**Nodejs server side language.**

**Nodejs is runtime environment for executing Javacript code.**

**We use node for build back end services.**

Node being a **single thread, and uses async function** calls to maintain the concurrency and can serve much larger number of requests compared to any traditional server like Apache HTTP Server

 Nodejs is not programming languge.

Nodejs is not framework.

All asynchronous method take fuction as last argument.

**NPM**:

Node Package Manager (NPM) is a **command line tool** that installs, updates or uninstalls Node.js **packages** in your application.

It is also an online repository for open-source Node.js packages. The node community around the world creates useful modules and publishes them as packages in this repository.

**Express.js**

1. Express.js is a **web application framework** for Node.js. It provides various features that make web application development fast and easy which otherwise takes more time using only Node.js.
2. **Allows you to create REST API server.**
3. **We can use to perform additional tasks on request and response.**

**body-parser:** to take input from form(reading data from form )

//Make sure you place body-parser before your CRUD handlers!

app.use(bodyParser.urlencoded({ extended: true }))

To serve static resources:  
It is easy to serve static files using built-in middleware in Express.js called **express.static.** Using express.static() method, you can server static resources directly by specifying the folder name where you have stored your static resources.

app.use(express.static('public'))//to load static resources like mages, CSS files, and JavaScript files in a directory named public:

### **res.render(view [, locals] [, callback])**

// send the rendered view to the client

res.render('index')

// pass a local variable to the view

res.render('user', { name: 'Tobi' });//important

)

**res.send([body])**

This method is used to send HTTP response.

res.send('<p>some html</p>')

res.send({ some: 'json' })

### **res.redirect([status,] path)**

Redirects to the URL derived from the specified path

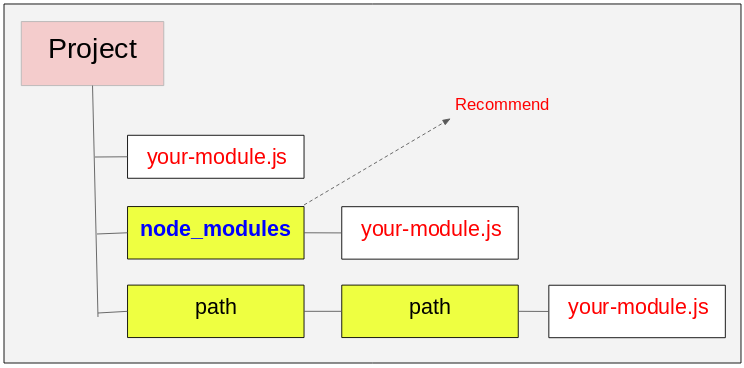
res.redirect('/foo/bar')

res.redirect('http://example.com')

## **Create your Module**

In the **NodeJS,**you can create a customized **module**. The  **Module** is a  **Javascript**file. please note to the position of this file because it affect the way for you to use it . You have 2 ways of choosing position of this file:

1. The **Module** file is put in the directory named **node\_modules** (subdirectory of the project). **This is encouraged position.**
2. The **Module** file is put in one of your directories, not **node\_modules**.



**Why we need Express instead of NodeJS**

Let take example:

var http = require('http');

var server = http.createServer(function (req, res)

{

if (req.url == '/'){

//perform action for root

}

if (req.url == '/data'){

//perform action like send response

}

});

server.listen(5000);

console.log('Node.js web server at port 5000 is running..')

See in above example we need use if again and again to check URL, so instead of checking using if again and again then Express come into the picture

var express = require('express');

**var app = express();//we use app var (don’t need to use createServer of http)**

var bodyParser = require("body-parser");

app.use(bodyParser.urlencoded({ extended: false }));

app.get('/', function (req, res) {

});

app.post('/submit-student-data', function (req, res) {

});

var server = app.listen(5000, function () {

console.log('Node server is running..');

});