|  |
| --- |
| Building Server Applications using Node.js |
| **Trainer** |
| **Magesh Kuppan**   * Profile - <http://in.linkedin.com/in/tkmagesh/> |
| **Overview** |
| This workshop teaches the participants how to design and develop server side applications using the event-driven, non-blocking model framework Node.js. This program inducts the participant in some of the advanced concepts of the JavaScript language so that the participant is well equipped to build end-to-end application using JavaScript. |
| **OBJECTIVES** |
| At the end of this training course, the participants will:   * Understand the non-blocking nature of Node.js * Writing asynchronous code with callbacks and streams * Learn how to modularize code using NPM and require() * Understand and take advantage of the built-in apis for building scalable few applications * Learn how to build web applications using Express.js * Understand how to handle authentication and authorization * Learn how to build real-time applications using WebSockets and socket.io |
| **suggested audience** |
| Web developer who wants to build data intensive scalable applications using Node.js |
| **Past occurances of this program** |
| * Amdocs, Pune * Yodlee, Bangalore |
| **DURATION** |
| 3 Days |
| **PARTICIPANT PREREQUISITES** |
| **Must Have**   * Experience in application development in any programming language * Intermediate to advanced knowledge of the JavaScript language   **Good To Have**   * Exposure to any server technology (J2EE, .NET, Ruby etc) |
| **infrastructure requirements** |
| * Node.js * Chrome * Internet Connection * Docker |
| **Case study** |
| * A Task Manager application * Real-time chat application using Socket.io * Building RESTFul services with Hapi.js |
| **Course contents** |
| * New in ES6   + Classes   + let and const   + Arrow functions   + Enhanced Object Literals   + Destructuring   + Spread Operator   + Generators and Iterators   + Symbols   + Comprehensions   + Async Programming using Promises   + Async Programming using Observables * Introduction to Node.js   + Installing Node.js   + Node’s Event Loop   + Alternatives to Node.js   + Writing asynchronous code * Modularizing code   + Understanding built-in modules   + Techniques for modularizing JavaScirpt code   + Using require() to modularize application code   + Using npm for third-party modules   + Handling Exceptions * Events and Streams   + Understanding Events   + EventEmitter class   + Understanding Streams   + Reading and writing streams   + Using pipe() * Accessing Local Resources   + Process Object   + Manipulating File System   + Understanding Buffers * Node.js and the web   + Handling web requests   + Building a web server   + Understanding the need for web sockets   + Realtime interaction using socket.io * Building web applications using Express.js   + Installing Express.js   + Routing   + Parameters and queries in routing   + Using blocks for layout   + Displaying data   + Working with forms   + Serving files   + Working with cookies and sessions   + Authentication and authorization   + Error Handling   + Persisting data in the database * Restful Server Applications using Hapi.js   + Creating Server   + Configuring Routes   + Handling Requests & Responses   + Serving Static Resources   + Using Plugins * Building Microservices   + Microservices Tenants   + Building Microservices   + Hosting Microservies in containers (Docker)   + API management using Swagger * Unit Testing   + Writing Tests Using Mocha.js   + Assertion helpers in Chai.js   + Mocks using sinon.js   + Setting up a test harness for Node applications * Scaling Node applications   + The Child process model   + exec, spawn, and fork functions   + Using the Cluster module * Node.js Eco System   + Node Packages   + Packages of interest   + Deciding Factors * Debugging Node Applications   + Node Inspector |