

# OSDE-14 - ANGULAR DATA ARCHITECTURES: REDUX E RXJS 5

Categoria: **Development**

## INFORMAZIONI SUL CORSO



Durata:  
3 Giorni



Categoria:  
Development



Qualifica Istruttore:  
Docente Senior (min.  
5 anni)



Dedicato a:  
Sviluppatore



Produttore:  
PCSNET

## OBIETTIVI

Imparare a sviluppare architetture dati e gestire flussi di dati complessi in Web Applications con Redux e RxJS 5 in Angular.

Angular Data Architectures è un corso diviso in 3 parti sulle più recenti ed avanzate tecnologie in merito alla creazione di architetture dati solide e scalabili in contesti Angular enterprise e non.

I partecipanti verranno preparati su RxJS5 ed Observables per sfruttare al meglio RxJS in scenari sia comuni che avanzati nello sviluppo di Web Applications quali gestione input utente, richieste multiple da e verso server dipendenti tra loro e molto altro.

Tramite Redux verranno inoltre esplorate tecniche per una gestione dello stato applicativo chiara, performante, prevedibile e scalabile. Man mano che un'applicazione cresce, come si garantisce che una variazione di stato in un modulo sia consistente e accurata in altri moduli indipendenti? Cosa succede se tali modifiche risultano in altre modifiche?

Programma tarato sul livello di conoscenza dei partecipanti

Preparazione allo sviluppo funzionale e reattivo in contesti enterprise Angular e Typescript

Comprensione nozioni in dettaglio su RxJS, Observables e Redux.

## PREREQUISITI

Per partecipare è necessario avere confidenza con:

- la maggior parte degli argomenti trattati durante i corsi Angular Base, Avanzato e Master Class
- concetto di stato applicativo di un'applicazione
- strumenti Git e npm v.3+

## CONTENUTI

### Day 1: Reactive Programming in RxJS 5

#### Reactive Programming In RxJS

- Concetti core Reactive Programming, differenze e use cases in relazione ad altri paradigmi
- Comprensione in dettaglio di Observables e Observers
- Creazione di custom Observables ed Observers in comuni casi d'uso
- Utilizzo degli operators più comuni e scenari d'uso per ciascuno
- Gestione eventi asincroni comuni ed avanzati quali HTTP

- Esercizi sulle basi di RxJS con tips & tricks

## Real-World RxJS

- Debugging in RxJS
- Combinazione di Observables ed operators
- Subjects e multicasting: cosa sono, differenze tra i diversi tipi di Subjects e casi d'uso per ciascuno
- BehaviourSubject in dettaglio: perché viene tanto usato e come evitare errori più comuni
- Studio problemi comuni nelle Web Applications con relative soluzioni in RxJS

## Angular & RxJS

- Integrazione RxJS in progetti Angular e Typescript
- Aspetti reactive del framework Angular: Http, Reactive Forms, Router, Componenti
- State management e application flow con RxJS
- Miglioramento performance di un'applicazione Angular con RxJS
- Build e distribution di un'applicazione Angular con RxJS

## RxJS Avanzato

- Unit testing con RxJS
- Gestione WebSockets in RxJS
- Dietro le quinte di RxJS: come è implementato
- Utilizzo di uno Scheduler in contesti ad alto carico di lavoro (livello alto avanzato)

## Day 2: Angular Data Architecture in Redux

### Concetti Core

- Problemi di comunicazione tra componenti Angular
- Gestione del data-flow deterministico, scalabile ed esplicito
- Caratteristiche del pattern Flux e come Redux evolve da esso
- Vantaggi dell'unidirezionalità
- Confronto con MVC
- Integrazione con Angular
- Pattern Smart e Dumb Components
- Come mantenere lo stato applicativo e Redux indipendente dai componenti
- Integrazione Redux + Typescript: tipizzazione dello stato applicativo

### Integrazione Redux e RxJS

- ChangeDetectionStrategy: come migliorare le performance dei componenti tramite Redux
- Utilizzo di librerie Redux in Angular: Angular-Redux (precedentemente denominato ng2-redux) e ngrx. Differenze e come scegliere consapevolmente.

### Real-World Redux con Unit Testing

- Principi fondamentali
- Action e action creator con unit testing
- Reducers con unit testing
- Caratteristiche dello Store immutabile
- Comporre lo stato iniziale e l'avvio dell'app
- Vantaggi dello Single state tree
- L'importanza dell'Immutabilità

## Redux Avanzato

- Composizione di reducers
- Redux Middlewares
- Side effects con redux-observable e ngrx/effects
- State selectors, memoization e computed data
- Accenni server-side rendering con Redux e Angular

## Day 3: Real World Application

Analisi di una applicazione reale per individuare gli aspetti legati all'utilizzo di RxJS e Redux

### Analisi e Risoluzione

- Analisi iniziale dell'applicazione prima di Redux + RxJS e studio dei problemi di stato applicativo e data-flow
- Risoluzione dei problemi tramite puro RxJS in Angular
- Analisi pro e contro puro RxJS
- Integrazione Redux con Observable Store nell'applicazione
- Analisi struttura dello stato applicativo: errori e design patterns più comuni
- Analisi struttura di un progetto Angular con Redux

### Operazioni

- Operazioni CRUD in Redux
- Creazione action creators e reducers con unit testing
- Creazione componenti smart & dumb connessi a Redux con unit testing della parte Redux
- Utilities e tips per la modifica dello stato applicativo in maniera concisa, performante e dichiarativa
- Gestione richieste HTTP al server
- Gestione validazione server ed optimistic updates con eventuale rollback
- Gestione autenticazione JWT in Redux
- Persistenza dello stato applicativo

## 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)