

Angular 18 et antérieures - Développement d'applications Web

Référence : OPS006C

Durée : 3 jours (21 heures)

Certification : Aucune

Connaissances préalables

- Avoir une bonne connaissance des langages du Web comme HTML, CSS et JavaScript.

Profil des stagiaires

- Développeurs et chefs de projets

Objectifs

- Utiliser la version 18 du framework Angular
- Développer et tester complètement une application
- Appliquer les bonnes pratiques de développement

Certification préparée

- Aucune

Méthodes pédagogiques

- 6 à 12 personnes maximum par cours, 1 poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions

Formateur·rice

- Consultant-Formateur expert Développement Internet

Méthodes d'évaluation des acquis

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation des compétences acquises envoyée au stagiaire
- Attestation de fin de stage adressée avec la facture

Contenu du cours

1. Jour 1 - Matin

-

2. Architecture, installation et premier test

- Architecture typique d'une application Angular
- Installation minimale d'Angular
- Test d'un code simple

3. TypeScript

- TypeScript : le JavaScript typé
- Les types, classes et modules
- L'asynchrone et les promesses

4. Rôle majeur des composants

- Définition d'un composant Angular
- Décorateurs
- Les vues
- Le modèle
- Exemples de travaux pratiques : Création d'une application et analyse du code généré

5. Jour 1 - Après-midi

-

6. Une première application Angular

- Démarrer "from scratch" avec Angular CLI
- Création d'un premier projet
- Exemples de travaux pratiques : Avec Angular CLI, création d'une application et modification des composants

7. Les templates

- Utiliser l'interpolation
- Property et event bindings
- Utiliser des variables locales
- Utilisation des pipes
- Nouvelle syntaxe @if, @for, @switch plus intuitive

8. Les services

- Fournir des fonctionnalités transversales avec les services
- Création de services
- Injection de services (par constructeur, par la fonction inject())
- Exemple de travaux pratiques : Travaux d'interpolation sur diverses zones d'affichage en relation avec les données métier alimentées via des services

9. Jour 2 - Matin

-

10. Les formulaires

- Création de formulaires avec : Le FormsModule, Le FormsBuilder
- Validation et gestion des erreurs
- Nouvelle API pour créer des formulaires fortement typés
- Les types FormGroup et FormRecord
- Exemples de travaux pratiques : Ajout de formulaires d'affichage et de saisie en utilisant FormsModule et FormBuilder

11. Jour 2 - Après-midi

-

12. Rôle de RxJS (Reactive extensions for JavaScript)

- Présentation des flux de données asynchrones
- Propagation des changements avec RxJS

13. Travail avec HTTP

- Le service HTTP
- Communication avec une API en asynchrone
- Exemples de travaux pratiques : Appel asynchrone à des API externes, intégration à un service Angular

14. Jour 3 - Matin

-

15. Le routage

- Les différentes versions du module de routage
- Fonctionnement du routage
- Configurer des routes et utiliser les directives
- Exemple de travaux pratiques : Ajout des tables de routage dans l'application Angular

16. Les "standalone components"

- Présentation du concept des composants autonomes
- Notion de SCAM (Single Angular Component Module)
- Les avantages de l'utilisation de composants autonomes
- Présentation des meilleures pratiques pour créer et utiliser des composants autonomes
- Exemple de travaux pratiques : Développement d'une application exploitant les standalone components

17. Jour 3 - Après-midi

- Présentation de Jasmine et Karma
- Exemple de travaux pratiques : Utilisation d'un outil de test

18. Les signaux

- Introduction
- Qu'est-ce qu'un signal ?
- L'importance des signaux
- Comparaison avec les Observables
- Syntaxe de base
- Exemple de travaux pratiques : Développement d'une application exploitant les signaux

19. Les tests

- Présentation de Jasmine et Karma
- Tester des composants
- Tester des services
- Les tests fonctionnels "end-to-end" avec Cypress, Playwright
- Exemple de travaux pratiques : Implémentation de tests unitaires et fonctionnels

Notre référent handicap se tient à votre disposition au [01.71.19.70.30](tel:01.71.19.70.30) ou par mail à referent.handicap@edugroupe.com pour recueillir vos éventuels besoins d'aménagements, afin de vous offrir la meilleure expérience possible.