

# Dokumentation - DiningManager - (DAT076)

Marcus Linderholm

Jonas Åkesson

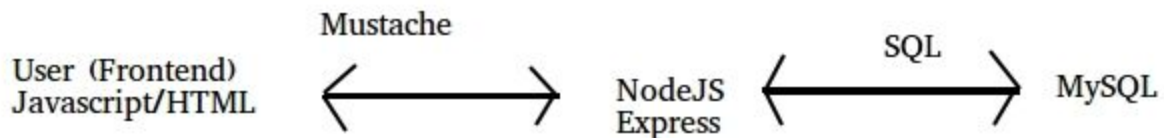
Sebastian Nilsson

DiningManager är skapat för att på ett väldigt simpelt sätt hantera bordsbokning på restaurang eller fik. Applikationen är inte unik i sin roll men de alternativ som finns<sup>1</sup> har dyra månadskostnader - här är ett gratis alternativ!

## Use-cases

Applikationen har två centrala use cases; antingen är du inte ännu medlem - och isåfall skapar du din användare samt din restaurang. Om du redan är medlem, loggar du istället in - och hanterar din restaurang och dina bokningar.

## Top-down



När klienten loggar in får vår server en SQL-query som ber om all information associerad till den aktuella användaren. Detta är, utöver namn och lösenord, är:

- En 2D-representation utav restauranggolvet i form av en tvådimensionell matris.
- En lista utav bord med respektive identifierare.
- En lista av bokningar, som innehåller tid och bord för bokningen, namn på person som bokat samt förväntad längd på vistelsen.

För att hantera vår databas har rena SQL-anrop använts, och datan har sedan templerats med Mustache.

Då klienten fått svaret målas, med matrisen som mall, restaurangen upp. Här tas även hänsyn till listan av bokningar, så att bokade bord målas i en annan färg än tillgängliga.

Klienten skickar regelbundet förfrågningar till databasen, för att se om nya bokningar blivit aktuella eller om äldre bokningar passerat.

---

<sup>1</sup> <https://easytablebooking.se/>

## Server eller klient

Att bestämma vad som ska ske på vår server och vad klienten själv ska göra kan vara komplicerat. Vår applikation sköter all beräkning på klientsidan, och vår server gör egentligen inget annat än att hämta och lämna data till databasen.

Som ett exempel på detta svarar vår server med en lista av bokningar för en särskild användare och klientsidan beräknar vilka av dessa som är bokade just nu.

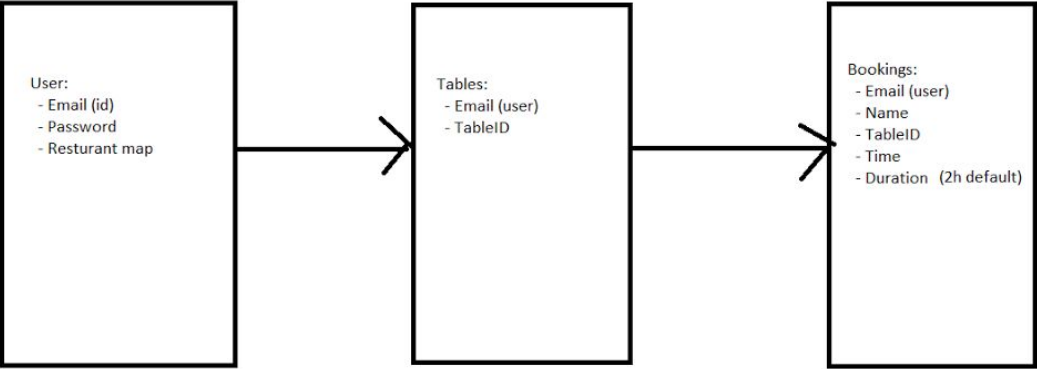
Allmänt sker interaktionen mellan server och klient på det sätt att användaren trycker på en submit-knapp som skickar data till express. Express skickar vidare denna data i form av anrop av databasfunktioner (ren SQL query). Svaret på Queryn hanteras efter att anropet har skett. Därefter skickas datan som hämtats till klienten via mustache.

## Databas

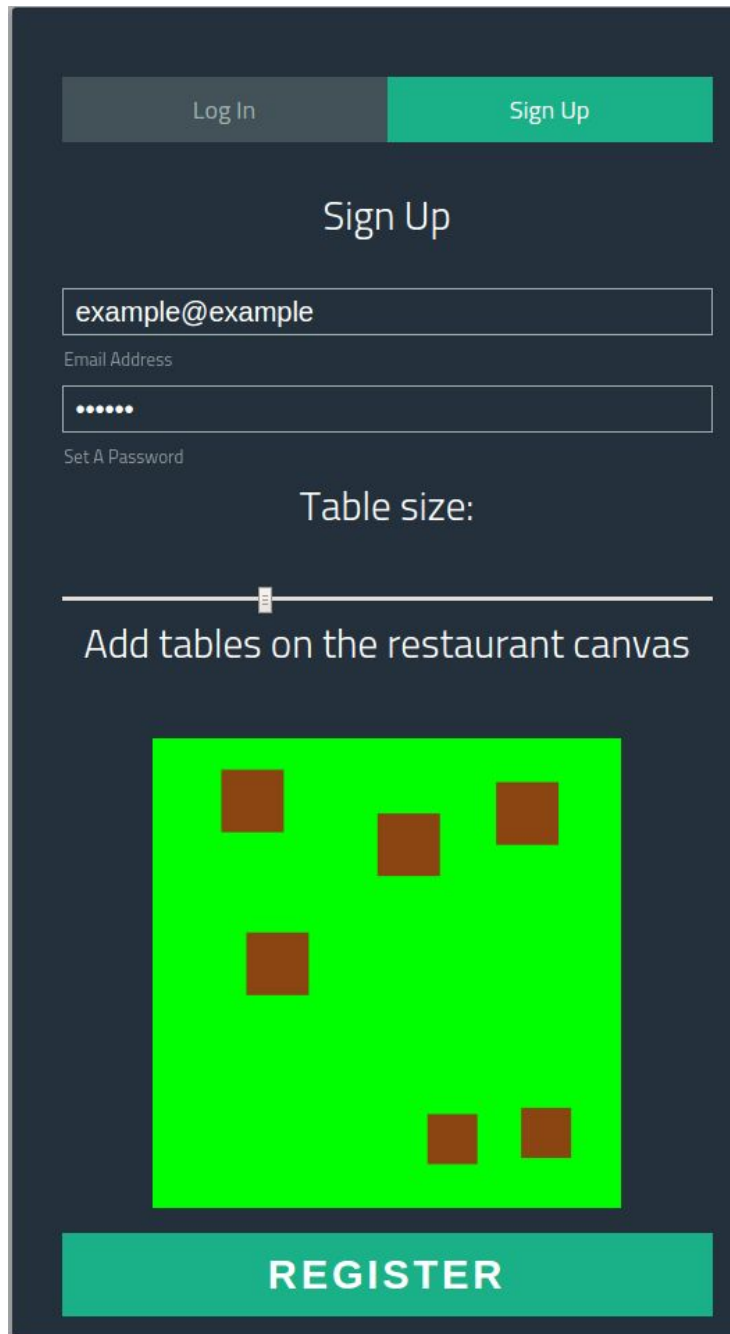
Databasen baseras på MySql och ligger för tillfället endast på localhost. Detta för smidighets skull.

Databasen består av 3 tabeller:

- User, där alla registrerade användare finns lagrade med Email som användar ID, ett lösenord samt restaurangen som användaren skapat vid inloggningen.
- Tables, där alla bord ligger lagrade med Email som nyckel till korrekt användare. Utöver det har varje bord ett unikt TableID inom användaren, dvs det finns aldrig två bord med samma ID inom samma användare.
- Bookings, där Email från användaren fortfarande finns med. Där finns även namn på gästen, TableID för att hålla koll på vilket bord som är bokat, tiden då bordet bokades samt duration på bokningen som vi har satt till 2h.



## “Sign up”-sidan



The image shows a 'Sign Up' form on a dark blue background. At the top, there are two buttons: 'Log In' (grey) and 'Sign Up' (teal). Below them is the title 'Sign Up'. The form includes an email input field with 'example@example' and a label 'Email Address'. Below that is a password input field with six dots and a label 'Set A Password'. A slider labeled 'Table size:' is positioned above the text 'Add tables on the restaurant canvas'. Below this text is a green square representing a restaurant canvas, with several brown squares indicating table positions. At the bottom is a large teal button labeled 'REGISTER'.

Bilden visar där man registrerar sig. Man väljer email och lösenord, sedan kan man välja olika bordsstorlekar och klicka ut dessa på mappen. Sedan registrerar man sig och mappen sparas ner tillsammans med email och lösenord till databasen. Värdet i textfälten skickas sedan till vår middleware med (Express) router post. Därefter hanteras värdet och används i våra databas-funktioner.

När du loggar in skickas värdet i textfälten på samma sätt till Express, därifrån skickas en SQL-query till databasen där användarnamn och lösenord valideras. Om valideringen stämmer sparas användaren i en (express)-session-variabel och användarsidan renderas med användarens (restaurangens) map över restaurangen och dagens bokningar.

## Restauranghanteringen

Logged in as: example@example

**Table 5 booked**

**TableID:**

**Name:**

**Time(hh):**

**Duration:**

**Table Time Name**

4 21:00:00 Sebbe

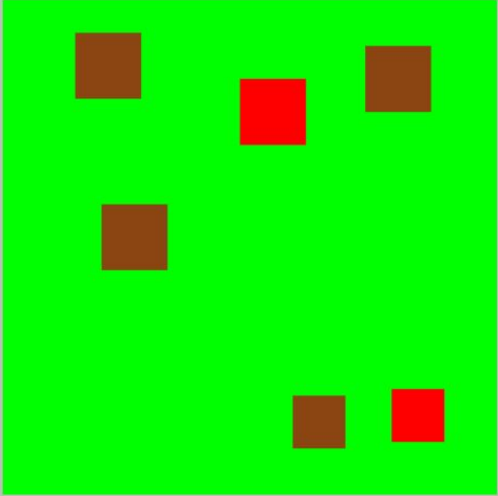
5 22:00:00 Marcus

2 23:00:00 Jonas

**Delete booking:**

**Table ID:**

**Time(hh):**



Bilden visar restauranghanteringen. Här finns det en map över restaurangen där borden som är bokade vid den aktuella tiden är röda och bord som inte är bokade just nu är bruna. Till vänster finns det möjlighet att boka bord vid en viss tid (hela timmar, och tiden är satt till 2 timmar). Vid bordsbokning skickas värdena i bokningsfälten till express. Där sker anrop av databasfunktionen bookTable som skickar en insert query till databasen. Efter att bokningen lagts till körs getBookings-funktionen som hämtar användarens bokningar och listar dem. Det finns även en tabell över alla bokningar som är aktiva samt vilka som kommer bli aktiva senare. Det finns också möjlighet till att ta bort en bokning. Genom att klicka på ett av borden på mapen får man det ID som tillhör det bordet.