

To-do applikation skrevet med C# .NET Core 2.1

WebService/Problemstilling:

Et interface muliggør oprettelse af korte tekstobjekter, som efterfølgende kan genkaldes med en parameter, der tillader en binær tilstandsændring.

Det er tænkt som til beskeder eller noter, som så kan markeres som sande eller falske eller relevante/irrelevante.

Httpprotokollen er forholdsvis universelt tilgængelig og velegnet til brug af en API defineret af såkaldte REST begrænsninger, som i form deklarative ruter tillader oprettelse og gengivelse af data i en database. Brugeren er frit stillet til at gengive indholdet med en mere eller mindre grafisk opsætning.

Benyttelse:

For at tilgå applikationen gøres brug af httpprotokollen, der eksempelvis er integreret i webbrowsere og som muliggør fremstilling og afsendelse af data ved hjælp af APIens resurser, og som benyttes med metoderne GET, POST, PUT, der tilsammen udgør behandling af tekstobjekterne i samlet omfang.

For at genkalde den samlede list med objekter foretages der en GET forespørgsel til api/todo og der returneres i form af et JSON formateret array.

Et kald til api/todo med metoden POST vil i stedet oprette et nyt tekstobjekt.

For at hente et bestemt objekt kaldes api/todo/id og ligeledes med GET metoden. Bruges der PUT i stedet, vil applikationen opdatere tekstobjektets førnævnte tilstandsparamter.

Databaseforbindelse:

Forbindelsen til databasen sker ved at pege på en overordnet database, enten lokal til test eller serverbeliggende som tiltænkt.

Filen appsettings.json oprettes og skal indeholde de nødvendige specifikationer.

Således med eksempel på en lokal database tilgang:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=(localdb)\\MSSQLLocalDB;Database=ToDoApp;"
  },
  "AllowedHosts": "*"
}
```

Derefter oprettes databasen og dens tabel ved at eksekvere kommandoerne:

```
dotnet ef migrations add InitialCreate // opretter Migrationsmappen med 3 filer
dotnet ef database update
```

Koden er skrevet i programmeringssproget C#, nærmere betegnet .NET Core 2.1

Installation består af:

- SDK 2.1.500
- Runtime 2.1.6

Browserbaseret brugerapplikation som eksempel:

Der er vedlagt en html side, som tillader brug af applikationen ved hjælp af en grafisk brugerflade. Siden indeholder alt nødvendig kode til at foretage de nødvendige opkald fra brugerens enhed.

På linje 255 ses hvordan tekstobjekternes indeks hentes og lagres på siden, således at de senere kan opdateres ved hjælp af et simpelt klik på musen.

```
checkbox.id = "item" + items[i].id;
```

Niels Gregersen 23-11-2018