

# OSDE-13 - ANGULAR MASTER CLASS

Categoria: **Development**

## INFORMAZIONI SUL CORSO



**Durata:**  
5 Giorni



**Categoria:**  
Development



**Qualifica Istruttore:**  
Docente Senior (min.  
5 anni)



**Dedicato a:**  
Sviluppatore



**Produttore:**  
PCSNET

## OBIETTIVI

Sviluppare Front-End e Single Page Application in Angular e Typescript utilizzando un approccio component-based, pattern e metodologie consolidate.

Al termine del corso i partecipanti saranno in grado di:

- Effettuare: binding, styling, forms, comunicare con il server, pipes, dependency injection
- Sviluppare Single Page Applications con un approccio totalmente component-based
- Sviluppare applicazioni multi-view con angular router, autenticazione JWT e gestione della sicurezza
- Creare direttive e componenti
- Organizzare progetti, applicare metodologie e best practices
- Scrivere codice scalabile, modulare ed integrare gli Unit Test

## PREREQUISITI

Per partecipare è necessario avere confidenza con la maggior parte degli argomenti trattati durante il corso Angular base e in particolare avere una certa familiarità con l'uso di Javascript, TypeScript, HTML, CSS, programmazione ad oggetti (OOP).

## CONTENUTI

### Day 1: Angular e TypeScript

#### **ES2015 e TypeScript**

- Introduzione a Javascript ES2015 e TypeScript
- Creazione e utilizzo dei moduli
- Array, oggetti e stringhe: le novità introdotte
- Fat arrow syntax, Template string, Destructuring
- Classi e Decorators
- Types e interfaces in TypeScript
- Utilizzo di Angular-Cli per la creazione di progetti Angular

#### **Dynamic Styling**

- Inline style e css "unit" suffix
- ngClass, ngStyle, styleUrls

- Animazioni con CSS Transition
- ViewEncapsulation e Shadow DOM

### Angular Fundamentals

- Creazione progetti con Angular-Cli
- Approccio Component-Based
- Il decorator @Component
- Binding and templates
- Structural Directives: \*ngIf, \*ngSwitch, \*ngFor
- Creazione componenti data-driven

### Custom Components

- Creazione componenti riutilizzabili
- Input properties
- Transclusion
- Components communication and Output emitter
- Reusable components

### Day 2: Form & Services

#### Services & Dependency Injection

- Dependency Injection in Angular
- Separazione delle competenze
- Classi e custom types in TypeScript
- Singleton e data model
- @Injectable e provider
- Organizzare servizi RESTful

#### Server-Side Communication

- Installazione e configurazione di un JSON mock server
- Utilizzo di REST API e il modulo @angular/http di Angular
- Utilizzo del servizio http per la comunicazione con il server
- Scenari e casi d'uso con form e operazioni CRUD
- Gestione data Model

#### Forms

- Angular Forms: 1-way and 2-way binding
- Local Properties
- ngModel, ngForm e ngSubmit
- Form validators
- Gestione messaggi di errori, dirty e touch state

#### Working with DOM

- @ViewChild e accesso agli elementi del DOM
- DOM Native Elements
- RxJS Observable and nativeElement

#### @Pipes

- Utilizzo dei filtri / pipes su date, numeri e stringhe

- Creazione Custom pipes

### **Day 3: Mastering Angular**

#### **Mastering @Components**

- 1-way data flow architectures e stateless components
- Ciclo di vita del componente
- Change Detection Strategies
- Nested component e differenti tecniche di comunicazione
- Angular animations
- Gestione errori ed eccezioni
- ViewChildren vs ContentChildren
- Integrazione 3rd party libraries / types

#### **Mastering Dependency Injection**

- forRoot() e custom modules
- @Injectable e provider
- @Inject and useValue
- OpaqueToken
- Configurare provider con useFactory

#### **Unit Test**

- Unit Test vs E2E
- Introduzione a Jasmine
- Introduzione e configurazione Karma
- Introduzione a Protractor
- Utilizzare TestBed
- Testare DOM e templates
- Mocking dependencies
- Spies

#### **Mastering @Directives**

- Structural directives vs attribute directives
- @HostBinding
- @HostListener
- Setter e Getter
- Structural directives: tag e asterisco \*
- Utilizzo di ElementRef, TemplateRef
- Creazione componenti a runtime con ComponentFactoryResolver e ViewContainerRef
- Integrazione 3rd party API (GSAP Tweenmax)

#### **Reusable Components**

- Si esplorano differenti tecniche, casi d'uso e approcci per la creazione di decine di componenti e direttive, per ognuno dei quali saranno integrati e descritti i relativi Unit Test:
- Maps: integrazione Google Map API e Leaflet
- Grid e layout components
- Integrazione HighChart
- Animate components with GSAP Tweenmax
- Typography, List, TabBar, Card

-Collapsable, Side Panel, Divider e molto altro

#### **Day 4: Real World Apps**

##### **Advanced Forms**

- Forms: custom validators @Directive
- Forms: creazione form multi-step
- Introduzione ai Reactive Forms

##### **Navigazione e Routing**

- MultiView Single page application in Angular
- Utilizzo di @angular/router
- Gestione stati e routes
- Passaggio parametri
- Route Navigation components
- Template injection
- protezione view (by token) con AuthGuard e canActivate
- Router Events

##### **Gestione Moduli**

- Applicazioni modulari in Angular con ngModule
- Organizzazione progetto
- Best practices

#### **Day 5: Hands-On**

##### **Auth e Security**

- Autenticazione e procedure di login
- JWT: login and token
- Utilizzo REST API protette da token
- Http "interceptors" in Angular
- Gestione errori
- CORS, preflight e OPTIONS

I partecipanti avranno la possibilità di fare molta pratica seguiti costantemente dal docente e si avrà modo di approfondire moltissime tematiche trattate durante le giornate precedenti.

## **INFO**

**Materiale didattico:** Materiale didattico e relativo prezzo da concordare

**Costo materiale didattico:** NON incluso nel prezzo del corso

**Natura del corso:** Operativo (previsti lab su PC)