MEAN Stack Development

By Dr. Vishwanath Rao

Topics

- Detailed coverage of Angular
- Essential MongoDB
- Introduction to Node.js
- Node.js applications
- Debugging and Unit testing on Node.js
- Express middleware
- Jade Template Engine
- Elements of performance tuning of Node.js applications on multi-core processors

Audience

Developers, Technical Leads, and Software Architects.

Prerequisites

Participants should have basic knowledge of JavaScript programming and web development.

Duration

5 days

1. Introducing Angular

- What is Angular?
- Central Features of the Angular Framework
- Appropriate Use Cases
- Building Blocks of an Angular Application
- Basic Architecture of an Angular Application
- Installing and Using Angular
- Anatomy of an Angular Application
- Running the Application
- Building and Deploying the Application

- Summary

2. Introduction to TypeScript

- Programming Languages for Use with Angular
- TypeScript Syntax
- Programming Editors
- The Type System – Defining Variables
- The Type System – Defining Arrays
- Type in Functions
- Type Inference
- Defining Classes
- Class Methods
- Class Constructors
- Class Constructors – Alternate Form
- Interfaces
- Working with ES6 Modules
- Visibility Control
- var vs let
- Arrow Functions
- Arrow Function Compact Syntax
- Arrow Function and Caller Context
- Template Strings
- Generics in Class
- Generics in Function
- Generics - Restricting Types
- TypeScript Transpilation
- Summary

3. Components

- What is a Component?
- An Example Component
- Creating a Component Using Angular CLI
- The Component Class
- The @Component Decorator
- Registering a Component to Its Module
- Component Template
- Example: HelloComponent Template
- Example: The HelloComponent Class
- Using a Component
- Run the Application

- Component Hierarchy
- The Application Root Component
- The Bootstrap File
- Component Lifecycle Hooks
- Example Lifecycle Hooks
- CSS Styles
- Summary

4. Component Templates

- Templates
- Template Location
- The Mustache {{ }} Syntax
- Setting DOM Element Properties
- Setting Element Body Text
- Event Binding
- Expression Event Handler
- Prevent Default Handling
- Attribute Directives
- Apply Styles by Changing CSS Classes
- Example: ngClass
- Applying Styles Directly
- Structural Directives
- Conditionally Execute Template
- Example: ngIf
- Looping Using ngFor
- ngFor Local Variables
- Manipulating the Collection
- Example - Deleting an Item
- Item Tracking with ngFor
- Swapping Elements with ngSwitch
- Grouping Elements
- Template Reference Variable
- Summary

5. Inter Component Communication

- Communication Basics
- The Data Flow Architecture
- Preparing the Child to Receive Data
- Send Data from Parent
- More About Setting Properties

- Firing Event from a Component
- @Output() Example - Child Component
- @Output() Example - Parent Component
- Full Two Way Binding
- Setting up Two Way Data Binding in Parent
- Summary

6. Template Driven Forms

- Template Driven Forms
- Importing Forms Module
- Basic Approach
- Setting Up a Form
- Getting User Input
- Omitting ngForm Attribute
- Initialize the Form
- Two Way Data Binding
- Form Validation
- Angular Validators
- Displaying Validation State Using Classes
- Additional Input Types
- Checkboxes
- Select (Drop Down) Fields
- Rendering Options for Select (Drop Down)
- Date fields
- Radio Buttons
- Summary

7. Reactive Forms

- Reactive Forms Overview
- The Building Blocks
- Import ReactiveFormsModule
- Construct a Form
- Design the Template
- Getting Input Values
- Initializing the Input Fields
- Setting Form Values
- Subscribing to Input Changes
- Validation
- Built-In Validators
- Showing Validation Error

- Custom Validator
- Using a Custom Validator
- Supplying Configuration to Custom Validator
- Sub FormGroups - Component Class
- Sub FormGroups - HTML Template
- Why Use Sub FormGroups
- Summary

8. Introduction to MongoDB

- MongoDB
- MongoDB Features
- MongoDB's Logo
- 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
- A MongoDB QL Example
- Data Inserts
- MongoDB vs Apache CouchDB
- Summary

9. MongoDB Data Model

- The BSON Data Format
- Database Terminology
- MongoDB Data Model
- The _id Primary Key Filed Considerations
- Indexes
- (Traditional) Data Modeling in RDBMS
- Data Modeling in MongoDB
- An Example of a Data Model in MongoDB
- MongoDB Data Modeling

- A Sample JSON Document Matching the Schema
- To Normalize or Denormalize?
- MongoDB Limitations
- Data Lifecycle Management
- Data Lifecycle Management: TTL
- Data Lifecycle Management: Capped Collections
- Summary

10. MongoDB Architecture

- MongoDB Nexus Architecture
- Blending the Best of Both Worlds
- What Makes MongoDB Fast?
- Pluggable Storage Engines
- Data Sharding
- Sharding in MongoDB
- Data Replication
- A Sample Sharded Cluster Diagram
- GridFS
- Getting a File from GridFS
- MongoDB Security
- Authentication
- Data and Network Encryption
- Summary

11. MongoDB Administration

- The Admin Shell
- Getting Help
- Admin Commands
- Starting MongoDB
- The mongod Web UI
- Running MongoDB .JS Scripts
- Shell Helper to JavaScript Equivalent Map
- A MongoDB Script Example
- Connecting to MongoDB
- User Authentication
- Authentication Options
- Import / Export Operations
- MongoDB Extended JSON and Strict Mode
- Examples of Strict Mode Encoding
- Examples of Data Import / Export

- The Robomongo Tool
- Robomongo Features
- Summary

12. Working with Data in MongoDB

- Reading Data in MongoDB
- The Query Interface
- Query Syntax is Driver-Specific
- Projections
- Query and Projection Operators
- MongoDB Query to SQL Select Comparison
- Cursors
- Cursor Expiration
- Writing Data in MongoDB
- An Insert Operation Example
- The Update Operation
- An Update Operation Example
- A Remove Operation Example
- Limiting Return Data
- Data Sorting
- Aggregating Data
- Aggregation Stages
- Accumulators
- An Example of an Aggregation Pipe-line
- Map-Reduce
- Summary

13. 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
- Examples of Creating Indexes with Index Properties

- Summary

14. Introduction to Node.js

- What is Node.js?
- Application of Node.js
- Installing Node.js and NPM
- "Hello, Node World!"
- How it Works
- Built on JavaScript – benefits
- Traditional Server-Side I/O Model
- Disadvantages of the traditional Approach
- Event-Driven, Non-Blocking I/O
- Concurrency
- Using Node Package Manager (NPM)
- Express
- Summary

15. Module and Dependency Management

- Nature of a Node.js Project
- Introduction to Modules
- A Simple Module
- Using the Module
- Directory Based Modules
- Example Directory Based Module
- Using the Module
- 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
- Summary

16. The File System Module

- Introduction
- Basic File Manipulation
- Getting File/Directory Meta Data
- Read an Entire File
- The Buffer Class
- Writing to a File
- Reading in Chunks
- Writing in Chunks
- The open() Method
- Stream API
- The Readable Interface
- Example Reading Data in Chunks
- The Writable Interface
- Summary

17. Basic Web Application Development

- Introduction to the HTTP Module
- The Request Handler Callback Function
- The Server Object
- Example Use of Server Object
- The Request Object
- The Response Object
- Parsing Request Body
- Serving Static Files
- The HTTP Client API
- Making POST/PUT/etc. Requests
- Where To go from Here?
- Summary

18. 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

- Connection Pooling
- Summary

22. Jade Template Engine

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

23. Introduction to Single Page Application

- What is a Single Page Application (SPA)?
- How is it Any Different?
- Why Create SPA?
- It's Not All or Nothing
- Challenges to SPA
- Implementing SPA Using Angular
- Simple SPA Using Visibility Control
- Dynamic Templates Using ng-include
- Example of Dynamic Template
- SPA Using the $route Service
- Summary

24. The Route Service

- Introduction to the Route Service
- Downloading the Route Service Code
- Using the Route Service
- Setting up the Route Table
- URL Fragment Identifier
- Showing the Views
- Navigation