Building Rich Internet Applications using Angular.js, HTML5, jQuery, Bootstrap, Karma & Jasmine

TRAINER

Magesh Kuppan

- Profile - http://in.linkedin.com/in/tkmagesh/
- Videos - http://bit.ly/angular-videos

OVERVIEW

This workshop aims at equipping the participants with the necessary knowledge and skills required to build rich internet applications using cutting edge technologies, frameworks and tools.

PAST OCCURANCES OF THIS PROGRAM

- Amdocs, Pune
- Ellucian, Bangalore
- Motorola, Bangalore
- Cisco, Bangalore & Pune

OBJECTIVES

At the end of this training course, the participants will:

- Be able to understand the exploit the functional and object-oriented programming techniques in JavaScript
- Understand and resolve scope issues in JavaScript
- Techniques for modularizing JavaScript code
- Be aware of the quirks in the JS language
- Be able build RIA using Angular.js
- Using Bootstrap (AngularUI) components
- TDD using Jasmine
- Using Karma Test Runner for testing against other browsers

CASE STUDIES

Following are the case studies that will be developed by the participants during the workshop:

- An “Expense Management” application using Angular.js

SUGGESTED AUDIENCE

Web developer who wants to build best-of-breed web applications with the simplicity and elegance of JavaScript.

DURATION

5 Days

PARTICIPANT PREREQUISITES

INFRASTRUCTURE REQUIREMENTS

- A good text editor (Textpad++/Sublime Text) or WebStorm
- Chrome
- Internet Connection (Mandatory)
- Node.js

DAY WISE SYLLABUS

HTML5 and CSS3
Duration : 1 Days

- Overview of HTML5
    - New in HTML5
    - New DOCTYPE and Character Set
    - New and Depreciated Elements
    - Semantic Markup
    - Simplifying Selection using the Selectors API
    - JavaScript Logging and Debugging
    - Window.JSON

- Using HTML5 Today
    - Currently available features
    - Browser Support
    - Detective native availability of features
    - Working with emulation

- Understanding HTML5 Markup
    - HTML5 Page Structure
    - HTML5 DOCTYPE
    - HTML5 markup structural elements
    - Semantic elements

- HTML5 Forms
    - HTML Forms VsXForms
    - Functional Forms
    - New Form Attributes and Functions
    - Placeholder Attribute
    - Autocomplete Attribute
    - Autofocus Attribute
    - List Attribute and datalist Element
    - Min and Max attributes
    - ValueAsNumber attribute
    - Required Attribute
    - Checking Forms with Validation
    - Validation Feedback

- HTML5 Canvas & SVG
    - Overview of Canvas Vs. SVG
    - Canvas coordinates
    - Context
    - Pixel Data
    - Working with Canvas APIs
    - Drawing Operations
    - Canvas Transforms
    - Understanding and Working with SVG

- Server Sent Events
    - Server Sent Events Support
    - Server Support
    - SSE Emulation
    - Broadcasting Information
    - Using the EventSource API

- WebSockets
    - Realtime and HTTP
    - Understanding HTML5 Web Sockets
    - Using HTML5 WebSockets API
    - WebSocket Events
    - Sending and Receiving Messages using WebSockets
    - Overview of Pusher

- Web Workers
    - Checking for Browser Support
    - Creating HTML5 Web Workers
    - Loading and Executing Additional JavaScript
    - Communicating with HTML5 Web Workers
    - Handling Errors
    - Stopping Web Workers
    - Using Timers

- Web Storage
    - Overview of HTML5 Web Storage
    - Browser Support for HTML4 Web Storage
    - Using the HTML5 Web Storage
    - Checking for Browser Support
    - Setting and Retrieving Values
    - Plugging Data Leaks

- HTML5 Offline Web Applications
    - Overview of HTML5 Offline Web Applications
    - Browser Support for HTML5 Offline Web Applications
    - Using HTML5 Offline Web Applications API
    - Creating a Sample Offline Application
    - Manifest Files
    - The applicationCache API

- CSS 3.0
    - Introducing CSS3
    - CSS3 Colors, Gradients and Multiple Backgrounds
    - Rounded Corners and Border-Radius
    - CSS3 Transforms
    - Transitions
    - Animations & 2D Transformations
    - Web Fonts with @font-face
    - CSS3 Multicolumn Layouts

JavaScript In Depth

Duration : 1 Days

- Expressive JavaScript
    - The Flexibility of JavaScript
    - JavaScript as a Loosely Typed Language
    - Functions as First-Class Objects
    - Object Mutability
- Functions In Depth
    - Function Objects
    - Function Literal
    - Function Invocation Patterns
    - Augmenting Types
    - Recursion & Closures
    - Functions as Callbacks
    - Function Currying & Memorization
- Closures In Depth
    - How Closures Work
    - Private Variables, Callbacks and Timers
    - Binding Function Contexts
    - Overriding Function Behavior
- Object Orientation With Prototypes
    - Object Instantiation
    - Constructors

- Inheritance and Prototype Chain
- Extending Object
- Extending Number
- Instantiation Issues
- Writing class-like code

- JavaScript Gotchas
  - Global Variables
  - Scope
  - Semicolon Insertions
  - Type Coercion

jQuery

- Introducing jQuery
  - Unobtrusive JavaScript
  - jQuery fundamentals
  - The jQuery wrapper
  - Utility functions
  - The document ready handler
- Creating the wrapped element set
  - Selecting elements for manipulation
  - Using basic CSS selectors
  - Using child, container, and attribute selectors
  - Selecting by position
  - Using custom jQuery selectors
  - Generating new HTML
  - Managing the wrapped element set
  - Determining the size of the wrapped set
  - Obtaining elements from the wrapped set
- Manipulating page contents with jQuery
  - Manipulating element properties and attributes
  - Manipulating element properties
  - Fetching attribute values
  - Setting attribute values
  - Removing attributes
  - Changing element styling
  - Adding and removing class names
  - Getting and setting styles
  - More useful style-related commands
  - Setting element content
  - Replacing HTML or text content
  - Moving and copying elements
  - Wrapping elements

    - Removing elements
    - Cloning elements
    - Dealing with form element values
- Events
    - The jQuery Event Model
    - Binding event handlers using jQuery
    - Removing event handlers
    - Inspecting the Event instance
    - Affecting the event propagation
    - Triggering event handlers
    - Other event-related commands

Angular.js

Duration : 2 Days

- Introduction to Angular.js
    - How Angular.js is opinionated
    - Difference between Backbone.js and Angular.js
- Angular.js Building Blocks
    - Controller Component
    - Model Component
    - View Component
    - Directives
    - Filters
    - Services
    - Providers
    - Factory
    - DI in Angular.js
- Anatomy of an Angular.js Applications
    - Creating Boundaries using ng-app
    - Model View Controller
    - Templates and Data Binding
    - Repeating elements in templates
    - Using Expressions, CSS Classes and Styles
    - Using Controllers for UI responsibility separation
    - Responding to model changes
- Data Binding in Angular.js
    - Understanding Built-in Directives
    - Scope resolution
    - One way and Two way data binding
- Using Filters
    - Filters Overview

    - Understanding Filter Expressions
    - Building custom Filters
- Services & Factories
    - Services Overview
    - Modularity using Services
    - Injecting Services
    - Creating Custom Factories and Providers
- Directives
    - Directives Overview
    - Creating Directives
    - The Directive Definition Object
    - Compilation and Linking
    - Creating Components
- Communicating with Servers
    - Communicating over $http
    - Configuring the requests
    - Sending Http Headers
    - Caching Responses
    - Request and Response Transformation
    - Using RESTful Resources
    - Communication over WebSockets
- Testing
    - Testing Models using Jasmine
    - Test considerations for Directives and Filters
    - Using Angular Mocks
    - Using Protractor for end to end testing
- Modular JavaScript
    - Techniques for modularizing JavaScirpt code
    - Using Require.js for loading dependent modules
- JavaScript Development Workflow
    - Using Grunt.js
    - Overview of Yeoman
    - Clientside dependency management using Bower.js

Using Twitter Bootstrap

Duration : 0.5 Days

- Overview
- Understanding Bootstrap components
- Using Angular-ui Bootstrap Modules
- Including Bootstrap components in web pages
- Layouts overview

- Creating a tabular layout using Bootstrap
- Responsive design
- Typography overview
- Creating and formatting tables for displaying tabular data
- Styling form elements
- Formatting buttons
- Using Icons
- Navigation
- Progress bar styles
- Breadcrumb trails
- Modal Popups

BDD using Jasmine & Karma
Duration : 0.5 Days

- Introduction
    - What is BDD?
    - BDD Basics
    - The “Red-Green-Refactor” cycle
- HANDS ON PROBLEM – 1
- Test goals
    - Boundary conditions
    - Check Inverse Relationships
    - Cross Checking Results
    - Force Error Conditions
    - Performance Characteristics
- Jasmine.js
    - Understanding Jasmine.js
    - Writing tests in Jasmine.js
    - Learning built-in matchers
    - Covering before and after
- HANDS ON PROBLEM – 2
- Advanced
    - Using Spy for Mocking
    - Nesting describe blocks
    - Writing Custom Matchers
    - Asynchronous support using “runs”
    - Testing UI using Jasmine.js
    - Jasmine jQuery Helpers
- Karma.js
    - Configuring Karma.js
    - Browser Specs
    - Karma.js Test Runner