**Node**

**Module Patterns:**

var counter = function(){

return 'this is good';

};

var adder = function (a,b){

return `The sum is ${a+b}`;

};

module.exports = {

counter : counter,

adder : adder

}

**To provoke the functions in different file** :

var count = require('./count');]

console.log (count.counter());

console.log (count.adder(3,5));

**Reading File And making Directory**

The Module is “fs”.

var fs = require('fs');

fs.mkdir('stuff', function(){

fs.readFile('readMe.txt','utf8', function(err, data){

console.log(data);

fs.writeFile('./stuff/writeMe.txt', data);

});

});

To delete a file simply write : fs.unlinke(‘filename’);

To delete a directory : fs.rmdir(‘dir\_name);

Note: A directory can not be romovd if its not empty.

Example:

var fs = require('fs');

fs.unlink('./stuff/writeMe.txt', function(){

fs.rmdir('stuff');

});

**Creating Server**

var http = require('http');

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

res.writeHead(200, {'Content-Type':'text/plain'}); //200 is the status

res.end('whats good');

});

server.listen(3000, '127.0.0.1'); // that’s local ip address.

console.log('listening to port 3000');

**Read and Write Stream**

var fs = require ('fs');

var myReadStream = fs.createReadStream( \_\_dirname + '/readMe.txt', 'utf8' );

var myWriteStream = fs.createWriteStream(\_\_dirname + '/writeMe.txt');

myReadStream.on('data', function(chunk){ // chunk is some part of the full data

console.log('reading it');

myWriteStream.write(chunk);

});

Hosting a html file

var http = require('http');

var fs = require ('fs');

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

res.writeHead(200, {'Content-Type':'text/html'}); // “change the type to html”

var myReadStream = fs.createReadStream( \_\_dirname + '/index.html', 'utf8' );

myReadStream.pipe(res);

});

server.listen(3000, '127.0.0.1');

console.log('listening to port 3000');

**Jason (Java Script Object Notation)**

ar http = require('http');

var fs = require ('fs');

var myObj = { // making the object to send

name : 'anik',

job : 'engineer',

age : 22

};

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

res.writeHead(200, {'Content-Type':'application/json'});

// change the type to “application/json” res.end(JSON.stringify(myObj));

});

server.listen(3000, '127.0.0.1');

console.log('listening to port 3000');

**Basic Routing**

var http = require('http');

var fs = require ('fs');

var myObj = {

name : 'anik',

job : 'engineer',

age : 22

};

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

console.log('request was made:'+ req.url);

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

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

var myReadStream = fs.createReadStream(\_\_dirname + '/index.html', 'utf8').pipe(res);

}

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

var ninjas = [{name : 'anik', age: 19},{name: 'ting', 'age':21}];

res.writeHead(200, {'Content-Type': 'application/json'});

res.end(JSON.stringify(ninjas));

}

else{

res.writeHead(404, {'Content-Type': 'text/plain'});

res.end('404');

}

});

server.listen(3000, '127.0.0.1');

console.log('listening to port 3000');

**Routing with Express & Ejs Template**

**Code in app.js:**

var express = require ('express');

var app= express();

app.**set**('view engine', 'ejs'); // setting ejs. Ejs by default reads the files in view folder

console.log('listening to 3000');

app.**get**('/', function(req, res){ // sending a file

res.**sendFile**(\_\_dirname + '/index.html');

});

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

res.**send**('its the contact page'); //sending a text

});

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

var data = { age: 22 , job: 'ninja', **hobbies:['ramen','mission','trainig'**]}; //makin Object

res.**render**('profile', {person : req.params.name, data : data}); //sending Object

});

app.get('/person/:id',function(req,res){ // Using Req Params

res.send('This is profile of '+req.params.id);

});

app.listen(3000);

**Code in profile.ejs:**

<!DOCTYPE html>

<html>

<head>

<style>

body{ background: skyblue; color: #fff; padding: 30px;}

h1{font-size: 40px; text-align: center}

p,li{font-size: 26px; text-align: center;}

</style>

<meta charset="utf-8">

<title>whats good</title>

</head>

<body>

<h1>this is profile of <%= person %></h1>

<p>His age is: <%= data.age %></p>

<p>His job is: <%= data.job %></p>

<h1>His Hobbies are: </h1>

<ul>

<% data.hobbies.forEach(function(item){ %> // throwing java script.

<li><%= item %></li>

<% }); %> // there is no ‘=’ sign when throwing js code

</ul>

</body>

</html>