

# Disposition - remote-jukebox

Projektets formål er at lave et system, hvor du kan styre musik i et rum via en webside. Du kan konfigurere en playliste og derefter lade den afspille de angivne sange uden behov for en vedvarende forbindelse til klienten. Dette opnås ved at have en intern loop, som kan skifte sang, når den sidste sang er færdig.

**Flask + flask\_socketio** = Håndterer brugerforbindelser (HTML, JavaScript, Socket.IO)

**werkzeug.utils** = Forhindrer farlige stier (f.eks. ../../../../etc/passwd → etc\_passwd)

**hashlib** = laver checksums

**sqlite3** = Databasehåndtering

**os** = Filhåndtering

**python-vlc** = Styrer musikafspilning

**threading** = Multithreading og tråd-låsning

For at gøre dette har jeg brugt **Flask** med tilføjelsen **flask\_socketio**. De to biblioteker håndterer alle klientforbindelser. Når brugeren uploader nye sange, bruges **werkzeug.utils** til at sanitere data for at forhindre path traversal. Efter filen er blevet hentet, bruges **hashlib** til at lave en checksum af filen, og navnet bliver gemt i **sqlite3**, hvilket har to formål:

1. Det gør det nemt at opdatere navnet uden at ændre andet end en celle i databasen.
2. Det er en nem måde at afgøre, om en ny sang faktisk er ny.

Efter filen er tjekket, bliver den flyttet til sangmappen af **os**, hvorfra **python-vlc** kan læse og afspille de angivne sange, som bliver opbevaret i en liste af database primary key kaldet playlist. På grund af en interessant værgle i **python-vlc** er jeg nødt til at bruge **threading** til at oprette en tråd, som modtager et event fra main thread, når en sang er færdig. Derefter finder den næste sang og loader den ind i afspilleren.