

Introduction to MEAN Stack Development

Duration: 5 Days | **Price:** \$2595 *California residents and government employees call for pricing.

Course Description: MEAN (MongoDB, Express, AngularJS, and Node.JS) represents a new breed of software platforms that creatively leverage a stack of production-ready technologies centered on JavaScript. This course introduces students to the essential concepts, design principles, idioms, and techniques behind the efficient use of the MEAN JavaScript Technology Stack. Upon completion of this training course, students will have solid understanding of MEAN'S' value proposition; they will also gain practical experience working with all the components of the MEAN stack. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Course Prerequisites: Knowledge of JavaScript equivalent to attending the [JavaScript Programming](#) course

Introduction to AngularJS

- Overview of AngularJS
- Scope and Goal of AngularJS
- Building Blocks of an AngularJS Application
- Using the Model View Controller (MVC) Pattern
 - The View
 - The Controller
- Data Binding
- Basics of Dependency Injection (DI)
- Other Client Side MVC Frameworks

AngularJS Controllers

- Controller Main Responsibilities
 - Defining a Controller
 - Using the Controller
 - Controller Constructor Function
- About Constructor and Factory Functions
- Using Scope Hierarchy
 - Modifying Objects in Parent Scope
 - Modified Parent Scope in DOM
- Handling Events
- Storing Model in Instance Property

Basic View Directives

- Introduction to AngularJS Directives
- Controlling Element Visibility
- Adding and Removing Elements
- Dynamically Changing Style Class

AngularJS Module

- What is a Module?
 - Benefits of Having Modules
 - Life Cycle of a Module
- The Configuration Phase
- The Run Phase
- Module Wide Data Using Value
- Module Wide Data Using Constant
- Module Dependency
- Using Multiple Modules in a Page

AngularJS Expressions

- Expressions
- Operations Supported in Expressions
- AngularJS Expressions vs JavaScript Expressions
- Considerations for Using src and href Attributes in Angular
- `ng-src` and `ng-href` Directives

Advanced View Directives

- The `ng-repeat` Directive
- Dynamically Adding Items
- Special Properties
- Using the `$index` Property

The `ng-class` Directive

- Setting an Image Source
- Setting Hyperlink Dynamically
- Preventing Initial Flash

Scope and Iteration

- Event Handling in Iterated Elements
- The `ng-switch` Directive
- Inserting an External Template Using `ng-include`

Working with Forms

- Forms and AngularJS
- Scope and Data Binding
- Role of a Form
- Using Input Text Box
- Using Radio Buttons
- Using Checkbox
- Using Select
- Disabling an Input
- Reacting to Model Changes in a Declarative Way
- Using the `ng-change` Directive

Formatting Data with Filters in AngularJS

- Overview of AngularJS Filters
- The Filter Syntax
- Using Filters in JavaScript
- The `date` Filter
- The `format` Parameter
- Using `limitTo` Filter
- The `filter` Filter
- Filter Performance Considerations

Communicating with Web Servers

- The Promise Interface
- Using `$http` Service
- Shortcut Functions
- Using `$http.get()`
- Using `$http.post()`
- Combining `$http` POST Request Data with URL Parameters
- Direct `$http` Function Invocation
- Setting Up HTTP Request Headers
- Caching Responses
- Setting the Request Timeout
- The `then()` Function of the Promise Object
- The Response Object
- Working with JSON Responses
- Using `success()` and `error()` for Callbacks
- Making Parallel Web Service Calls

AngularJS Services

- Overview of Services
- Defining a Service
- The `factory()` Method Approach
- The `service()` Method Approach
- Using a Service
- The `provider()` Method Approach
- Configuring a Service Using its Provider

Introduction to MongoDB

- MongoDB Features
- Positioning of MongoDB
- MongoDB Applications
- MongoDB Data Model
- MongoDB Limitations
- MongoDB Use Cases
- MongoDB Query Language (QL)
- The CRUD Operations
- The `find` Method
- The `findOne` Method
- Data Inserts
- MongoDB vs Apache CouchDB

MongoDB Data Model

- The BSON Data Format
- Database Terminology
- MongoDB Data Model
- The `_id` Primary Key Field Considerations
- Indexes
- Data Modeling in RDBMS
- Data Modeling in MongoDB
- Normalize vs. Denormalize
- MongoDB Limitations
- Data Lifecycle Management: TTL
- Data Lifecycle Management: Capped Collections

MongoDB Architecture

- MongoDB Nexus Architecture
- Pluggable Storage Engines
- Data Sharding
- Sharding in MongoDB
- Data Replication
- Sharded Cluster Diagram
- GridFS
- Getting a File from GridFS
- MongoDB Security
- Authentication
- Data and Network Encryption

Working with Data in MongoDB

- Reading Data in MongoDB
- The Query Interface
- Query and Projection Operators
- MongoDB Query to SQL Select Comparison
- Cursor Expiration
- Writing Data in MongoDB
- The Insert Operation
- The Update Operation
- Limiting Return Data
- Data Sorting
- Aggregating Data
- Aggregation Stages
- Accumulators
- Aggregation Pipe-line
- Map-Reduce

Introduction to Node.js

- Overview of Node.js
- Application of Node.js
- Installing Node.js and NPM
- Benefits of Node.js
- Traditional Server-Side I/O Model
- Event-Driven, Non-Blocking I/O
- Concurrency
- Using Node Package Manager (NPM)
- Express

The File System Module

- Basic File Manipulation
- Getting File/Directory Meta Data
- Read an Entire File

MongoDB Administration

- The Admin Shell
- Admin Commands
- Starting MongoDB
- The mongod Web UI
- Running MongoDB .JS Scripts
- Shell Helper to JavaScript Equivalent Map
- Connecting to MongoDB
- Authentication Options
- Import/Export Operations
- MongoDB Extended JSON and Strict Mode
- Strict Mode Encoding
- The Robomongo Tool
- Robomongo Features

Indexes in MongoDB

- Indexes in MongoDB
- Creating an Index
- Listing Existing Indexes
- The Sort Order
- Using the Explain() Method
- Main Index Types
- The `_id` Field
- Single and Compound Indexes
- Multikey Indexes
- Other Index Types
- Index Properties
- Creating Indexes with Index Properties

Module and Dependency Management

- Introduction to Modules
- Directory Based Modules
- Making a Module Executable
- Core Modules
- Loading Module from `node_modules` Folders
- Dependency Management Using NPM
- Installing a Package
- About Global Installation
- Setting Up Dependency
- Package Version Numbering Syntax
- Updating Packages
- Uninstalling Packages
- Alternate Dependency Management

Basic Web Application Development

- Introduction to the HTTP Module
- The Request Handler Callback Function
- The Server Object

- The Buffer Class
- Writing to a File
- Reading in Chunks
- Writing in Chunks
- The `open()` Method
- Stream API
- The Readable Interface
- The Writable Interface
- The Request Object
- The Response Object
- Parsing Request Body
- Serving Static Files
- The HTTP Client API
- Making POST/PUT Requests

Debugging and Unit Testing

- Problem Determination Options
- Using `console.log`
- Using the `debug` Logging Package
- Configure Logging
- The Node Inspector Debugger
- Basic Usage of the Debugger
- Unit Testing Node.js Applications
- Getting Setup
- Writing a Test Script
- Running Unit Test
- Testing Asynchronous Code
- Using the Chai Assert API
- The Chai Expect API

Express Middleware

- Introduction to Express Middleware
- Writing a Middleware Function
- Binding to a Path
- Order of Execution
- Raising Error
- Handling Error
- Serving Static Files
- Handling POST Request Body
- Enable Response Compression

Jade Template Engine

- Introduction to Jade
- Using Jade
- Passing Data to a Template
- Basic HTML Tag Rendering
- Rendering Values
- Conditional Rendering
- Rendering a List
- Layout Templates
- Creating a Layout Template
- Creating a Content Template

The Route Service

- Introduction to the Route Service
- Downloading the Route Service Code
- Using the Route Service
- Setting up the Route Table

Introduction to Express

- Defining Routing Rules
- Route Path
- The Response Object
- Supplying URL Parameters
- Ordering of Routes
- Defining Catch All Route

Accessing MongoDB from Node.js

- The Connection URL
- Obtaining a Collection
- Inserting Documents
- Updating a Document
- Querying for Documents
- Deleting a Document
- Connection Pooling

Introduction to Single Page Applications

- What is a Single Page Application (SPA)?
- Why Create SPA?
- Challenges to SPA
- Implementing SPA Using AngularJS
- Simple SPA Using Visibility Control
- Dynamic Templates Using `ng-include`
- SPA Using the `$route` Service

Clustering and Failover

- Process Management
- Managing the Process Using OS Tools
- Installing a Service in Window
- Create an Upstart Script in Ubuntu

- URL Fragment Identifier
- Showing the Views
- Navigation
- The `$location` Service
- Programmatic Navigation
- Controllers for the Views
- Passing URL Parameters
- Accessing Query Parameters
- Configuring Route Parameters
- Accessing Route Parameters
- HTML5 Mode
- Using the HTML5 Mode
- Bookmarking HTML5 Mode URL
- Process Management Using `forever`
- Clustering Basics
- Child Process Failover

Hands On Technology Transfer, Inc.
1 Village Square, Suite 8
14 Fletcher Street
Chelmsford, MA 01824
1-800-413-0939 | 1-978-250-4299
www.traininghott.com

Copyright© 2016 Hands On Technology Transfer, Inc.