

## Nodejs- Day -1: Nodejs

"What is nodejs?"

Npm packages

- In build
- Third-party
- Custom

**Node Introduction :**

Node.js is an open-source, cross-platform **JavaScript runtime environment** that allows developers **to run JavaScript code outside of a web browser**.

It is designed to be efficient and scalable, making it well-suited for building server-side and networked applications.

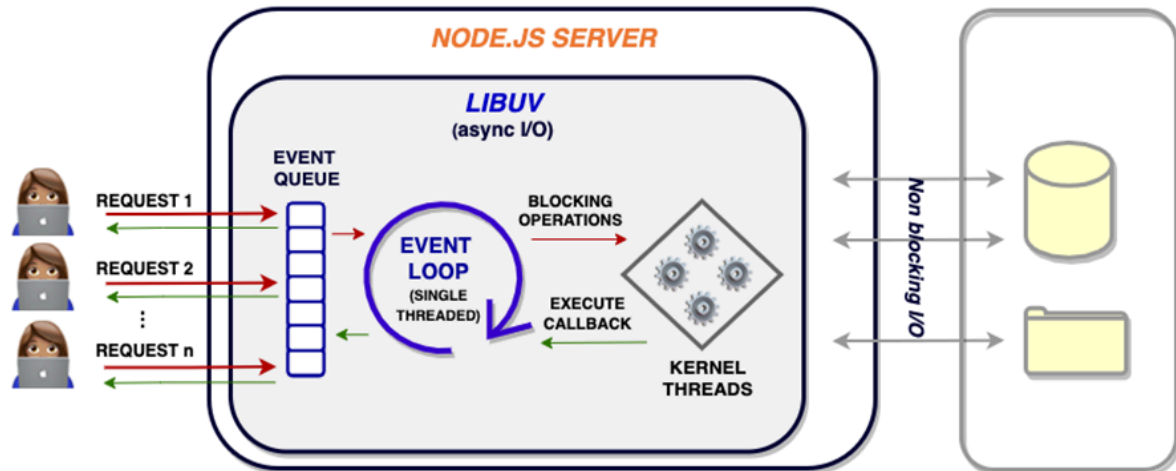
**Node.js is built on the V8 JavaScript engine**, the same engine that powers Google Chrome, and it uses a **non-blocking, event-driven architecture** that makes it ideal for handling asynchronous tasks and I/O operations.

**Non-blocking I/O: (async operation)**

Node.js uses an **event loop to handle I/O operations in a non-blocking manner**. This allows it to efficiently handle many **concurrent connections without the need for multiple threads**.

## Event-driven:

Events triggered by user actions or system events



## In-Built Packages:

- These are packages that come with Node.js, and you can use them **without installing** anything.

**Http, fs, path, router**

## Third-Party Packages:

- These are packages created by the Node.js community or other developers. You **need to install** them using npm.

**Express, cors, jwt, bcrypt, nodemon, mongoose**

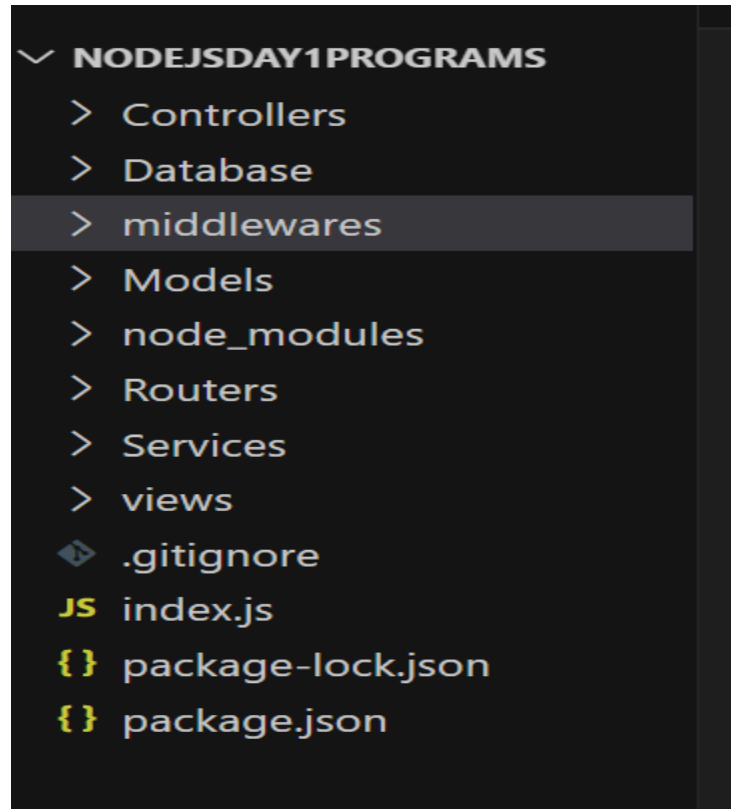
## Custom Packages:

- These are packages that you or your team **may create** for specific use cases.

## Session flow/plan:

### Third-Party Packages:

API creation using express and show postman demo . And  
**Real-time folder structures.**



### In-Built Packages:

**Task oriented example** & deploy . It helps learners deploy their first Nodejs task onwards.

**Task Explanation:** **File system**

## Using Third Parties:

```
import fs from 'fs' // default/ built-in package
```

```
import { format } from 'date-fns'; // third party
```

## Using Inbuilt Functions:

```
fs.writeFileSync(filePath,timeStamp,'utf8')
```

```
fs.readFileSync(filePath,'utf8')
```

### **writeFileSync and readFileSync:**

When you switch to using writeFileSync and readFileSync from the Node.js fs module, the code works correctly **because these are synchronous versions of the file system and do not require callbacks, and they do not return promises**