Express JS

Lesson 01 :
Introduction of ExpressJs

Capgemini

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

➢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.

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.

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');
```

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);
```

Demo

express01