

# Temel ve İleri Seviye Angular Eğitimi

## Eğitim Hakkında

Digital Vizyon Akademi'nin "Temel ve İleri Seviye Angular Eğitimi", katılımcılara ES6, TypeScript ve npm ve Webpack dahil modern ön uç araçları kullanarak uygulamaların nasıl oluşturulacağını öğretir.

Katılımcılar ayrıca Angular'da uygulama mimarisi ve tasarım en iyi uygulamaları hakkında bir anlayış kazanırlar, bir Angular uygulamasında kimlik doğrulama, birim test etme ve uygulama durumunu yönetmeyi öğrenirler.

## Neler Öğreneceksiniz

- Understand how single-page web application architectures are different than traditional web application architectures
- Use new JavaScript (ES6) language features including Classes, Modules, and Arrow Functions
- Use new TypeScript language features including Types, Decorators, Interfaces, and Generics
- Learn Angular coding and architecture best practices including project layout and using container and presentation components
- Understand and use Angular model-driven forms, observables, dependency injection, and routing
- Communicate with a backend server using Angular's HttpClient to load and save data
- Configure the router and navigate between components
- Unit test all parts of an application including Components, Services, and Pipes
- Implement Authentication and Authorization in an Angular Application
- Optimize Angular Performance by changing Change Detection Strategies
- Setup new projects from scratch using the Angular CLI
- Scaffold modules, components, services, models, routes, and unit tests in accordance with best practices using the Angular CLI
- Write End-to-End Tests (optional; taught only if this applies to your group)
- Upgrade an existing application from AngularJS to the current Angular version (optional; taught only if this applies to your group)

## Eğitim İçeriği

### Introduction

### TypeScript and ECMAScript 6 (ES6) Fundamentals

- TypeScript Installation, Configuration & Compilation
- Type Annotations
- Classes
- Scoping using let, var, and const Keywords
- Arrow Functions
- ES Modules
- Decorators
- Template Literals
- Spread Syntax and Rest Parameters
- Destructuring

## Angular Overview

- Benefits of Building using Angular
- Understanding Angular Versions
- Single-page Web Application Architectures vs. Traditional Server-side Web Application Architectures
- Angular Style Guide
- Angular Architecture
- Angular Compared to Other JavaScript Libraries and Frameworks (React, VueJS, etc...)
- Your First Angular Application

## Components

- Understanding Components
- Component Properties & Methods
- Templates: Inline, Multi-line, and External with Component-relative Paths

## Angular Modules (NgModule)

- Angular Modules vs. ES Modules
- Organizing your code into Feature Modules

## Project Set-Up (Using the Angular CLI)



- Angular CLI Features
- Creating a New Project (with new CLI Prompts)
- Generating Code
- Customizing the Angular CLI

## Data Binding

- Interpolation
- Property binding
- Event binding
- Two-way data binding

## Directives

- Structural: ngFor, ngIf, ngSwitch
- Attribute: ngClass, ngStyle

## Pipes

- Built-in Pipes: Using, Passing Parameters, Chaining

## Advanced Components

- Component Communication using @Input, @Output
- Component Architecture
- Component Styles
- Component Lifecycle Hooks
- Evaluating UI Component Frameworks & Libraries

## Services & Dependency Injection

- Using a service to access data
- Using a service to encapsulate business logic
- Understanding the scope of services

## Dependency Injection

- Understanding Dependency Injection
- Angular's Dependency Injection System
- Registering
- Injecting

## Model-driven Forms (Reactive Forms)

- Importing the ReactiveFormsModule
- FormControl, FormGroup, and AbstractControl
- Binding DOM Elements to FormGroups and FormControls
- Validation Rules, Messages, and Styles
- Refactoring Reactive Forms for Reuse
- Custom Validators

## Communicating with the Server using the HttpClient Service

- Deciding between Promises or Observables (RxJS)
- Making an HTTP GET Request
- Sending data to the server using Http POST and PUT Requests
- Issuing an HTTP DELETE Request
- Intercepting Requests and Responses

## Router

- Importing the RouterModule
- Configuring Routes
- Displaying Components using a RouterOutlet
- Navigating declaratively with RouterLink
- Navigating with code using the Router
- Accessing parameters using ActivatedRoute

## Deploying an Angular Application to Production

- Building an application using the Angular CLI
- Differential loading: creating a modern build (ES2015) and a legacy build (ES5)
- Deploying to a web server

## Upgrading to the latest version of Angular from earlier versions

- 2.x and above
- Update Guide
- Deprecation Guide
- Patching Dependencies: npm audit fix
- Looking for AngularJS to Angular upgrades? See optional topics below.

## RxJS and Observables

- What is an Observable?
- Creating Observables
- What is an Observer?
- Observer Example
- Operators: map, switchMap, debounceTime, distinctUntilChanged
- Practical Application of using RxJS
- Subscriptions
- Unsubscribing from Observables in Angular (unsubscribe, Async Pipe, takeUntil)
- Subject
- Subject Example
- Subject Variants (AsyncSubject, BehaviorSubject, ReplaySubject)
- EventEmitter or Observable

- RxJS Operators and HTTP

## Unit Testing

- Tools: Jasmine, Karma
- Jasmine Syntax: describe, it, beforeEach, afterEach, matchers
- Setup and your First Test
- Testing Terminology: Mock, Stub, Spy, Fakes
- Angular Testing Terminology: TestBed, ComponentFixture, debugElement, async, fakeAsync, tick, inject
- Simple Component Test
- Detecting Component Changes
- Testing a Component with properties (inputs) and events (outputs)
- Testing a Component that uses the Router
- Testing a Component that depends on a Service
- Testing a Service and Mocking its HTTP requests
- Testing a Pipe

## Security

- Best Practices
- Preventing Cross-site Scripting (XSS)
- Trusting values with the DOMSanitizer
- HTTP Attacks (CSRF and CSSI)
- Authentication using JSON Web Tokens (JWT)
- Authorization: Router Guards

## Change Detection

- Understanding Zone.js and Change Detection
- Change Detection Strategies Default and OnPush

## Advanced Angular CLI

- Customizing a build using Builder APIs in the CLI
- Generating web workers

## Advanced Routing

- Lazy-loading Angular Modules (using Dynamic Imports)
- Nested or Child Routes
- Providers
- Hierarchical Injection
- providedIn options: root, module, platform, any

## Pipes

- Creating a custom Pipe using PipeTransform
- Understanding Pure and Impure Pipes

## Conclusion

- Choose any two additional topics. If desired, the course can be customized to include more than two of these topics if other topics are scaled back or removed.
  - npm QuickStart
    - Installing Dependencies
    - Understanding package.json and package-lock.json
    - Using npm as a Build Tool
  - Benefits Overview
  - Three Principles of Redux: Single Source of Truth, State is Read-Only, Pure Functions
  - Examples of Pure Functions
  - Reducers
  - Simple ngrx Example
  - Time-traveling with Redux Devtools
  - Full ngrx Example Application
  - Upgrade Strategies from AngularJS
    - High-level Approaches
    - Concept Mapping AngularJS to Angular
    - UpgradeAdapter

- What can be Upgraded or Downgraded
- What cannot be Upgraded or Downgraded
- UpgradeAdapter and Dependency Injection
- End-to-End Testing
  - What is Protractor?
  - Why Protractor?
  - Using Locators
  - Page Objects
  - Debugging E2E Tests