



Technology Series

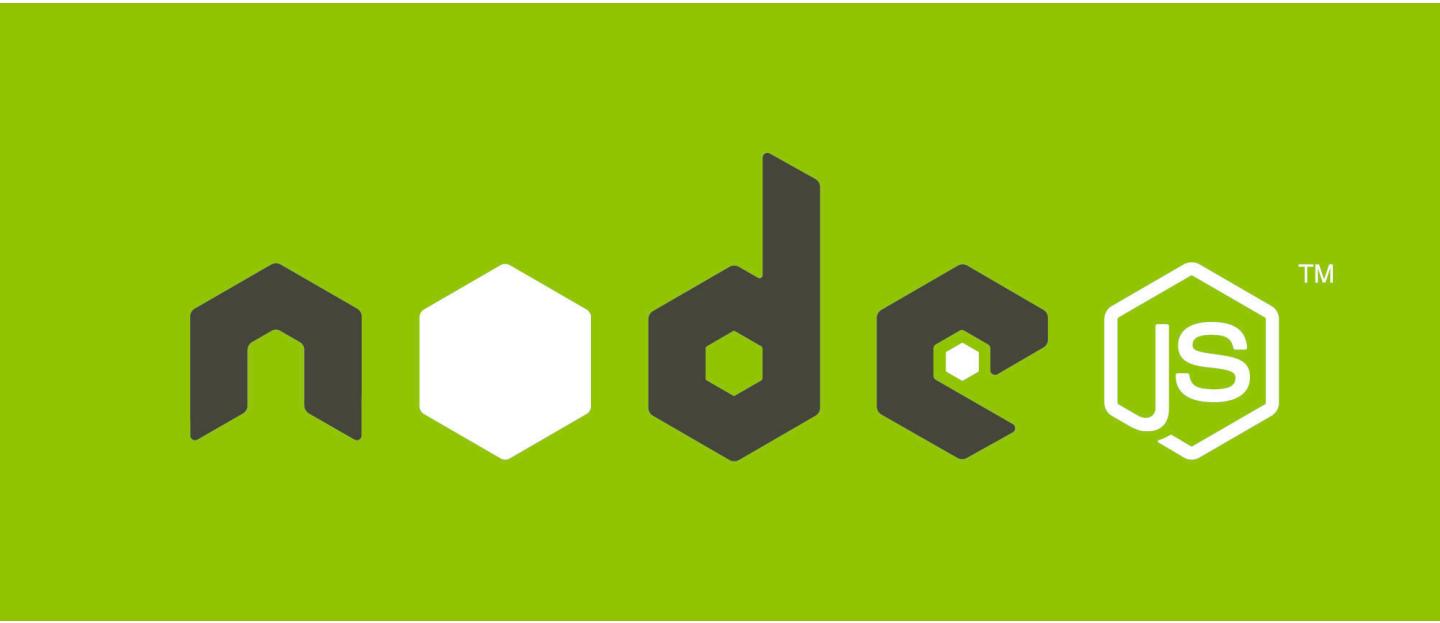
NodeJS

06

Presented by
Subhash EP



Subhash



Subhash

Node.js HTTP

- `net / require('net')`: provides the foundation for creating TCP server and clients
- `dgram / require('dgram')`: provides functionality for creating UDP / Datagram sockets
- `http / require('http')`: provides a high-performing foundation for an HTTP stack
- `https / require('https')`: provides an API for creating TLS / SSL clients and servers

http module

- Configure with `createServer`,
 - which takes a callback
 - returns an HTTP server
 - On each client request
 - the callback is passed in two arguments
 - request stream
 - response stream

http module

- To start the returned HTTP server
 - simply call its listen function
 - passing in the port number to listen on.

HTTP Demo

Connect

- core Node.js modules provide features for building web applications.
- There are quite a few web frameworks available on NPM that build on this.
- A very popular one is connect
 - (npm install connect),
 - a middleware framework.

Middleware

- sits between
 - application and
 - some low level API
- Connect extends the built-in HTTP server functionality
 - adds a plug-in framework.
 - The plug-ins act as middleware

Bare-Bones Connect Application

- At the heart of connect is connect function
 - creates the connect dispatcher.
 - connect dispatcher takes the request/response arguments
 - the dispatcher can be used as an argument to http.createServer

Basic Connect

```
var connect = require('connect')
, http = require('http');

// Create a connect dispatcher
var app = connect();

// Register with http
http.createServer(app)
.listen(3000);
console.log('server running on port 3000');
```

404 by default

- If you run this server,
 - it will return a 404 for every client request.
 - built-in behavior provided by connect.
- In the absence of middleware to handle the client request,
 - connect will return a 404



Creating a Connect Middleware

- To register a middleware with connect,
- call the ‘use’ member method
 - on the connect dispatcher
 - passing in a function that takes three arguments—
 - a request,
 - a response, and
 - a next callback

Next?

- next allows you to
 - optionally pass the control
 - onto the next middleware registered with connect
 - OR inform connect about an error

Connect / Middleware Demo

That's all



End of Session

©

subhash.ep@gmail.com
linkedin.com/in/subhashep