

Programmering – Cereal API

Tid: 35-50 timer

Formålet med denne opgave er følgende:

- 1) Udvikle første prototype af et funktionelt Cereal API
- 2) Vise dit programmeringsniveau i det sprog du har mest erfaring med.
- 3) Udfordre dig til at skrive et program der håndterer informationer til og fra en database
- 4) Obs: Der er fokus på den kode, test mm ikke frontend. Når det er sagt ville en simpel frontend gøre godt til denne opgave

Aflevering: Du skal aflevere Cereal API kildekoden. Derudover skal du lave en kort PowerPoint præsentation (15-20min) som opsummerer de væsentligste punkter fra din proces med at udvikle scriptet.

Baggrund for opgaven:

Hvis du kan lide at spise morgenmadsprodukter, så gør dig selv en tjeneste og undgå denne datasæt for enhver pris. Efter at have set disse data vil det aldrig være det samme for dig at spise Fruity Pebbles igen.

Indhold af datasættet:

Name: Name of cereal

mfr: Manufacturer of cereal

A = American Home Food Products;

G = General Mills

K = Kelloggs

N = Nabisco

P = Post

Q = Quaker Oats

R = Ralston Purina

type:

cold

hot

calories: calories per serving

protein: grams of protein

fat: grams of fat

sodium: milligrams of sodium

fiber: grams of dietary fiber

carbo: grams of complex carbohydrates

sugars: grams of sugars

potass: milligrams of potassium

vitamins: vitamins and minerals - 0, 25, or 100, indicating the typical percentage of FDA recommended.

shelf: display shelf (1, 2, or 3, counting from the floor)

weight: weight in ounces of one serving

cups: number of cups in one serving

rating: a rating of the cereals (Possibly from Consumer Reports?)

Opgaven:

- 1) Udfyld udleverede kravspecifikation og få den godkendt hos din opgavestiller
- 2) Følgende krav til din løsning:
 - a. Opret en database (frit valg af teknologi) fra det udleverede dataark
 - b. Skriv en parser til data, så du ville kunne opdatere den med flere informationer
 - c. Lave et endpoint, hvor man kan lave GET-requests på og uden ID for alle produkter
 - d. Tilføj en filtrering således at man ville kunne sortere i flere parameter. Fx så man kan se alle produkter med fedt = 0g, skelne gerne mellem varm og kold
 - e. Tilføj POST-requests til dit endpoint. Den skal acceptere data på samme format som GET-requestet returnerer. Hvis den kaldes med et ID, skal programmet tjekke om objektet findes, og i så fald skal det opdateres. Hvis objektet ikke findes, skal der sendes en fejl om at man ikke selv kan vælge ID på nye objekter. Hvis man vil oprette et nyt objekt, skal der sendes et POST-request uden ID.
 - f. Tilføj DELETE til endpoint.
 - g. Tilføj yderligere filtrering til GET-request, så man kan f.eks. kan lave \geq , \leq , $<$, $>$, \neq eller sortere output baseret på en kategori.
 - h. Tilføj rettigheder, så alle kan bruge GET-request, men man skal have brugernavn og password for at sende POST og DELETE.
 - i. Tilføj et andet endpoint, hvor man kan lave GET-requests på et ID, og så får man et billede af produktet. For at spare tid, kan der bruges det samme billede for alle produkter fra samme producent, men endpointet skal stadig fungere for ID'et for produktet.
- 3) Følg din Kravspecifikation og udvikle din løsning, når din Kravspecifikation er godkendt får du udleveret billederne af morgenmadsprodukterne
- 4) Udarbejd en PowerPoint præsentation

God Arbejdslyst.