

Certifiante Sessions garanties disponibles

Formation Angular : développement avancé et performance – certification en option

Réf. SF33146

Approfondissez votre maîtrise d'Angular pour architecturer des applications complexes, performantes et maintenables avec Signals, NgRx et le mode Zoneless ! En option, validez vos acquis avec la certification Angular Mid-level Developer.

3 j Durée	2 250 € HT Inter / participant	4 500 € HT Intra (≤12 pers.)	13 Sessions à venir
--------------	-----------------------------------	---------------------------------	------------------------

Objectifs pédagogiques

- ✓ Architecturer une application Angular moderne avec standalone components et injection de dépendances avancée
- ✓ Implémenter le mode Zoneless et optimiser la détection de changements
- ✓ Maîtriser les Signals avancés (writable, computed, linkedSignal, inputs signalés) et leur interopérabilité avec RxJS
- ✓ Gérer l'état applicatif avec NgRx et les API standalone (provideStore, provideEffects)
- ✓ Mettre en œuvre le rendu côté serveur (SSR) avec hydration incrémentale
- ✓ Sécuriser et optimiser le routage avec guards fonctionnels, resolvers et lazy loading avancé
- ✓ Écrire des tests avancés avec Jest/Vitest, TestBed et marble testing

Public visé & prérequis

Public visé Architectes logiciels, Développeurs

Prérequis- Expérience pratique de développement avec Angular (minimum 3-6 mois de projet) - Bonne maîtrise de TypeScript (génériques, types utilitaires) - Connaissance des bases de RxJS (Observables, pipe, subscribe) - Notions de Signals Angular (signal, computed, effect) - Pratique des tests unitaires (Jasmine ou équivalent)

Programme détaillé

Architecture moderne et standalone components

- Architecture applicative Angular : organisation d'un projet standalone-first, suppression des NgModules et structuration par feature
- Injection de dépendances avancée : InjectionTokens personnalisés, providers hiérarchiques, stratégies d'encapsulation des services
- Workspace Nx : introduction au monorepo, génération de projets et gestion des dépendances entre librairies
- Composants avancés : projection de contenu (ng-content), vues dynamiques (NgComponentOutlet), décorateurs @ViewChild et @ContentChild

Activités pratiques :

- Mise en place d'un projet Angular moderne avec architecture standalone et injection avancée
- Création d'un composant générique réutilisable avec projection de contenu

Zoneless et détection de changements

- Zone.js : rôle historique dans la détection de changements et limites de performance
- Mode Zoneless : architecture sans Zone.js pour un rendu plus rapide et un contrôle explicite des mises à jour
- Stratégies de détection : Default vs OnPush, utilisation de ChangeDetectorRef pour le contrôle manuel
- Optimisation des cycles de rendu : identification et suppression des re-rendus inutiles, impact sur les performances réelles

Activités pratiques :

- Migration d'une application utilisant Zone.js vers le mode Zoneless

- Profiling de la détection de changements avec Angular DevTools et comparaison des stratégies

Signals avancés et interopérabilité

- Writable signals : gestion de l'état local des composants avec signal(), set(), update()
- Computed signals : réactivité fine avec computed(), dépendances automatiques et évaluation paresseuse
- Effects et linkedSignal : effect() pour les effets de bord, linkedSignal pour la synchronisation d'état dérivé
- Inputs signalés : communication parent/enfant via input signals comme alternative aux décorateurs @Input classiques
- Interopérabilité RxJS/Signals : toSignal(), toObservable(), stratégies de migration progressive depuis les Observables

Activités pratiques :

- Réécriture d'un composant basé sur des Observables en utilisant exclusivement les Signals
- Implémentation d'une communication inter-composants complexe avec inputs et outputs signalés
- Migration progressive d'un service RxJS vers un pattern hybride Signals/RxJS

RxJS avancé et patterns réactifs

- Subjects spécialisés : BehaviorSubject, ReplaySubject et AsyncSubject — cas d'usage et choix du bon type
- Opérateurs avancés : switchMap, mergeMap, concatMap, exhaustMap pour l'orchestration de flux asynchrones
- Combinaison de flux : combineLatest, forkJoin, withLatestFrom pour synchroniser des sources de données multiples
- Gestion des souscriptions : stratégies d'unsubscribe, takeUntilDestroyed et bonnes pratiques pour éviter les fuites mémoire
- Marble testing : tests d'Observables avec la syntaxe marble pour valider les comportements asynchrones

Activités pratiques :

- Implémentation d'un service réactif avec orchestration de flux complexes (recherche avec debounce et annulation)
- Écriture de marble tests pour valider le comportement d'un opérateur personnalisé

Gestion d'état avec NgRx

- Architecture NgRx : Store, Actions, Reducers et le flux unidirectionnel de données
- Effects : gestion des effets de bord (appels HTTP, navigation) de manière déclarative
- Selectors avancés : composition de sélecteurs, mémoïsation et optimisation des performances
- API standalone : provideStore() et provideEffects() pour une intégration sans NgModule
- NgRx et Signals : utilisation conjointe de NgRx pour l'état global et des Signals pour l'état local des composants

Activités pratiques :

- Mise en place d'un store NgRx complet pour gérer l'état d'une feature applicative
- Intégration de NgRx avec les Signals pour un pattern hybride global/local

Routage avancé et sécurisation

- Configuration avancée du Router : provideRouter() avec options, secondary routes et named outlets
- Guards fonctionnels : CanActivate, CanActivateChild, CanDeactivate sous forme de fonctions standalone
- Resolvers : pré-chargement de données avant la navigation pour une expérience utilisateur fluide
- Lazy loading avancé : loadComponent(), loadChildren(), stratégies de pré-chargement et redirections asynchrones

Activités pratiques :

- Sécurisation d'une application multi-rôles avec guards fonctionnels et redirections conditionnelles
- Implémentation d'un resolver pour pré-charger les données d'une page de détail

Server-Side Rendering et hydration

- SSR Angular : rendu côté serveur activé par défaut dans les nouveaux projets, configuration et bénéfices
- Hydratation incrémentale : réactivation progressive des composants côté client pour un temps d'interactivité optimal
- Event replay : capture et rejeu des interactions utilisateur pendant la phase d'hydratation
- Prérendu statique : génération de pages HTML statiques pour les contenus peu dynamiques

Activités pratiques :

- Mise en place d'une application Angular avec SSR et hydratation incrémentale
- Configuration du prérendu statique pour les pages publiques d'une application

Tests avancés et optimisation des performances

- Jest ou Vitest : configuration comme runner de tests moderne en remplacement de Karma
- TestBed avancé : tests de composants standalone, mocking de services et de HttpClient, tests asynchrones
- Marble testing : validation du comportement des Observables et des flux NgRx avec la syntaxe marble
- Optimisation de production : tree-shaking, bundle analysis, directives @defer pour le chargement conditionnel
- Profiling : utilisation d'Angular DevTools pour identifier les goulots d'étranglement et mesurer l'impact des optimisations

Activités pratiques :

- Écriture de tests avancés pour un composant intégrant NgRx et des Signals
- Audit de performance d'une application et mise en œuvre d'optimisations ciblées

Prochaines sessions

DATE	LIEU	STATUT
27/10/2026	Paris	Sur confirmation
27/10/2026	à distance	Sur confirmation
27/10/2026	Lille	Sur confirmation
27/10/2026	Nantes	Sur confirmation
16/11/2026	Lyon	Sur confirmation
16/11/2026	à distance	Sur confirmation

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

Note moyenne des participants de la filière Développement Web & Fullstack : 4,9/5 — 39 avis

Maxime V. — session très agréable, dense mais toujours agréable. formateur très à l'écoute, prends le temps de répondre aux questions. on apprend l'utilisation des fonctions mais aussi les processus mis...

Cécile L. — Très bon instructeur qui explique toujours en montrant du code, aucune slide et une présentation vivante ce qui permet de rester intéressé et concentré tout au long !...

Votre formateur

Romain R. — Formateur React & JavaScript

Romain intervient sur des formations portant sur le développement web et fullstack. Parmi les formations qui lui sont confiées figurent notamment « C# » et « Java ». Son activité...

Ils nous font confiance

Cette année notre filière Développement Web & Fullstack a accompagné plus de 111 entreprises dont :

ARPEGE APSIDE TOP CEGEDIM CONSERVATOIRE NATIONAL DES ARTS ET METIERS ECONOCOM APPS, CLOUD & DATA ORANGE BUSINESS
ORANGE BUSINESS SERVICES SA SOPRA STERIA ADVENTIEL AGATE AGENCE ALPINE DES TERRITOIRES +101 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com