

Project Backend Development – Node.js

Een multi-user real-time applicatie.

1 Inleiding

Node.js leent zich vooral voor het maken van **real-time** toepassingen met **meerdere** gebruikers die simultaan een relatief kleine hoeveelheid informatie uitwisselen of samenwerken.

Aan de hand van dit project is het de bedoeling de verschillende aspecten van node, die aan bod komen in de cursus, uit te testen of in te oefenen. Daarnaast verwacht de opgave inbreng van een eigen bestudeerd element. Gedurende een aantal labo's wordt tijd voorzien om het gekozen onderwerp te verantwoorden, eraan te werken en daarop de geziene leerstof toe te passen.

2 Onderwerp van de applicatie.

De opgave verwacht het aanmaken van een **backend** in node.js. Deze opgave loopt in samenwerking met een andere module. Hoofdzakelijk is dit de module Front Webtechnologie, maar evengoed kan dat met de domotica module of module voor mobiele applicaties zoals Android.

Onderwerp en Frontend komen m.a.w. vanuit een andere module. Wie geen passende andere module volgt, voorziet een eigen frontend, die wel eenvoudig maar toch acceptabel is. Evengoed kan je een backend maken voor een andere student die enkel frontend volgt.

Er mag met meerdere personen aan het onderwerp gewerkt worden, op voorwaarde dat een duidelijke taakverdeling en taakrealisatie gerapporteerd wordt. Wie liever alleen werkt, mag het project alleen uitvoeren.

1.1 Vanuit frontend (ook geldig voor backend)

Voor het onderwerp verwijst ik naar de ppt van frontend met volgende elementen:

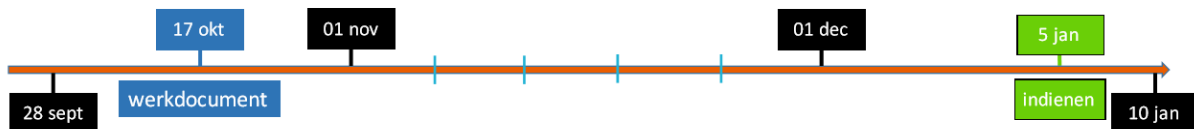
- Technisch vernieuwend: laatst beschikbare versies dus
- Een WOW-onderwerp: uitgebreid, boeiend, alles omvattend, uithangbord voor je portfolio, iets wat je dus interesseert.
- Categorieën: health, game, IoT, Education, MMedia (Music, Photo, Video), sport, people, game
- Voorbeelden: <http://rumpetroll.com/>, <http://hellomydear.be/>, weergave van realtime data in een score board, multi-user Q&A, realtime telemetry toepassingen, realtime API mockup (=combineren van meerdere API's), ...

1.2 Typisch backend:

- Voor elk van de vermelde applicaties worden gegevens (of een deel van de gegevens) **persistent gestockeerd**.
- **Realtime communicatie** wordt mogelijk gemaakt via sockets.

- Een niet geregistreerde gebruiker kan slechts een **deel** van de applicatie uitvoeren. Dit deel is meer dan enkel een startpagina en dient als teaser. Om over alle eigenschappen van de applicatie te beschikken moet de gebruiker zich eerst registreren.

3 Planning en tijdslijn



Het werkdocument wordt jouw centraal project dossier. Dit dossier wordt continu aangepast. Wie frontend en backend volgt, dient enkel in op de frontend dropbox van LEHO. Wie enkel backend volgt dient het document in op de backend dropbox van LEHO. Op het einde (5jan) schetst het werkdocument een duidelijk beeld van het projectverloop.

Je kan drie elementen onderscheiden in jouw werkdocument:

- Requirements document (verwachtingen en eisen): Hiermee vraag je de goedkeuring aan voor jouw onderwerp (17okt voor FE, 27okt voor wie enkel BE volgt). Begin dus niet met programmeren tot je document goedgekeurd is Deze aanvraag bevat:
 - o Project Titel, beschrijving
 - o Doelgroep en verwachtingen.
 - o Teammembers met zowel hun technische verantwoordelijkheid (FE-CSS en designer, BE-tester...) als planningsverantwoordelijkheid (scrummaster, secretaris voor verslag bij cruciale beslissingen...). Maximaal 4 personen per groep.
 - o Eerste korte technische analyse
 - o Kritische milestones in de planning. Dit zijn een beperkt aantal momenten die cruciaal zijn. Worden ze niet gehaald dan wordt vanaf dit punt de planning aangepast. Of kan vooraf al een alternatief plannings-pad voorzien worden.
 - o Referentie waar we het project kunnen volgen (git, URL...)
- Implementatie document: Van zodra je project goedgekeurd is, begin je het uit te werken (=implementeren). Gedurende deze implementatie kan je overwegen om elementen anders uit te voeren. Technische wijzigingen, aanpassingen en aanvullingen op de eerste korte technische analyse horen hierin thuis.
- Realisatie document: Bevat elementen die naar het einde toe, bij de aflevering van het project naar boven komen:
 - o Gerealiseerde werktijd tegenover de geschatte/geplande werktijd.
 - o De moeilijkste struikelstenen, die je ondervond tijdens het project.
 - o Waar ben je vooral tevreden over?
 - o Wat zou je volgende keer anders doen.

Noot: Het project forum op Backend-LEHO kan gebruikt worden om onder elkaar informatie uit te wisselen of vragen te stellen.

4 Technische elementen (backend)

De applicatie	In eerste instantie een (mobiele) webapplicatie maar ook een standalone desktop applicatie (Electron) of een IoT applicatie (Raspberry, Arduino, Cylon) zijn mogelijk	
De server-taal	Er wordt gebruik gemaakt van javascript - node.js met aandacht voor performantie en object based programmeren in een MVC-structuur. Er wordt asynchroon geprogrammeerd. Afwijkingen en verantwoording hierop worden vermeld in het implementatie document. Alternatieven: protocols zoals ZigBee, KNX, ZWave	
Framework	Een bestaand framework naar keuze mag (moet niet) worden gebruikt: In de labo's: Express Alternatieven: Sails.js , Hapi , Flatiron, Rhapsody Fullstack alternatief: Mean, Meteor, Derby	
Foutcontrole	Voorzie een deftige foutcontrole voor asynchrone (kritische) functies, zodat je applicatie niet crasht en je een duidelijkbeeld krijgt van fouten die zich voordoen. Een library mag maar moet niet.	
Persistent stockeren	Gebeurt in een NoSQL database naar keuze: In de labo's: MongoDB Alternatieven: CouchDB , Cassandra , InfluxDB	
Realtime websockets	Gebruik van een library naar keuze: In de labo's: socket.io Alternatieven: websocket , express-io , socket stream , pub/sub messaging	
Unit-testing	Gebruik van een testomgeving (testrunner) naar keuze: In de labo's: Mocha , Alternatieven: Expresso , nodeunit	
Registratie	Enkel gebruikers die ingelogd zijn kunnen de volledige applicatie gebruiken. Gebruik van OAuth login pakket naar keuze. Alternatieven: caching databases: Redis , Riak .	
Deployment	Op een omgeving naar keuze: azure , modulus of Heroku	
Source Control	Naar keuze: Github, TFS, Bitbucket	

Er wordt minstens gebruik gemaakt van **één alternatief element**. Een alternatief element is een element, patroon of node pakket, dat niet gebruikt werd tijdens de cursus. Je mag hiervoor ook zelf een niet vermeld of eigen studie element aanbrengen. Vermeld het nodige hierover in het implementatie document.

4 Praktische punten:

- Eerste indienen van het finaal afgewerkte project gebeurt ten laatste op LEHO tegen 5 januari in de dropbox van het vak. Er kunnen nog wijzigingen aangebracht worden tot de dag vóór de presentatie. Indienen gebeurt met een rar of zip file met volgende naam:

projectNaam-NaamVoornaam.

Wie FE en BE volgt dient enkel in op de FE dropbox.

- Voorbeeld: "ChatterBox-VerbruggeJoe.zip" of "chatterBox - VerbruggeJoe.rar"
 - De inhoud van de rar/zip:
 - Volledig uitvoerbaar project inclusief database
 - Werkdocument: Werkdoc-NaamVoornaam.docx
-