

ANGULAR JS 1

DURATION: - 5 DAYS

Day – 1:

Introduction to Angular

Understanding MVC

Angular Vs Backbone/EXT JS/Jquery & Ember JS

Anatomy of an AngularJS App :

- Preparing the Example Project

- Working with Modules

- Setting the Boundaries of an AngularJS Application

- Using Modules to Define AngularJS Components

- Defining Controllers

- Defining Directives

- Defining Filters

- Defining Services

- Using Modules to Organize Code

- Working with the Module Life Cycle

Using Binding and Template Directives :

- Why and When to Use Directives

- Preparing the Example Project

- Using the Data Binding Directives

- Performing One-Way Bindings (and Preventing Them)

- Creating Two-Way Data Bindings

- Using the Template Directives

- Generating Elements Repeatedly

- Repeating Multiple Top-Level Elements

- Working with Partial Views

- Using the ng-include Directive as an Attribute

- Conditionally Swapping Elements

- Hiding Unprocessed Inline Template Binding Expressions

Using Element and Event Directives:

- Preparing the Example Project

- Using the Element Directives

- Showing, Hiding, and Removing Elements

- Managing Classes and CSS

- Handling Events

- Creating a Custom Event Directive

- Managing Special Attributes

- Managing Boolean Attributes

- Managing Other Attributes

Working with Form:

- Preparing the Example Project

- Using Form Elements with Two-Way Data Bindings
- Implicitly Creating Model Properties
- Checking That the Data Model Object Has Been Created
- Validating Forms
- Performing Basic Form Validation
- Providing Form Validation Feedback
- Using CSS to Provide Feedback
- Using the Special Variables to Provide Feedback
- Deferring Validation Feedback
- Using the Form Directive Attributes
- Using Input Elements
- Using Text Areas
- Using Select Elements

Using Controllers and Scopes:

- Why and When to Use Controllers and Scopes
- Preparing the Example Project
- Understanding the Basics
- Creating and Applying Controllers
- Setting Up the Scope
- Modifying the Scope
- Organizing Controllers
- Using a Monolithic Controller
- Reusing a Controller
- Using Controller Inheritance
- Using Multiple Controllers
- Using Scope-less Controllers
- Explicitly Updating the Scope
- Setting Up jQuery UI
- Controlling the Button State
- Counting the Button Clicks

DAY 2

Using Filters

- Why and When to Use Filters
- Preparing the Example Project
- Downloading the Localization File
- Filtering Single Data Values
- Formatting Currency Values
- Formatting Other Numeric Values
- Formatting Dates
- Changing String Case
- Generating JSON
- Localizing Filter Output
- Filtering Collections
- Limiting the Number of Items
- Selecting Items
- Sorting Items
- Chaining Filters
- Creating Custom Filters
- Creating a Filter That Formats a Data Value
- Creating a Collection Filter
- Building on Existing Filters

Creating Custom Directives:

- Why and When to Create Custom Directives
- Preparing the Example Project
- Creating a Custom Directive
- Defining the Directive
- Implementing the Link Function
- Breaking the Data Property Dependency

- Handling Data Changes
- Working with jqLite
- Navigating the Document Object Models
- Modifying Elements
- Creating and Removing Elements
- Handling Events
- Other jqLite Methods
- Accessing AngularJS Features from jqLite
- Replacing jqLite with jQuery

Creating Complex Directives:

- Preparing the Example Project
- Defining Complex Directives
- Defining How the Directive Can Be Applied
- Using Directive Templates
- Using a Function as a Template
- Using an External Template
- Selecting an External Template with a Function
- Replacing the Element
- Managing Directive Scopes
- Creating Multiple Controllers
- Giving Each Directive Instance Its Own Scope
- Creating Isolated Scopes

Advanced Directive Features

- Preparing the Example Project
- Using Transclusion
- Using Compile Functions
- Using Controllers in Directives
- Adding Another Directive
- Creating Custom Form Elements
- Handling External Changes
- Handling Internal Changes
- Formatting Data Values
- Validating Custom Form Elements

Day – 3:

Working with Modules and Services

- Why and When to Use and Create Services and Modules
- Preparing the Example Project
- Using Modules to Structure an Application
- Maintaining a Single Module
- Creating a New Module
- Creating and Using a Service
- Using the Factory Method
- Using the Service Method
- Using the Provider Method
- Using the Built-in Modules and Services

Services for Global Objects, Errors, and Expressions

- Preparing the Example Project
- Accessing the DOM API Global Objects
- Why and When to Use the Global Object Services
- Accessing the Window Object
- Accessing the Document Object
- Using Intervals and Timeouts
- Accessing the URL
- Performing Logging
- Dealing with Exceptions
- Why and When to Use the Exception Service
- Working with Exceptions
- Working Directly with the Exception Service
- Implementing a Custom Exception Handler
- Working with Dangerous Data
- Why and When to Use the Dangerous Data Services
- Displaying Dangerous Data
- Using an Unsafe Binding
- Explicitly Trusting Data
- Working with AngularJS Expressions and Directives
- Why and When to Use the Expression and Directive Services
- Converting Expressions into Functions
- Interpolating Strings
- Compiling Content

Services for Ajax and Promises:

- Why and When to Use the Ajax Services
- Preparing the Example Project
- Making Ajax Requests
- Making the Ajax Request
- Receiving Ajax Responses
- Configuring Ajax Requests
- Setting Ajax Defaults
- Using Ajax Interceptors
- Working with Promises
- Getting and Using the Deferred Object
- Consuming the Promise
- Understanding Why Promises Are Not Regular Events
- Chaining Outcomes Together
- Grouping Promises

Services for REST:

- Why and When to Use the REST Services
- Preparing the Example Project
- Creating the RESTful Service
- Creating the AngularJS Application
- Using the \$http Service
- Listing the Product Data
- Deleting Products

- Creating Products
- Updating Products
- Testing the Ajax Implementation
- Hiding the Ajax Requests
- Installing the ngResource Module
- Using the \$resource Service
- Configuring the \$resource Service Actions
- Creating \$resource-Ready Components
- Services for Views
 - Why and When to Use the View Services
 - Preparing the Example Project
 - Understanding the Problem
 - Using URL Routing
 - Installing the ngRoute Module
 - Defining the URL Routes
 - Displaying the Selected View
 - Wiring Up the Code and Markup
 - Using Route Parameters
 - Accessing Routes and Routes Parameters
 - Configuring Routes
 - Using Controllers with Routes
 - Adding Dependencies to Routes
 - AngularUI Router
- Services for Animation and Touch
 - Preparing the Example Project
 - Animating Elements
 - Why and When to Use the Animation Service
 - Installing the ngAnimation Module
 - Defining and Applying an Animation
 - Avoiding the Perils of Parallel Animation
 - Supporting Touch Events
 - Why and When to Use Touch Events
 - Installing the ngTouch Module
 - Handling Swipe Gestures
 - Using the Replacement ng-click Directive
- Services for Provision and Injection:
 - Why and When to Use the Provision and Injection Services
 - Preparing the Example Project
 - Registering AngularJS Components
 - Managing Injection
 - Determining Function Dependencies
 - Obtaining Service Instances
 - Simplifying the Invocation Process
 - Getting the \$injector Service from the Root Element

Day – 4:

Localization

- angular-translate
 - Installation
 - Teaching Your App a New Language
 - Multi-language Support
 - Switching the Language at Run Time
 - Loading Languages
- angular-gettext
 - Installation
 - Usage
 - String Extraction
 - Translating Our Strings
 - Compiling Our New Language
 - Changing Languages

Caching

- What Is a Cache?
- Angular Caching
- Caching through \$http
- Setting Default Cache for \$http
- Security
- Strict Contextual Escaping: the \$sce Service
- Whitelisting URLs
- Blacklisting URLs
- \$sce API
- Configuring \$sce
- Trusted Context Types

AngularJS and Internet Explorer

- Ajax Caching
- SEO with AngularJS
- Getting Angular Apps Indexed
- Server Side
- Options for Handling SEO from the Server Side
- Taking Snapshots
- Using Zombie js to Grab HTML Snapshots
- Using grunt-html-snapshot
- Prerender io
- <noscript> Approach

Building Angular Chrome Apps

- Understanding Chrome Apps
- Building our Chrome App
- Building the Skeleton
- manifest json
- tab html
- Loading the App in Chrome
- The Main Module
- Building the Home Page
- Sign Up for Wunderground's Weather API

- A Settings Screen
- Implementing a User Service
- City Autofill/Autocomplete
- Sprinkling in Time Zone Support

DAY 5

Optimizing Angular Apps

- What to Optimize
- Optimizing the \$digest Loop
- Optimizing ng-repeat
- Optimizing the \$digest Call
- Optimizing \$watch Functions
- Optimizing Filters
- Tips for Optimizing Page Load
- Debugging AngularJS
- Debugging from the DOM
- Debugger
- Angular Batarang

Getting ready for production :

- Working with Yeoman, Grunt & Bower in Angular

Best Practices :

Organizing your Code

- Introduction
- File Organization
- Understanding Modules
- Organizing Modules
- Naming Overview
- Namespacing
- Naming Objects

Writing Controllers & Services

- Introduction
- Overview
- Minification
- Understanding Controllers
- Controllers - Design Guidelines
- Understanding Services
- Filters as Services
- Services - Design Guidelines
- Services - Specific Types

Writing Directives and Views

- Introduction
- Directives - Naming
- Directives - Purposes
- Directives - Controllers vs Link
- Directives - Requiring Controllers
- Directive Interfaces
- Avoiding FOUC in Views

- Writing valid HTML with Angular
- Thinking Declaratively Part 1
- Thinking Declaratively Part 2
- Treating HTML as A DSL
- Dealing with Scope
 - Introduction
 - Batarang
 - Scope Overview
 - Creating Scopes
 - Sharing Scope
 - Inheriting Scope
 - Isolating Scope
- Communicating Between Components
 - Introduction
 - Communicating with Inherited Scopes
 - Communicating with Events
 - Communicating with Services
 - Conclusion
- Breaking Down A Page into Components
 - Introduction
 - Inline Controllers
 - Controllers and ngInclude
 - Directives
 - Conclusion
- Communicating with The Server
 - Introduction
 - Using \$resource
 - Using \$http to Read Data
 - Using \$http to Save Data
 - \$http, \$resource, and Promises
 - HTTP Configuration
 - Transforms
 - Interceptors
 - Using Restangular
- Dealing with Models
 - Introduction
 - Creating Models with \$http
 - Creating Models with \$resource
 - Creating Models with Restangular
 - Angular and Backbone Model Together
 - Creating Models with Backbone Model - Retrieving Data
 - Creating Models with Backbone Model - Saving Data
 - A Brief Overview of Breeze
 - Conclusion
- Unit Testing:
 - Why and When to Unit Testing
 - Preparing the Example Project
 - Installing the ngMocks Module

Creating the Test Configuration
Creating the Example Application
Working with Karma and Jasmine
Running the Tests
Understanding the Mock Objects
The Test Objects and APIs
Testing a Controller
Arranging the Test
Using the Mock Objects
Mocking HTTP Responses
Mocking Periods of Time
Testing Logging
Testing Other Components
Testing a Filter
Testing a Directive
Testing a Service

Introduction to E2E with