

## Get Things Done Permanently

För denna inlämningsuppgift har jag valt att göra API enligt MVC samt gränssnitt med Javascript. Att koda API enligt MVC gör att separationen görs automatiskt genom att använda Models och Controllers. Detta gör att det blir lättare att koda samt felsöka.

Åtkomstpunkterna för mitt Api är:

**<https://localhost:44319/api/User>**

Post: Skapar användare

Delete: Tar bort användare

Get: Visa alla användare

**<https://localhost:44319/api/Login>**

Post: Inloggning, skapar Jwt-Token

**<https://localhost:44319/api/WorkList>**

Post: Skapar en WorkList

Get: Tar fram alla WorkList

Id/ Get : Ta fram enskilt WorkList mha Id

Id/ Del: Ta bort WorkList

Id/ Put: Ändra namn på WorkList

/id/Collaborator Put: Lägg till/ta bort Collaborator för WorkList

**<https://localhost:44319/api/WorkList/Id/task>**

Post: Skapar task

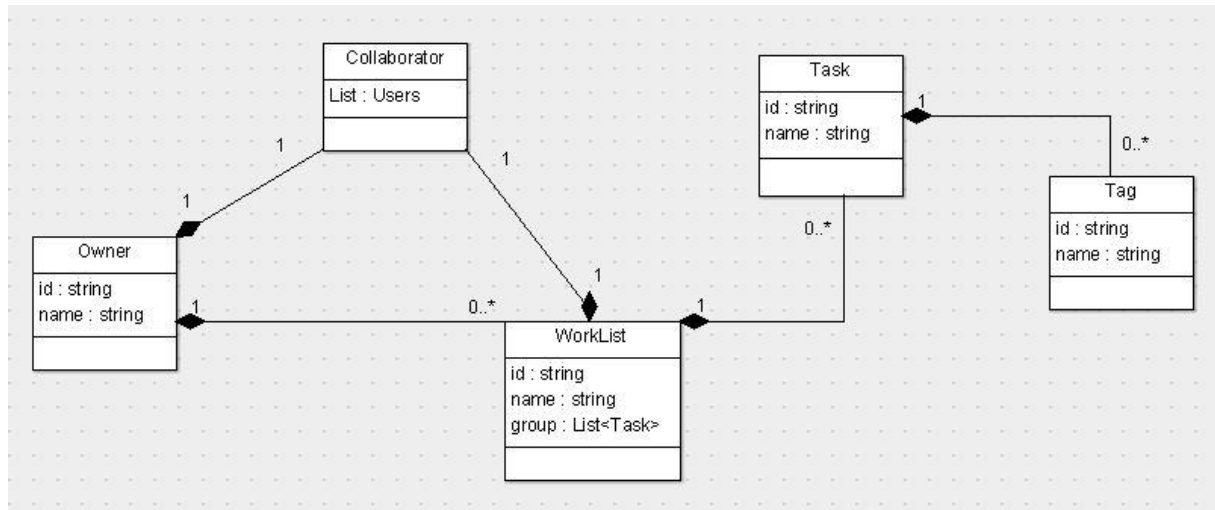
Get: hämtar alla task

/taskid Del: ta bort tak

/taskid Put: ändra status/namn på task

/taskid Post: skapar Tag till en task

För att underlätta arbetet har följande skiss tagits fram.



#### Reflektion:

Uppgiften upplevdes inte svår men det var många krav som behövdes uppfyllas. Därför känns det från början att uppgiften skulle ta lång tid pga. allt som skall implementeras.

All data skapades och sparas i en SQL serverdatabas. Jag var noggrann i min skiss att undvika n:n relationer då dessa kan skapa problem.

För mina användare använder jag Identity.EntityFrameworkCore som hjälper mig att lagra användarna i databasen. Jag fick lägga lite jobb att implementera själva skapandet och borttagande av användare, vilket kunde underlättas genom scaffolding om jag hade haft gränssnittet i MVC. Överlag var det inte svårt, det gäller bara att ta reda på vad UserManager/SignInManager gör.

För autentisering ville jag testa att göra token istället för Basic inloggning där användarnamn och lösenord skickas. Token gör att det bli säkrare då användaruppgifter enbart skickas i intervaller. Överlag var det inga problem att få en fungerade token inloggning.

För användargränssnittet så fick jag fetcha min endpoints och fylla ut sidan efterhand med information. Jag kunde lösa de flesta kravspecs men det är några som jag fick problem med.

När jag i PostMan lägger till en Collaborator till en worklist så visar den all information som svar.

(<https://localhost:44319/api/worklist/1/collaborator>) Put name:test@hotmail.com

Men när jag sedan gör en Get på worklist/1 så kommer problemet. Det visar bara id för Collaborator och inte namnet (null).

Jag behöver få fram alla namn för att kunna markera dessa som Collaborators till min Worklist. Men i och med den bara visar null har jag inget sätt att gå vidare. Jag misstänker att det är något som saknas när jag skickar worklist från databasen till mina controllers (försökte med include och Theninclude men verka inte funkar).

Jag kunde inte lösa det och därför kan jag inte heller i gränssnittet visa vilka användare som är Collaborater(även om de är inlagd enligt postman). Förstår inte varför det visas i postman som svar när de lägger till Collaborator till lista men inte när man i efterhand använder Get.

Detta återkommer sedan även för min task, samma där task visas som null med Get, men inte när man skapar det.

**<https://localhost:44319/api/WorkList/Id/task/1>**

Dessa två null för Collaboraters och task gör att visa kravspecs inte kunde uppfyllas i gränssnittet.

- Man ska via en åtkomstpunkt kunna söka på uppgifter i alla sina listor efter vilka taggar den har.
- En inbjuden användare kan se och markera uppgifter i listan som avklarade.
- Bara användaren som skapade listan kan ta bort en uppgift eller listan.
- Endast användaren som skapade listan ska ha möjlighet att skapa nya uppgifter.

Punkt 3-4 är löst i API då worklist är bunden till en användare(owner), därför kan jag göra en check mot user.Identity göra så endast ägaren kan ta bort och skapa lista/uppgift.

Av: Fon To

Github: <https://github.com/fonto1/ToDo.git> , Branch 2,0

