

Sveriges Förenade Filmstudios

...

SFF API V1

Uppdraget

I detta uppdrag så ska ni jobba för Sveriges Förenade Filmstudios (SFF) och utveckla ett API och en hemsida där SFF kan registrera filmutlån till anslutna organisationer och föreningar.

SFF fungerar på så sätt att filmintresserade personer bildar föreningar som sedan blir medlemmar i SFF, som är förbundet för hela Sverige. SFF har rätt från filmbolagen att låna ut ett visst antal exemplar av olika filmer till dom lokala föreningar som sedan visar dem på exempelvis kulturbiograferna i sina städer. Förut skedde detta via blanketter och dyr transport av fysiska filmrullar - men idag sker detta såklart digitalt.

Det finns dock filmer som inte hunnit blivit digitaliserade ännu och de måste skickas med post och finns bara i ett visst antal! De digitala filmerna har vanligtvis också ett max antal som får vara utlånade samtidigt.

SFF önskar nu att du tar fram en lösning för att hålla koll på vilka filmer som finns tillgängliga för föreningarna att låna!

API och gränssnitt

API:et och klientgränssnittet ska bara användas av administratörer som jobbar på SFFs huvudkontor i Stockholm. De har kontakt med olika filmföreningar genom många olika kanaler och vill använda programmet för att uppdatera den nuvarande statusen på de olika filmerna.



API och gränssnitt

- Applikationen skall först och främst bestå av ett fungerande WebAPI, så att SFF lätt kan integrera informationen med deras andra existerande system.
- De vill dock gärna ha ett användarvänligt webbgränssnitt där de skall kunna se status på filmerna, men gärna också kunna administrera allt. Detta webbgränssnitt kan tänkas köras internt, så det finns i nuläget inget krav på autentisering.
- SFF har ännu inte tagit beslut om hur ett eventuellt klientgränssnitt skall se ut, men ber dig att göra klart API:et under tiden de tar beslut.
- Använd dig under tiden av Rest-klienten i VS Code, Swagger eller PostMan för att testa ditt projekt.

Tekniska krav - API

- Din API-lösning ska vara byggt i ASP.NET Core 5.0
- Alla endpoints ska ta emot eller returnera JSON-data
- I API:et ska resursen **film** finnas:
 - Det ska gå att lägga till en ny film
 - Det ska gå att ändra hur många studios som kan låna filmen samtidigt
 - Det ska gå att markera att en film är utlånad till en filmstudio (man får inte låna ut den mer än filmen finns tillgänglig (max-antal samtidiga utlåningar)
 - Det ska gå att ändra så att filmen inte längre är utlånad till en viss filmstudio (lämna tillbaka)
- En annan resurs som ska finnas är **filmförening**:
 - Det ska gå att registrera en ny filmförening
 - Det ska gå att ta bort en filmförening
 - Det ska gå att byta namn, och ort på en filmförening
 - Via API:et ska man kunna få fram vilka filmer som en viss förening har lånat
- När filmföreningar visat filmerna rapporterar de in ett betyg och ibland även en **triva** på film
 - Det ska gå att skapa ett nytt trivia objekt som håller koll på en liten anekdot för en viss film (Kan vara en string , kolla på IMDB för exempel)
 - Det ska gå att ta bort inskriven trivia
 - Det ska också gå att rapportera in ett betyg mellan 1 och 5. Det är viktigt att komma ihåg vilken studio som gav betyget för vilken film.

Exempel på endpoints:

[get] /api/v1/films

[post] /api/v1/films/14

Tips

- **Läs kravspecen noga** och skriv en egen Feature-lista i stil med “Lägg till ny film”, “Låna film” osv.
- Läs kravspecen och analysera vilka objekt du kommer behöva. Minns våra tidigare övningar där vi analyserade vilka substantiv och verb som fanns med i beskrivningarna.
- Notera att det inte måste gå att ta bort filmer, eller ändra deras detaljer.
- Det är ok att ni skapar extra resurser och klasser än de som finns beskrivna här, men detta är nuvarande minimumkrav från SFF i detta projekt.
- Innan alla objekt är klara och specificerade är det smidigt att använda en InMemory-databas, men använd sen SQLite som datalagring (eller annan lokal databas).
- Tänk på att kunden är en ovan beställare av datasystem. Kraven kan komma att ändras, eller att de är lite otydligt formulerade. Det går att ställa öppna frågor i Fråge-kanalen på Discord.
- Fokusera inte för mkt på exakta formuleringar i kravspecen, exempelvis "Det ska gå att markera att en film är utlånad till en filmstudio". Det är funktionaliteten med utlåning som är det viktiga, inte att det just är någon form av markering det har att göra med. Ibland kan en kundformulerad spec vara skriven på ett sätt som inte använder sig av de ord vi vanligtvis använder inom programmering för att beskriva saker, så då är det bra att fundera på vad är det egentligen de vill åt för funktionalitet.

Rapport

Du ska lämna in en sedvanlig teknisk rapport över din lösning. *Huvudfokus är att reflektera och motivera dina designval.* Inlämning: 21 April.

Krav på innehåll:

- En kort dokumentation över åtkomstpunkterna i API:et och vad de returnerar för information
- En kort reflektion över utformningen av API:et för att uppfylla kraven
- Beskriv dina Models och hur de hänger ihop med Entity Framework och databasen
- Beskriv projektets upplägg i sin helhet: Ett projekt eller flera, hur ser klass-strukturerna ut, använder du lager-arkitektur eller inte? Vad var motivet bakom det upplägg du valt?
- Identifierade du några alternativa möjliga lösningar när du skapade API:et?
- Beskriv för och nackdelar med de alternativa lösningarna

Svara förslagsvis på om åtkomstpunkterna var väl valda för att lösa uppgiften. Tänk på att motivera dina ställningstaganden.

Frågor och svar

- Ska API:t täcka både digitala och fysiska filmer eller enbart digitala?
 - Det spelar egentligen ingen roll, det kan vara digital eller fysisk. Oavsett har de ett max antal tillåtna utlåningar. Är de fysiska är det ju begränsat av faktiska föremål, är de digitala är de begränsade av licensavtal. Dock kan det vara bra att hålla koll på om det är en fysisk film eller inte, den fysiska måste ju skickas med posten, och det kan då vara bra att en förening kan registrera en adress.
- Hur fungerar det med uthyrningen? Hur länge har studion filmen och kan man köra flera visningar på en utlåning?
 - Det står inget i specen om uthyrningstid eller visningar, men om ni tänker att det är ju något som behövs så kan ni lägga till det.