Period-2 Node.js, Express + JavaScript

NodeJs = Server

Express = Minimal and flexible NodeJS web application framework.

Why would you consider a Scripting Language as JavaScript as your Backend Platform?

**Fordele:**

* NodeJS tillader programmører at skrive JavaScript på både Front-end og Back-end. Dette formindsker fejl og besvær i samarbejdet mellem Front-end og Back-end, da begge lag programmeres i samme sprog. Dette betyder at man kan bruge lignende patterns og nogle tilfælde samme biblioteker for både Front-end og Back-end. Det er et stort plus, når det kommer til udviklerens produktivitet, vedligeholdelse, og tid til markedet.
* En anden god fordel ved at bruge NodeJS, er at NodeJs bruger såkaldt Non-Blocking I/O, hvor en tråd udføre alle opgaverne igennem, efterhånden som de dukker op. Med traditionelle web-applikationer vil en tråd stå og vente på, at en opgave eller et kald til databasen er afsluttet, før den er klar til næste opgave.

**Ulemper:**

* Et af de største ulemper i NodeJS er at NodeJs selv skal hente svarende hjem med call-back. Hver eneste gang man har noget I/O, bliver man nødt til at specificere et call-back og have en funktion, der køre videre derfra. Dette er en af svaghederne, rent vedligeholdelsesmæssigt, da det godt kan være svært at holde styr på alle de call-backs.

Explain Pros & Cons in using Node.js + Express to implement your Backend compared to a strategy using for example Java/JAX-RS/Tomcat?

Hvis NodeJS+Express skulle sammenligenes med Java/JAX-RS/Tomcat, så vil NodeJS være nemmere at håndtere, da man i Node ikke skal tænke over tråde og blokkeringer. Sprogmæssig vil Java være fortrukket da sprog arkitekturen er mere menneskeligt.

**Fordele:**

* Non-blocking I/O, asynkron arkitektur
* Hurtig processing
  + - *Node bruger V8 JavaScript Runtime, som Google Chrome også bruger. Både Node og V8 JavaScript engine er skrevet i C, hvilket gør det virkelig hurtigt. Det er meget hurtigere end Ruby, Python og Perl.*
* Bruger JavaScript, som er forholdvis nemt at lære
  + - *Node er ikke et nyt sprog, det er JavaScript, så JavaScript udvikler vil ikke bruge meget tid til at lære Node.*
* Bruger samme kode på både Front-end og Back-end
* Mange Node packaged modules, og bliver ved med at blive større

**Ulemper:**

* Samarbejdet med relational databases
* Call-backs
* Uden en dyb viden om JavaScript, kan give store problemer i Node
* Node er ikke egnet til CPU-intensive opgaver. Den er kun velegnet til I/O-ting (ligesom webservere).