1)Inroduction

Nodejs is run time which allows us to run javascript on server.

We mostly use javascript in client(browser). To understand how js code runs on server we have to understand how it runs on browser. Browser developers provide engines in browsers which interpret js code. Such a engine is chrome’s v8 engine. It is developed by google.

They realized that js is slow and they could speed it up if they transpile js code into machine code and them execute it . this is one of reason why v8 is fast. This v8 engine is used by chrome. It is open source and it is same engine that powers nodejs. If you have engine you can run js anywhere , it dnt need to browser.

Ofcourse node has made some adjustments because we need some functionalities when running it on server, when creating a server that we dnt need in browser and so on.but ingeneral nodejs uses v8 engine to run js files that you give to it. In future videos we will see how to setup a server with nodejs.

Why we need nodejs? Is it better that php , .net,ruby? well it depends. There are certain advnatges and disadvantages.

Huge advantage of node is you can use same language on front end and backend.

Amother advantage of node is that it is non-blocking which means it uses one thread but is asynchronous by nature. So it is very fast. It is very good at handling multiple thousands of request at same time.

What it is not good at is handling a single data intensive request. There are better languages. It can be slow , it’s not fast in every situation. But it is generally fast when serving multiple clients at same time, multiple request at same time.

Another disadvantage is that node js is not very mature. So you cannot find many resourses of node.

2)First Application

Idownload nodejs and run its installer. You dnt need to install any server. We will create our own server with nodejs.

Now create app.js file. Here we will cwrite our server. Nodejs is run time executing our js code.node js provide certain functionalities which allows us to create server functionality. For ex we can tell nodejs that it should listen to certain port on a certain domain and all request coming in should be handles in a certain way. This will what we will do in this file. We will tell node js to listen to incoming request and do something upon such request.

So nodejs will execute this server,js file. It will run through it once and then finish, but in this file we will kind of start loop where node js continues. So you listen to request.

So first we import or require http module.

var http = require('http');

We will come back to basic js things that you should know when using nodejs. later. For now think it like we are importing a functionality that we need.then we use createServer method to create a server. Then we chain another method to listen to port 8000 -

http.createServer().listen(8000);

now all request port 8000 will be handled by this server.

Now server is doing nothing, so I specify a method, that should handle these request. This functions expects 2 arguments. These arguments are passed by nodejs.

# 3)JavaScript Modules & Anonymous Functions

# In this lecture we define 2 modules and we imported them in server.js and used them there. In module 1 we expoted some properties of module, but in module 2 we exported everything in that file. To export we use module.export.

# Now http module is built in module provided by node, that is why we can use it without creating it.

# 5)Rendering HTML as Response

# Now instead of returning a text, I want to return a index.html file to browser. So send it to client, first we need to grab this file on server. To access file system node js provide us fs module. We import it in app.js. now we want to read index.html in our application then send it to client.

# We use readFile method to read file. This method takes 3 argumets- path of file, options and call back. This callback will be executed when nodejs has completed reading the file. This callback gets 2 arguments first one is error and other one is data read from file.

# Now if we run our code we see html file is displayed in browser. But we want browser to interpret this file and show us web page. for this change ‘Content-Type’ header.

response.writeHead(200,{'Content-Type': 'text/plain'});

# to

response.writeHead(200,{'Content-Type': 'text/plain'});

# here important thing is we are calling response.end() inside our callback. This is important, if we call it outside our callback then may be we send the response even before we are done with reading file.

# 6)Routing

# Here we will create separate js file which will handle request and do routing. We create app.js.

# Here we set up our headers. Then we need to get path url, so that we can decide which page to render, depending on path of url. For this we use another core module of nodejs- url.

# After getting path in variable named path we use switch statement for routing.

# This is how we implement routing in nodejs.i wan to say straight way at this point-We are doing this with pure nodejs so you can get the basics. So you understand the way nodejs works. We will very soon use express framework which will make lot of these things we are doing by hand a lot easier. For example we use switch statement to implement routing. We have got uch better ways to do this when using actual framework on top of nodejs.

# 7)What is express and why should we use it

# We have to write lot of code for routing using pure nodejs. We have to write every logic- parsing routes, parsing urls, creating routing actions- all things have to written by us.

# We should focus on business logic and not on repetitive tasks. So that is why we use frameworks in general in all languages.

# Very popular node framework is express. Express built upon node and it just offers us lot of tools which makes writing node apps more fun and easy. Express also allows u or require us to use templating engines to creating HTML files, to dynamically inject values in these files.

# So for next of section I will use express. We will set up and look at basic folder structure.

# 8)Setup of Express

# We will use express generator to quickly create my express projects. This is npm module which will set us up a default folder structure with nodejs, express, already setup templating engine, with some example routes. Lets install express-generator-

# npm install -g express-generator

it will create folder structure. In bin folder we have www file. It is our server file. Here we create our server pass express app to it and give port. If we want to change ort w ecan do in this file.

Now we have to install dependencies. Run –

**npm install**

now to run our server, go to project path and run-

**npm bin/www**

another way to start our server is to run-

**npm start**

in project path(we have define start command in script in package .json)

9)Express Project Structure & First little app

First we will look at [www.js](http://www.js).

var port = normalizePort(process.env.PORT || '8000');

app.set('port', port);

normalizePortthis function converts the string into number. App is our express app , then we set port in express application.

Then we create server and pass express app to it.telling that expresss will be responsible fioor handling all the request.

server.listen(port);

server.on('error', onError);

server.on('listening', onListening);

we give port to our server. Last 2 are functions which listen to certain events- error event and listening event. Both of these call functions onError and onListening which are defined in the same file.

onError method takes care of whatever error may appear during listening, for example when port is already in use.

Now lets look at app.js, this is our express application-