**Instructor Notes:**

Add instructor notes here.

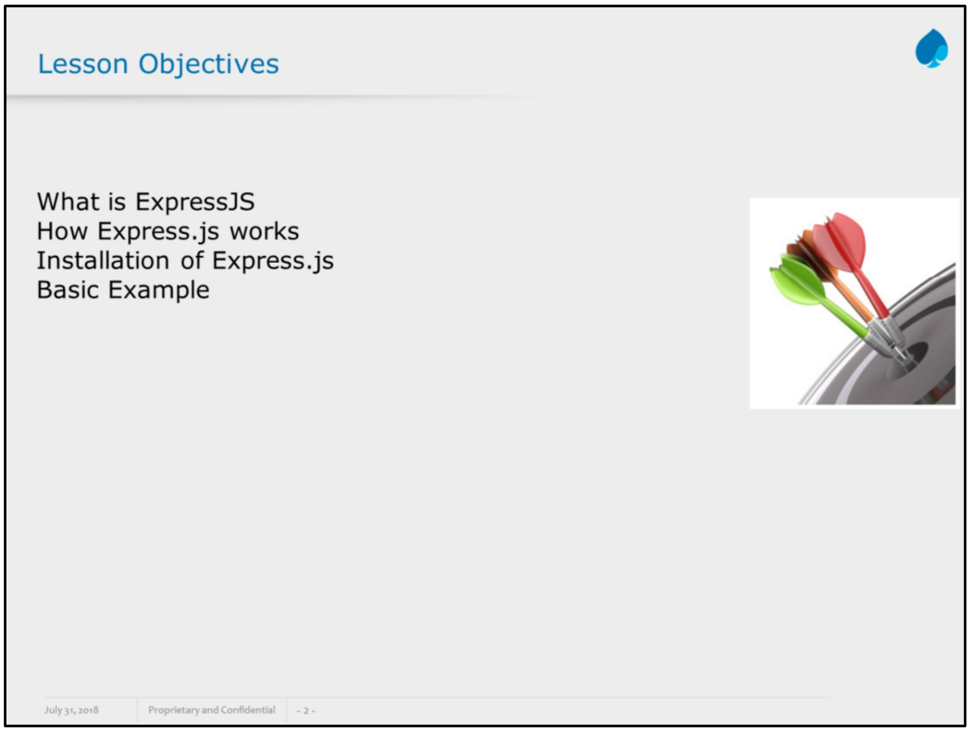
# Express JS

Lesson 01 :
**Introduction of ExpressJs**

Capgemini

**Instructor Notes:**

Add instructor notes here.



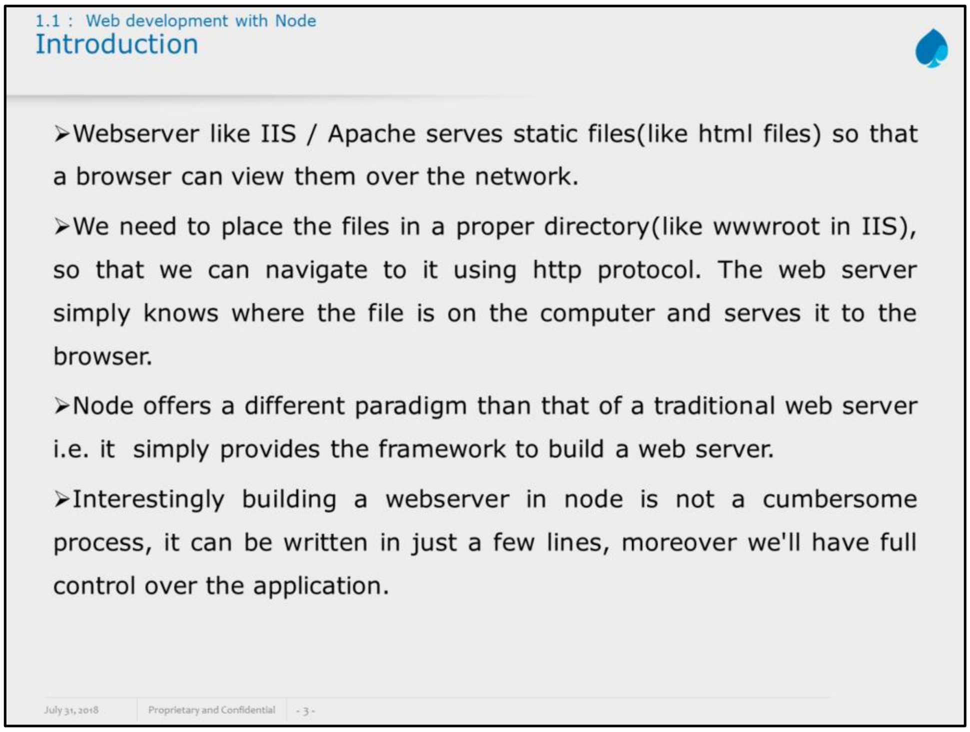
## Lesson Objectives

What is ExpressJS
How Express.js works
Installation of Express.js
Basic Example

July 31, 2018 | Proprietary and Confidential | - 2 -

**Instructor Notes:**

Add instructor notes here.



1.1 : Web development with Node

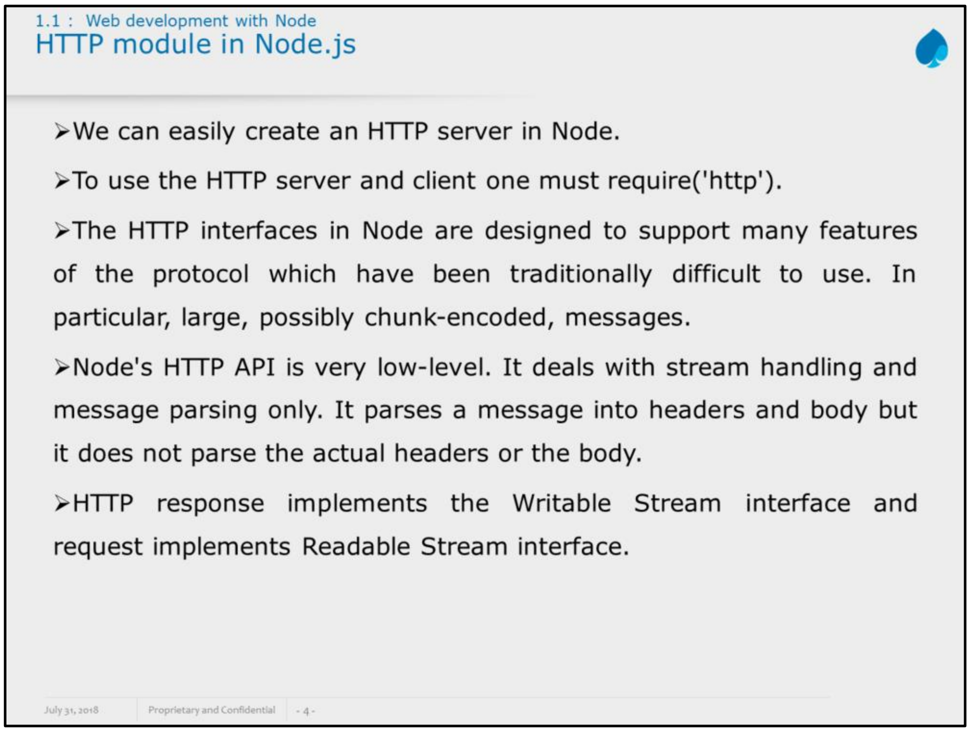
## Introduction

- ➢Webserver like IIS / Apache serves static files(like html files) so that a browser can view them over the network.
- ➢We need to place the files in a proper directory(like wwwroot in IIS), so that we can navigate to it using http protocol. The web server simply knows where the file is on the computer and serves it to the browser.
- ➢Node offers a different paradigm than that of a traditional web server i.e. it simply provides the framework to build a web server.
- ➢Interestingly building a webserver in node is not a cumbersome process, it can be written in just a few lines, moreover we'll have full control over the application.

July 31, 2018 | Proprietary and Confidential | - 3 -

**Instructor Notes:**

Add instructor notes here.



1.1 : Web development with Node

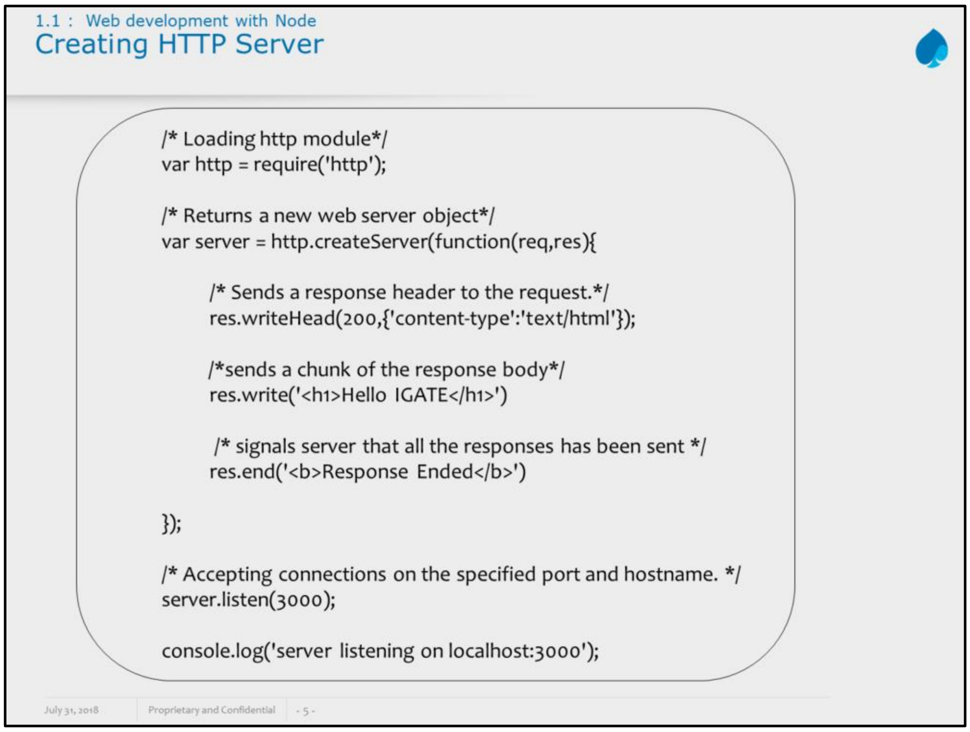
## HTTP module in Node.js

- We can easily create an HTTP server in Node.
- To use the HTTP server and client one must require('http').
- The HTTP interfaces in Node are designed to support many features of the protocol which have been traditionally difficult to use. In particular, large, possibly chunk-encoded, messages.
- Node's HTTP API is very low-level. It deals with stream handling and message parsing only. It parses a message into headers and body but it does not parse the actual headers or the body.
- HTTP response implements the Writable Stream interface and request implements Readable Stream interface.

July 31, 2018 Proprietary and Confidential - 4 -

**Instructor Notes:**

Add instructor notes here.



1.1 : Web development with Node

## Creating HTTP Server

```
/* Loading http module*/
var http = require('http');

/* Returns a new web server object*/
var server = http.createServer(function(req,res){

        /* Sends a response header to the request.*/
        res.writeHead(200,{'content-type':'text/html'});

        /*sends a chunk of the response body*/
        res.write('<h1>Hello IGATE</h1>')

         /* signals server that all the responses has been sent */
        res.end('<b>Response Ended</b>')

});

/* Accepting connections on the specified port and hostname. */
server.listen(3000);

console.log('server listening on localhost:3000');
```

July 31, 2018 Proprietary and Confidential - 5 -

**HTTP status codes**

**1xx**
The 1xx series of status codes is classified as Informational, and is used for conveying provisional response from the server.
The available codes in this series are: 100, 101, and 102.

**2xx**
The 2xx series of status codes is classified as Success, and is used for conveying a successful request for a resource on the server.
The available codes in this series are: 200, 201, 202, 303, 204, 205, 206, 207, 208, 250, and 226.

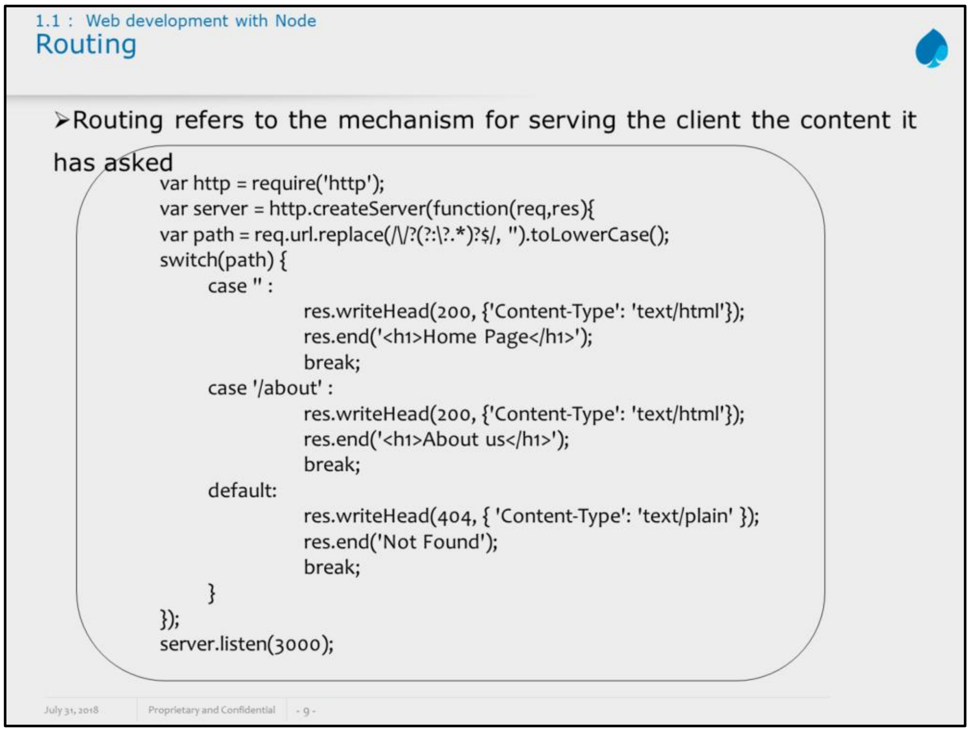
**3xx**
The 3xx series of status codes is classified as Redirection, and is used for information by the user agent about taking additional action to retrieve the requested resource. The available codes in this series are: 300, 301, 302, 303, 304, 305, 306, 307, and 308.

**4xx**
The 4xx series of status codes is classified as Client Error, and is used for informing the user agent of its erroneous requests to the server.
The available codes in this series are: 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 422, 423, 424, 425, 426, 428, 429, 431, 444, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 494, 495, 496, 497, and 499.

**5xx**
The 5xx series of status codes is classified as Server Error, and is used for informing the user agent that the server has encountered an error because of which the request was not fulfilled. The available codes in this series are: 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 551, 598, and 599.

**Instructor Notes:**

Add instructor notes here.



1.1 : Web development with Node

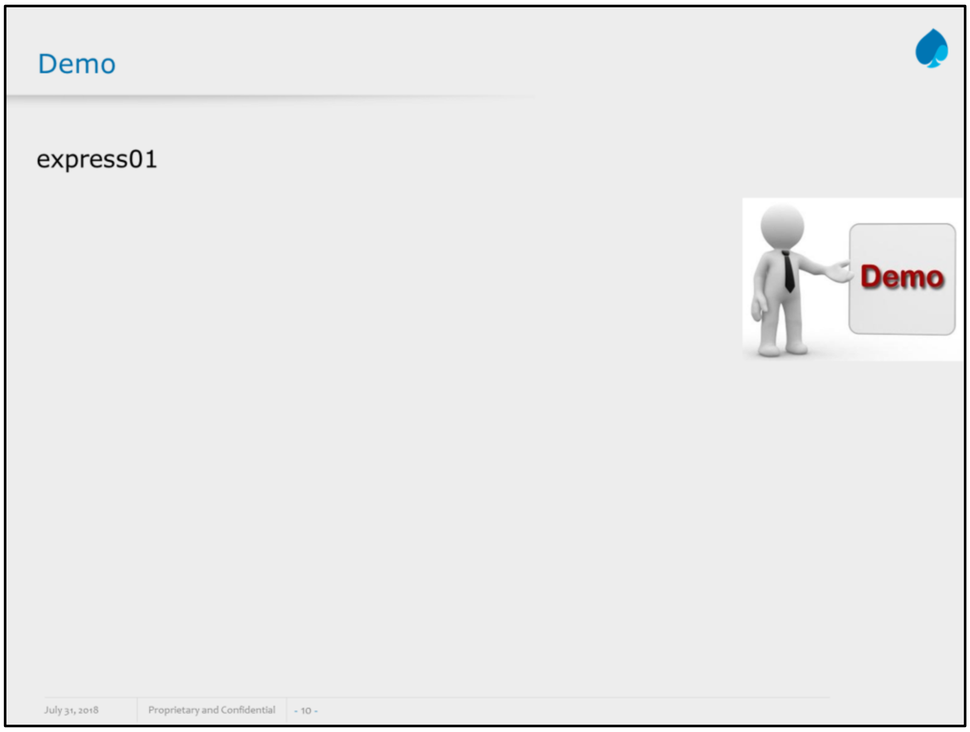
## Routing

➢Routing refers to the mechanism for serving the client the content it has asked

```
var http = require('http');
var server = http.createServer(function(req,res){
var path = req.url.replace(/\/?(?:\?.*)?$/, '').toLowerCase();
switch(path) {
        case '' :
                        res.writeHead(200, {'Content-Type': 'text/html'});
                        res.end('<h1>Home Page</h1>');
                        break;
        case '/about' :
                        res.writeHead(200, {'Content-Type': 'text/html'});
                        res.end('<h1>About us</h1>');
                        break;
        default:
                        res.writeHead(404, { 'Content-Type': 'text/plain' });
                        res.end('Not Found');
                        break;
        }
});
server.listen(3000);
```

July 31, 2018 Proprietary and Confidential - 9 -

**Instructor Notes:**



## Demo

express01