**Programme de Formation Angular 18**

**Module 1: Introduction à Angular**

* **Introduction à Angular**
  + Qu'est-ce qu'Angular?
  + Historique et versions d'Angular
  + Architecture d'Angular : Modules, Composants, et Services
  + Configuration de l'environnement de développement
  + Introduction à Angular CLI
* **Installation et Configuration**
  + Installation de Node.js et Angular CLI
  + Création d'une nouvelle application Angular
  + Structure des dossiers et fichiers dans une application Angular
  + Introduction au fichier angular.json

**Module 2: Les Bases d'Angular**

* **Composants Angular**
  + Création d'un composant
  + Structure d'un composant (template, style, et logique)
  + Communication entre les composants (Input, Output)
  + Cycle de vie des composants
* **Templates et Data Binding**
  + Interpolation et Property Binding
  + Event Binding
  + Two-Way Data Binding avec ngModel
  + Directives structurelles (\*ngIf, \*ngFor, \*ngSwitch)
  + Directives d'attributs ([ngStyle], [ngClass])
* **Services et Dependency Injection**
  + Introduction aux services
  + Création et utilisation d'un service
  + Injection de dépendances dans Angular
  + Le rôle des providers

**Module 3: Gestion des Formulaires**

* **Formulaires Template-Driven**
  + Création d'un formulaire
  + Validation des formulaires
  + Gestion des erreurs et affichage des messages d'erreur
  + Utilisation de ngForm, ngModel, et les directives associées
* **Formulaires Reactive**
  + Introduction aux formulaires réactifs
  + Création d'un formulaire réactif avec FormBuilder
  + Validation réactive et gestion des états des champs
  + FormGroup, FormControl, et FormArray

**Module 4: Gestion des Données et HTTP**

* **HTTP Client et API REST**
  + Introduction à HttpClientModule
  + Envoi de requêtes HTTP (GET, POST, PUT, DELETE)
  + Gestion des Observables avec RxJS
  + Traitement des erreurs HTTP
* **RxJS et Programmation Réactive**
  + Introduction à RxJS
  + Concepts de base : Observable, Observer, et Subscription
  + Opérateurs RxJS courants : map, filter, mergeMap, switchMap
  + Manipulation des flux de données asynchrones

**Module 5: Routage et Navigation**

* **Routage de base**
  + Configuration des routes dans Angular
  + Navigation entre les routes
  + Paramètres de route et query parameters
  + Protection des routes avec AuthGuard
* **Routage avancé**
  + Chargement paresseux (Lazy Loading) des modules
  + Préchargement des modules
  + Utilisation des routes enfants

**Module 6: Gestion de l'État avec NgRx**

* **Introduction à NgRx**
  + Concepts de base : Store, Actions, Reducers, Selectors
  + Configuration de NgRx dans un projet Angular
  + Création et gestion du Store
  + Utilisation d'Effects pour gérer les effets secondaires

**Module 7: Tests Unitaires et End-to-End**

* **Tests unitaires avec Jasmine et Karma**
  + Introduction à Jasmine et Karma
  + Écriture de tests pour les composants, services, et directives
  + Test des formulaires et des services HTTP
* **Tests End-to-End avec Protractor**
  + Introduction à Protractor
  + Écriture de tests E2E pour les composants et le routage
  + Configuration et exécution des tests

**Module 8: Optimisation et Déploiement**

* **Optimisation de la Performance**
  + Optimisation du chargement des modules
  + Compression et minimisation des bundles
  + Utilisation de Change Detection Strategy
* **Déploiement d'une Application Angular**
  + Build et préparation pour la production
  + Déploiement sur Firebase, Netlify, ou un serveur web
  + Configuration d'un serveur pour le déploiement (Nginx, Apache)

**Module 9: Projet Final**

* **Développement d'une application complète**
  + Définition des exigences du projet
  + Mise en œuvre des concepts appris
  + Présentation du projet et retour d'expérience

**Durée et Méthodologie**

* **Durée:** Le programme peut être étalé sur 8 à 10 semaines, en fonction du rythme d'apprentissage des participants.
* **Méthodologie:** Chaque module doit comprendre des sessions théoriques suivies de travaux pratiques. Les participants doivent être encouragés à poser des questions et à résoudre des exercices pour renforcer leur compréhension.

**Matériel Requis**

* Ordinateur avec un environnement de développement (VS Code, Node.js, Angular CLI)
* Accès à une API REST pour les exercices pratiques (ou la création d'une API fictive)
* Documentation officielle Angular et ressources en ligne pour approfondir les sujets