

Sikkerhed og Performance: Vores Valg

Her er de sidste noter, der forklarer, hvordan vi holder systemet sikkert og hurtigt.

Sikkerhed mod Falske Scripts (XSS)

- **Risikoen, kort fortalt:** Vi skal forhindre, at en bruger kan skrive skadelig kode (f.eks. JavaScript) i et tekstfelt som Brugernavn, som så afvikles i browseren hos en receptionist.
 - **Vores forsvar:** Vores stærkeste våben er **Blazor** selv. Når Blazor viser data, omdanner den automatisk farlig kode til harmløs tekst på skærmen. Det er en indbygget sikkerhedsfunktion, vi stoler på. For en sikkerheds skyld sørger API'en også for basal validering af data.
-

Et Hurtigt API: Hvorfor vi bruger Eager Loading

- **Udfordringen:** Når vi henter en booking, har vi også brug for info om brugeren og værelsestypen. Spørgsmålet er, hvordan vi henter det smartest.
- **Den "dovne" metode (Lazy Loading):** Man kan hente bookingen først, og bagefter hente brugeren, og bagefter værelsestypen. For 10 bookinger kan det hurtigt blive til 20-30 ture til databasen. Det er langsomt og belaster serveren.
- **Vores "ivrige" metode (Eager Loading):** Vi bruger i stedet `.Include()` til at bede databasen om at hente **alt**, vi skal bruge, på **én gang**. Én bookingforespørgsel, én tur til databasen. Det er hurtigt, effektivt og det oplagte valg for vores API.