Formation Angular v14

Des bases solides pour développer des applications avec Angular

N° ACTIVITÉ : 11 92 18558 92 | TÉLÉPHONE : 01 85 77 07 07 | E-MAIL : inscription@hubformation.com

Angular est un des plus anciens et des plus répandus des frameworks frontend moderne. C'est un framework très complet, pensé pour être productif, tout en étant pensé pour maintenir les applications sur le long terme.

Référence	ANGLR
Durée	5 jours (35h)
Tarif	2 500 €HT

Objectifs

| Développer une application Angular
| Mettre en place des tests pour garantir sa maintenabilité dans le temps
| Gérer correctement l'état de l'application

Public

| Développeurs frontend
| Développeurs Full-stack type JavaScript / Node.js
| Développeur backend type Java intéressé par le développement Angular front-end
| Développeur web intéressé par le développement d'une application Angular
| Tout public intéressé par le développement d'une application Angular

Prérequis

| Connaissances ECMAScript 6: POO, classes, arrow function, scopes - en particulier const/let et this en JS, Promise
| Connaissances TypeScript aideront
| Connaissances RxJS aideront

SESSIONS PROGRAMMÉES

A DISTANCE (FRA)

du 16 au 18 septembre 2024
du 16 au 18 octobre 2024
du 20 au 22 novembre 2024
du 18 au 20 décembre 2024

VOIR TOUTES LES DATES

Programme de la formation

Introduction
| Introduction du cours
| Présentation des speakers et tour de table
| Objectifs et agenda de la formation
| Présentation du Framework Angular (historique, status, versions, outils, philosophie...), de l'écosystème, des uses cases, les points forts et différenciants

Initiation à TypeScript
| Présentation de TypeScript
| Superset d'ECMAScript 6
| Les classes en TypeScript
| class vs interface vs type
| Comprendre les types génériques
| Configuration du mode strict
| Labs : Premiers pas avec TypeScript

Initiation à RxJS
| Présentation de RxJS
| Promise vs Observables
| Logique de stream avec les Observables
| Pipe et opérateurs de base (map, filter, mergeMap, switchMap, debounceTime, distinctUntilChanged, tap), multiplexing (share) et factory (of, combineLatest)

| Conserver le stream
| Labs en TDD : Premiers pas avec RxJS

Une architecture à base de composants et modules
| Architecture d'une application Angular
| Command Line Interface (CLI)
| Structure d'un composant
| Présentation de l'utilisation de la directive *ngFor
| Labs : Créer un projet Angular avec la CLI
| Créer son premier composant Angular
| Mettre en oeuvre une première directive : *ngFor

Exploration des composants
| Le cycle de vie d'un composant
| Data Binding : associer les données du contrôleur et la vue
| Events : réagir aux actions de l'utilisateur(click, saisie, etc...)
| Faire communiquer des composants entre eux
| Labs en TDD : Data binding, Events, Components communication

Rendre son application dynamique
| Ajouter de la navigation entre plusieurs page via un routeur
| Utiliser fetch pour récupérer des données côté serveur
| Labs : Navigation, Server communication (with fetch)

L'injection de dépendance
| Providers et dependancy injection : comprendre l'Inversion of Control en Angular
| Introduction de la fonction inject
| Récupérer des données via HttpClient
| Labs en TDD : Providers et dependancy injection, Fonction inject, Server communication (with HttpClient)

Au delà des composants
| Utiliser les pipes pour formater les données dans la vue
| Utiliser des directives pour augmenter les composants
| Labs en TDD : Pipe, Directive

Les formulaires template driven
| Template driven forms : gérer ses formulaires par des contraintes côté template
| Labs : Template driven forms

Les formulaires réactifs
| Reactive form : gérer ses formulaires côté contrôleur
| CustomFormControl : créer sa propre brique de formulaire
| Labs en TDD : Reactive form, CustomFormControl

Navigation avancée
| Charger dynamiquement une partie de l'application via du lazy loading
| Le contrôle d'accès aux pages via les guards
| Labs : LazyLoading, Guards

Interceptors
| Agir sur toutes les requêtes http via interceptors
| Labs en TDD : Interceptors

Le template en détail
| Quelques éléments avancés pour gérer le template de son composant
| Agir sur l'élément sur lequel est attaché la directive via Host Binding
| Manipuler la structure du DOM avec les Structural Directive
| Labs en TDD : Templating, Host Binding, Structural Directive

State management avec Ngrx
| Gérer l'état de son application avec Ngrx
| Labs en TDD : State management avec Ngrx

State management avec Ngxs
| Gérer l'état de son application avec Ngxs
| Labs en TDD : State management avec Ngxs

HUB Formation - 1-7, Cours Valmy - 92923 Paris - La Defense
01 85 77 07 07 - www.hubformation.com - SIREN 525 287 785

State management avec Elf
| Gérer l'état de son application avec Elf
| Labs en TDD : State management avec Elf

Détection de changement
| Comment agir sur la détection de changement d'Angular ?
| Labs : Change detection

Tester une application Angular de bout en bout
| Concept : tests unitaires VS tests de bout en bout (e2e)
| Faire un test e2e avec Cypress
| Labs : tests e2e

Modern Angular
| Standalone components ou comment se passer des NgModules
| Se passer des classes et utiliser des fonctions pour les guards avec canMatch **NEW**
| Se passer des classes et utiliser des fonctions pour les interceptor
| Labs : Standalone components, Functional guards / canMatch, Functional interceptor

Progressive Web App
| PWA : comment donner un côté plus natif à nos applications
| Labs : PWA

SSR
| Server Side Rendering : servir nos applications sans JavaScript côté client
| Labs : Server Side Rendering

Méthode pédagogique

Chaque participant travaille sur un poste informatique qui lui est dédié. Un support de cours lui est remis soit en début soit en fin de cours. La théorie est complétée par des cas pratiques ou exercices corrigés et discutés avec le formateur. Le formateur projette une présentation pour animer la formation et reste disponible pour répondre à toutes les questions.

Méthode d'évaluation

Tout au long de la formation, les exercices et mises en situation permettent de valider et contrôler les acquis du stagiaire. En fin de formation, le stagiaire complète un QCM d'auto-évaluation.

Suivre cette formation à distance

Voici les prérequis techniques pour pouvoir suivre le cours à distance :

| Un ordinateur avec webcam, micro, haut-parleur et un navigateur (de préférence Chrome ou Firefox). Un casque n'est pas nécessaire suivant l'environnement.
| Une connexion Internet de type ADSL ou supérieure. Attention, une connexion Internet ne permettant pas, par exemple, de recevoir la télévision par Internet, ne sera pas suffisante, cela engendrera des déconnexions intempestives du stagiaire et dérangera toute la classe.
| Privilégier une connexion filaire plutôt que le Wifi.
| Avoir accès au poste depuis lequel vous suivrez le cours à distance au moins 2 jours avant la formation pour effectuer les tests de connexion préalables.
| Votre numéro de téléphone portable (pour l'envoi du mot de passe d'accès aux supports de cours et pour une messagerie instantanée autre que celle intégrée à la classe virtuelle).
| Selon la formation, une configuration spécifique de votre machine peut être attendue, merci de nous contacter.
| Pour les formations incluant le passage d'une certification la dernière journée, un voucher vous est fourni pour passer l'examen en ligne.
| Pour les formations logiciel (Adobe, Microsoft Office...), il est nécessaire d'avoir le logiciel installé sur votre machine, nous ne fournissons pas de licence ou de version test.
| Horaires identiques au présentiel.

Accessibilité

HUB Formation - 1-7, Cours Valmy - 92923 Paris - La Defense
01 85 77 07 07 - www.hubformation.com - SIREN 525 287 785

Les sessions de formation se déroulent sur des sites différents selon les villes ou les dates, merci de nous contacter pour vérifier l'accessibilité aux personnes à mobilité réduite.
Pour tout besoin spécifique (vue, audition...), veuillez nous contacter au 01 85 77 07 07.