Sveriges Förenade Filmstudios

SFF API V1

Uppdraget

I detta första uppdrag så ska ni jobba för Sveriges Förenade Filmstudios (SFF) och utveckla ett API som filmstudion kan anropa för att beställa film.

SFF fungerar på så sätt att lokala filmintresserade bildar föreningar som ingår 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.

De filmer som inte hunnit digitaliseras ännu är dock reglerade för hur många filmer som får lånas ut samtidigt, så SFF önskar nu att du tar fram ett API för att hålla koll på vilka filmer som finns tillgängliga för föreningarna att låna digitalt!

API och gränssnitt

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

Krav

- Din API-lösning ska vara byggt i ASP.NET Core 3.0 eller 3.1.
- Alla endpoints ska ta emot eller returnera JSON-data
- I API:et ska resursen **filmer** 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
- En annan resurs som ska finnas är **filmstudios:**
  - Det ska gå att registrera en ny filmstudio
  - Det ska gå att ta bort en filmstudio
  - Det ska gå att byta namn, och ort på en filmstudio
  - Via API:et ska man kunna få fram vilka filmer som en viss studio har lånat
- När filmstudiorna har 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.
- Notera att det behöver inte gå att ta bort filmer, eller ändra deras detaljer.
- Det är ok att ni skapar extra resurser och klasser än de som finns här, men detta är nuvarande minimumkrav från SFF i detta projekt.
- Innan allt är klart och specificerat är det smidigt att använda en InMemory-databas.
- Det finns inga krav på autentisering
- Tänk på att kunden är en ovan beställare av datasystem. Kraven kan komma att ändras.
- 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.

Klientgränsnitt

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.

Detta betyder att du får sätta upp en testmiljö i Postman att använda under tiden.

Tips: Det går att automatisera API-anrop och köra flera anrop i rad i Postman.

Rapport

Du ska lämna in en sedvanlig teknisk rapport över din lösning. *Huvudfokus är att reflektera och motivera dina designval.* Inlämning: Någon gång mellan 20-22 April.

Krav på innehåll:

- En länk till gitrepot med implementationen av REST APIet
- En kort dokumentation över åtkomstpunkterna i API:et och vad dom returnerar för information
- En kort reflektion över utformningen av API:et för att uppfylla kraven
- Ge exempel på hur du använt dig av LINQ för att accessa eller modifiera data i ditt projekt
- Beskriv dina Models och hur de hänger ihop med Entity Framework och databasen
- Identifierade du några alternativa möjliga lösningar när du skapade APIet?
- Beskriv för och nackdelar med de alternativa lösningarna
- Vad var motivet bakom de lösningar du valde?

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

Bra-att-ha-länkar:

Entity Framework Core

https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx

Linq-to-Entities Query:
https://www.entityframeworktutorial.net/querying-entity-graph-in-entity-framework.aspx

One-to-many conventions

https://www.entityframeworktutorial.net/efcore/one-to-many-conventions-entity-framework-core.aspx

One-to-one conventions

https://www.entityframeworktutorial.net/efcore/one-to-one-conventions-entity-framework-core.aspx

Snabbt klar?

Här är några extra features som du får lägga till om du är klar tidigt:

- Skriv ett eget middleware för autentisering. Det kan vara ett väldigt enkelt och osäkert sätt, poängen är att skriva ett eget middleware.
- Koppla din API-server till en riktig SQL-databas
- Lägg till möjligheten att ladda upp en bild som thumbnail till filmerna
- Skriv ett enkelt konsolprogram som kan fungera som klient
- Alternativt gör ett grafiskt användargränssnitt istället för konsolprogram

Frågor och svar

- Ska APIt 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.
- 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.

Tidsplanering

Onsdag 1 april

Vecka 14

- I grupper om fem:
  - Repetera designmönster: Vilka designmönster ingår i koden för ert API?
  - Fundera kring motivationen till att just de designmönstren används
  - Vilka endpoints behöver ert API för att uppfylla kravspecen?
  - Diskutera och ta fram klassdiagram
  - Vilka one-to-one eller one-to-many relationships kan ni behöva?
  - Om flera alternativ finns, motivera och välj den mest lämpade lösningen
- Börja implementera och utveckla SFF API V1

Torsdag 2 april

- Fortsättning av implementation
- Tid för enskild handledning. Boka tid via spreadsheet:
  https://docs.google.com/spreadsheets/d/1clxn4tFlN32s1BFIRlT_1CpAcSgFNOU4jFtbzrHygZA/edit?usp=sharing

Vecka 15

Måndag 6:e april

- Förmiddag: Föreläsning/genomgång efter önskemål (Ex Entity Framework/ORM)
- Eftermiddag: Fortsätt med kodning

Tisdag 7:e april

- Tid för enskild handledning. Boka tid via spreadsheet: https://docs.google.com/spreadsheets/d/1clxn4tFlN32s1BFIRlT_1CpAcSgFNOU4jFtbzrHygZA/edit?usp=sharing

Vecka 16

Onsdag 15:e april

- Förmiddag: Presentation av nya kundkrav!
- Eftermiddag: Fortsätt med kodning

Torsdag 16:e april

- Eventuellt föreläsning på fm
- Tid för enskild handledning. Boka tid via spreadsheet: https://docs.google.com/spreadsheets/d/1clxn4tFlN32s1BFIRlT_1CpAcSgFNOU4jFtbzrHygZA/edit?usp=sharing

Fredag 17:e april

- Kodgranskning i grupper. Underlag till rapport.

Kompetens

- Identifiera alternativa lösningar vid skapandet av applikationer.
- Självständigt jämföra och värdera alternativa lösningar
- Motivera och välja den mest lämpade lösningen
- I grupp med andra diskutera olika designlösningar på ett givet problem med ASP.NET
- Bygga och utvärdera en objektorienterad applikation i ASP.NET
- Förstå och applicera för uppgiften relevanta designmönster.

Färdigheter

- Använda ett ORM-ramverk
- Kommunicera med andra datakällor via OOP
- Integrera befintlig programmeringskod