# NodeJS Essentials Notes

## Chapter 1

Useful to know – JS and CLI

Node JS is an open source, cross platform JavaScript environment used for executing JS code outside a browser. We use node to build APIs – Application Programming Interfaces. These are the services that power client applications e.g. web app or mobile app. Client apps are just what the user sees – they are the surface. They talk to a server to get data, send stuff etc.

Node is ideal for building highly scalable, data intensive and real time back end services. Node is easy to start, can be used for prototyping and agile development. It is fast and highly scalable.

Paypal use it. Uses JS everywhere, cleaner and consistent codebase. Largest ecosystem of open source libraries.

### NodeJS History

Created by Ryan Dawe – JS run time based on chrome v8 engine

2011 NPM v1 to allow sharing of open source node library

Nodejs foundation formed 2015 – IBM, Microsoft, PayPal – open source community like Linux

### How NodeJS works and why is it so fast?

### Two restaurants analogy

#### Restaurant 1 Apache Steaks and Chops

Big restaurant – every new guest represents a new user, making order is like making a request

If I place an order – I need to hire a waiter – in this case the waiter is a thread. This is similar to how apache works, when he has the order the waiter takes the order to the kitchen and then just waits till the food is ready to take. So if you want to order anything else, I can’t do that until the chef is ready and the waiter comes back. In this analogy – the chef represents the file system or a data store. In apache, the single thread waits for file system to finish and read files. Waiter brings me food and then order class of water – when thread is finished the waiter is then fired i.e. thread released. So when really busy, every guest has their own waiter but there is a lot of latency as the waiters will be hanging around the kitchen. If the restaurant gets really buys or wants to expand it will require lots of space for all the waiters.

#### Restaurant 2 – chez node

There is only waiter – nodejs is single threaded. So if we order then the waiter will also take orders from all other guests in the restaurant. So when food ready, the waiter goes to get it as soon as he can and then can take orders from other guests and get their food when ready. We can say this waiter behaves asynchronously. Everything a waiter needs to do represents a new event – new table, new order, deliver order - waiter does not wait and things get sorted in the order they are raised. Are waiter is non blocking – it does not wait and is busy, busy, busy.

Non-blocking, event driven I/O – single thread responds to events in the order that they are raised. Thread behaves asynchronously – does not have to wait until resources are finished whilst it does something else.

If this restaurant gets busy we can franchise it – chez node can be expanded by duplicating or forking the restaurant into a neighbouring space. This is how we host nodejs applications into the cloud. Nodejs is single threaded, all the users are sharing the same thread, the events are raised and recorded in an event queue, nodejs is asynchronous which means it can do more than one thing at a time.

## Chapter 2

NodeJS runs JS files which are text

Two ways to update node – either with nvm or with a helper called -n

All notes should be available on the code now and in the