Developing SPA with Angular 4

Workshop Details:

|  |  |
| --- | --- |
| Duration: | 5 Days |
| Description: | Angular 4 is used to develop and deploy SPA applications. |
| Objectives: | This course helps you to develop Single Paged Applications using Angular 4 and helps you to understand the new features added in Angular 4 and CLI. |
| Participants’ Entry Profile: | Participants attending this course must have development experience on:   * Good Knowledge of JavaScript * RESTful services * HTML 5 and CSS 3 |
| Training Methodology: | The workshop will follow Synergetics methodology of   * Concept Visualization * Active Experimentation * Application Development   The workshop will be 100% Hands-On with each participant having access to system during the session |

Setup Requirements:

|  |  |
| --- | --- |
| Hardware and Software Requirements: | Participant’s as well as Trainer’s Machine are required to have:  Hardware   * Intel Pentium 4 [2+ GHz recommended] * 4 GB RAM * 50 GB HDD space * LAN connectivity * Good Internet connectivity and bandwidth   Software [Installed]   * Windows 8.1 or later * NodeJS 6.10 or later * Visual Studio Code * Chrome latest, Firefox latest |
| Training Lab Requirements: | Whiteboard 6 feet by 4 feet (minimum)  Whiteboard markers – Red, Blue, Green, Black  Video Projector (1024 X 768 resolutions) |
| Virtual Lab Requirements:  [Optional] | Virtual labs can be provided for participants, that provides completely configured platform to work with. |

Course Contents:

Day 1

TypeScript Introduction

* Why TypeScript
* Getting Started with TypeScript
* TypeScript Playground
* Basic Types
* Classes in TypeScript
* Inheritance
* Accessors
* Static Property
* Optional, Default & Rest Parameters
* Template Strings
* Arrow Functions
* Function Overload
* Generics
* Interfaces
* Modules
* Declaration file (.d.ts)
* definitelytyped.org

Modules

* Grammar conventions
* @NgModule decorator
* Import and export modules
* Declarations in module
  + Registering Components
  + Registering Pipes
* Registering providers in module
* Bootstrapping component

Components and Directives

* Built-in Components
  + NgIf
  + NgSwitch
  + NgStyle
  + NgClass
  + NgFor
  + Getting an index
  + NgNonBindable
  + NgPluralCase
* Custom Components
  + Component Decorator
  + Exporting components
  + Templates and TemplateUrl
  + Styles and styleUrls
  + [Inputs] and [outputs]
  + @Input and @Output
  + Accessing elements using Reference (#ref)
  + Defining events
    - Event target properties

Day 2

Pipes

* Built-in pipes
  + json
  + slice
  + uppercase
  + lowercase
  + number
  + percent
  + currency
  + date
  + async
* Custom pipes

Dependency Injection

* Injecting on components
* Global injection on modules
* Providers
* Hierarchical injection

Injectable Services

* Built-in Services
* Creating Custom services
  + Registering services in module
  + Registering services in Component
* Injecting Services in Constructor
* Need for @Injectable

Forms

* Template Driven Forms
  + NgForm and NgModel
  + FormsModule
* Reactive Forms
  + FormControl and FormGroup
  + ReactiveFormsModule
* Using Validators
  + Custom validators
* Watching form changes

Day 3

Lifecycle Hooks

* OnInit interface
* OnInit and OnDestroy
* OnChanges
* Constructor vs ngOnInit

Stream programming with RxJS

* EventEmitter
* Creating observables
  + Observer
  + Subscribers
* Promise

Routing and Navigation

* Importing RouterModule
* Creating Routes
  + Path and Components
  + Exporting Routes
* Passing Route parameters
  + ActivatedRoute
  + Params.snapshot
  + Subscribing parameters
  + Passing multiple parameters
* RouterOutlet using <router-outlet>
* RouterLink using [routerLink]
* Defining child routes

Day 4

Http Services

* Importing HttpModule
* Requesting REST services
  + GET, POST, PUT and DELETE
  + RequestOptions
  + Consuming Results
* JSONP

Angular 7 updates