



**TEMEL VE İLERİ SEVİYE
ANGULAR EĞİTİMİ
5 GÜN**



Digital Vizyon
Akademi

www.digitalvizyon.net



İçindekiler

| | |
|--|---|
| Eğitim Hakkında..... | 3 |
| Neler Öğreneceksiniz? | 3 |
| Ön Koşullar | 4 |
| Kimler Katılmalı..... | 4 |
| Outline | 4 |
| TypeScript and ECMAScript 6 (ES6) Fundamentals | 4 |
| Angular Overview | 5 |
| Components | 5 |
| Angular Modules (NgModule)..... | 5 |
| Project Set-Up (Using the Angular CLI)..... | 5 |
| Data Binding | 5 |
| Directives | 6 |
| Pipes | 6 |
| Advanced Components | 6 |
| Services & Dependency Injection..... | 6 |
| Dependency Injection..... | 6 |
| Model-driven Forms (Reactive Forms) | 6 |
| Communicating with the Server using the HttpClient Service | 7 |
| Router..... | 7 |
| Deploying an Angular Application to Production | 7 |
| Upgrading to the latest version of Angular from earlier versions | 7 |
| RxJS and Observables | 7 |
| Unit Testing..... | 8 |
| Security..... | 8 |
| Change Detection | 8 |
| Advanced Angular CLI..... | 9 |
| Advanced Routing..... | 9 |
| Pipes | 9 |
| Conclusion | 9 |

Eđitim Hakkında

Temel ve İleri Seviye Angular Eđitimi, Angular uygulamalarınızı etkili ve karmaşık bir şekilde geliřtirmek için temel ve ileri düzey teknikler öğrenmek isteyen geliřtiriciler için ideal bir kaynaktır. Bu kapsamlı eđitim, Angular'ın temel ve ileri düzey özelliklerini ayrıntılı olarak inceler.

Eđitimin ilk bölümünde, Angular'ın temel kavramları ve özelliklerine odaklanıyoruz. Katılımcılara, Angular bileşenleri, modülleri, direktifleri ve servisler gibi temel konseptlerini ve bunların uygulamalar üzerindeki etkilerini anlatıyoruz.

Daha sonrasında, temel düzeyde bir Angular uygulaması oluşturmayı öğreniyoruz. Formlar, HTTP istekleri ve temel yönlendirme gibi konuları ele alarak, uygulamaları nasıl oluşturacaklarını ve genel web geliřtirme süreçlerini nasıl yöneteceklerini öğrenirler.

İleri düzey bölümünde, Angular'ın ileri düzey özelliklerini ve tekniklerini inceliyoruz. Reaktif programlama, ileri düzey bileşenler, direktifler ve performans optimizasyon teknikleri üzerine duruyoruz. Ayrıca, Angular'da güvenlik ve erişilebilirlik gibi kritik konuları ele alıyoruz.

Eđitim ayrıca, Angular'ın performans optimizasyonu ve ölçeklendirme tekniklerine de yer veriyor. Lazy Loading, Change Detection ve bundling gibi ileri düzey tekniklerin nasıl kullanılacağını ve uygulamaların yüksek trafikli ortamlarda nasıl optimize edileceđi ve ölçekleneceđi üzerine odaklanıyoruz.

Temel ve İleri Seviye Angular Eđitimi, gerçek dünya örnekleri üzerinde çalışma fırsatı sunar. Katılımcılar, Angular ile ilgili öğrendikleri temel ve ileri düzey kavramları uygulamalı projelerde uygularlar. Bu, Angular'ın geniş kapsamlı özelliklerinin, performans optimizasyonunun ve en iyi uygulamaların daha iyi anlaşılmasına yardımcı olur.

Temel ve İleri Seviye Angular Eđitimi, Angular uygulamalarınızı etkili ve performanslı bir şekilde geliřtirmek için gereken geniş kapsamlı bilgi ve becerileri kazandırır.

Neler Öğreneceksiniz?

Angular eđitimlerinde, řu konular üzerinde durulabilir:

- Component yapısı ve yapılandırması: Angular componentlerinin nasıl tanımlanması, yapılandırılması ve bir araya getirilmesi gerektiđi öğrenilir.
- Veri yönlendirme ve form oluřturma: Angular ile veri yönlendirme ve form oluřturma konuları ele alınır ve uygulamalı olarak çalışılır.
- Servisler: Angular ile servislerin nasıl tanımlanması, yapılandırılması ve kullanılması öğrenilir.
- API entegrasyonu: Angular ile RESTful API'lerin nasıl entegre edileceđi ve kullanılacağı öğrenilir.
- Veri iletiřimi: Angular componentleri arasında veri iletiřimi konuları ele alınır.



- Uygulama performansının iyileştirilmesi: Angular ile uygulama performansını nasıl iyileştireceğiniz ve uygulamanızı hızlandıracak yollar hakkında bilgi sahibi olursunuz.
- En son Angular trendleri ve best practices: Angular ile ilgili en son trendler ve best practices hakkında bilgi sahibi olursunuz.

Ön Koşullar

Angular eğitimlerine katılabilmek için genel olarak şu ön koşullar bulunabilir:

- HTML, CSS ve JavaScript temel bilgisi: Angular, HTML, CSS ve JavaScript ile çalışır, bu dillerin temel kavramlarını anlammanız gerekir.
- Programlama deneyimi: Angular, bir programlama dili olarak kullanılır ve programlama deneyimi olması öğrenmenizi kolaylaştırabilir.
- TypeScript bilgisi: Angular, TypeScript dili ile yazılır, bu dille temel bilgi sahibi olmanız faydalı olabilir.

Bu ön koşullar sadece bir örnektir ve farklı Angular eğitim verilen yerlerde farklılık gösterebilir. Ancak, Angular'ı daha verimli bir şekilde öğrenmenizi sağlamak için bu ön koşulların sağlanması faydalı olabilir.

Kimler Katılmalı

Angular eğitimlerine katılmak isteyen kişiler arasında şunlar bulunabilir:

- Front-end geliştiriciler: Angular ile dinamik ve verimli web uygulamaları oluşturmak isteyen front-end geliştiriciler.
- Full-stack geliştiriciler: Angular ile full-stack geliştirme yapmak isteyen ve Angular'ın tüm özelliklerini kullanmak isteyen full-stack geliştiriciler.
- Web tasarımcılar: Angular ile kullanıcı dostu ve interaktif web uygulamaları tasarlamak isteyen web tasarımcılar.
- Öğrenmeye açık herkes: Angular ve web geliştirme ile ilgilenen ve öğrenmeye açık herkes.

Angular eğitimlerine katılma nedeniniz ne olursa olsun, Angular ile ilgili temel kavramları ve ileri seviyedeki uygulamaları öğrenmek isteyen herkes bu eğitimlere katılabilir.

Outline

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 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

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

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 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 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 Test