

LABO FRAMEWORKS VOOR SERVERAPPLICATIES

GROEPSOPDRACHT: ORM, NODEJS EN WEBSOCKETS

oktober - november 2024

1 Inleiding

De groepsopdracht uit drie delen:

- het maken van een datalaag met ORM in het typescript-framework [TypeORM](#),
- een REST-API voor deze datalaag definiëren en implementeren in NodeJS,
- een webapplicatie maken in NodeJS die gebruik maakt van websockets en de REST-webservices.

2 Datalaag met ORM

Maak een datalaag in typescript die de principes van ORM toepast in het javascript-framework [TypeORM](#). Deze keuze van de databank is vrij. Hou er wel rekening mee dat de applicatie (en dus ook de databank) gepubliceerd worden op een linux-server (zie puntje 5) en dat de REST-webservices in NodeJS geschreven zullen worden.

De voorwaarden waaraan de datalaag moet voldoen zijn de volgende:

- De datalaag bestaat uit minstens één DAO-object waaraan je objecten kan opvragen, toevoegen, aanpassen, ...
- De datalaag bevat minstens één 1-1-relatie, 1-n-relatie en één n-n-relatie. Van deze relaties is er minstens één met cascade en één zonder cascade.
- De datalaag bevat overerving.
- Met de datalaag kan je één object toevoegen.
- Met de datalaag kan je een verzameling van objecten toevoegen.
- Met de datalaag kan je objecten opvragen. Voorzie zowel een “lazy” als niet “lazy” opvraging.
- De datalaag heeft ook een opvraging die gebruik maakt van parameters.
- Met de datalaag kan je objecten aanpassen.

Hoe je een startproject voor TypeORM kan maken, vind je op <https://typeorm.io/readme#quick-start>.

3 Specificatie en implementaties REST-webservices

De functionaliteit die ontwikkeld werd in de datalaag moet toegankelijk gemaakt worden via REST-webservices. Pas hiervoor de “API first approach” toe. Dit betekent dat je eerst functionaliteit van de REST-webservices vastlegt en pas daarna start met implementeren. Je bepaalt dus eerst op papier welke functionaliteit REST-webservices moeten bieden.

Werk de REST-webservices uit in NodeJS met het Express-framework en laat ze gebruik maken van de datalaag uit puntje 2. Om vlot Express met TypeOrm te laten samenwerken, maak je best gebruik van typescript om je REST-webservices te implementeren. Je kan hiervoor het project uit het vorig puntje uitbreiden. Meer info op <https://typeorm.io/example-with-express>.

4 Webapplicatie in NodeJS

Ontwikkel een webapplicatie in node. Breid hiervoor de bovenstaande applicatie uit. De webapplicatie maakt gebruik van minstens één van de webservices uit puntje 3 en krijgt ook geregeld updates voor één of meerdere van zijn webpagina's van de server via websockets.

5 Publiceren op de Virtual Wall

De webapplicatie moeten gepubliceerd worden op Virtual Wall. Om de applicatie te deployen krijgen jullie het IP-adres van een machine en een login en wachtwoord. Je kan via een consolevenster inloggen op de server gebruik makende van `ssh`. ([handleiding](#))

```
ssh -p 2222 login@ipadres
```

Om mappen en bestanden te kopiëren naar de server maak je gebruik van `scp`. ([handleiding](#))

```
scp -P 2222 padnaarmap login@ipadres:/padopserver
```

Gebruik je liever een grafische omgeving, dan is [WinSCP](#) een optie.

Je kan ook “git pull” gebruiken om code te kopiëren van je repository naar de server.

Op de server staat poort 3000 open voor HTTP. Je kan de webpagina's opvragen door in de browser het adres `http://ipadres:3000/pad` in te geven.

6 Indienen opdracht

Bewaar geregeld jullie oplossing in jullie gitlab-repository. De locatie van deze repository is `https://gitlab.stud.atlantis.ugent.be/frameworks2425/groupX`, waarbij X vervangen wordt door het groepsnummer. Zorg ervoor dat elk teamlid zijn code pusht onder zijn naam.

Voeg een `.gitignore`-bestand toe aan je repository zodat enkel je broncode bewaard wordt op gitlab. Meer info over `.gitignore` vind je op [manual gitignore](#) en [voorbeeld .gitignore-bestand](#).

Je oplossing moet ten laatste afgewerkt zijn op vrijdag 8 november 2024 om 22u00. Beschrijf in de `README.md` hoe de verschillende onderdelen getest kunnen worden. Voorzie ook voldoende dummy data om de onderdelen te kunnen testen.