



---

# FILMSTUDION

---

Inlämningsuppgift - REST och anrop från användare av API:er



DEN 28 FEBRUARI 2021

WIVIANNE GRAPENHOLT  
grwi20ql@student.ju.se

## Innehåll

<b>REST</b> .....	2
PUBLIC .....	2
USER .....	2
AUTENSIERADE ANVÄNDARE .....	2
<b>Implementation</b> .....	3
<b>Säkerhet</b> .....	3
<b>Klientgränssnitt</b> .....	4

## REST

### PUBLIC

Jag har valt att designa om en hel del, nu finns det en Public-controller för att uppfylla kraven på alla publika åtkomstpunkterna samt anpassat resurserna som returnerats.

I PublicMovieResource exkluderar jag allt som har med FilmCopies att göra eftersom det inte ska exponeras för icke autentiserade användare.

Dessa åtkomstpunkter anser jag följer REST eftersom de alltid returnerar samma respons.

### USER

För att det ska gå att registrera en FilmStudio så har jag en åtkomstpunkt för att registrera FilmStudio och User samtidigt. genom att använda automapper för att mappa RegisterFilmStudioModel med både FilmStudio och User. Jag sätter IsAdmin till false direkt i resursen för att vara säker på att ingen användare kan ha dubbla roller.

RegisterAdminUserModel är en separat åtkomstpunkt för att registrera User(admin). Där IsAdmin sätts till true och FilmStudiold till 0 direkt i resursen för att vara säker på att ingen användare kan ha dubbla roller.

Jag tycker att det var den mest logiska lösningen då det inte finns något krav (eller behov av) att registrera en användare som varken är FilmStudio eller admin eller att en Filmstudio skall kunna ha flera användare.

Det finns också en åtkomstpunkt för autentisering av användare som även returnerar JWT token.

### AUTENSIERADE ANVÄNDARE

Jag har korrigerat adressen till åtkomstpunkten och resursen för att hämta en specifik Movie. MovieResource inkluderar dels antalet FilmCopies och en array av dessa.

För att uppfylla kravet på att inloggad användare ska kunna se tillgängliga filmer har jag lagt till en GET metod i Movie-controllern som returnerar en array MovieResource.

För request body till att skapa eller uppdatera en Movie använder jag en separat resurs, CreateUpdateMovieResource, där jag validerar en del av inputen och returnerar sedan Created / Ok samt en MovieResource. Främst eftersom antalet tillgängliga kopior och de tillgängliga kopiorna är till större nytta för användaren än bara maximala antalet kopior.

För att hyra och återlämna filmer använder jag PUT, eftersom IsRented och FilmStudiold ändras (uppdateras) känns det som det mest logiska. I FilmCopyResource har jag lagt till MovieName för att resursen enkelt skall användas för att i klientgränssnitt kunna visa filmstudions alla hyrda filmer, återlämnad eller hyrd film. En förbättring är också att det inte längre krävs någon request body. Åtkomstpunkten svarar med Created samt den uppdaterade resursen. Jag är dock inte helt övertygad om att Created är det bästa svaret men det fungerar.

Över lag är jag nöjd med och anser att mina åtkomstpunkter och resurser uppfyller kraven på Webb-API:et på ett REST-fullt sätt.

## Implementation

Jag har valt att ha fyra domänklasser: User, FilmStudio, Movie och FilmCopy. Jag tolkar "synlig" som en responsresurs, hoppas att det är korrekt uppfattat.

För **User** finns det ingen synlig resurs utan här returneras en förenklad User: Id, Username(email) och JWT token.

**FilmStudio** representeras av PublicFilmStudioResource där endast FilmStudiold exkluderas eftersom det inte finns någon anledning att visa detta.

**Movie** har två PublicMovieResource och MovieResource som jag nämnde tidigare har den publika resursen ingen information om antalet tillgängliga kopior medan MovieResource är uppdaterad med en IEnumerable AvailableFilmcopies och AvailableCopies som är antalet tillgängliga kopior. Jag gjorde bedömningen att det var mer användbart för användare än AmountOfCopies som endast är relevant vid skapande eller uppdatering av en film.

**FilmCopy** har en resurs FilmCopyResource som används som response för filmstudios hyrda filmer, hyra samt återlämna. Här valde jag att utesluta IsRented och lägga till titeln på filmen i MovieName för att enklare kunna presentera hyrda filmer för användare på ett snyggt sätt.

I MovieResource AvailableFilmcopies exponeras FilmCopy direkt då det mer är tänkt för admins att ha koll via postman.

## Säkerhet

Jag har korrigerat tidigare brister gällande registrering av FilmStudio och admin, se under REST. Beslutet att skapa en PublicController har gjort det smidigt att skilja på vilka åtkomstpunkter som kräver autentisering. Resterande kontrollers kräver autentisering, bortsett från åtkomstpunkterna i UsersController.

Jag använder Core Identity för att kontrollera User och där det behövs kontrolleras typ av autentiserad användaren kontrollerar jag vad IsAdmin är satt till. För att säkerställa vilken filmstudio som är inloggad används numera user.FilmStudiold.

Inloggning i klientgränssnittet sker via ett formulär som läser in och skickar nödvändig data i en request-body med POST till /users/authenticate responsen tas emot och sparar Bearer token i app.globaltoken samt user.id i app.id. Vid utloggning nollställs helt enkelt dessa.

## Klientgränssnitt

Här måste jag få erkänna att jag är väldigt nöjd!

Det som kändes viktigast att få med och det jag startade med var de publika åtkomstpunkterna, dvs att det gick att visa filmstudior och filmer.

Jag anser att jag uppnått ett användarvänligt klientgränssnitt genom tex enkel layout, tydliga instruktioner till användaren, att registrering leder till att logga in formuläret visas som vid inloggning i sin tur automatiskt renderar filmstudions "min sida" med eventuellt lånade filmer.

När en inloggad filmstudio väljer "filmer" i menyn visas även antalet tillgängliga kopior. Om det inte finns tillgängliga kopior visas texten "inge tillgängliga kopior" överst på film-"kortet" och hela kortet har en dämpad style jämfört med övriga filmer.

Jag valde att visa "återlämna film" under hyrda filmer och "låna film" under "filmer" eftersom det är där informationen om filmen (filmens id) finns tillgänglig för användaren.

För att ytterligare höja användbarheten har jag ansträngt mig för att användaren alltid skall få information från olika funktioner även när tex filmstudion inte har några lånade filmer eller bekräftelser på lånad / återlämnad film.

Jag har sett till att hålla designen enkel och tydlig och det går utmärkt att navigera på sidan med endast tangenter.