

title: Examenvragen 2018

# Vragen Webdevelopment

Deze vragen zijn reeds gesteld op vorige examens van het vak Webdevelopment.

Deel I: Ruben

## Opgave 1

Vraag van een gast op stackoverflow (te lang om volledig te hertypen, maar het komt hier op neer)

Wil Rest-API maken voor sensors. Gebruikers kunnen via een server info opvragen over de sensors. Om dit te doen moet de gebruiker het nummer van de sensor kennen. Als de sensor niet bestaat, welke status code moet er gebruikt worden?

200 (OK, met een foutboodschap) 404 (Not Found) 500 (Internal Server error)

Dit hangt denk ik af van de structuur van de URL

`www.serversite.com/deviceinfo?id=123` `www.serversite.com/devices/123/info`

In het eerste geval zou een 404 betekenen dat de hele device info pagina niet bestaat, in het tweede geval dat enkel die sensor niet bestaat.

Hoe moet ik dit doen?

Vragen

- Welke code moet je gebruiken en waarom? 404 -> fout ligt bij de gebruiker (vroeg waarom geen 500 -> fout ligt bij server)
- Hangt dit idd af van de structuur van de URL Nee, de twee URL zouden hetzelfde moeten doen aangezien ze naar dezelfde resource wijzen (zijn antwoord, kan dus niet helemaal accuraat zijn. Mijn redenering was niet juist)
- Hoe zou je te werk gaan om aan te tonen dat jouw uitleg juist is (of zo iets) Double checken op documentatie van HTTP en deze documentatie ook bij mijn antwoord linken
- Welke ongerelateerde fout maakt de schrijver hiervan tegen REST? gebruiker moet nummer kennen -> moet met hypermedia controls

## Opgave 2

- Geef en beschrijf 3 interfaces om linked data te publiceren
  - Data dump (+ korte beschrijving)
  - SPARQL-endpoint (+ korte beschrijving)
  - TPF (moest ik niet meer uitleggen/beschrijven om een of andere reden, wel opgeschreven) (\*eventueel document stuff)
- Geef enkele voor en nadelen van deze interfaces (zo veel mogelijk, ik heb er denk ik meer opgeschreven op het examen, maar ik weet ze nie allemaal meer)
  - data dump
    - weinig rekenwerk voor server
    - veel bandbreedte nodig (client en server)
    - veel opslagruimte nodig voor client
  - SPARQL-endpoint
    - lichte queries -> weinig bandbreedte
    - veel rekenwerk voor server
    - hoge downtime -complexere queries meestal niet mogelijk (veel users tegelijk, zou hoge rekenkracht nodig hebben)
  - TPF compromis tussen twee vorige: voor en nadelen van beide in mindere mate
- Met welke factoren zou je rekening houden om een van deze methoden te kiezen
  - bandbreedte
  - wat een acceptabele uptime/downtime is
  - hoe krachtig de server is
  - ...

bijvragen

- Relatie/verschil RDFS/OWL beide ontologies. RDFS -> beschrijven van andere ontologies. OWL -> uitbreiding RDFS, meer focus op relaties tussen resources
- nog een vraag, kan ik mij niet meer herinneren

## Deel 2: Peter

### Opgave 3 (video in het web)

- Wat zijn mediafragments en hoe leidde deze tot de W3C dingetjes voor URI dingetjes (vraag was ingewikkeld gesteld, my apologies)
  - uitleg over verschillende soorten fragments (spatiaal/temporeel/track)
  - uitleg over hoe de URI deze fragments interpreteerd (t=.../xywh=.../track=...)
  - was niet voldoende, moest nog iets bij, weet niet meer wat
- Vergelijking fragment URI vs query URI (of zoiets, da stond zo in een tabelleke in de slides) m.b.t. caching en < nog iets > (ik ben nog altijd verbaasd da ik dit juist had)
  - fragment query
  - eventueel cacheable niet cacheable
  - moet uitdrukbaar zijn in byteranges moet niet uitdrukbaar zijn...
- Bespreek in detail het verwerkingsproces (eresolvinge) voor de URI: ehttp://foo.com/video.ogv#t=11,19e in het geval van mapping aan server-zijde op een proxy cacheable manier (geval E) en de impact op het huidige web (of zoiets)
  - Eerste deel geskipped (wist ik nie)
  - Impact -> minder latency aangezien data dichterbij client zit
- Nog een vraag denk ik, maar ben vergeten
- Bespreek de basisprincipes van adaptive http streaming
  - stream opgedeeld in chunks (2 - 10 sec)
  - manifest wordt meegestuurd
  - client lees manifest een haalt aan de hand hiervan nodige chunks op (is mogelijks onvolledig)

### Opgave 4 (metadata)

- iets in de trant van: wat is een xml based metadata standaard, wat zijn nadelen ivb met semantiek, hoe kan RDF dit oplossen en hoe kan RDF in het hedendaags web geintegreerd worden
  - ~\(\_)\_/`
- Voor da laatste gewoon wa over Schema.org zeveren