Cognixia®

# COURSE CONTENT

www.cognixia.com

## COURSE NAME

**Full Stack Engineer II (Angular)**

## COURSE OUTLINE

**HTML5, CSS3, JavaScript & TypeScript**

**The Web: HTML & CSS**

- Different versions of HTML
- What is DOCTYPE?
- Standards and Do's and Don'ts's in HTML
- Block-level and Inline Elements
- HTML Attributes
- HTML Entities
- DOM
- Forms
- HTML Layout Elements
- HTML Responsive Web Design (meta tag)
- HTML Web Storage
- HTML Web Workers
- HTML5 New Features

**Styling with CSS**

- Concepts, Features, Syntax, Properties
- Comments
- Selectors – id, class, grouping of selectors
- Three ways to insert CSS
- Text alignment
- CSS font – family, size, style
- CSS tables, lists
- Box Model
- Backgrounds
- CSS Properties
- CSS Units
- CSS Responsive
- CSS Variables
- Pseudo-Classes and Pseudo-Elements

**Working with JavaScript**

- A Brief Introduction to JavaScript
- Linking a JavaScript File
- Values and Variables
- Data Types
- let, const, and var
- Basic Operator
- Operator Precedence
- Strings and Template Literals
- Taking Decisions: if / else Statements
- Type Conversion and Coercion
- Truthy and Falsy Values
- Equality Operators: == vs. ===
- Boolean Logic
- Logical Operators
- The Switch Statement
- Statements and Expressions
- The Conditional (Ternary) Operator
- JavaScript Releases: ES5, ES6+, and ESNext
- Activating Strict Mode

**JavaScript Functions**

- Function Declarations vs. Expressions
- Arrow Functions
- Functions Calling Other Functions
- Reviewing Functions
- Introduction to Arrays
- Basic Array Operations (Methods)
- Introduction to Objects
- Dot vs. Bracket Notation
- Object Methods
- Iteration: The for Loop
- Looping Arrays, Breaking, and Continuing
- Looping Backwards and Loops in Loops
- The while Loop

**Debugging JavaScript**

- Setting up Prettier and VS Code
- Installing Node.js and Setting Up a Dev Environment
- Debugging (Fixing Errors)
- Debugging with the Console and Breakpoints

**Performing DOM Manipulation**

- What's the DOM and DOM Manipulation
- Selecting and Manipulating Elements
- Handling Click Events
- Manipulating CSS Styles
- Refactoring Code: The DRY Principle
- Working With Classes
- The JavaScript Engine and Runtime
- Execution Contexts and The Call Stack
- Destructuring Arrays
- Destructuring Objects
- The Spread Operator (...)
- Rest Pattern and Parameters
- Short Circuiting (&& and ||)
- Logical Assignment Operators
- Looping Arrays: The for-of Loop
- Enhanced Object Literals
- Looping Objects: Object Keys, Values, and Entries

**Working with Sets and Maps**

- Sets
- Maps: Fundamentals
- Maps: Iteration
- Working With Strings
- String Methods Practice
- Default Parameters
- How Passing Arguments Works: Value vs. Reference
- First-Class and Higher-Order Functions
- Functions Accepting Callback Functions
- Functions Returning Functions
- The Call and Apply Methods
- The Bind Method

- Immediately Invoked Function Expressions (IIFE)
- Closure

**Working with TypeScript**

- Getting started with TypeScript
- Working of TypeScript
- Type Annotations
- Classes
- Fields
- Parameter Properties
- Member visibility (public, private, protected)
- Automatically Import Modules
- Nullable Parameters
- Using Interfaces in TypeScript

**GIT**

**Introduction**

- Introduction
- About Version Control
- Centralized vs. Distributed VCSs
- History of Git
- Differences between Git and other VCSs

**Configuring Git**

- Installing Git
- git help
- git config
- Setting User Identity

**Working with Git**

- Initializing a Repository in a Directory
- Cloning an Existing Repository
- Checking the Status of Files
- Tracking New Files
- Staging Modified Files
- Ignoring Files
- Moving & Removing Files

- Working with Remote Repositories
- Committing Changes
- Viewing Commit History
- Rolling Back Changes

**Branching with GIT**

- Branches in Git
- Creating a Branch
- Switching Branches
- Merging Branches
- Merge Conflicts

**SDLC & Agile**

- The SDLC: The Life Cycles
- Overview of SDLC
- Overview of Project Management
- Overview of Business Analysis
- Agile Overview from Industry Practices
- Introduction to Agile Project Management
- Fundamentals of the Scrum Framework
- Identifying the roles and their responsibilities
- Engineering Practices Overview

**JIRA**

**Introduction**

- Introduction to Jira
- Getting Started
- Project Administration

**Jira for Agile Development**

- Agile Development
- Managing User Stories, Epics, and Themes
- Product Backlog Management

**Jira in Scrum Processes**

- Scrum Framework Integration
- Sprint Planning

- Daily Scrum
- Tracking Progress: Burndown Charts & Velocity
- Sprint Retrospective Meetings
- Sprint Review Practices

**Jira Boards and Reporting**

- Scrum Board
- Generating Agile Reports
- Exploring Reports and Dashboards

**Angular Framework**

**Introduction to Angular Framework**

- Introduction to Angular Framework, History & Overview
- Environment Setup, Angular CLI, Installing Angular CLI
- NPM commands & package.json
- Bootstrapping Angular App, Components, AppModule
- Project Setup, Editor Environments
- First Angular App & Directory Structure
- Angular Fundamentals, Building Blocks
- MetaData

**Essentials of Angular**

- Component Basics
- Setting up the templates
- Creating Components using CLI

**Nesting Components**

- Data Binding - Property & Event Binding, String Interpolation, Style binding
- Two-way data binding
- Input Properties, Output Properties, Passing Event Data

**Template-Driven and Reactive Forms**

- Template-Driven vs Reactive Approach
- Understanding Form State
- Built-in Validators & Using HTML5 Validation
- Grouping Form Controls
- FormGroup, FormControl, FormBuilder

- Forms with Reactive Approach
- Predefined Validators & Custom Validators
- Showing validation errors

**Routing with Angular**

- Component Life Cycle Hooks
- Navigating with Router links
- Understanding Navigation Paths
- Navigating Programmatically
- Passing Parameters to Routes
- Passing Query Parameters and Fragments

**HTTP Requests / Observables**

- HTTP Requests
- Sending GET Requests
- Sending a PUT Request
- Using the Returned Data
- Catching Http Errors
- Basics of Observables & Promises

**Angular-Spring Boot Integration**

- Introduction to how Angular front-end communicates with Spring Boot back-end
- Why combine Angular with Spring Boot?

**Setting Up the Backend in Spring Boot**

- Creating RESTful services with Spring Boot
- Exposing API endpoints using @RestController
- Handling different HTTP methods (GET, POST, PUT, DELETE)

**Configuring CORS in Spring Boot**

- Understanding Cross-Origin Resource Sharing (CORS)
- Configuring Spring Boot to handle CORS for Angular communication

**Sending HTTP Requests from Angular**

- Setting up Angular services to interact with the Spring Boot REST API
- Using Angular's HttpClient for sending GET, POST, PUT, DELETE requests
- Handling API responses and error handling

**Data Binding with Angular and Spring Boot**

- Passing data from Angular forms to Spring Boot backend
- Binding back-end responses to Angular components

**Managing Authentication and Security**

- Integrating JWT (JSON Web Token) Authentication in Spring Boot
- Securing API endpoints in Spring Boot
- Handling authentication flow in Angular (login, token storage, and authorization)

**Form Handling with Angular and Spring Boot**

- Submitting Angular forms to Spring Boot API
- Using Reactive and Template-driven forms with backend validation
- Showing validation errors from the Spring Boot backend

**Routing and Data Fetching**

- Navigating between components based on backend data
- Passing parameters and using route guards in Angular based on API responses

**Working with Observables and Promises**

- Handling asynchronous data using Observables in Angular
- Using Promises and chaining API requests

**MongoDB**

**Introduction to MongoDB and NoSQL**

- What is NoSQL?
- CAP Theorem and why it matters
- Types of NoSQL Databases
- Document Databases and their advantages
- Why choose MongoDB for your projects?
- Comparing SQL Terms vs MongoDB Terms
- Overview of JSON and BSON

**Setting up MongoDB**

- Installing and configuring MongoDB
- Getting started with MongoDB: Command Line and Compass
- Understanding the MongoDB environment

**CRUD Operations in MongoDB**

- Overview of CRUD Operations
- Creating Documents: Single and Multiple Inserts
- Retrieving Data with Queries and Projections
- Updating Documents: One and Many
- Deleting Documents: Safely and Efficiently

**Demos on CRUD Operations**

- Using Compass for CRUD Operations
- Command Line: CRUD Demonstrations
- Best practices in CRUD operations

**MongoDB Data Handling**

- Read and Write Concerns: Ensuring Data Integrity
- Atomic Operations and Transactions
- Indexing for Performance Improvement
- Aggregation Framework: Simplifying Complex Queries

**Relating MongoDB to SQL Concepts**

- SQL vs MongoDB: Differences in Structure
- SQL Semantics vs MongoDB Semantics
- Mapping SQL relationships in MongoDB
- Which database model is right for your use case?

**Spring Framework**

**Introduction**

- Introduction to Spring Framework
- Understand the core principles of Spring
- Explore the various spring modules
- The IOC Container
- Perform Dependency Injection

**Types of Configurations**

- Understand different types of Configurations
- XML Configuration
- Java Configuration

- Annotation Configuration

**Spring Beans**

- Understand bean lifecycle
- Understand bean scopes
- Create Singleton and Prototype scoped beans.

**AOP**

- Aspect Oriented Programming Core Concepts
- Understanding AOP Advice Types
- Before Aspect Example
- AOP Pointcut Methods
- AOP JoinPoints
- AOP After Advice Example
- AOP Around Aspect Example

**Spring MVC Framework**

- Introduction to MVC Architecture
- Understand the model component
- Understand the view component
- Understand the Controller component
- Creating a Spring MVC application

**Getting started with Spring Boot**

- Introduction to Spring Boot
- Features of Spring Boot
- Using Spring Initializr
- Understanding Spring Boot Auto Configuration
- Exploring Spring Boot Annotations
- Configuring application properties
- Understanding profiles
- Using Spring Boot developer tools

**Introduction to Web Services**

- Understand SOA
- Define a web service
- Explore the need for web services
- Identify the characteristics of a web service

- Architecture of web services

**Building RESTful web services with Spring Data JPA**

- Introduction to REST
- The GET, POST, PUT, and DELETE methods
- The Accept and Content type headers
- Creating a CRUD REST API
- Implementing JPA based repositories
- Perform validation and exception handling
- Documenting RESTful web services
- Versioning RESTful web services

**Responsive Web Design**

**Introduction to Responsive Web Design**

- What is responsive Web design?
- Why should I use RWD?
- Where did RWD come from?
- Who is responsible for RWD?
- When should I use RWD?
- How do I implement RWD?

**Fundamental Techniques of RWD**

- Feature detection
- The viewport element
- Responsive layouts vs. adaptive layouts
- Media queries
- Responsive images
- Responsive tables

**Introduction to Feature Detection**

- What is Modernizr?
- Using the Modernizr JavaScript library
- How Modernizr works
- Using the Modernizr object and API
- HTML5 shim and Modernizr

**CSS Feature Queries**

- CSS @supports
- CSS.supports() method

**Understanding the Viewport**

- What is the viewport?
- Pixel width of the viewport
- The viewport meta tag

**Media Attribute and Media Queries**

- Specifying a media type
- HTML media attribute
- What is a media query?
- Syntax and features of media queries

**The CSS Box Model**

- What is the CSS box model?
- CSS reset
- Box-sizing and size calculations

**The Display Property**

- Inline vs block elements
- Important display properties

**Overview of RWD Layout Techniques**

- Fixed layout vs liquid layout
- Elastic layouts
- Responsive vs Adaptive layouts

**Flexbox Layout**

- What is Flexbox?
- Flexbox in Bootstrap 4
- Flexbox properties and usage
- Creating a responsive flexbox layout

**Grid Layout**

- Introduction to Grid layout
- Grid terminology and behavior
- Using grid properties and named lines

**CSS Frameworks and Bootstrap**

- Responsive frameworks
- What is Bootstrap?
- Why use Bootstrap?
- Bootstrap 4 and Bootstrap 5 differences

**Fundamentals of Layout with Bootstrap**

- Containers
- Media queries and responsive breakpoints

**Bootstrap Components**

- Typographic styles
- Table styles
- Image styles
- Modals, Forms, Cards

**Responsive Images**

- Scaling images with CSS
- Max-width, cropping, and server-side solutions

**Responsive Tables**

- Techniques for responsive tables
- Hiding and scrolling table data

**Testing with Jest and Enzyme**

**Overview**

- Why testing is important
- Overview of Jest and Enzyme
- Basic testing terminology

**Setting Up the Testing Environment**

- Creating a React project with create-react-app
- Installing Jest and Enzyme
- Basic configuration in package.json

**Writing Your First Tests with Jest**

- Structure of a Jest test file
- Writing simple test cases with it() or test()
- Running tests with npm test

**Testing React Components with Enzyme**

- Introduction to Enzyme
- Setting up Enzyme in your project
- Shallow rendering of components
- Writing basic component tests

**Jest Matchers and Assertions**

- Common Jest matchers (toBe, toEqual, toContain, etc.)
- Writing meaningful assertions

**Snapshot Testing**

- Understanding snapshot tests
- Creating and updating snapshots
- When to use snapshot testing

**Mocking Basics**

- Introduction to mocking
- Mocking simple functions with Jest
- Testing components with mocked props

**Testing User Interactions**

- Simulating clicks and other events with Enzyme
- Testing component state changes

**Asynchronous Testing**

- Testing components with asynchronous behavior
- Using async/await in tests

**Best Practices and Common Patterns**

- Organizing test files
- Naming conventions
- What to test and what not to test