

Notes on Node.JS:

1. Module Declaration:

movies.js:

```
function printAvatar()  
{  
    console.log("Avatar: 01");  
}  
function printChappie()  
{  
    console.log("Chappie:02");  
}  
module.exports.avatar=printAvatar;
```

app.js:

```
var movies=require('./movies');  
movies.avatar();
```

output:

"C:\Program Files\nodejs\node.exe" C:\Users\Azizul56\IdeaProjects\azizul\app.js

Avatar: 01

Process finished with exit code 0

2. Multiple Modules(Another technique of declaring module):

movies.js

```
module.exports=  
{  
    printAvatar: function()  
{  
    console.log("Avatar");  
},  
    printChappie: function()  
{  
    console.log("Chappie");  
},  
    azizul: "Azizul"  
}
```

app.js

```
var movie=require('./movies');  
movie.printAvatar();
```

```
movie.printChappie()  
console.log(movie.azizul);
```

output:

"C:\Program Files\nodejs\node.exe" C:\Users\Azizul56\IdeaProjects\azizul\app.js

Avatar

Chappie

Azizul

Process finished with exit code 0

3.Shared State of modules:

Movies.js

```
module.exports=  
  {  
    favmovie: ""  
  }  
}
```

Azizul.js

```
var movies=require("./movies");  
movies.favmovie="The Titanic";  
console.log("Azizul's favourite movie is"+movies.favmovie);
```

Ashik.js

```
var movies=require("./movies");  
console.log("Zahirs's favourite movie is"+movies.favmovie);
```

App.js

```
var movies=require("./movies");  
console.log("Zahirs's favourite movie is"+movies.favmovie);
```

Output:

Azizul's favourite movie is The Titanic

Zahirs's favourite movie is The Titanic

4.Object Factory:

Movies.js:

```
module.exports = function()  
{  
  return  
  {  
    favMovies: ""  
  }  
};
```

Azizul.js:

```
var movies = require("./movies");
var azizulsMovie=movies();
azizulsMovie.favMovie="The Notebook";
console.log("Azizul's favourite movie is"+azizulsMovie.favMovie);
```

Ashik.js:

```
var movies= require('./movies');
var ashiksMovie=movies();
ashiksMovie.favMovie="The Catbook";
console.log("Azizul's favourite movie is"+ashiksMovie.favMovie);
```

App.js:

```
require("./Azizul");
require("./Ashik");
```

Output:

*Can't solve the bug.

5.Different Core modules:

```
var fs=require('fs');
fs.writeFileSync("corn.txt","Corn is good,corn is life");
console.log(fs.readFileSync("corn.txt").toString());
var path=require('path');
var website="Desktop/Azizul/CUET/index.html";
var websiteabout="Desktop/Azizul/CUET/about.html";
console.log(path.normalize(website));
console.log(path.dirname(website));
console.log(path.basename(website));
console.log(path.extname(website));
setInterval(function()
{
    console.log("Chicken");
},2000);
console.log(__dirname);
console.log(__filename);
```

Output:

Corn is good,corn is life

Desktop\Azizul\CUET\index.html

Desktop/Azizul/CUET

index.html

.html

C:\Users\Azizul56\IdeaProjects\azizul

C:\Users\Azizul56\IdeaProjects\azizul\app.js

Chicken

Chicken

Chicken

6. Creating a basic Server:

```
var http=require('http');
function onRequest(request,response)
{
  console.log("A user mode is requested"+request.url);
  response.writeHead(200,{"Context-Type":"text/plain"});
  response.write("Here is some data");
  response.end();
}
http.createServer(onRequest).listen(8888);
console.log("Server is running now");
```

Output:

Server is running now

A user mode is requested/

A user mode is requested/favicon.ico



7. Simple web file server:

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Azizul's Website</title>
</head>
<body>
  The website is awesome!
</body>
</html>
```

Server.js:

```
var http=require('http');
var fs=require('fs');
```

```
function send404Response(response)
{
    response.writeHead(404, {"Context-Type": "text/plain"});
    response.write("Error 404:page not found");
    response.end();
}
function onRequest(request, response)
{
    if(request.method == 'GET' && request.url=='/')
    {
        response.writeHead(200, {"Context-Type": "text/html"});
        fs.createReadStream("./index.html").pipe(response);
    }
    else {
        send404Response(response);
    }
}
http.createServer(onRequest).listen(8888);
console.log("Server is running now");
```

Output:

"C:\Program Files\nodejs\node.exe" C:\Users\Azizul56\IdeaProjects\azizul\server.js

Server is running now



8.Connect:

Server.js:

```
var connect=require('connect');
var http=require('http');
var app=connect();
function doFirst(request,response,next)
{
    console.log("Azizul");
    next();
}
function doSecond(request,response,next)
```

```
{
  console.log("Nafis");
  next();
}
function profile(request,response)
{
  console.log("Profile is requested!")
}
function forum(request,response)
{
  console.log("Forum is requested!")
}
app.use(doFirst);
app.use(doSecond);
app.use('/profile',profile);
app.use('/forum',forum);

http.createServer(app).listen(8888);
console.log("Server is running perfectly.");
```

Output:

Server is running perfectly.

Azizul

Nafis

Azizul

Nafis

Profile is requested!

Azizul

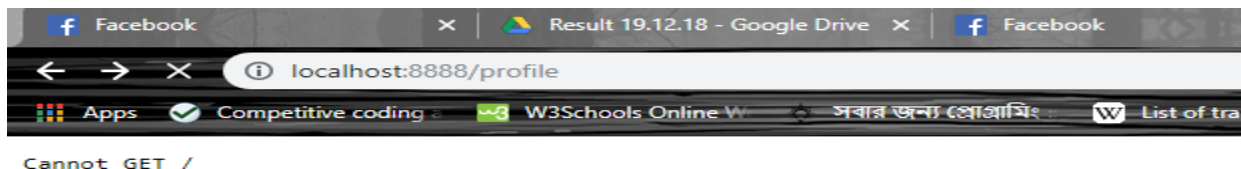
Nafis

Forum is requested!

Azizul

Nafis

Forum is requested!



9.Model View Route Structure:

www.js:

```
* Module dependencies.
*/
var app = require('../app');
var debug = require('debug')('ahc:server');
var http = require('http');

/**
 * Get port from environment and store in Express.
 */
var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

/**
 * Create HTTP server.
 */
var server = http.createServer(app);

/**
 * Listen on provided port, on all network interfaces.
 */
server.listen(port);
server.on('error', onError);
server.on('listening', onListening);

/**
 * Normalize a port into a number, string, or false.
 */
function normalizePort(val) {
  var port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
```

```

    // port number
    return port;
}

return false;
}

/**
 * Event listener for HTTP server "error" event.
 */

function onError(error) {
  if (error.syscall !== 'listen') {
    throw error;
  }

  var bind = typeof port === 'string'
    ? 'Pipe ' + port
    : 'Port ' + port;

  // handle specific listen errors with friendly messages
  switch (error.code) {
    case 'EACCES':
      console.error(bind + ' requires elevated privileges');
      process.exit(1);
      break;
    case 'EADDRINUSE':
      console.error(bind + ' is already in use');
      process.exit(1);
      break;
    default:
      throw error;
  }
}

/**
 * Event listener for HTTP server "listening" event.
 */

function onListening() {
  var addr = server.address();
  var bind = typeof addr === 'string'
    ? 'pipe ' + addr
    : 'port ' + addr.port;
  debug('Listening on ' + bind);
}

```

Index.js:

```

var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

module.exports = router;

```


users.js:

```
var express = require('express');
var router = express.Router();

/* GET users listing. */
router.get('/', function(req, res, next) {
  res.send('respond with a resource');
});

module.exports = router;
```

index.ejs:

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <p>Welcome to <%= title %></p>
  </body>
</html>
```

errors.ejs:

```
<h1><%= message %></h1>
<h2><%= error.status %></h2>
<pre><%= error.stack %></pre>
```

10. Handling user Request:

www.js- Basic configuration of the server.

```
var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);
```

app.js:

```
app.use('/', indexRouter);
app.use('/users', usersRouter);
```

```
var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
```

index.js:

```
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});
```

users.js:

```
router.get('/', function(req, res, next) {
  res.send('respond with a resource');
});
```

index.ejs:

```
</head>
<body>
  <h1><%= title %></h1>
  <p>Welcome to <%= title %></p>
</body>
```

11.Connect:

Server.js:

```
var connect=require('connect');
var http=require('http');
var app=connect();
function doFirst(request,response,next)
{
  console.log("Azizul");
  next();
}
function doSecond(request,response,next)
{
  console.log("Nafis");
  next();
}
function profile(request,response)
{
  console.log("Profile is requested!")
}
function forum(request,response)
{
  console.log("Forum is requested!")
}
app.use(doFirst);
app.use(doSecond);
app.use('/profile',profile);
app.use('/forum',forum);

http.createServer(app).listen(8888);
console.log("Server is running perfectly.");
```

Output:

Server is running perfectly.

Azizul

Nafis

Azizul

Nafis

Profile is requested!

Azizul

Nafis

Azizul

Nafis

Forum is requested!

12.Intro to www-app-view-route:

www-configuration basic server and assign port to 3000.

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

var app = require('../app');
var debug = require('debug')('ahc:server');
var http = require('http');

/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

/**
 * Create HTTP server.
 */

var server = http.createServer(app);

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);

/**
 * Normalize a port into a number, string, or false.
 */

function normalizePort(val) {
  var port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }
}
```

```

    if (port >= 0) {
      // port number
      return port;
    }

    return false;
  }

  /**
   * Event listener for HTTP server "error" event.
   */

  function onError(error) {
    if (error.syscall !== 'listen') {
      throw error;
    }

    var bind = typeof port === 'string'
      ? 'Pipe ' + port
      : 'Port ' + port;

    // handle specific listen errors with friendly messages
    switch (error.code) {
      case 'EACCES':
        console.error(bind + ' requires elevated privileges');
        process.exit(1);
        break;
      case 'EADDRINUSE':
        console.error(bind + ' is already in use');
        process.exit(1);
        break;
      default:
        throw error;
    }
  }

  /**
   * Event listener for HTTP server "listening" event.
   */

  function onListening() {
    var addr = server.address();
    var bind = typeof addr === 'string'
      ? 'pipe ' + addr
      : 'port ' + addr.port;
    debug('Listening on ' + bind);
  }

```

App.js:

Connects with the route.

```

var indexRouter = require('./routes/index');
var aboutRouter = require('./routes/about');

```

```

app.use('/', indexRouter);
app.use('/about', aboutRouter);

```

index.js(route):

connects with the view and links view and app.js.

```
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express',
    name: 'My favourite band',
    videodata: vd });
});
```

header.ejs:

```
<a href="/">Home</a> |
<a href="/about">About</a> |
<a href="#">Link 3</a>
```

Index.ejs(view):

The main html view.

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <% include Template/header.ejs%>
    <h1><%= videodata.categoryName %></h1>
    <ul><%= videodata.categories.forEach(function(item)
      {<%=
<li><%= item.categoryName %></li>

    <%=});%></ul>

    <ul></ul>
  </body>
</html>
```

Output:

```
Home | About | Link 3
Appetizers & Sides
  • Apps and Side Dishes (Laura)
  • Side Dishes
  • Side Dishes (bt)
  • Side Dishes for Barbecue
  • Soups and Salads (Laura)
```

13. Adding an extra page:

app.js:

```
var aboutRouter = require('./routes/about');
```

```
app.use('/about', aboutRouter);
```

about.js(route):

```
var express = require('express');
var router = express.Router();

/* GET about page. */
router.get('/', function(req, res, next) {
  res.render('about', { title: 'About myself',
    name: 'My personal information' });
});

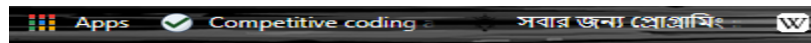
module.exports = router;
```

about.ejs(view):

```
<!DOCTYPE html>
<html>
<head>
  <title><%= title %></title>
  <link rel='stylesheet' href='/stylesheets/style.css' />
</head>
<body>
<% include Template/header.ejs%>
<h1><%= title %></h1>
<p>Hey! I am Azizul Hakim Chowdhury. Have a nice day!</p>

</body>
</html>
```

Output:



14. json file management:

videodata.json:

```
{
  "categoryName": "Appetizers & Sides",
  "categories": [
    {
      "categoryID": "294",
      "parentID": "304",
      "subjectID": "7",
      "categoryName": "Apps and Side Dishes (Laura)",
      "categoryDescription": "Learn to make amazing appetizers and side dishes with Laura in the Kitchen.",
      "videosCount": "101",
      "forumCategoryID": "163"
    },
    {
```

```

        "categoryID": "285",
        "parentID": "304",
        "subjectID": "7",
        "categoryName": "Side Dishes",
        "categoryDescription": "Side dish recipes for salads, vegetables, sauces with
Hilah cooking.",
        "videosCount": "38",
        "forumCategoryID": "163"
    },
    {
        "categoryID": "337",
        "parentID": "304",
        "subjectID": "7",
        "categoryName": "Side Dishes (bt)",
        "categoryDescription": "Side dish recipes with Byron Talbott.",
        "videosCount": "5",
        "forumCategoryID": "163"
    },
    {
        "categoryID": "301",
        "parentID": "304",
        "subjectID": "7",
        "categoryName": "Side Dishes for Barbecue",
        "categoryDescription": "Barbecue side dish recipes done on the grill by the BBQ
Pit Boys!",
        "videosCount": "43",
        "forumCategoryID": "163"
    },
    {
        "categoryID": "297",
        "parentID": "304",
        "subjectID": "7",
        "categoryName": "Soups and Salads (Laura)",
        "categoryDescription": "Looking for the perfect recipe to start your meal? Or
are you looking to eat something on the lighter side? These are sure to have you
covered!",
        "videosCount": "70",
        "forumCategoryID": "163"
    }
]
}

```

App.js:

No operation is changed.

Index.js:

```

var vd=require('../videodata.json');

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express',
    name:'My favourite band',
    videodata:vd });
});

```

index.ejs:

```

<!DOCTYPE html>
<html>

```

```

<head>
  <title><%= title %></title>
  <link rel='stylesheet' href='/stylesheets/style.css' />
</head>
<body>
<% include Template/header.ejs%>
  <h1><%= videodata.categoryName %></h1>
  <ul><% videodata.categories.forEach(function(item)
    {%>
<li><%= item.categoryName %></li>

    <%});%></ul>

  <ul></ul>
</body>
</html>

```

Output:

[Home](#) | [About](#) | [Link 3](#)

Appetizers & Sides

- Apps and Side Dishes (Laura)
- Side Dishes
- Side Dishes (bt)
- Side Dishes for Barbecue
- Soups and Salads (Laura)

15.Declaring a global variable/object:

```
app.locals.points="8300";
```