AKHIL SHARMA

NodeJS

An Introduction

SCALER 

AKHIL SHARMA

Flow

- 15 Mins Presentation and briefing.
- 15 Mins installation guides
- 5 Mins session doc walkthrough
- 10 Mins Project setup
- 10 Mins Github project overview
- 10 Mins Break
- 2 Hour Coding
- 10 Mins Q and A

- 2 Hrs 55 Mins.

AKHIL SHARMA

About Session

- Basic Level session
- Understand the basics of NodeJS
- Build a CRUD App, backend APIs

Expectations

- Don't be harsh on your self
- 3 Hours won't make you an expert, requires months of practice.
- Today's session will only accelerate your learning, that's it!

AKHIL SHARMA

IMPORTANT

- Javascript basics absolutely necessary
- Basic NodeJS knowledge is helpful
- VS Code installation is helpful
- Do NOT have to install NodeJS
- Coding ALONG with me is NOT COMPULSORY. (If you're not comfortable, just sit back and just observe.)

AKHIL SHARMA

ERRORS

- We will build a simple CRUD app, the code will be very usable - use it across your apps.
- Since you're coding for the first time, you will get hundreds of errors. It is ALRIGHT!
- DO NOT PANIC!
- If you get an error, DO NOT paste it in the chat
- You have to DEBUG on your own.

AKHIL SHARMA

DEBUGGING

- Session is only 3 hrs, focus on understanding the concepts even if your code doesn't work now.
- If you're unable to debug, compare your code with my github code, AFTER the session.
- You can copy the errors and paste them in google, you will always find something on stack overflow to fix it.

AKHIL SHARMA

REPEAT

- If you don't understand much, do not panic, build the project after the session from scratch by referring my code.
- This session will repeat in a month or so, join back again in case you're unable to follow at all.
- DO NOT DISTURB the other learners by spamming or sending errors in the chat. It will affect their learning.

AKHIL SHARMA

IMPORTANT

Local Vs. Sandboxes

Install Vs. Gitpod

Code Vs. Github

Intro

- Use Javascript in the server side
- Uses V8 engine by google, written in C++
- Has a command line tool
- Great for concurrent environments
- NOT a framework or a library, just a runtime environment for JS on V8 to be used on servers
- Event driven, non-blocking, lightweight
- NodeJS is single threaded
- Create HTTP server, TCP server, DNS server, Static file server, Web chat apps, Video streaming servers (real time servers)

Intricacies

Libraries as Packages

- Access thousands of libraries, simply import them as packages whenever needed.

NPM

- Special software and platform, NPM for managing packges in projects

Launch Own Modules

- Simple to create your own modules and launch for the entire world to use

Packages

Install

- npm install "modulex"

Import

- var modulex = require('modulex')

Run local

- npx nodemon

Package.json

- List of dependencies

Package-lock.json

- Dependencies of dependencies

node-modules

- Actual code of the dependencies

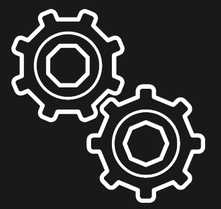
Use cases

APIs

Create general purpose CRUD APIs very easily.

Integrations

Create integrations, set events and triggers for events etc.



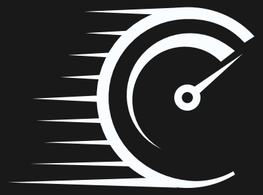
Layered business logic

Layered, complex business logic apps are a perfect fit.

When not to use

CPU intensive

Heavy calculations (event-loops are CPU hungry)



Low level stuff

Not fast enough for low level stuff

Parallelization

Processor utilization, parallelization not possible. No support for parallelization and is also single-threaded :(

Popular Frameworks

Express, hapi, koa

- Easy creation of APIs

Moleculer

- Fast development and deployment of microservices.

Nest, Sails, Loopback

- Create CRUDs and standard apps with ease