HTTP Protokol

HTTP Protokollen bruges i dette tilfælde, til at oprette kommunikation mellem REST servicen og klienten, dette gøres ved at der sendes et request, hvori der ligger et metodekald, hvilket vores REST service videre kan bruge til at håndtere kaldet, og sende den korrekte data retur.

Request

Et request er hovedsageligt bygget op af følgende:

Request-line med: Metode, URI, Version.

Header-field

Body

Metode fortæller vores REST service, hvilken metode vi gerne vil have fat i (GET, POST, PUT, DELETE)

URI fortæller hvilken ressource vi gerne vil have fat i.

Version handler om hvilken http version vi har fat i.

Headeren kan bruges til flere ting, men bruges hovedsageligt til at fortælle vores REST hvordan vi regner med at responset kommet tilbage, såsom karaktersæt, sprog mm.

Det kan også bruges til diverse sikkerhedsmæssige muligheder, ved at REST'en kun accepterer f.eks data fra en given adresse, eller at det SKAL inkludere specifikke headers.

Body bruges ikke altid, men hvis man skal bruge f.eks POST, altså tilføje et nyt objekt, kan data til det nye objekt skrives i http'ets body.

Response

Response er bygget op nogenlunde på samme måde, der er dog lidt forskel helt i toppen, et response har en Response Line, hvori man finder følgende:
Version, Status Code, Status Message.

Version er igen direkte relateret til hvilken version af http.

Statuscode er nogle koder der symboliserer hvordan det gik til med vores request;

Som regel bruger vi 200 eller 400-koder i vores Rest

De gode responses er mellem 200 og 299, så alt hvor det gik fint

Fejl-koderne er mellem 400 og 499.

REST

Vores REST service bruger data annotations til at fortolke vores http request-metoder til kode som vi nemt kan forstå og håndtere, det kan f.eks være
[HttpGet], hvilket kaldes når der modtages et Get request på den gældende controller, men vi kan også [HttpGet("{id}")], hvilket betyder, at vi regner med at der kommer et id med, i dette tilfælde fra URI'en