MEAN Stack | MongoDB, ExpressJS, Angular 7 and NodeJS   
5 Days By Vijay Shivakumar

****

ObjectivesMEAN is popularly known combination of MongoDB, Express.js, Angular7, and Node.js, which has emerged as one of the leading technology stacks for developing dynamic web

DescriptionThis is a Hands On Training workshop using the stable versions of Angular 7. NodeJS, MongoDB and ExpressJS.

Starting out with the fundamentals of the MEAN stack technologies, you will learn how to build a strong solution architecture for automated builds and distributed development. You’ll then dive deep into its relative technologies to become a full- stack MEAN developer.

AudienceDevelopers with knowledge of JavaScript, JQuery, HTML and CSS syntaxes.

participants are also expected to understand the Ajax concepts and usage of libraries in JavaScript.

Hardware Windows8 or above with at least 4GB RAM. internet to download libraries.

Software Visual Studio Code, MongoDB (fresh installation), NodeJS

Browsers Chrome latest, Firefox with firebug

**NOTE : Participants must have admin privileges to install modules as and when needed both from npm and github.**

**Participants and trainer need internet to access exercises files and resources.**

**The training will use existing libraries but may create few as needed the exercises files used in the classroom will be made available to the participants at the end of the training only, It is recommended that the participants will participate in creating their own copy of the lessons covered in the training.**

**Course Coverage**

**DAY 1**

**Introduction to the Course**

Why MEAN stack

contemporary challenges and advantages

**ES5, ES6**

Introduction and features

ES5, ES6, ES7 and ES8

Using the ES Transpilers

Block scopes and hoisting

Destruction

Array Methods

Object Creation

Functions

Default parameters and rest parameters

Spreads

Working with Modules in ES6

**NodeJS**

Understanding CommonJS Design Pattern  
Creating Modules

Exporting Modules

Working with External Modules  
Downloading from NPM  
Contributing to NPM

**Putting it all together in a project**

Planning a project

Steps to complete a project with MEAN architecture

**DAY 2**

**Event Driven Architecture**

Understanding Node Architecture

Advantages of events

Using custom events

Understanding triggers

Streams

Working with streams

**Web with Node**

Creating web server with nodeJS

HTTP / HTTPS

Basic werbserver with Node modules

**ExpressJS**

Installing Express.js

Installing Express.js and Command-line Tools

Bootstrapping

Structure of an ExpressJS App

**Introducing Template Engine**

Jade Template Engine

Introducing Stylus

Bootstrap with Jade and Stylus

Public and Static Files

Other Templating Options

**Structure of Our Application**

Installing the Necessary Modules

Creating Our Endpoints

**Routing**

Parameters and queries in routing

Displaying data

Working with forms

Serving files

Working with cookies and sessions

**MongoDB**

Installing and Configuring MongoDB

Wiring Up Mongoose.js

Creating Our Models

Differences between MongoDB and a Relational Database

Collecting Data

**DAY 3**

**Angular**

Overview of Single page Apps

What is new in Angular 7

The Angular CLI   
Commands of CLI

Building blocks of Angular 7 framework

Why Angular 7 ?

Working with NPM

Setup Angular 7 project and application

Introduction to Angular Module

Root Component

Writing Re-usable components

Structure of a component

Application bootstrapping process

Annotations of @Module and @Component

Lifecycle hooks

Template interpolation

**Binding**

One way data binding concept

Property Binding

Event Binding

Two way Binding

Repeat elements with ngFor

Conditional elements with ngIf

Template variables

**Template**

Template Driven Forms

Two way binding Input Fields

Accessing the Form Object

ngSubmit and submit events

Displaying Form Validation State

Displaying Field Validation State

Displaying Validation State Using CSS Classes

Overview of Model driven forms

**DAY 4**  
 **Component Communication**

Nested Components

Using @input and @output

Component communication using Observable

**Pipes**

What are Pipes

Built-In Pipes

Chaining Pipes

Custom Pipes

**Dependency Injection**

Need for Dependency Injection(DI)

Writing services with @Injectable

Registering providers

Inject services in components

Other provider configurations

Inject service within a service

**DAY 5**

**Routing**

Routing and Navigation

Component Router Terminology

Setting up routes for an application

Set URL Links with RouterLink

Creating Routes with Route Parameters

Default routes

**Customizations**

Creating Custom Pipes

Custom Directives

Custom Modules

**Authentication and authorization**

Passport.js Authentication

Understanding Sessions

Using 3rd Party for Authentication

Error Handling

**Full Project with MEAN**