

Programme de formation Angular

• Objectifs

Notre formation sur le framework Angular permet une montée en compétences rapide et pérenne. Grâce à une méthode pédagogique moderne alternant mises en pratique, retours d'expérience et échanges en continu, les participants acquièrent les bons réflexes pour développer sereinement et de façon autonome avec Angular. A l'issue des 3 jours de formation, ils sont capables de concevoir des applications propres et robustes quel que soit leur contexte professionnel. Les nouveaux standards du web (TypeScript, ES6 et les Web Components notamment) sont abordés via des exemples concrets de mises en application, pour faciliter l'assimilation des bonnes et mauvaises pratiques. Un support de cours reprenant tous les points importants de la formation sera remis à chacun des apprenants.

• Pré requis

Connaissances en JavaScript, HTML et CSS.

• Durée

3 jours

• Public

Architectes, Chefs-de-projet, Consultants, Développeurs

• Plan de formation

Introduction à la formation Angular

Présentation générale et objectifs de cette formation Angular

Décryptage de la philosophie voulue par Google

Les apports d'Angular dans le monde du développement web

Versionning : d'AngularJS à Angular

Le langage TypeScript, Babel et les versions standardisées de JavaScript (EcmaScript 2015, ES7, ES2018...) : ce qu'il faut savoir avant de démarrer sur Angular

Comprendre la notion de Web components du W3C (custom elements, shadow DOM, HTML templates et imports)

Atelier : Présentation du formateur et des stagiaires, échanges sur les différents contextes professionnels et perspectives d'application des acquis de la formation.

Premier pas : comprendre la structure du framework

Concepts fondamentaux et vue d'ensemble de l'architecture : composants, services, vues...

Comment est organisée une application Angular ?

Organisation du code avec les modules : les conteneurs NgModules et l'encapsulation

Templates, les nouvelles syntaxes

Fonctionnement du databinding pour gérer les données dynamiques

Rôles des pipes et directives

Principes de l'injection de dépendances avec Angular

Les packages à installer, les nouvelles fonctionnalités d'Angular CLI et l'intérêt de Webpack

Exemples de cas pratiques : Configuration de l'environnement de développement, installation des outils et création d'une première application « from scratch » avec Angular CLI (en version stable depuis Angular 4).

Composants, templates et liaison de données (databinding)

Cycle de vie et structure des Angular components

Démarrer un projet et créer une hiérarchie de composants

Les mécanismes d'interaction entre composants

Annotations et décorateurs, leurs différences

et propriétés

Ajouter des styles aux composants (comprendre la ViewEncapsulation)
L'interpolation et les différents binding de données (propriétés et événements)
Le two-way binding dans un composant
Travailler avec les Angular Elements
Exemples de cas pratiques : Création de composants Angular, ajout de styles et mise en oeuvre des différents types de databinding., transformation de composants en custom elements.

Directives et pipes

Directives structurelles (ngFor, ngIn et les ngSwitch)
Directives par attribut (ngModel, ngStyle, ngClass...)
Construire ses propres directives
Utiliser les pipes natifs (DatePipe, UpperCasePipe, CurrencyPipe...)
Créer ses propres pipes
Le pipe keyValue
Exemples de cas pratiques : Utilisation de directives, manipulation des pipes (paramètres, notion de Change Detection) pour enrichir ses composants.

Services et injection de dépendances

Retour sur le système d'injection de dépendances
Créer un service basique
Les injecteurs et instances dans Angular
Enregistrer un service : les différentes options
Déclarer des services avec les tree-shakable providers
Exemples de cas pratiques : Développement et déclarations de services personnalisés.

Formulaires

Les différentes façons de créer des formulaires : template-driven ou reactive ?
Comprendre la gestion du flux de données
Valider et gérer les erreurs : considérations pour la sécurité
Créer des validateurs personnalisés et les ajouter à un formulaire
Exemples de cas pratiques : Création d'un formulaire selon les deux approches possibles, enregistrement des contrôles et mise en oeuvre de tests.

Les Observables et la bibliothèque RxJS

Concepts de programmation réactive
Angular et RxJS (Reactive Extensions for JavaScript)
Comprendre et utiliser les observables et observateurs
L'intérêt des Subjects
Les principaux opérateurs : map(), filter(), throttleTime(), scan() et reduce()
Gestion des requêtes HTTP avec l'API HttpClient
Exemples de cas pratiques : Utilisation des observables et comparaison avec d'autres techniques (promises, event handlers, arrays), interactions avec un serveur Firebase.

Routing et navigation

Vue d'ensemble du routage Angular
Déclarer et configurer des routes et URLs
La navigation avec routerLink et navigate
Paramètres de routes
Gérer les redirections
Configurer des guards
Exemples de cas pratiques : Définition des routes et navigation dans une application.

Tests

Configurer l'environnement de test
Karma et Jasmine
Utiliser Protractor pour des tests end-to-end
Les bonnes pratiques pour tester sous Angular
Exemples de cas pratiques : Tests unitaires et aperçu de la mise en oeuvre d'une démarche TDD (Test Driven Development)

Progressive Web Apps, Service Workers, Schematics : nouveautés Angular concepts avancés

Focus sur les nouveautés et changements apportés par la dernière version
Qu'est-ce qu'une PWA ?
Introduction aux services workers, pris en charge depuis Angular 5
UI Design avec Angular Material et le Component Dev Kit (CDK)
Migration automatique avec ng update
ng add et l'utilisation des Schematics
Composants avancés
Zones
Compilation Ahead of Time (AoT)



Le lazy-loading

Exemples de cas pratiques : Ajout de service workers, démonstration de l'implémentation d'une PWA avec Angular.