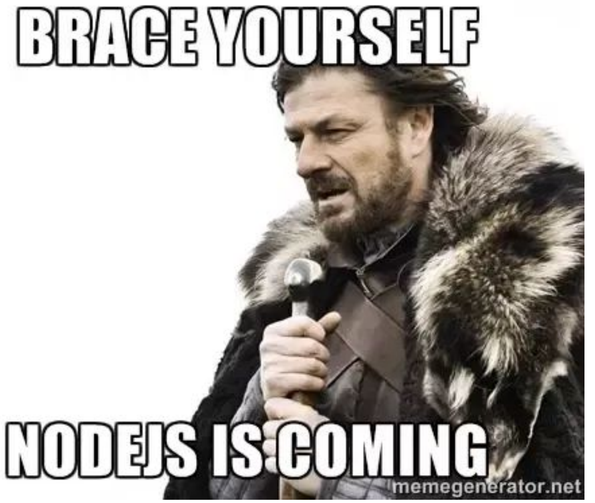
Developer Student Clubs

Create API's and Use them using NODE JS and MongoDB

Mohit Gupta
DSC CGC
@guptamohit_004

Google Developers

Introduction to NODE JS



What is NODE JS ?

NODE JS is an open source runtime environment build built on Chrome's V8 JavaScript engine.

Used developing server-side and networking applications.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent

Node.js = Runtime Environment + JavaScript Library

.

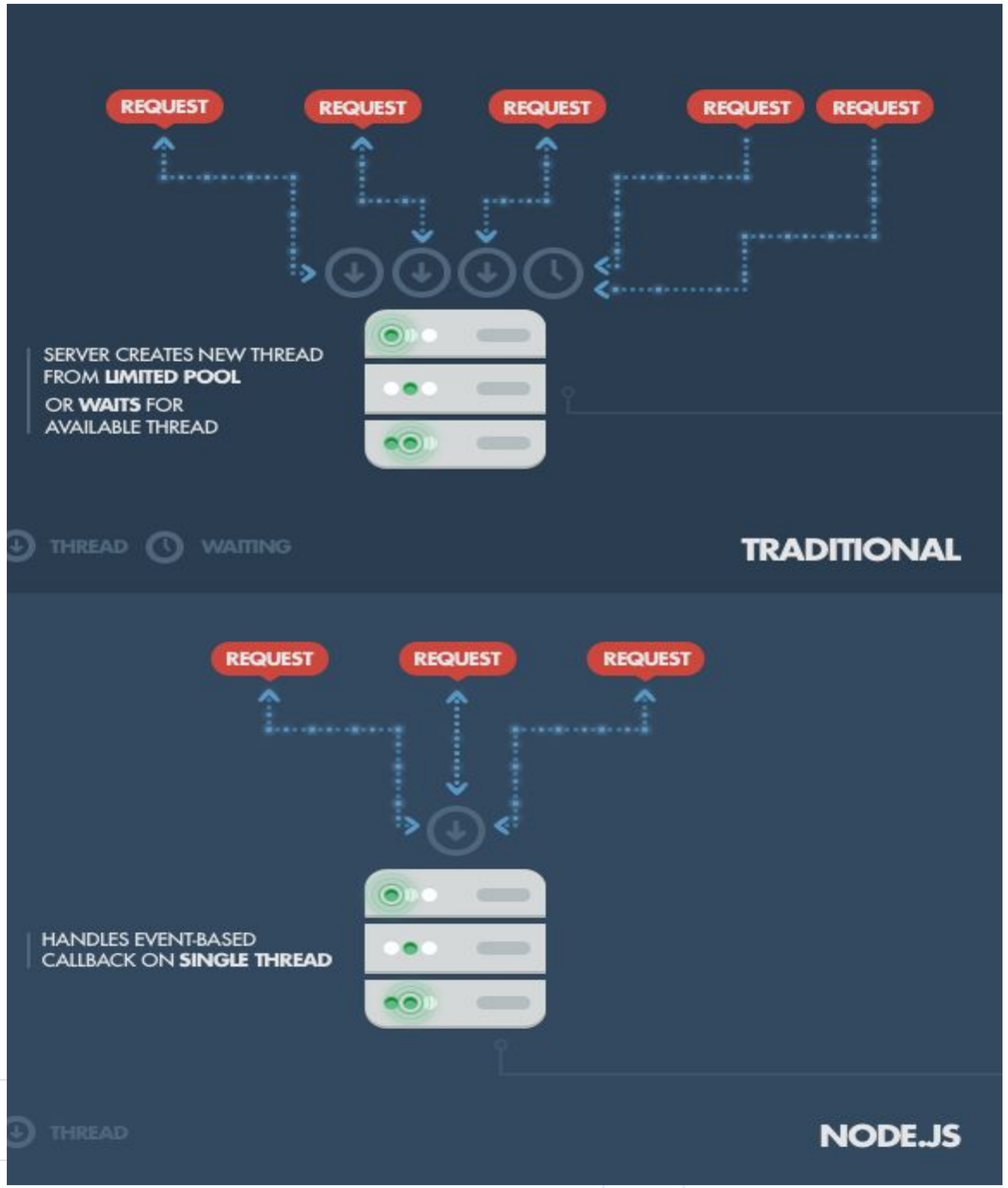
Why NODE JS-

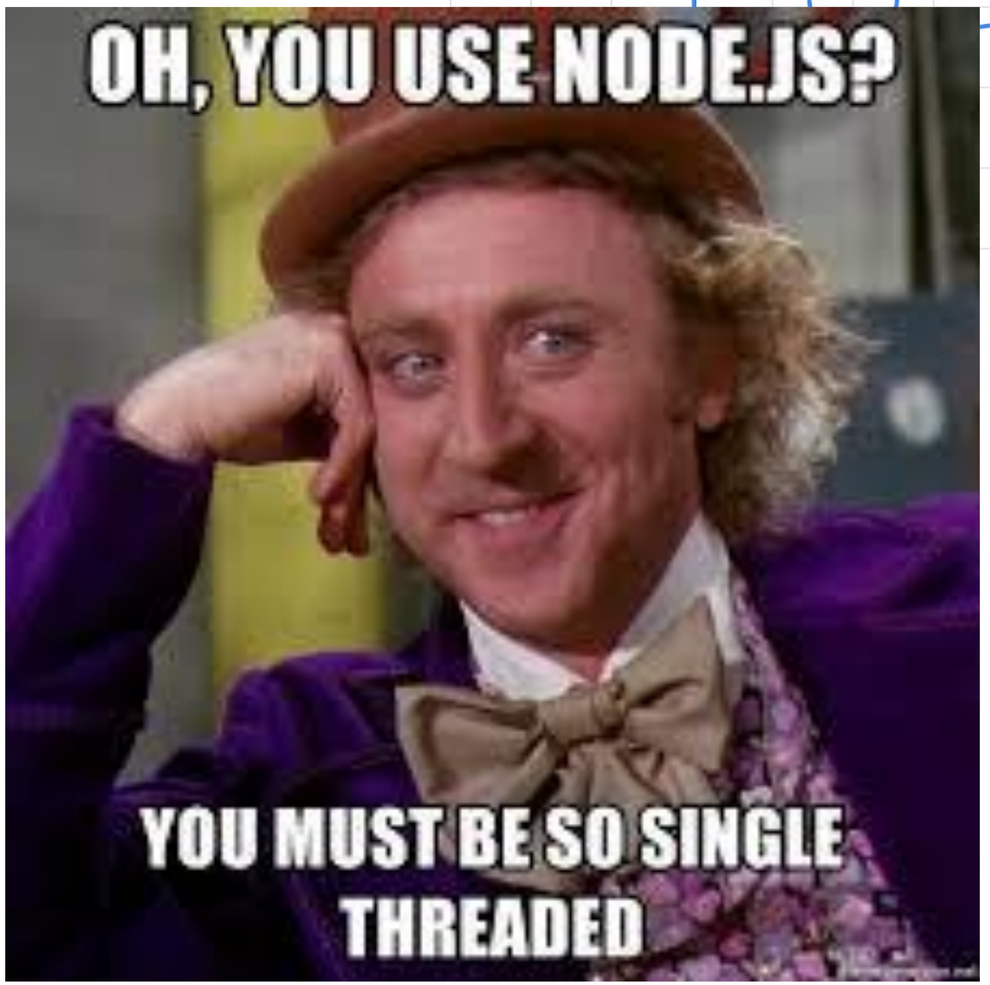
The main idea of Node.js: use non-blocking, event-driven I/O to remain lightweight and efficient in the face of data-intensive real-time applications that run across distributed devices.

Features-

- Asynchronous
- Cross-Platform Development
- Very Fast
- Performance
- Single Threaded but Highly Scalable.
- npm Enterprise
- Scalability
- No Buffering

SINGLE THREADED-





Developer Student Clubs

INSTALLATION of NODE JS

How to install NODE JS ?

Go to the site https://nodejs.org/en/download/ and download the necessary binary files.

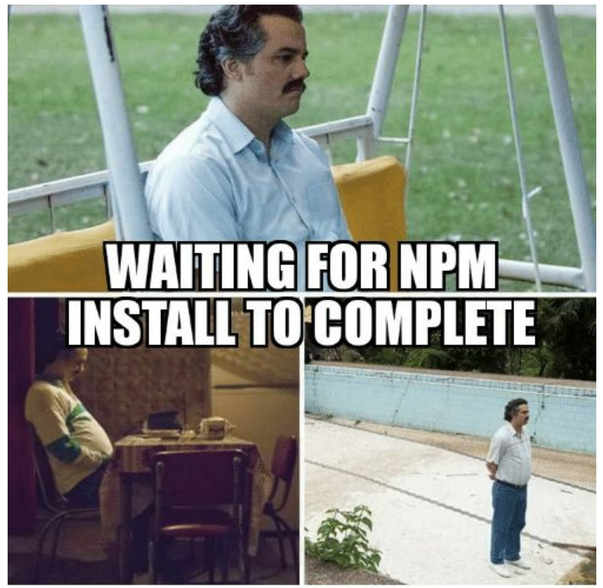
Confirm INSTALLATION ?

Run any of the two commands to confirm -

- Node -v
- Npm -v

NPM ?? What is npm ?? Let us Discover

NPM ?? What is NPM ??



npm install

- npm is the world's largest Software Registry.
- The registry contains over 800,000 code packages.
- npm is free to use.
- You can download all npm public software packages without any registration.
- npm is installed with Node.js.
- Install any module from https://www.npmjs.com/.
- Any module can be downloaded using command npm install <package-name>.
- Most Used NPM modules.
  1. Express
  2. Lodash
  3. Async
  4. Request
  5. Moment

Install NPM Package and Use it.

- Go to CLI in your System.
- Get the name of package, you want to install.
- Type npm install <package-name> and hit enter.
- Wait for the package to get installed.

Example-

Package Name - moment.

Installation - npm install moment

Description - Parse,validate,manipulate and display dates and times in JavaScript.

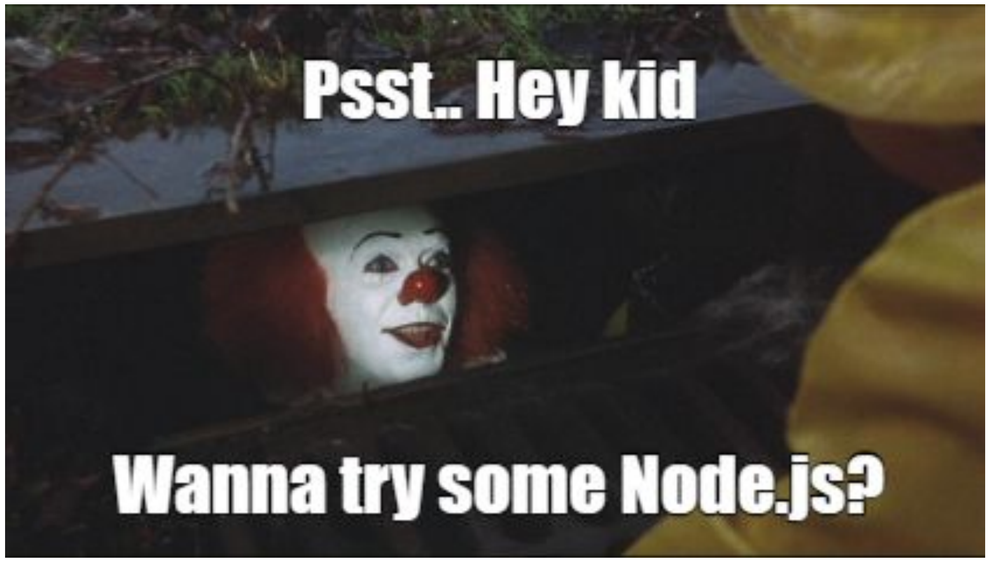
```
C:\Users\mohit>npm -v
6.4.1

C:\Users\mohit>npm install moment
npm WARN saveError ENOENT: no such file or directory, open 'C:\Users\mohit\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\mohit\package.json'
npm WARN mohit No description
npm WARN mohit No repository field.
npm WARN mohit No README data
npm WARN mohit No license field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wa

+ moment@2.24.0
added 1 package from 6 contributors and audited 25324 packages in 49.447s
found 1079 vulnerabilities (1033 low, 3 moderate, 43 high)
  run `npm audit fix` to fix them, or `npm audit` for details

C:\Users\mohit>node
> var moment = require('moment');
undefined
> moment().format();
'2020-04-20T00:26:01+05:30'
>
> moment().format("dddd, MMMM Do YYYY, h:mm:ss a");
'Monday, April 20th 2020, 12:26:57 am'
```

My First NODE JS Program-



- Open your favorite code editor.
- Create a file named as hello.js.
- Type in console.log(“Hello World”);
- Save the file.
- Open terminal and run command (node hello.js).
- Hello World got printed on console ??
- Yipee, you successfully created your first Node JS program.

Package.JSON in a NODE Application

- All npm packages contain a file, usually in the project root, called `package.json` - this file holds various metadata relevant to the project.
- It can also contain other metadata such as a project description, the version of the project in a particular distribution, license information, even configuration data .
- The `package.json` file is normally located at the root directory of a Node.js project.
- The `dependencies` field is used to list all the dependencies of your project that are available on `npm`. When someone installs your project through `npm`, all the dependencies listed will be installed as well.
- **It is the most important file for a NODEJS application.**

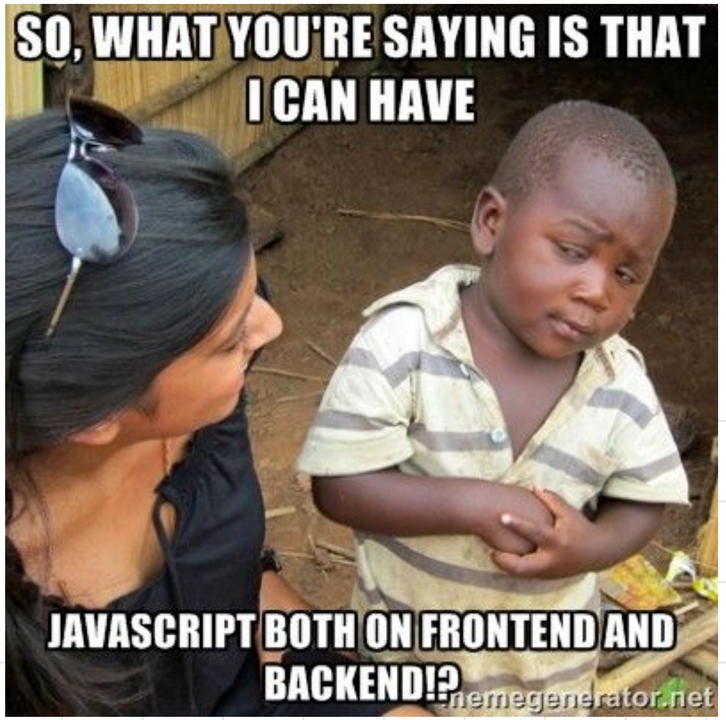
Our junior developer had no idea what's package.json so he decided to do this

Node JS on Frontend.

Node.js is a runtime environment, which **let users choose how to use, whether frontend or backend**, and one common language can be used as backend and front end.

USAGE-

These days, the code that we write in our text editors is not at all identical to what is present in our production bundles. This especially applies to the front-end projects - where we're not writing raw HTML and CSS anymore, and even our JavaScript gets severely transformed and minified before reaching the web browser.



Frontend Frameworks-

- Angular Js
- React Js
- Vue Js

Developer Student Clubs

Node JS and Database.

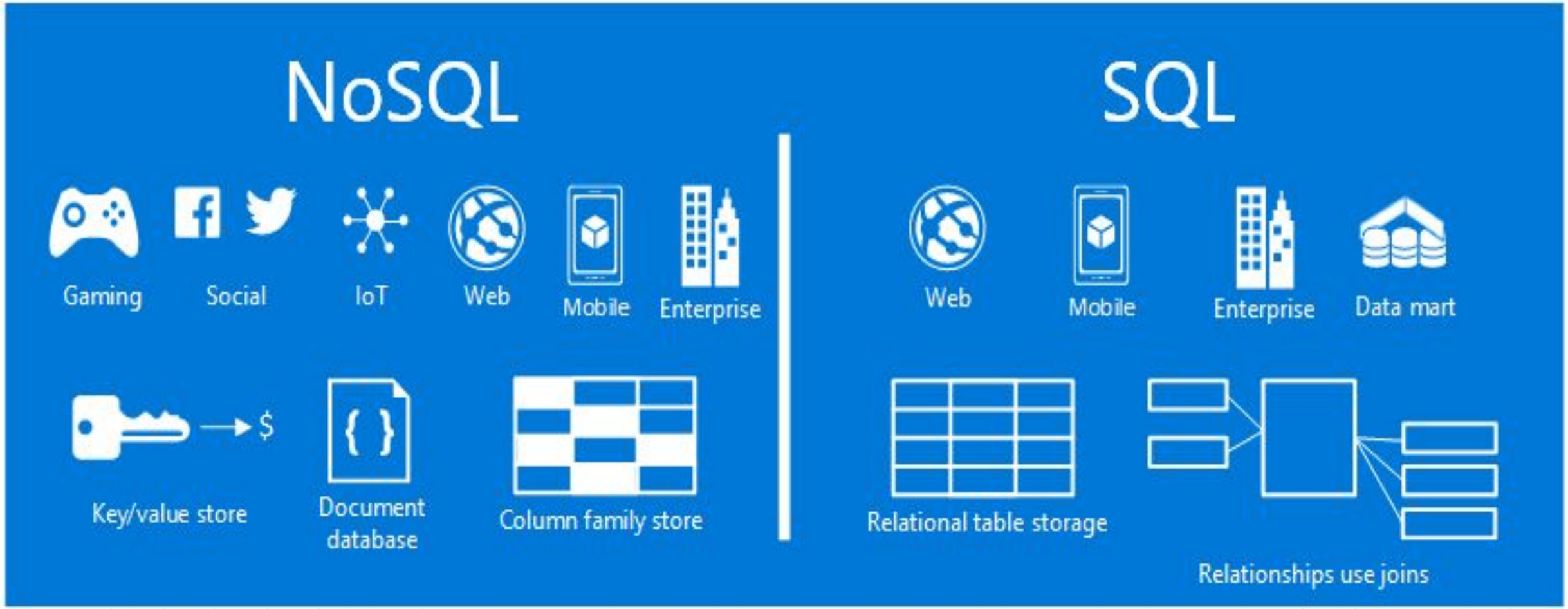
Node.js supports all kinds of databases no matter if it is a relational database or NoSQL database. However, NoSQL databases like MongoDb are the best fit with Node.js.

To access the database from Node.js, you first need to install drivers for the database you want to use.

Different Database Supported by Node JS

- MS SQL Server
- MySQL
- MongoDB
- Neo4j
- Redis

SQL VS NOSQL


NoSQL
Gaming
Social
IoT
Web
Mobile
Enterprise
Key/value store
Document database
Column family store
SQL
Web
Mobile
Enterprise
Data mart
Relational table storage
Relationships use joins

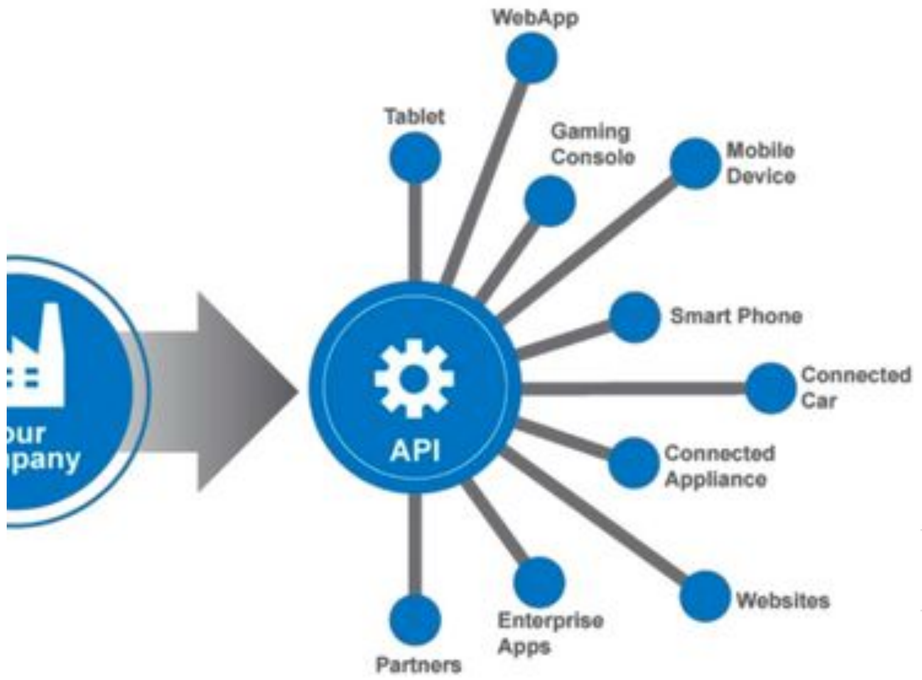
Node JS and MONGODB.

Why MONGODB ?

- Though Node.js works well with MySQL database, the perfect combination is a NoSQL like MongoDB wherein the schema need not be well-structured. MongoDB represents the data as a collection of documents rather than tables related by foreign keys.
- Document-based data storage is the main aim of using a non-structured database like NoSQL. MongoDB is a distributed database which allows ad-hoc queries, real-time integration, and indexing efficient. Moreover, MongoDB is open-source and perfect for frequently changing data. It also offers server-side data validation.
- Companies using MONGODB -**Uber, Lyft,Codecademy.**

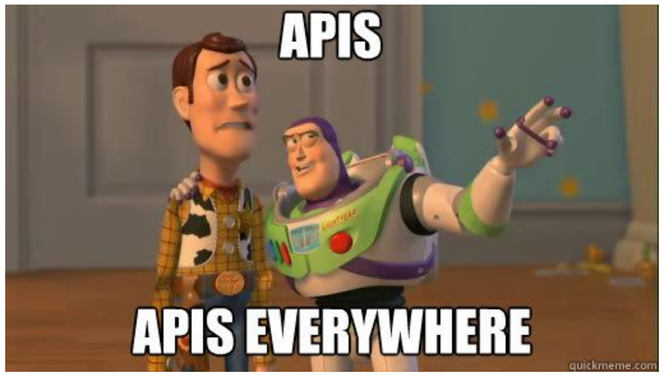
Application Program Interface (API)

- An API is a set of definitions and protocols for building and integrating application software. API stands for application programming interface.
- APIs let your product or service communicate with other products and services without having to know how they’re implemented.
- Different Approaches to APIs-
  - Private
  - Public
  - Partner



NODE JS and APIS-

- Today, we will be learning how to create and use Rest API's using NODE.JS.
- REST stands for REpresentational State Transfer. REST is web standards based architecture and uses HTTP Protocol.
- A REST Server simply provides access to resources and REST client accesses and modifies the resources using HTTP protocol. Here each resource is identified by URIs/ global IDs.
- REST uses various representation to represent a resource like text, JSON, XML but JSON is the most popular one.
- HTTP methods
  - GET – This is used to provide a read only access to a resource.
  - PUT – This is used to create a new resource.
  - DELETE – This is used to remove a resource.
  - POST – This is used to update a existing resource or create a new resource.



Creating APIS and use them-

- Let’s start our journey by creating our First API.. Excited ??
- Fire up your terminal and create a new folder for the application.
- Initialize the application with a package.json file.
- Install dependencies.
- Setting up the web server-
  - const express = require('express');
  - const bodyParser = require('body-parser');
  - const app = express();
  - app.get('/', (req, res) => {res.json({"message": "My First API."});
    });
  - app.listen(3000, () => { console.log("Server is listening on port 3000"); });

CRUD operations in a File

- What is **CRUD**?

  Create, Read, Update and Delete are the four basic functions of persistent storage.

- CRUD Operations-
  - **CREATE :** Adds in a new Data.
  - **READ :** Reads the Data.
  - **UPDATE :** Updates the Data.
  - **DELETE :** Deletes the Data.

- The Node.js file system module allows you to work with the file system on your computer.
- To include the File System module, use the `require()` method:
  - `var fs = require('fs');`

Connect Node JS with Mongodb

To connect to MongoDB from Node.js using **Mongoose** package, call connect() function, on the variable referencing to mongoose, with MongoDB Database URI passed as argument to the function.

- Get the URL to your Database. (Install mongodb locally and get db url or use MLAB or MONGO ATLAS).
- To get a reference to the database specified, use connection() function on Mongoose reference, as shown below :
- To check if the connection is successful or not, you may use callback functions : on() and once().
- One can also uses Javascript promosies to check if connection was established or not.

CRUD operations in a Mongo Database

Create, read, update and delete (as an acronym CRUD) are the four basic functions of persistent storage. Using some methods of Mongoose we will implement CRUD operations.

Lets view the methods for CRUD operations using the MONGODB (NOSQL).

- **CREATE** - Create function present in Mongoose to create a Document.
- **READ** - find(), findOne(), findById() functions present in Mongoose to Read data from a document.
- **UPDATE -** update(), findOneAndUpdate(), findByIdAndUpdate() functions are present in Mongoose to update the data of a document.
- **Delete -** remove(), findOneAndRemove(), findByIdAndRemove() functions are present in Mongoose to delete the data from a document.

Developing CRUD API's

So, let's start with developing our CRUD API using Node JS to create,update,read and delete data from a MONGODB database.

- MVC pattern will be used to develop the API.
- Schema would be desgined.
- All the operations (CRUD) will be done on the database using the API's.

Testing and Usage of API's-

- All the API's will be tested using POSTMAN.
- The request and response of the operartions would be senn in real time.
- All the updates can be seen in the database.

Questions ???

Resources for NODE JS-

- GITHUB Repo - https://github.com/DSC-CGC/NODE-JS-WEBINAR
- Presentation - Create API's and Use them using NODE JS and MongoDB
- MONGO ATLAS - https://www.mongodb.com/cloud/atlas