**Node JS**

-JavaScript runtime build on googles open-source V8 JavaScript engine.

* Single-threaded, based on event driven, non-blocking Input/Output model
* Perfect for building fast and scalable data-intensive apps

Use node for:

* API with database behind it (preferably NoSQL)
* Data streaming
* Real-time chat app
* Server-side web app
* JavaScript across the entire stack
* NPM: huge library of open-source packages available free
* Very active dev community

Don’t use node for:

* App with heavy server-side processing

-Da bih pokrenuo Node, dovoljno je u terminalu da upisem ‘node’ I enter. Ako nakon toga kliknem ‘tab’ dobicu sve dostupne globalne variable.

-Da bih izvrsio neki file, kucam ‘node <ime file-a>’.

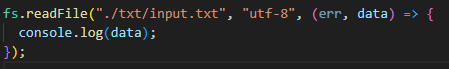
-Da bih (Sync) procitao neki file iz filesystem-a koristim ‘fs’ module:



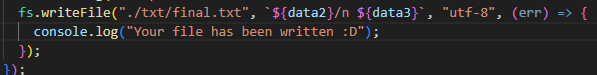
-Da bih (Sync) zapisao neki file:



-Synchronous code je ‘blocking’, zato najcesce procese odvijamo u ‘pozadini’ uz pomoc Asynchronous code-a.

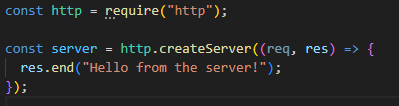


Ili za pisanje:

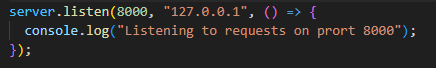


- Pravljenje web servera:

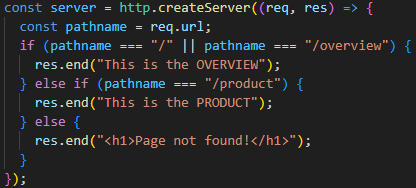
* Kreiranje servera



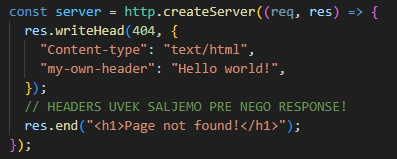
* Slusanje servera



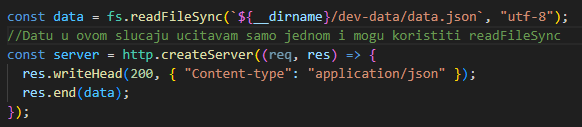
- Routing:



-Headers je informacija koju saljemo nazad o response-u



- API je servis od kojeg zahtevamo neku datu



- Parsing variables from URLs



- Kreiranje modula:

* Napravim file (u modules folderu)
* U njemu pisem funkciju 
* Tamo gde zelim da ga koristim 

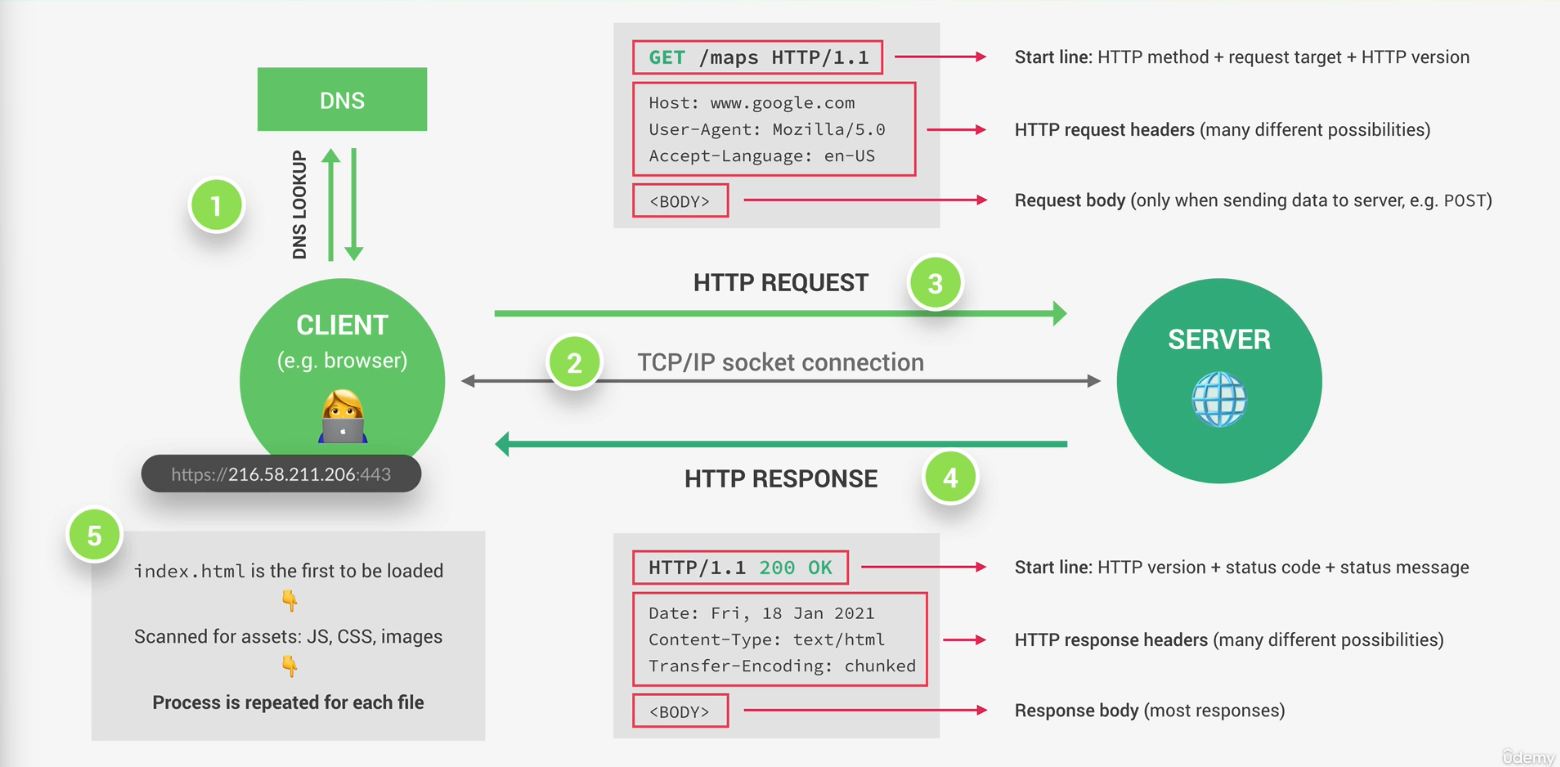
- Npm:

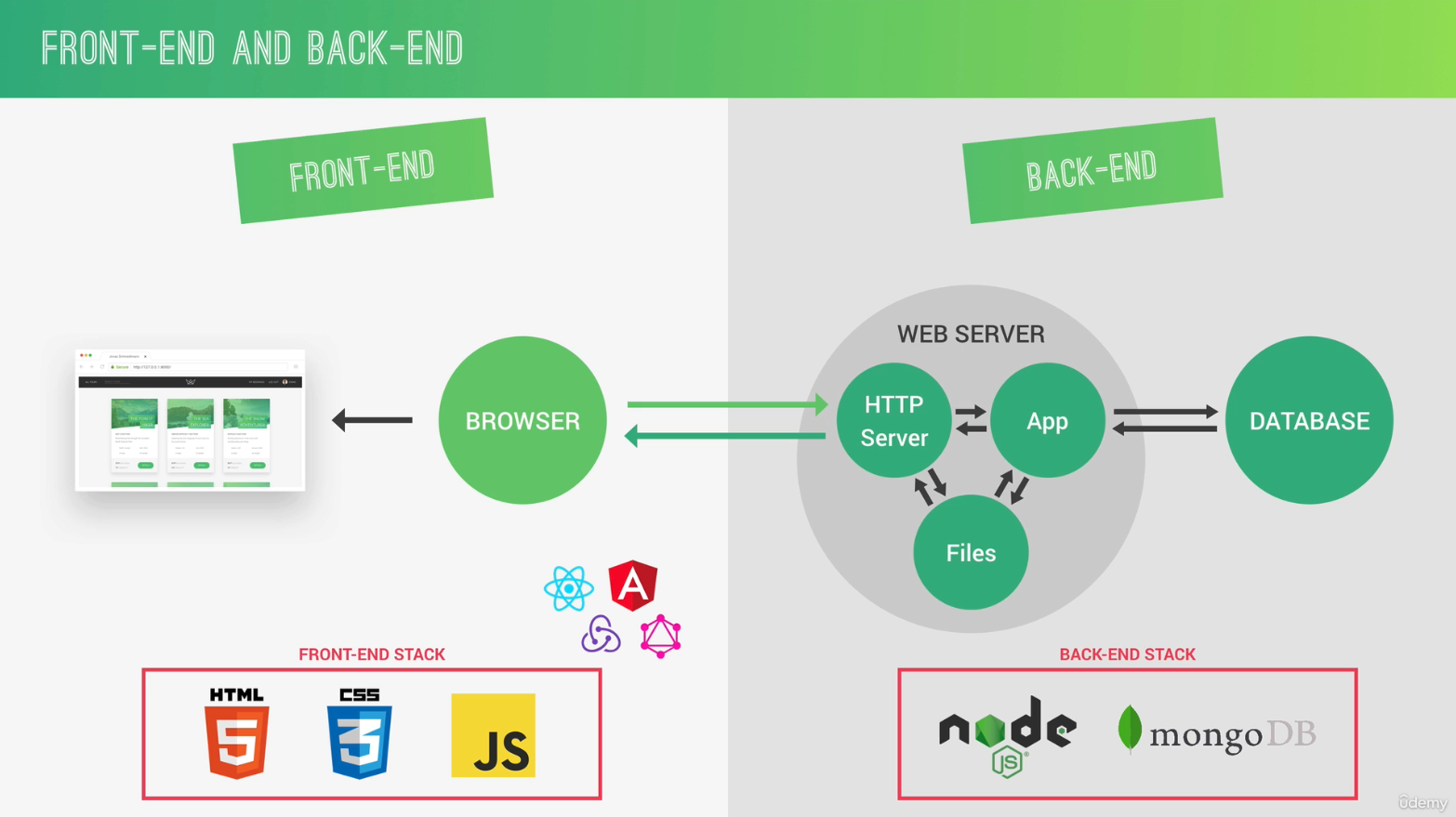
* U terminal kucam npm init (kreira package.json file)
* Postoje dve vrste paketa, devDependencies i regular dependencies
* npm un/install <dependency name> ili npm un/install <dependency name> --save-dev ili npm un/install -g <dependency name>
* da bih koristio npm modul u code-u pisem
* 
* Npm outdated kucam da proverim da li ima novih verzija

- Prettier:

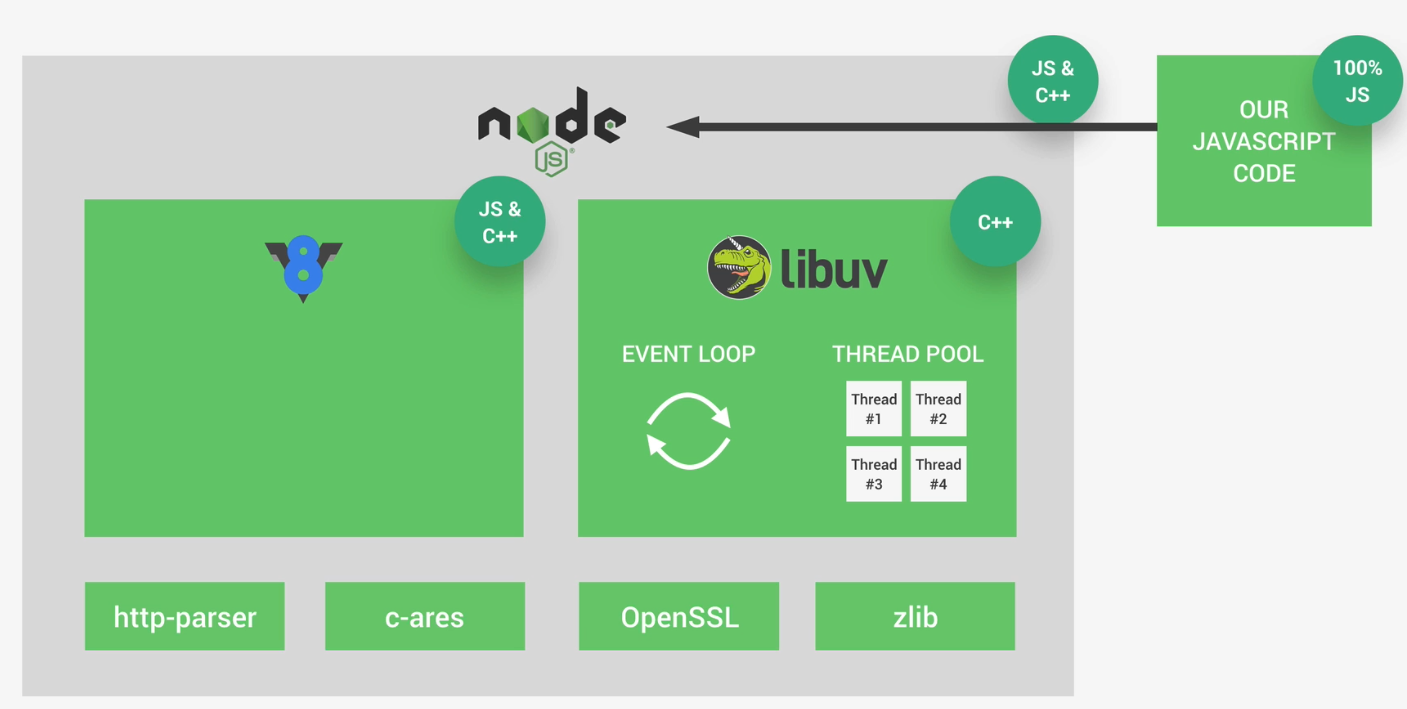
* Pravim file u root-u
* Config file popunim po zelji {}

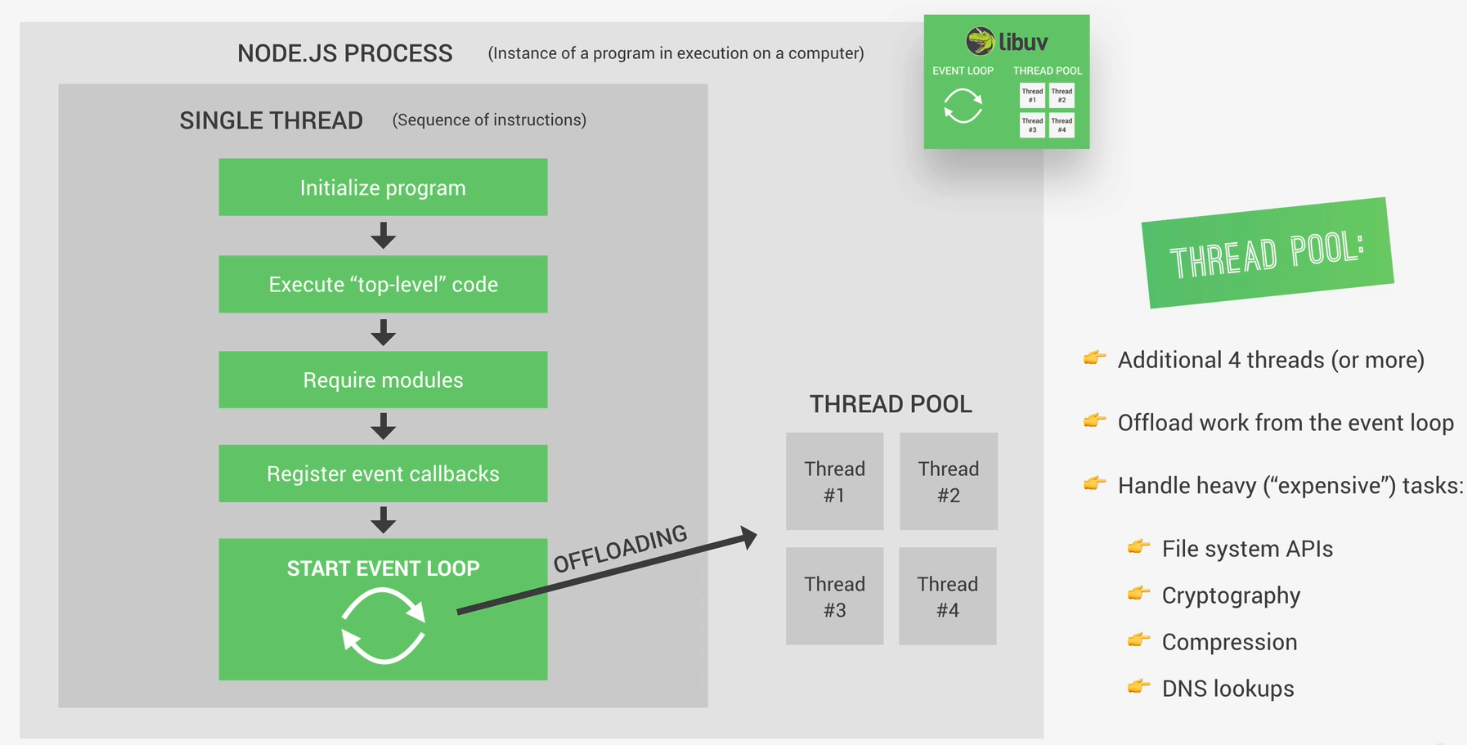
-How the web works:

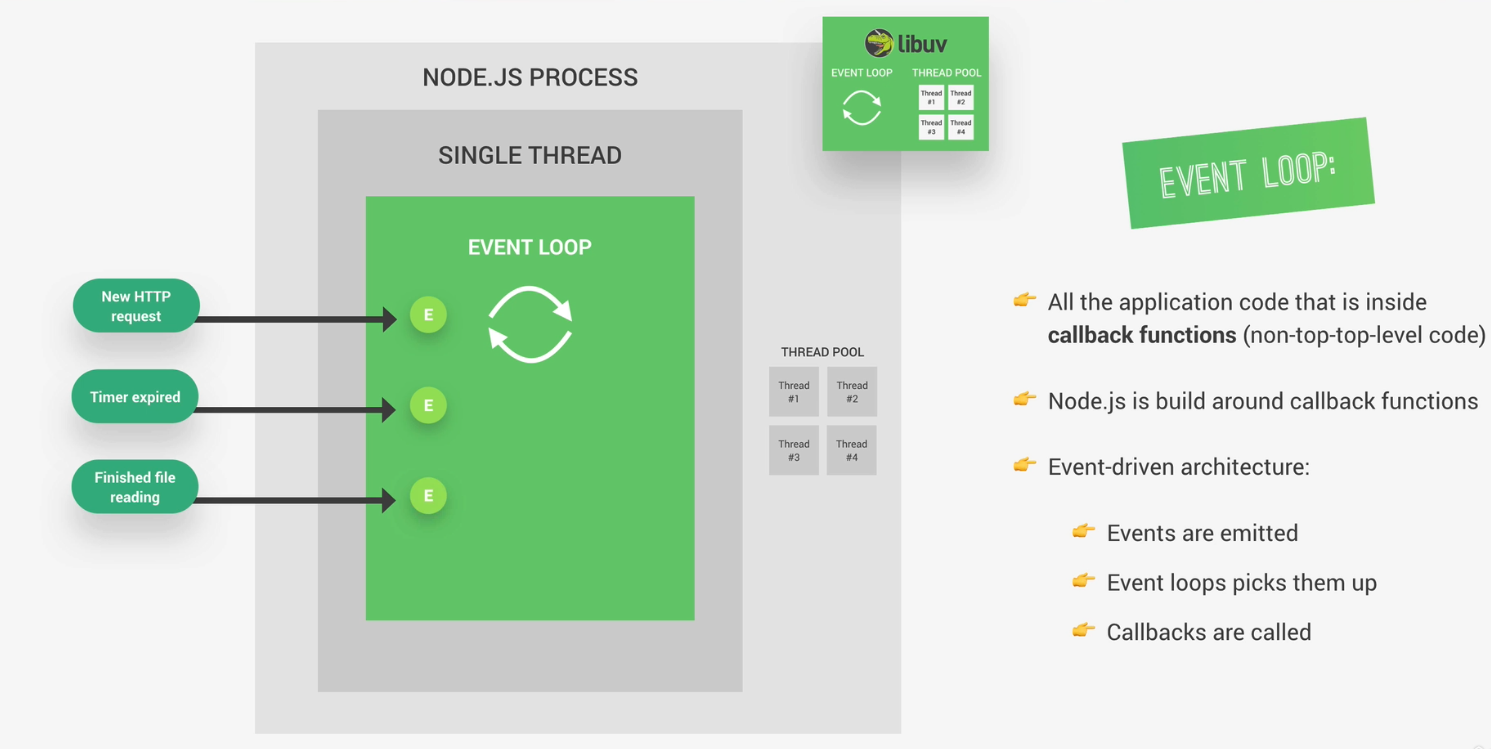


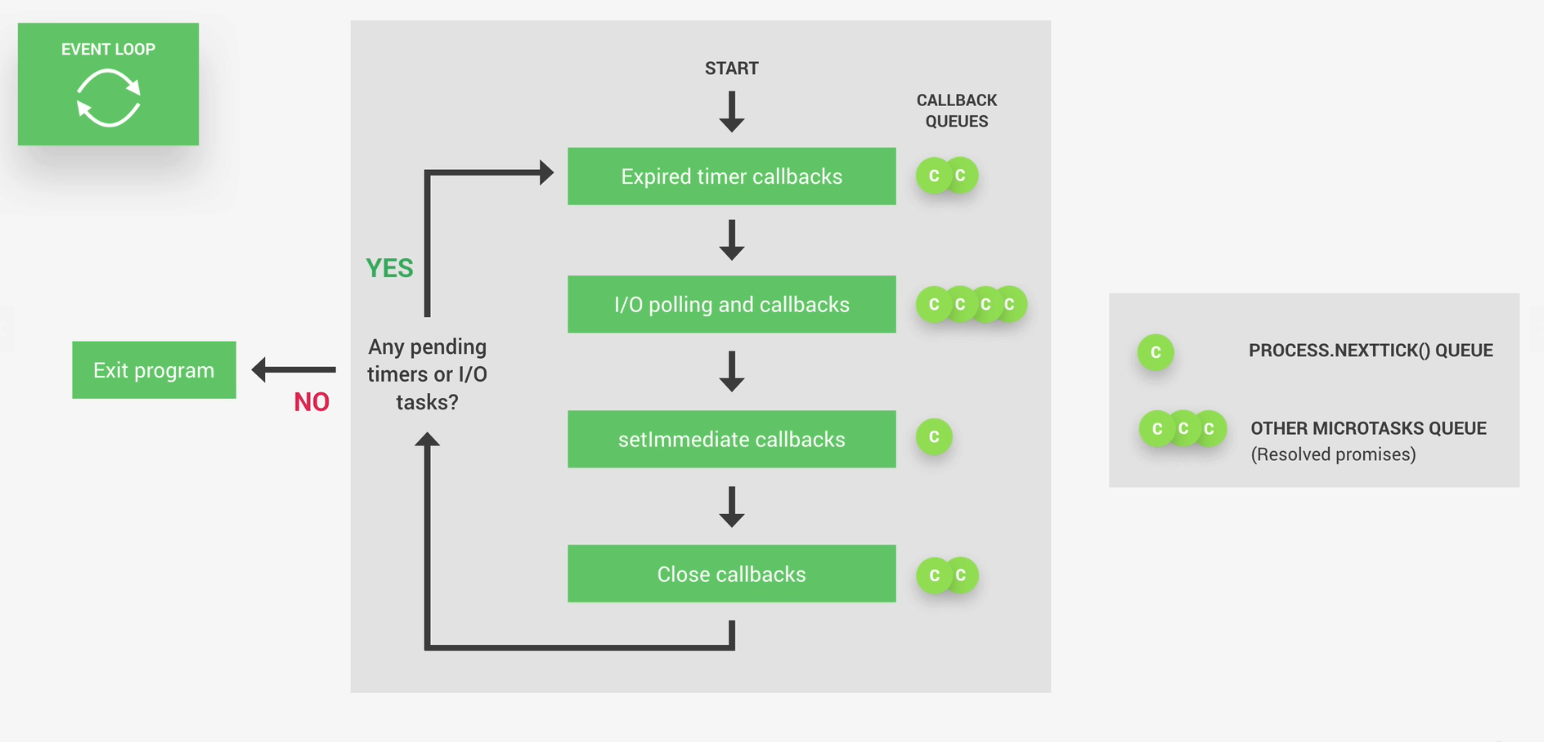


- Node arhitektura:



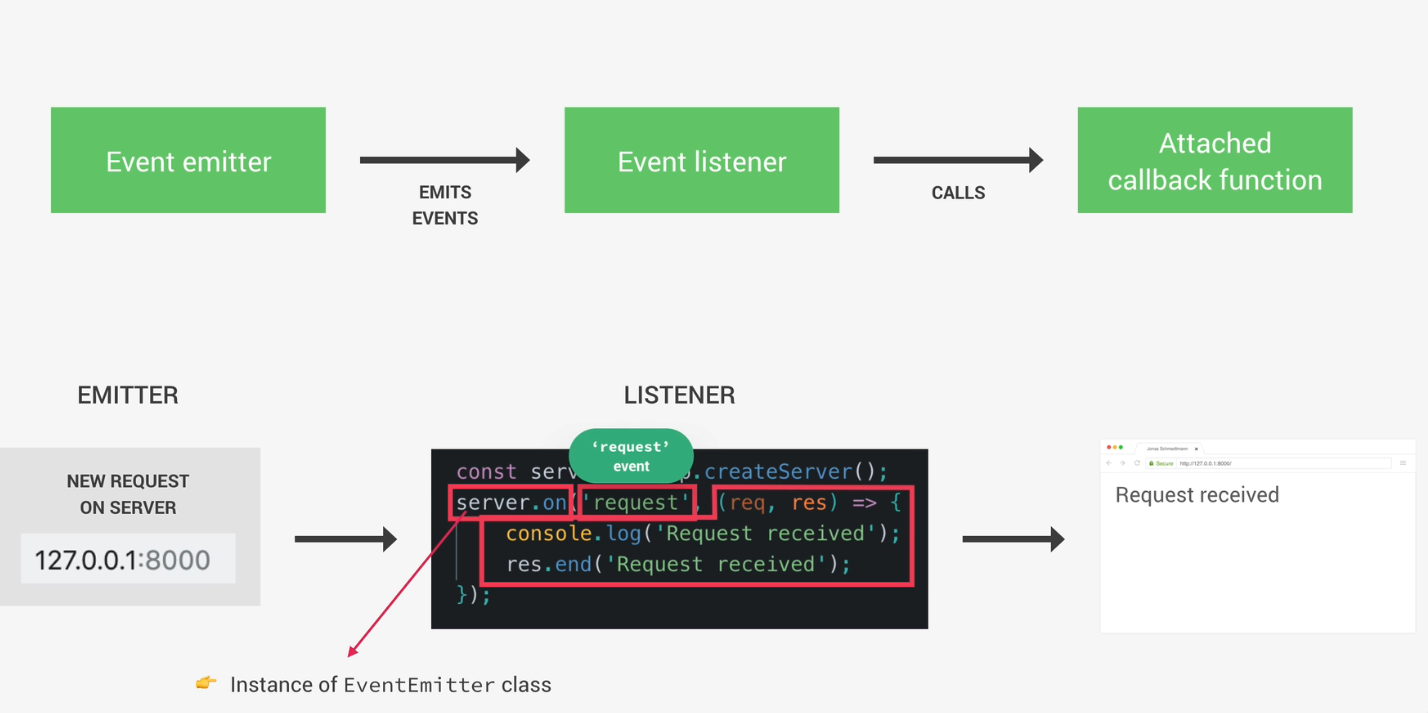




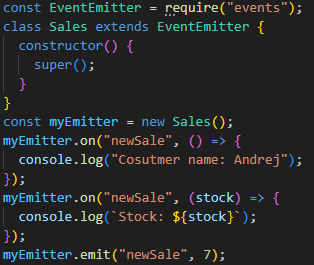


* Ako zelim da promenim threadpool size: 

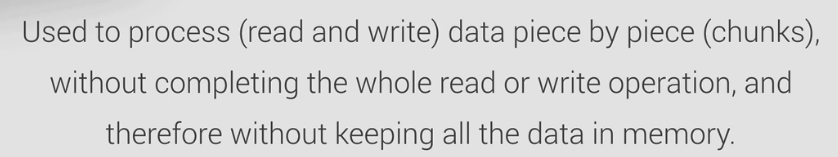
- Events and Event-driven architecture:

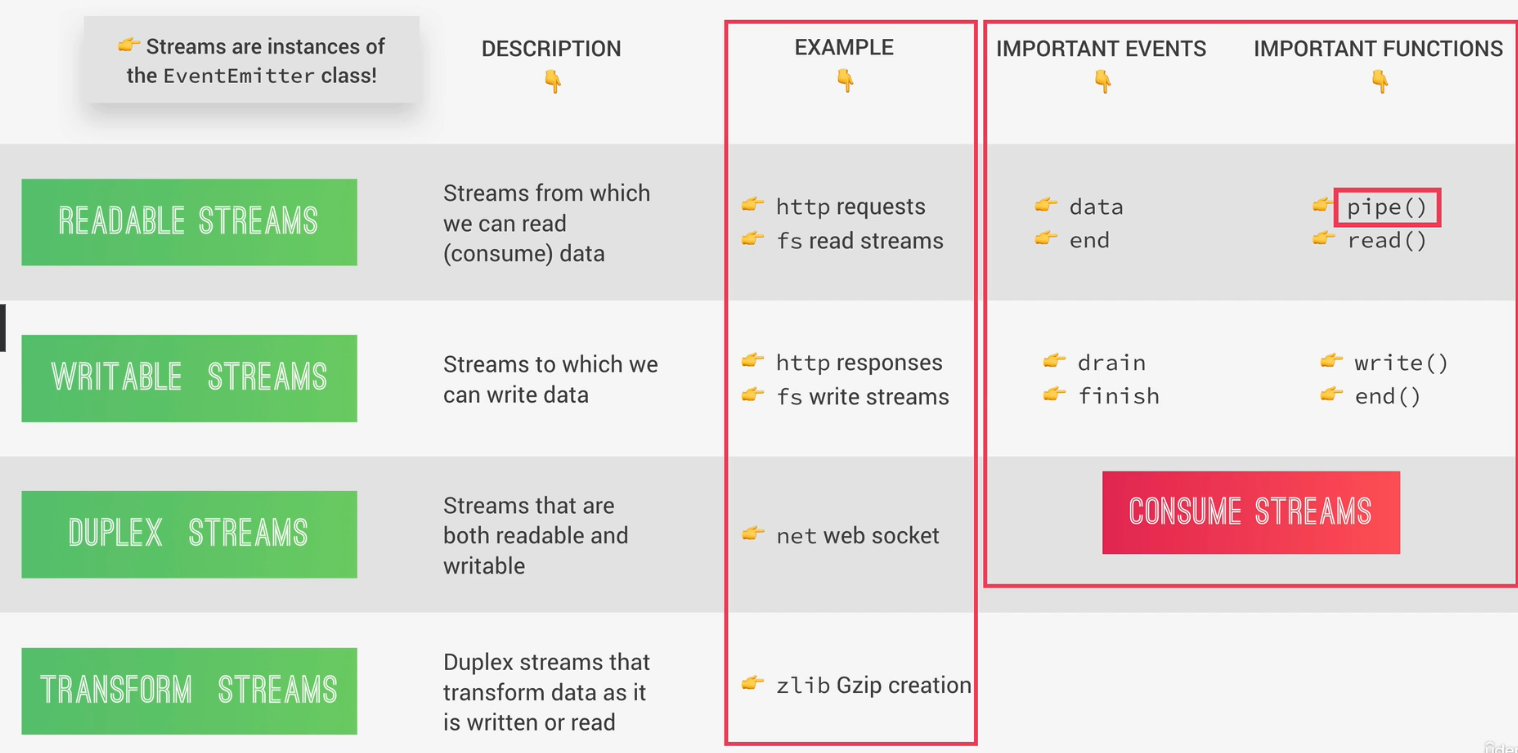


- Events u praksi:

* Kreirati class koja ce naslediti EventEmitter 

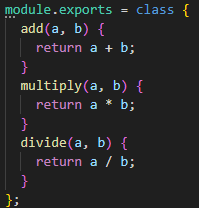
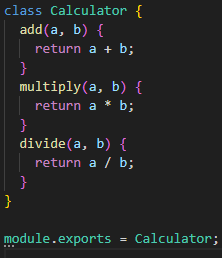
- Streams:







- Modules in practice:

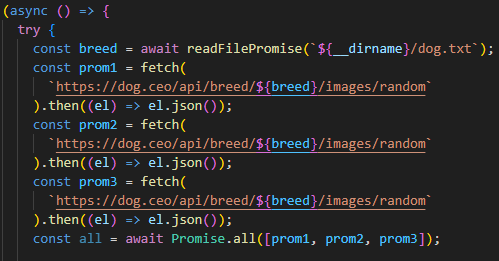




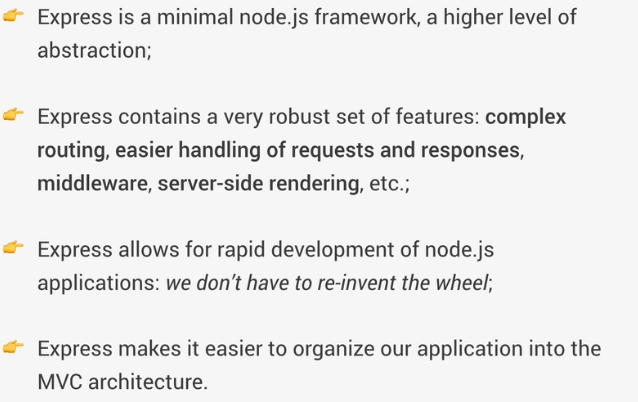
- Asynchronous JavaScript. From Callback Hell to Promises. async/await



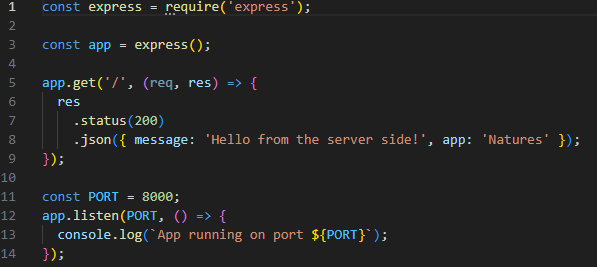
-Promise. all()



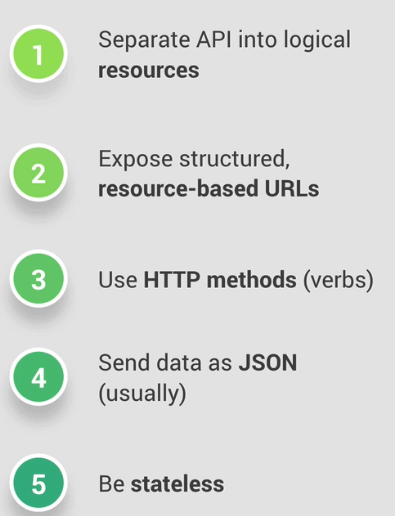
**Express**

****

- Basic setup:



- Rest API:



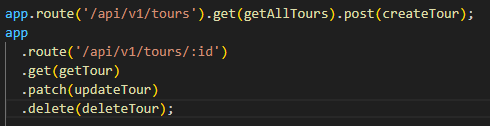
- Razlika izmedju PUT I PATCH:

* PUT menja ceo objekat
* PATCH menja samo deo koji je poslat

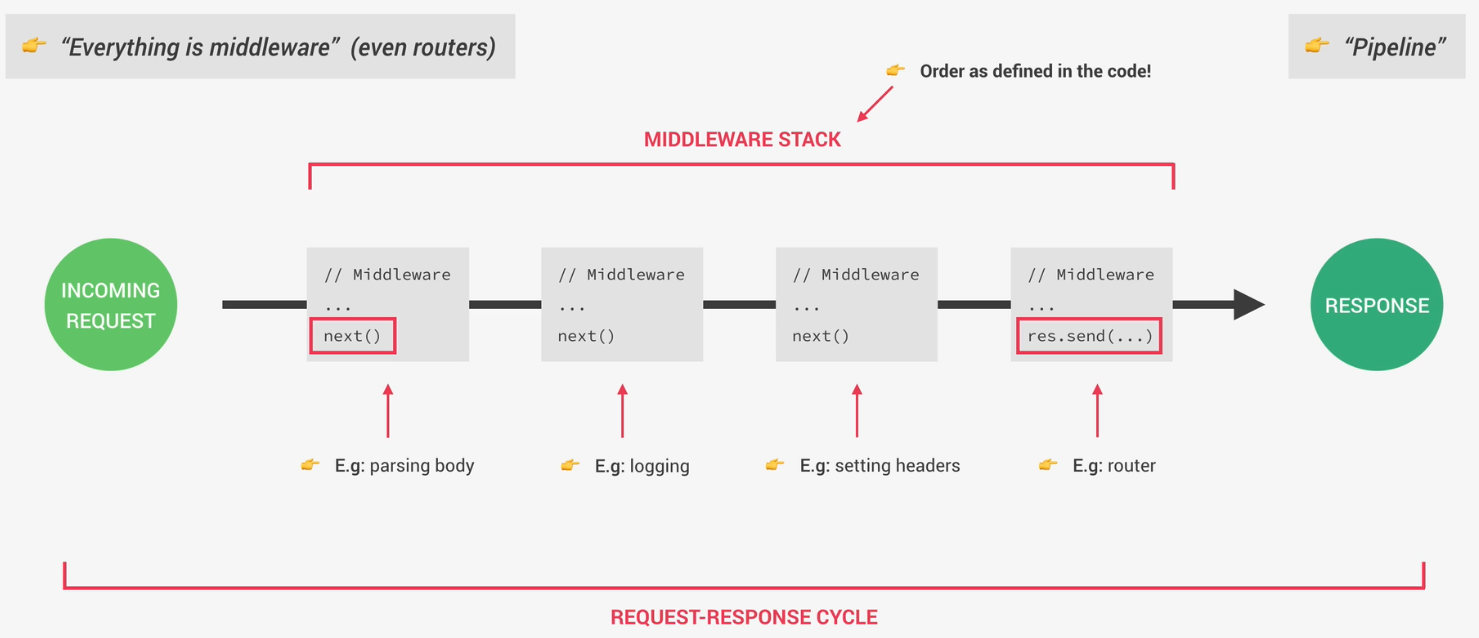
- Kod POST, express ne ‘stavlja’ body data u request, pa moramo da koristimo middleware



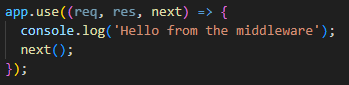
- Express routes:



- Middleware and the Req-Res Cycle:



* Da bih koristio middleware kucam app.use()



* Morgan middleware – pomaze pri dev-u

