|  |
| --- |
| Building RIA using JavaScript and HTML5 |
| **Overview** |
| This workshop is aimed at imparting the required knowledge required to build Rich Internet Applications (RIA) using JavaScript and HTML5. |
| **Past occurances of this program** |
| * Amdocs, Pune * Honeywell, Bangalore (3 occurrences) |
| **OBJECTIVES** |
| At the end of this training course, the participants will:   * Understand the architectural difference between a traditional web application and a Rich Internet application. * Understand and appreciate the dynamic typed nature of JavaScript language * Be able to understand the exploit the functional programming features in JavaScript * Harness the power of jQuery for DOM manipulation * Know how to modularize and loadJavaScript files using Require.js * Be able to use Handlebar templates for UI templating * Know the techniques for organizing UI code using Backbone.js * Know how the exploit the infrastructure (storage, real time communication, threading) capabilities offered by HTML5 * Reduce the need of images using CSS3 * Using API Controllers in ASP.NET MVC 4.0 to build REST endpoints. |
| **suggested audience** |
| Web developer who wants to build best-of-breed web UIs with the simplicity and elegance of JavaScript. |
| **DURATION** |
| 4 Days |
| **PARTICIPANT PREREQUISITES** |
| **Must Have**   * Web development experience with HTML, CSS, JavaScript.   **Good To Have**   * Exposure to any server technology (J2EE, .NET, Ruby etc) |
| **infrastructure requirements** |
| * Any Web Server (IIS or Tomcat) * Chrome * Internet Connection |
| **DAY WISE SYLLABUS** |
| **Day – 1**  **RIA and Advanced JavaScript**   * RIA Architecture Overview   + Difference between traditional web application and RIA architectures   + Role of JavaScript & related frameworks   + Overview of REST * 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 * Using Templates   + Avoid string concatenation for dynamic generation of UI   + Template syntax   + Exploring Handlebar.js   + Binding data to templates   + Using pre-compiled templates * Modular JavaScript   + Techniques for modularizing JavaScirpt code   + Using Require.js for loading dependent modules   **Day – 2**  **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   + Managing jQuery chains * Manipulating page contents with jQuery   + Manipulating element properties and attributes   + Manipulating element properties   + Fetching attribute values   + Setting attribute values   + Removing attributes   + Cloning elements   + Dealing with form element values * Events   + Understanding the browser event models   + 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 * Animations and Effects   + Showing and hiding elements   + Implementing a collapsible list   + Toggling the display state of elements   + Animating the display state of elements   + Showing and hiding elements gradually   + Fading elements into and out of existence   + Sliding elements up and down * Talk to the server with Ajax   + Introduction to Ajax   + Loading content into elements   + Making GET and POST requests   + Getting data with jQuery   + Getting JSON data   + Making POST requests   + Taking full control of an Ajax request   + Making Ajax requests with all the trimmings   + Setting request defaults   + Global functions * Extending jQuery with custom plugins   + The jQuery plugin authoring guidelines   + Naming files and functions   + Managing Parameters   + Writing custom utility functions   + Adding new wrapper methods   **Day – 3**  **HTML5 and CSS3**   * 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   **Day -4**  **Building RIA**   * Introduction to Backbone.js   + Need to organize UI code   + Backbone Fundamentals   + View   + Model   + Collection   + Router * Model Basics   + Implement a simple Model   + Modify Model attributes   + Use a Model in the Console * Common Model Methods   + get(key)   + set({key:value})   + isNew()   + toJSON() * View Basics   + Implement a simple View   + Understanding template rendering   + Using View in the console   + Using HandleBar View Engine * Collections Basics   + Implement a collection   + Use underscore methods to manipulate a collection   + Understanding Collection Events * Combining Models Views and Collections   + Extend an existing View   + Writing a template   + Render Models in a Collection   + Using jQuery sub-search in a View   + Adding Content to the DOM   + Render on the Collection’s reset event * Routes   + Constructor Route   + Router Redirection   + Evented Routes * Unit Testing   + Testing Models using Jasmine |