# Angular 4 Advanced Training

**Coverage: Angular 2 and Angular 4**

Angular is a productive Web Frameworks for developing complex Single Page Application with RESTFul backend server. Software Developer should learn Angular correctly to use in production applications, to get all the benefits and features offered by Angular. This training includes extensive workshop sessions, in-depth content, with hands-on from an experienced Chief Software Architect, Developer & Trainer who has architected, developed more than 20+ applications & products for Web, Mobile, Desktop.

**Advanced Angular Training covers introductory to intermediate to advanced topics, 80% of the course is hands on, 20% for trainer instructions, demos.**

This course is designed to meet day-to-day challenges faced by software developers while developing Angular Application.

**The course sessions are designed to build a project in iterative mode with best practices followed in the industry, close to real application. By end of the training, developers shall have experience of applying all concepts learned on a project, ready to apply skills on production applications.**

**Course Abstract and Summary:**

Day 1 of Angular training focus on setting and clearing ground to developers with latest technologies around JavaScript, building foundation for modern JavaScript Development, Introduction to Single Page Application, discussion about Thick vs Thin server, thin vs thick client, ES6, TypeScript, introduction to Node.js, Node Package Manager, Angular Building Blocks, Angular Components, Component Communications, Angular Templates, Angular Pipes, Custom Pipes and Introduction to Services in Angular.

Day 2 of Angular training focus on intermediate level topics, that includes creating custom, reusable modules, Promises, RxJS, Routing, Navigation, HTTP, HTML5 Forms, 2-way data binding, Form Validations. Day 2 shall have interactive with RESTful backend server, editing/deleting resources on RESTful APIs.

Day 3 of Angular training shall focus on advanced topics, Custom Form Validation, Custom Attribute and Structural Directive Creations, Authentication and Authorization with Angular, Introduction to AoT, Introduction to Build Process, Angular-cli, Unit Testing with Angular, Testing Component, Services, Templates, Introduction to Jasmine and Karma.

|  |  |
| --- | --- |
| **Course Duration** | 3 Days |
| **Timings** | 9:00 AM to 6:00 PM |
| **Course Level** | Intermediate/Advanced |
| **Trainer** | Gopalakrishnan Subramani |
| **Hands-on** | Yes |
| **Fundamentals** | Yes |
| **Backend REST API** | 1. Node.js/Express REST API |
| **Angular** | 2.x or 4.x |

# About Trainer

Gopalakrishnan Subramani is a Consultant, Software Architect, Developer and Mentor with over 14+ years of working experience in Full stack web Applications, RESTful Web Services, Backend services, Single Page Web Applications and Hybrid Mobile Applications. He has architected, developed 20+ applications on Web, Mobile and Backend servers. He has good experience in architecting and developing Node.js, Express JS, Angular JS 1.x, Angular, RESTful web services, API Design, MEAN JS, ASP.NET MVC, Spring MVC application with noSQL databases and mobile application development using Ionic Framework, Phonegap/Cordova platform.

He teaches experienced software professionals on intermediate and advanced concepts, best practices, writing clean code, modules, testable applications, hands-on training with use cases that can be used in development projects.

Gopalakrishnan Subramani actively participate in JavaScript, Web, IoT local Meetups, conduct workshops, talks on Web, Nodejs, Angular, Ionic Framework, speak in conferences and developer summits.

|  |  |
| --- | --- |
| **Pre-requites for Participants** | |
| Tools | 1. Install NodeJS 6.10 LTS from <https://nodejs.org/en/download/> 2. Google Chrome Version 55 or later   <https://www.google.com/chrome/browser/desktop/>   1. Git - https://git-scm.com 2. Visual Studio Code <https://code.visualstudio.com/> 3. Angular-cli - https://github.com/angular/angular-cli |
| IDE | Visual Studio Code  + Debugger for Chrome Plug-in  + TypeScript Angular code snippets for Visual Studio Code |
| Developer Skill | Basic JavaScript (/Intermediate preferred) level  HTML, CSS Basic level  Web Technologies  Below skills are preferred, but can be done along with course as well.  RESTful APIs (basic level) (GET/POST/DELETE methods)  Awareness of ES6 (ECMA 2015) preferred  <http://es6-features.org/> |
| Internet | Course need fulltime internet connectivity for Developers. Trainer shall use DataCard or Company given WIFI connectivity |
| Firewall | If you have firewall setup, you should enabled npm registry, github website or provide a proxy settings with username/password for installing software modules |
| System | Laptop/Desktop PCs with one of the operating system   1. Windows 7.0 or later 2. Mac OS 10.11 3. Ubuntu Linux Desktop 15.10 or later |
| Administrator Account | All the systems must have Administrator privileges for developers attending training. |

Below are deliverables for Angular Advanced Training.

|  |  |
| --- | --- |
| **Deliverables** | |
| Presentations | All course presentations as PDF documents at end of the day |
| Example Source Code | All Working Example source codes at end of the day |
| Instructor Demo Source Code | Source code work out by instructor by end of the training |
| Feedbacks | Feedback forms to measure training effectiveness at end of the day |
| Assignments | Assignments for participants at the end of the day |
| Starter Templates | Starter Templates and Files shall be provided to participant for hands-on though github |

|  |  |
| --- | --- |
| **Day 1** | |
| Introduction | Course Objective  Angular Introduction  Single Page Application  MVC Pattern  Single Page vs Multi-page application  Introduction to Development environment  Introduction to Node.js  Introduction to npm (Node Package Manager)  Introduction to CommonJS  Introduction to Transpilers |
| Environment Setup | Node.js  HTTP RESTful API Server for Workshop  Setup NPM project with quick start configuration  Visual Studio Code  Debugger for Chrome  Debugging with Visual Studio Code |
| Foundation  Type Script &  ECMA 2016/ES6 | ES6 features, arrow functions  Template Literals  Block Scope, Constant, Variables  ES6 Classes, Inheritance, Interfaces  ES6 modules, export, import  Promise  TypeScript static type checking  TypeScript Generics  TypeScript Compiler  @types, Definitely Typed project  Angular 1 vs Angular 2 vs Angular Overview |
| Foundations | Components  Zone.js  Reflect.js  Core.JS/Shim  RxJS  Webpack  Angular Branding => 1.x vs 2.0 vs 4.0 |
| Angular Get Started with SystemJS Universal Module Loaders | Angular App Component  Angular App Module  Angular Main Module, dynamic bootstrapping  Bootstrapping an Application component  Bootstrapping Process  Introduction to System JS  Introduction to require module loading, CommonJS |
| Angular-Cli | Introduction to Webpack  Webpack and Module loaders for Angular  Angular-cli  Create projects  Create Services/Components, Helps  Build and Serve |
| Angular Components | Creating Components  Providers  Component selector  Component Templates  Component Styles  Property Binding [One way, Two way]  Event Binding  Get data into Component through Input  Subscribing to components events through output  Nested Component  Deeper Nesting of Components  Components Events  Component Life Cycle  Accessing Other Components |
| Directives | Attribute Directives (NgStyle, NgClass)  Structural Directives (NgIf, NgFor, NgSwitch) |
| Angular Templates | Template Syntax  Function Calls  Property Binding Syntax  Two Way Binding  One-way Binding  Interpolation  Syntactic Sugars in Angular |
| Pipes | Pipes Introduction  Built in Pipes (Currency, Date, Number, Percentages, JSON)  Creating custom Pipes  Passing Arguments to Pipes  Registering Pipes  Chaining Pipes  Http Web Service Calls with Pipes  Async Pipes  Handling Promises in Pipes |
| Services | Implementing Services in Angular  Service Dependency  Dependency Injection  Injectable decorators  Provider definition  Provider configuration at Angular Bootup  Singleton & Multiple Services Instance  Http Web Service calls with Services |

|  |  |
| --- | --- |
| **Day 2** | |
| Angular Module | Creating a custom Module  Module Declarations  Imports  Exports  Providers  Reusing Modules, Components, Directives, Services |
| Rxjs | Introduction to Reactive Extension (RxJS) Library  What is Reactiveness  Observables  Subscription  Events  Streaming in Observables  Http Web Service with Observables |
| Promises | ES6 Promise Introduction  Custom Promises  Promises vs Observable  Using HTTP Web Services with Promises |
| HTML5 Forms | Form Controls  Input  Select  Buttons  ngSubmit  HTML 5 Validations  Angular Styles, special form valid, pristine, dirty features  Two Way Binding  Angular Basic Validations |
| Routing | Route Introduction  Configuring Routes  Redirection  Linking  Passing values/parameters between Routes  Router Outlet  Nested Routing /Child Routes  Handling Query Parameters  CanActivate Guard  CanDeactivate Guard |
| HTTP | Angular HTTP  HTTP backend REST API/Web API Communication with GET, POST, PUT, DELETE methods using Angular HTTP  HTTP GET, POST, PUT and DELETE  URL Query Parameters  Promises  HTTP with Observables |

|  |  |
| --- | --- |
| **Day 3** | |
| Topic |  |
| Custom Directives | Creating Custom Attribute Directives  Creating Custom Structural Directives  HostListener  HostBinding |
| Angular Reactive  Forms | Reactive Forms Introduction  Control Groups  Form Validation  Form Builder  Form Builder Validation  Custom Validation |
| Angular Security | Cross Origin Request  CORS Headers Introduction  JSONP |
| Authentication  Authorization | JWT Token based Authentication  Storing Tokens in HTML5 Storage  Authorization with Angular  Handling Multiple Roles |
| CSS/SCSS | Handling SCSS files [Optional]  Scoped styles  In-lining CSS in components  Handling CSS Files using webpack |
| HTML5  Storage | Introduction to localStorage  Introduction to sessionStorage  Security with HTML5 storage  Cookie fundamentals  HTML5 Storage for Application Cache |
| Unit Testing Angular | Test Driven Development  Introduction to Jasmine  Introduction to Karma  Jasmine Test Suites, Specs, Expectations  Jasmine Matchers  Jasmine Custom Matchers  Setup Injection, Environment for Angular testing  Testing Pipes  Testing a Components  Testing a Service  Testing with HTTP  Mocking HTTP, Services  Testing Directives |
| Angular  AoT Compilation | AoT Compilation  AoT Module Bootstrapping |
| Setup Release Environments | Release Build  Bundle Creation  Source map generation with ts-config |