# Mock2

## Opgave 1

**Hvilke protokoller og porte anvendes af din service?**

Vores service RestCoinService benytter REST protokollen sammen med HTTP protokollen

http protokollen benytter metoderne GET, POST, PUT, DELETE på et webservice.

* GET anvendes når et dokument skal hentes.
* POST anvendes når man skal sende noget til en webservice, f.eks. for at indsætte et nyt produkt
* PUT anvendes når noget skal opdateres på en webservice
* DELETE anvendes når noget skal slettes fra en webservice

Rest bruges i forbindelse med http og standardiserer måden at bruge http metoderne på.

* Man benytter primært JSON objekter (nogen gange XML) i forbindelse med at læse/sende data til en webservice
* REST er tilstandsløs, dvs. den ikke husker en aktiv forbindelse

**Beskriv kort indholdet af en REST request og response**

* <https://localhost:44335/api/Coin>
* Min webservice kan findes på stien localhost med porten 44335, den indeholder en Coin api med Coin controller knyttet som styrer Coins på webservicen

Et REST request kan typisk indeholde

* Request Line
  + Første linje skal altid starte med en kommando/metode og derefter kommer filnavn, og derefter protokollen (hver linje slutter med \r\n for at angive ny linje)
  + GET /index.html HTTP/1.1
* Header
  + Indeholder ekstra oplysninger om request, f.eks. hvem kommer requestet fra (host)
  + Host: URL
* Data
  + Indeholder data som skal sendes til webservice i tilfælde af POST/PUT

Et REST response kan typisk indeholde

* Status line
  + Viser status på sin request, f.eks. om det er OK og gik igennem
  + Starter med protokol, derefter status code, og status phrase
  + HTTP/1.1 200 OK
* Header
  + Indeholder ekstra oplysninger om response, f.eks. last-modified tidspunkt
  + Last-Modified
    - Tidspunkt på hvor filen sidst er ændret på serveren
  + Content-Type
    - text/html eller application/json
* Data
  + Fx json object i et get request

## Opgave 3

## c) Benyt Postman til vise requests/response

**Request**

GET /api/Coin HTTP/1.1  
Host: restcoinservice20181102095913.azurewebsites.net

**Response**



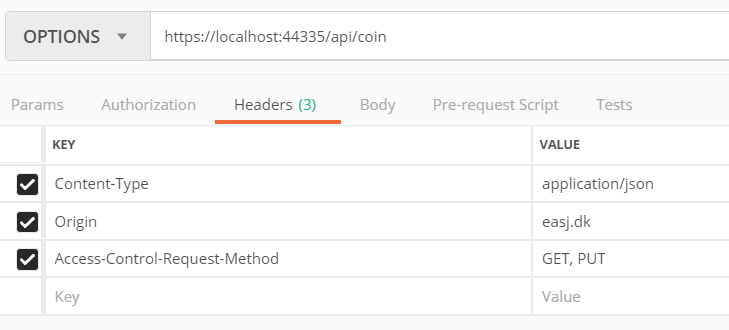
Response header



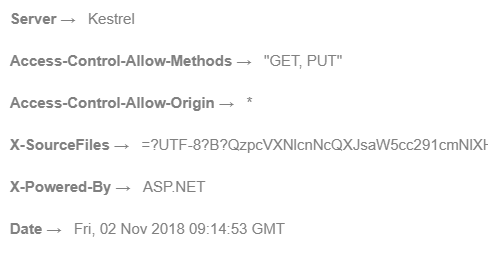
## Opgave 4

c) Benyt Postman til at sende en OPTIONS request, for at vise CORS er sat korrekt op

**Request headers**



**Response headers**



## Opgave 5

c) Hvilken sammenhæng er der mellem TypeScript og JavaScript

* TypeScript er bygget ovenpå javascript af Microsoft for at have en typestærk udgave af Javascript
* Da det er bygget ovenpå javascript er al javascript gyldig typescript kode

d) Forklar hvordan/hvor TypeScript eksekveres

* TypeScript sproget skal hentes ned i ens projekt først, dette gøres ved at intallere node.js som er en javascript packagemanager (blandt andet).
* Efter at have hentet node.js, kan typescript installeres på ens projekt og køres
* For at køre ens typescript skal man installere en pakke kaldet webpack, som ”kompilerer” al typescript kode og forvandler det til almindelig javascript, som browseren kan læse