

# REST. Implementer en simple REST API

## Tidligere opgaver

- [Design REST API](#)

## Opgave 1: Fra design opgaven implementer en simple REST Service

### Opgave 1.1: Opret projekter og model-klasse

Du skal oprette et nyt projekt (ASP.NET Core Web API), hvor du skal implementere din REST-service

Du skal desuden oprette et nyt projekt (Class Library – til C#) inde i din solution, hvor du skal implementere din modelklasse (**Du vælger én af dine modelklasser**)

Husk at oversætte dit library (dvs. build), samt i dependencies i dit REST-service projekt skal du tilføje en reference til dit library.

### Opgave 1.2: Opret funktionalitet og at data kan 'gemmes'

Du skal i dit REST-projekt oprette en mappe 'Managers', hvor du skal definere et interface (fx IxxxxxManager, hvor xxxxx er navnet på din modelklasse), som skal indeholde metoder til de 4 CRUD funktioner, samt 'Get' fra et specificeret Id:

- GetAll()
- Get(int Id) – kan også være en string eller andet
- Create(XXXX xxxx) – XXXX er din model-klasse
- Delete(int id) – samme kommentar som ved get(int id)
- Update(int id, XXXX xxxx) -- samme kommentar som ved Get(int id) og Create(XXXX xxxx)

Du skal tillige implementere dit interface i en manager-klasse fx: xxxxManager, denne implementering skal bygge på en statisk liste, der benyttes til at 'gemme' data.

### Opgave 1.3: Opret en Controller til at udstille din REST-service (lave API'en)

Du skal i mappen controller oprette en ny klasse ('API Controller with read/write actions' – ikke MVC). Navngiv den XXXXs**Controller** (husk flertal på navnet af din modlklasse).

Opret et privat instans felt i din controller klasse af din manager klasse. Dette objekt kan du benytte til at implementere dine controller-metoder (= REST API).

### Opgave 1.4: Prøv din REST-service

Kør din REST-service og prøv de forskellige metoder igennem den brugergrænseflade (swagger) der popper op.

Virker REST-servicen?

## Opgave 1.5: Forklar din REST Service

Forklar :

- Hvorledes er din URI til fx Get?
- Hvorledes angiver du i C# koden at du benytter Get hhv POST (eller de andre)?
- Ved Post, hvorledes får Controller-metoden adgang til det nye objekt, der skal oprettes?
- Kan du ændre på din URI uden at 'rename' din controller-klasse? hvis ja hvordan?
- Hvad sker der hvis din liste i Manager-klasse **IKKE er statisk?**

*Hvis du ønsker en mere Tutorial tilgang [se denne opgave](#) der har en Item-model-klasse*