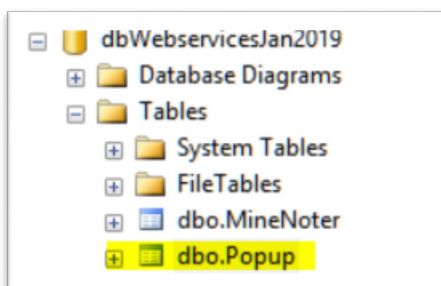


Opgave 5 – Søgefunktion i tekster med brug af webservice

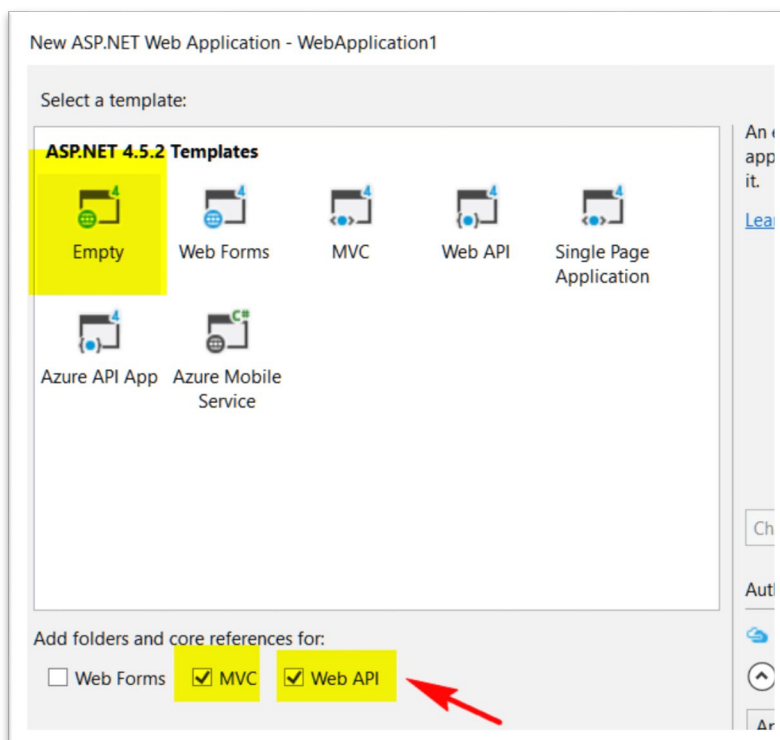
I denne opgave brug fx

- En database-tabel med en del tekst - fx Citater/Quotes, faktabokse/popup-opgaven, skuespillere eller jokes – kig dine databaser/tabeller igennem.

I eksemplet herunder har jeg valgt at (gen)bruge tabellen fra popup-opgaven (JavaScript/jQuery) med faktabokse, som jeg har scriptet ind i databasen, hvor jeg også har note-tabellen



- Et webprojekt med WebAPI-funktioner – dvs. oprettet, hvor du har sat kryds i Web API:

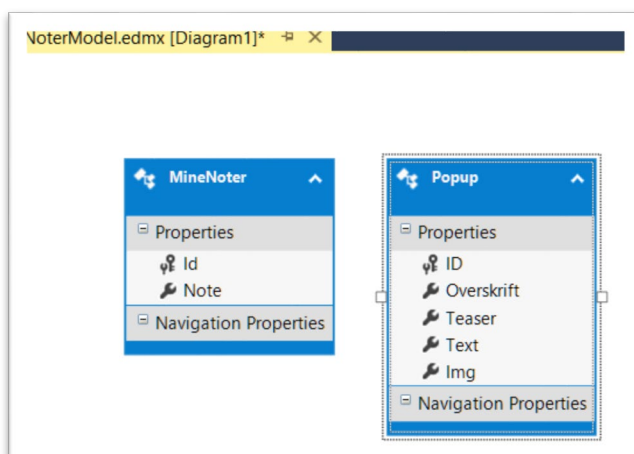
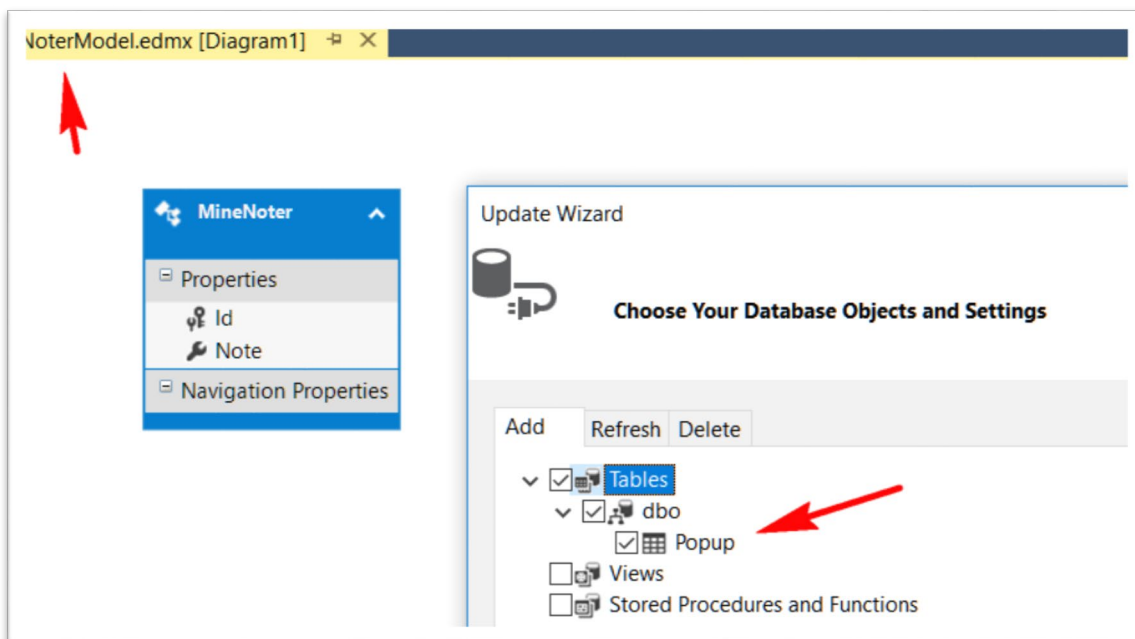


... (gen)brug gerne det webprojekt, hvor du har arbejdet med Noter (opret, ret og slet noter med brug af WebAPI) – du må gerne have flere modeller tilknyttet samme projekt.

- **Lidt basic viden om at arbejde med et WebAPI/en webservice** og at bruge JavaScript (Fetch, XMLHttpRequest eller jQuery Ajax) til GET/POST/PUT/DELETE data

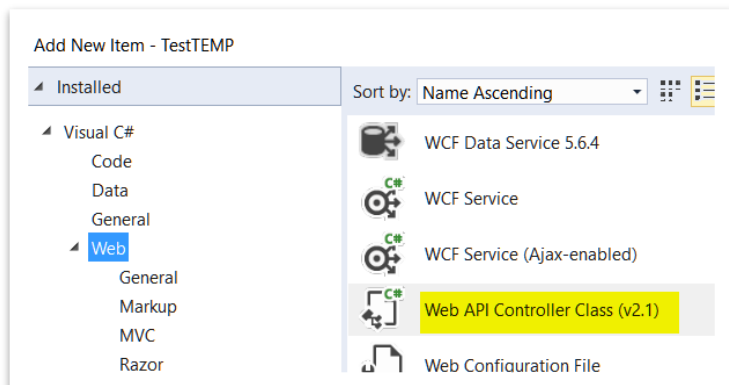
Modellen til din tabel med tekster

Hvis du har lagt tabellen med tekster, som du vil søge i, ind i den database, du allerede bruger, så skal du blot opdatere din projekt-Model – "Update Model from Database" – så den ekstra tabel bliver lavet til en model:

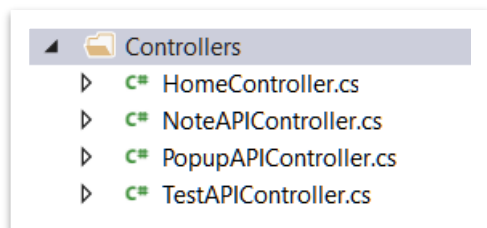


APIController

1. Når modellen, Layoutpage og alt det andet basic er på plads, så **opret en ApiController** (højreklik på Controllers-mappen >> New Item >> Web API Controller Class



2. **Giv api-controller'en et navn som matcher dataene** – jeg valgte navnet PopupApiController.cs. Husk at indsætte ordet API i dine api-controller'e, så du nemt kan se forskel på de almindelige controller'e og api-controller'ne:



3. **Tilpas den første Get-metode** (den der leverer en liste/IEnumerable), så den udtrækker alle data/tabel-rækker fra tabellen:

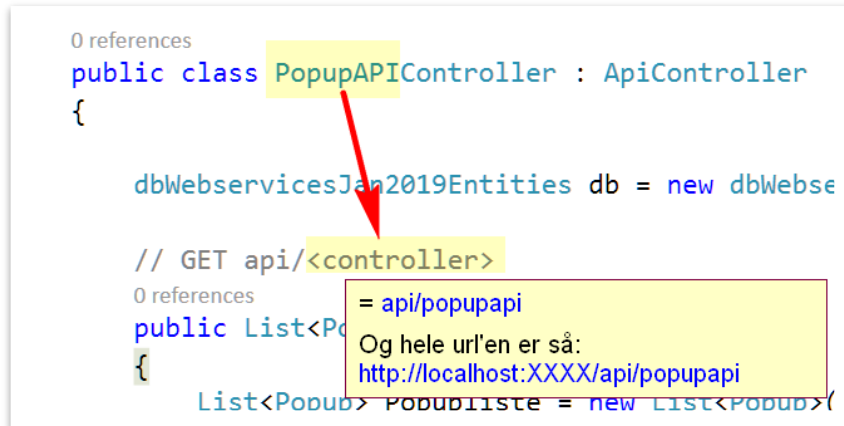
```
namespace TestTEMP.Controllers
{
    0 references
    public class PopupApiController : ApiController
    {
        dbWebservicesJan2019Entities db = new dbWebservicesJan2019Entities();

        // GET api/<controller>
        0 references
        public List<Popup> Get()
        {
            List<Popup> Popupliste = new List<Popup>();
            Popupliste = db.Popup.ToList();
            return Popupliste;
        }

        // GET api/<controller>/5
    }
}
```

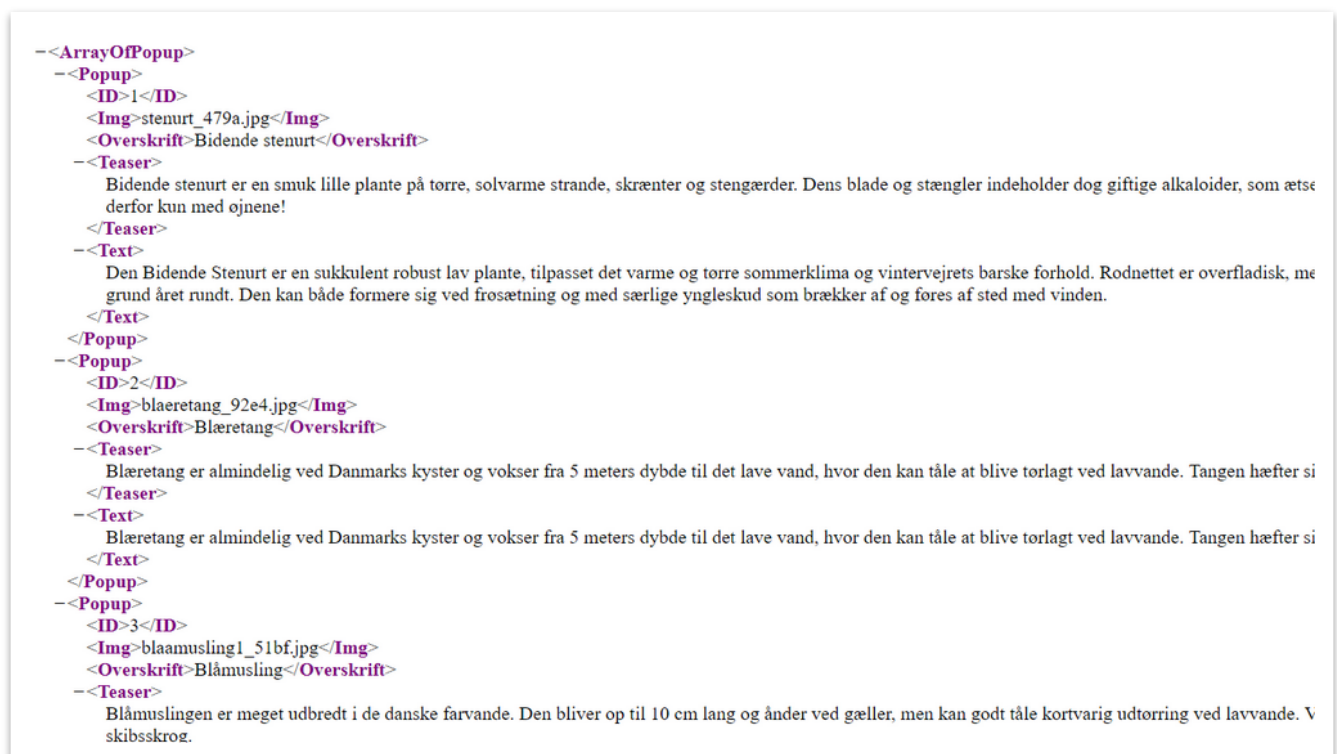
4. **TEST din webservice/Get-metode ...** Sørg for at dit webprojekt er startet, og indtast så adressen til din get-metode i browseren – for mit projekt er det:

<http://localhost:11369/api/popupapi>



HUSK AT BUILDE FØRST - og herefter ctrl+F5 ... eller brug debugging (som automatisk builder for dig)

5. **Tjek, at de viste data matcher det, du forventer at se** – her er det alle faktabokse/popup-tekster:



6. Hvis du ikke får teksterne frem, må du i gang med at **lede efter fejl ... gå ikke videre, før det virker.**

Spørg om hjælp, hvis du er tvivl 😊 ...

7. **Næste step er så at lave et helt almindeligt View** (husk først at oprette et `ActionResult` i fx HomeController (almindelig controller – IKKE api-controller):

```
0 references
public ActionResult FaktabokseSoeg()
{
    return View();
}
```

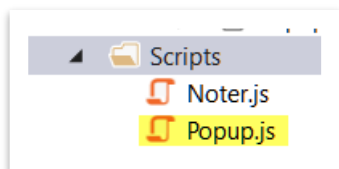
8. **I viewet/cshtml-siden – gør klar til at modtage data fra JavaScript/webservice** – fx noget af dette:

```
<h2>Søgning i fakta-tekster</h2>

Skriv et søgeord: <input type="text" value="" placeholder="Søg noget" />

<div id="faktabokse">
    <div id="resultat"></div>
</div>
```

9. **Opret en js-fil og tilføj den til dit view/din side (eller til layout-siden)** – kald den fx `Popup.js`



10. **I js-filen** – opret din foretrukne metode til at hente data fra en webservice – `Fetch()`, `jQuery.ajax` eller `XMLHttpRequest`. Genbrug din kode fra tidligere projekter – det er NÆSTEN kun url'en til webservicen, du skal ændre i.

11. Nu har du url'en/adressen til din webservice.

Og du har set et eksempel på, hvilke data du får, når du kalder webservicen.

Og du har en del eksempler fra tidligere opgaver på, hvordan du looper data fra en webservice (fx `StarWars-api`'et, `postnumre-api`'et, `OpenWeatherMap` osv.).

... så **prøv nu at bruge eksemplerne og din viden fra de tidligere opgaver til at kalde din egen webservice ... og loop alle faktabokses tekster ind på siden i en div eller hvad du nu foretrækker.**

Opgaven opsummeret

1. Lav først en Get-metode (eller genbrug den, der er der) i din ApiController, som henter alle dine tekster – og loop teksterne ud på en side/et View. Ignorer input-feltet – det skal du først tage dig af i punkt 2 herunder.
Gå ikke videre før det virker ...

Punkt 1 herover svarer til alt det, der er beskrevet herover ...

Fra punkt 2 herunder er du lidt mere på egen hånd:

2. Lav herefter en Get-metode mere (eller genbrug en af dem, der er der) i din ApiController, som kan modtage en parameter (søgeordet), og som herefter returnerer alle de tabelrækker, som kan matche det medsendte søgeord.
3. Og bagefter en side/et View med et søge-inputfelt ... som ved keyup (eller oninput) kalder en **javascript-funktion** som:
4. Modtager inputtet (value) fra inputfeltet.
5. Med fetch, XMLHttpRequest eller jQuery/Ajax kalder (requester) din webservice (og sender søgeordet/input-value med)
6. Modtager responset webservicen (alle matchende tekster) – test først resultatet med console.log()
7. Looper resultatet (alle de matchende tekster) ud på siden i en <div> eller lignende.
8. Tjek for fejl og mangler – sørg for, at det også fungerer ved 0/nul søgeresultat
9. Finpuds resultatet – style det evt. lidt

Tip

Hvis du vil lave et udtryk (i controller) som kan SØGE efter noget, dur det ikke med == for så søges der helt specifikt. I stedet har du brug for Contains eller StartsWith (lidt afhængig af, hvordan du vil søge). Fx:

```
db.Popup.Where(q => q.Text.Contains(soegeord)).ToList();
```

Eller hvis du vil søge i flere felter (både i Overskrift, Teaser og Text):

```
Db.Popup.Where(q => q.Text.Contains(soegeord) || q.Teaser.Contains(soegeord) || q.Overskrift.Contains(soegeord)).ToList();
```