

Emma De Koster (202186541)

- ☒ Web Services: GITHUB URL
 - [GitHub repository](#)
 - [Online versie](#)

Logingegevens

- Gebruikersnaam/e-mailadres:
- Wachtwoord:

Projectbeschrijving

De link naar het originele ERD:

https://kroki.io/erd/svg/eNpNjs8KgkAQxu_zFJ6VhXoGz7HRVSIIm_awB3ZXZkbCnT5cob8N8_37NBcl4Vg52pVI6CjyCkilgFObxDqW3THXsQK3YQIV8BehtdVLjt3MX49meh6ycwMNOmFRafOt8X_MQVZCoE9i2v6omMSR6KI_J92eNBglkywTfb11U6Y8zb_-xi9K5Y5FR9t-jc2WRox9aaU6-

Het idee achter mijn API was om een applicatie te maken die mensen helpt bij het volgen van een bepaald dieet. Sporters, vegans of mensen met allergiën kunnen via deze app vereisten opgeven voor een bepaalde meal (bv: ik ben vegan (dietRestrictions) en ik moet nog avondeten (typeOfMeal) eten en ik moet nog 30gram proteïne (grams of protein) binnen krijgen)

Elke maaltijd behoort tot een restaurant. Als uitbater kan je zo dus je restaurant en de bijbehorende maaltijden in de API toevoegen. Als gebruiker kan je dan jouw vereisten opgeven en zo een gerecht bekomen in een bepaald restaurant!

Screenshots

(heb geen front-end, dus dit is niet relevant voor mij)

Behaalde minimumvereisten

Web Services

- **data laag**
 - ☒ voldoende complex (meer dan één tabel)
 - ☒ één module beheert de connectie + connectie wordt gesloten bij sluiten server
 - ☒ heeft migraties
 - ☒ heeft seeds
- **repository laag**
 - ☒ definieert één repository per entiteit (niet voor tussentabellen) - indien van toepassing
 - ☒ mapt OO-rijke data naar relationele tabellen en vice versa
- **servicelaag met een zekere complexiteit**

- ☒ bevat alle domeinlogica
- ☒ bevat geen SQL-queries of databank-gerelateerde code

- **REST-laag**

- ☒ meerdere routes met invoervalidatie
- ☒ degelijke foutboodschappen
- ☒ volgt de conventies van een RESTful API
- ☒ bevat geen domeinlogica
- ☒ degelijke autorisatie/authenticatie op alle routes

- **varia**

- [] een aantal niet-triviale testen (min. 1 controller >=80% coverage)
- ☒ minstens één extra technologie
- ☒ duidelijke en volledige **README.md**
- ☒ maakt gebruik van de laatste ES6-features (object destructuring, spread operator...)
- ☒ volledig en tijdig ingediend dossier

Projectstructuur

Web Services

Ik heb een duidelijke mappenstructuur gecreëerd: eerst is er de inderverdeling van de config en de src files. vervolgens binnen de src heb ik het lagen-principe toegepast (core, data, repo, rest, service).

Extra technologie

Web Services

De extra gebruikte technologie voor mij is dat ik Objection.js heb gebruikt om de data op te halen en om te vormen tot Models. Via de Model-files haalt hij de juiste informatie op uit de database.

Testresultaten

Web Services

Ik heb redelijk wat problemen ervaren bij het uitvoeren van mijn testen. Ondanks dat ik alles geschreven heb, zoals ons in de les werd aangetoond en uitgelegd + uren heb getroubleshoort kwam ik niet van de error : "Cannot read properties '...' of undefined".

Uiteindelijk heb ik besloten de testen wel volledig te schrijven zoals het hoort, ondanks deze foutmelding. De gevraagde screenshots kunnen dus niet worden toegevoegd.

De testen hebben betrekking tot de controller 'Restaurant' en testen de volgende routes: GET /api/restaurants GET /api/restaurants/:id POST /api/restaurants PUT /api/restaurants/:id DELETE /api/restaurants/:id

Gekende bugs

Web Services

Zoals eerder beschreven is er een fout binnen de test code waardoor ik steeds dezelfde foutmelding krijg. Jest kan ingevoerde variabelen niet lezen.