**Angular 2/4/5 (Duration: 5 days)**

**Overview**

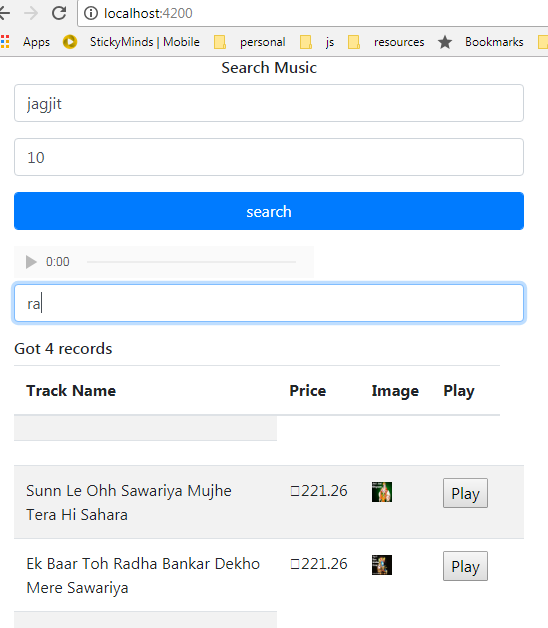
This Comprehensive Angular training class teaches students the skills and best practices they need to design, build, test, and deploy applications that provide rich end-user experiences similar to a desktop application while still offering the ease of deployment of a web application.

**Goals**

1. Understand how single-page web application architectures (including Angular) are different than traditional web development architectures
2. Use new JavaScript (ES6) language features including Classes, Modules, and Arrow Functions
3. Use new TypeScript language features including Static Types, Interfaces, and Generics
4. Build an application from scratch using Angular 5
5. Learn Angular coding and architecture best practices
6. Understand and use Angular Model-driven Forms, Observables, Dependency Injection, and Routing
7. Communicate with a backend server using Angular's HttpClient service to load and save data.
8. Unit test all the parts of an Angular application including Modules, Components, Services, and Pipes
9. Upgrade an existing application from AngularJS to Angular 5 over time by running both frameworks in the same project
10. Start a new Angular project and scaffold modules, components, services, models, routes, and unit tests following best practices using the Angular CLI
11. Build and deploy an Angular application including combining and minifying JavaScript and HTML files, Tree-shaking unused code, and doing Ahead-of-Time compilation to reduce the size of the Angular framework.

**Pre-requisites: Trainees must have strong knowledge on Javascript, CSS , HTML and Restfull web-service**

**The entire training is based on developing a form-based SPA application (refer below screenshot) with live-coding. 70% time will be dedicated to live development.**



<https://itunes.apple.com/search?term=jack+johnson&limit=25>

**Content:**

1. **Why Angular?**
   1. User Experience similar to a Desktop Application
   2. Productivity and Tooling
   3. Performance
   4. Community
   5. Full-featured Framework
2. **Understanding Angular Versions**
   1. AngularJS (Angular 1.x)
   2. Angular
   3. Angular 2
   4. Angular 5
3. **TypeScript and ECMAScript 6 (ES6) Fundamentals**
   1. Classes
   2. ES Modules
   3. Arrow Functions
   4. Template Literals
   5. Scoping using let and const Keywords
   6. Spread Syntax and Rest Parameters
   7. Destructuring
   8. Decorators (JavaScript Aspect-Oriented Programming)
4. **Angular 5 Basics**
   1. Components
   2. Templates
      1. Inline Templates
      2. Multi-line Templates using ES6 Template Literals
      3. External with Component-relative Paths
   3. Modules
      1. Angular Modules vs. ES Modules
   4. Models
5. **Template Syntax**
   1. HTML in templates
   2. Interpolation
   3. Binding syntax
   4. Property binding
   5. Event binding
   6. Two-way data binding
   7. Attribute, class, and style bindings
   8. Built-in Directives
      1. Built-in attribute directives: NgClass, NgStyle, NgModel
      2. Built-in structural directives: \*ngIf, \*NgFor
   9. Template Input Variables
   10. The NgSwitch Directives
   11. Template Reference Variables
   12. Input and output properties
   13. Template Expression Operators
   14. Pipe ( | )
6. **Components**
   1. Component Lifecycle Hooks
      1. Implementing the OnInit Lifecycle Hook
   2. Component Communication
      1. Input properties
      2. Output properties: Custom Events using EventEmitters
7. **Services and Dependency Injection**
   1. Using a services to access data
   2. Using a service to encapsulate business logic
   3. Understanding the scope of services
8. **Dependency Injection**
   1. Understanding Dependency Injection
   2. Angular's Dependency Injection System
   3. Registering
   4. Injecting
   5. Hierarchical Injection
9. **Model-driven Forms (Reactive Forms)**
   1. Importing the ReactiveFormsModule
   2. FormControl, FormGroup, and AbstractControl
   3. Binding DOM Elements to FormGroups and FormControls
   4. Validation Rules, Messages, and Styles
   5. Refactoring ReactiveForms for Reuse
   6. Custom Validators
10. **RxJS and Observables**
    1. What is an Observable?
    2. Observable Operators
    3. Creating Observables Using Static Operators
    4. What is an Observer?
    5. Observer Example
    6. Subject
    7. Subject Example
    8. EventEmitter or Observable
11. **Communicating with the Server using the HttpClient Service**
    1. Deciding between Promises or Observables (RxJS)
    2. Making a HTTP GET Request
    3. Sending data to the server using Http POST and PUT Requests
    4. Issuing a Http DELETE Request
    5. Intercepting Requests and Responses
12. **Router**
    1. Importing the RouterModule and Routes
    2. Configuring Routes
    3. Displaying Components using a RouterOutlet
    4. Navigating declaratively with RouterLink
    5. Navigating with code using the Router
    6. Accessing parameters using ActivatedRoute
    7. Organizing your code into Modules
13. **Unit Testing**
    1. Tools: Jasmine, Karma
    2. Jasmine Syntax: describe, it, beforeEach, afterEach, matchers
    3. Setup and your First Test
    4. Testing Terminology: Mock, Stub, Spy, Fake
    5. Angular Testing Terminology: TestBed, ComponentFixture, debugElement, async, fakeAsync, tick, inject
    6. Simple Component Test
    7. Detecting Component Changes
    8. Testing a Component with properties (inputs) and events (outputs)
    9. Testing a Component that uses the Router
    10. Testing a Service and Mocking its Http requests
    11. Testing a Pipe
14. **Advanced Components**
    * 1. Component Styles
      2. using MetaData properties: Styles and StyleUrls
      3. Encapsulation Strategies
    1. Change Detection Strategies
    2. Component Lifecycle Hooks
15. **Advanced Routing**
    1. Lazy-loading Angular Modules
    2. Location Strategies
    3. Nested or Child Routes
    4. Route Guards
16. **Pipes**
    1. Built-in Pipes: Using, Passing Parameters, Chaining
    2. Creating a custom Pipe using PipeTransform
    3. Understanding Pure and Impure Pipes
17. **Creating, Building, and Deploying an Angular Application**
    1. Manually
    2. Using the Angular CLI
       1. Overview
       2. Features
       3. Installation
       4. Generating a New Project
       5. Generating Code
       6. Builds
       7. Customizing Builds
       8. Angular Material Setup
       9. Eject
18. **Template-driven Forms**
    1. NgSubmit Directive
       1. FormsModule
       2. NgForm, NgModel, and NgModelGroup Directives
       3. Validation Directives
          1. Displaying validation messages
          2. Styling validation messages
19. **npm QuickStart**
    1. Installing Dependencies Locally
    2. Using npm as a Build Tool
20. **Webpack Guide**
    1. Installation
    2. Building/Bundling
       1. JavaScript
       2. CSS
       3. HTML
       4. Images
    3. Development Builds
    4. Production Builds
21. **Conclusion**