:

- Username
- Password
- Role
- Last_Login

Det vanligaste man börjar med är att skapa en ny variabel som ska spara ens token. Detta speglas direkt in på databasen och sätter NULL i alla rader. Inte vidare bra.

Om Entiteten måste ha en Java variabel som *inte* ska serialiseras, använd *@Transient*

Ett bättre sätt är dock att manipulera entitetet i DAO och Model byggnationen.



SCENARIO #2



Databasen har växt sig stor. Helt plötsligt säger FrontEnd teamet att de vill ha dels en sök funktion på en viss kategori samt en top-25 lista.

Detta *borde* åtgärdas i transaktionslagret men medans man väntar på tillstånd att utöka med frågor, använder man sig av getAll och eventuellt sparar in data i JPA cachén med hjälp av @Transient så man kan skapa en lista med top25 / kategori beroende på någon befintlig kolumn i produkttabellen.