[Sf=ir] Institute

SFEIR Institute | ANGULAR

Angular

Des bases solides pour développer des applications avec Angular

5jours / 35h

Objectifs pédagogiques

- Développer une application Angular
- Réaliser des tests pour garantir une maintenabilité dans le temps
- Appréhender l'écosystème d'Angular dans sa globalité (librairies externes)

Modalités d'évaluation : Les objectifs pédagogiques sont évalués à travers la réalisation des parties pratiques (labs dirigés) sous la supervision du formateur délivrant la session de formation.

Public cible

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

Prérequis

- Modern Javascript (scope, arrow function, class, notion du this, programmation asynchrone avec les promises)
- Notion sur Typescript (typage basiques)
- Notion sur ce qu'est un Observable vs une Promise

Programme

Jour 1

A la découverte d'un nouveau framework et première prise en main des fondations du framework

Module 1: Initiation à Angular

- Configuration de son IDE: Webstorm vs Vscode
- Architecture globale d'une application Angular
 - o module
 - o composants
- Les différents modules bundler et executor
- Les bases du CLL

Ateliers:

- Bootstrapper une application Angular
- Créer votre premier composant à l'aide du CLI
- Imbriquer les composants

Module 2: Rendu dynamique et cycle de vie

- DataBinding
 - o les différents types de databindings
 - o Différence entre propriété et attributs
- Les différents cycle de vie d'un composant

Ateliers:

- Afficher une personne dans une card
- Réagir à l'événement click d'un bouton pour afficher une autre personne de manière aléatoire.

Module 3: Réaliser des requêtes réseaux

- Le client Http d'Angular
 - o Les différentes manières d'enregistrer le client HttpClient
 - o Les différentes méthodes du client Http
 - o Les différents types de réponse possible
- Comprendre comment lire les diagrammes Marble
- Comprendre les différents opérateurs de base d'RxJs

- Réaliser un appel au serveur pour afficher une personne
- Réaliser un appel au serveur lors de l'évènement click sur un bouton

Module 4: Réaliser un routing de base

- Appréhender le routing
 - o La philosophie de routing dans une app de type CSR
 - o La configuration de base du routing en Angular
- Configurer un routing
- Naviguer dans son application

Ateliers:

- Mettre en place un système de routing au sein d'une application Angular
- Naviguer par défaut sur une route
- Naviguer vers une route en particulier

Module 5: Le code flow d'Angular

- Les directives structurelles built-in
 - o nglf
 - o ngFor
 - ngSwitch
- Le nouveau code flow d'Angular
 - o @if
 - o @for
 - o @switch

Ateliers:

• Utilisation de la directive *ngFor pour créer la vue d'une liste de personne

Module 6: Communication composants - Parents / Enfants

- Communication Parent Enfant
 - L »annotation @Input
 - La configuration de l'annotation
 - o La transformation d'un Input de manière réactive
 - Rendre un input required
- Communication Enfant Parent
 - L'annotation @OutPut

- o Émettre des données à l'aide de emit
- o Rendre un OutPut totalement réactif

• Créer un composant customisable à l'aide d'Input et d'Output

Jour 2

Vers des notions plus poussées du framework mais vitales pour créer des applications plus poussées comportant des formulaires et ayant des UIs plus riches.

A la fin des ces deux jours, le candidat aura les connaissances nécessaires pour passer la certification level 2 de l'organisme Angular Training.

Module 1: L'injection de dépendances

- Les services en Angular
 - o L'annotation @Injectable
 - o Les différentes manières d'enregistrer un service
 - Injecter son service
 - Le bubbling d'injection
 - La configuration de l'injection
- Le fonctionnement de la Dependency Injection en Angular
 - Le rôle de l'Injector
 - o Le rôle du DI Token
 - o Le rôle du Provider
 - o Les différentes résolutions de valeurs d'injections

Ateliers:

- Créer et utiliser son propre service
- Réaliser un traitement à l'initialisation de l'application

Module 2: Les pipes

- Introduction aux Pipes
 - Qu'est ce qu'un Pipes ?
 - o A quoi servent-ils?
 - Les Pipes dit Pure
 - Les Pipes dit Impure
 - o Les différents pipes built-in d'Angular
- Créer ses propres pipes

- Formater une date correctement grâce au pipe date
- Créer son propre pipe permettant d'afficher une valeur par défaut dans le cas où aucune valeur existe

Module 3: Les directives

- Introduction aux directives
 - Qu'est ce qu'une directives
 - Les différents type de directives
 - o Les différents sélecteurs d'une directives
- Interagir avec les éléments du DOM
 - La class ElementRef
 - La class Renderer

Ateliers:

- Utiliser la directive ngSwitch
- Utiliser le nouveau control flow ngSwitch
- Créer une directive affichant une icon en fonction d'une condition

Module 4: Les formulaires de type Template Driven

- Les formulaires
 - Pourquoi les formulaires
 - o Les différents types de formulaires
- Les formulaire de type Template driven
 - Le module FormsModule
 - Les directives ngForm / ngModel
 - o Deep dive dans le code de ngModel
 - La validation
 - o Le retour utilisateurs des erreurs de validations
- Les formulaire de type Réactive
 - Le module ReactiveFormsModule
 - Les directives FormGroup et FormControl
 - La validation
 - Les formulaires réactifs autonomes
- La validation custom

- Créer formulaire de création d'une personne (template driven)
- Créer un formulaire d'update de la personne (template driven)
- Rendre les formulaires réactives
- Créer son propre validator
- Rendre les formulaires autonomes

Jour 3

Dans cette troisième journée, les notions d'expertise d'Angular seront abordées.

Module 1: Vers un routing plus avancés

- Le lazy loading
 - o Qu'est ce que le lazy loading
 - o A quoi sert le lazy loading
 - o Comment lazy loader un module
 - La notion de routes enfants
- Les Guards
 - o Qu'est ce qu'un guards
 - o A quoi sert un guards
 - o Les différents guards et leur utilisation

Ateliers:

- Mettre en place un lazy loading
- Mettre en place un guard
- Mettre en place un resolver

Module 2: Intercepter vos requêtes HTTP

- L'Interceptor
 - o Qu'est ce qu'un Intercepteur?
 - A quoi sert un intercepteur ?
 - o Dans quel cas en a t'on besoin?
 - o Comment écrire et enregistrer un intercepteur
- Le context HTTP
 - Qu'est que c'est ?
 - Comment et quand l'utiliser

- Mettre en place un intercepteur
- Mettre en place un context http de cache

Module 3: Vers des directives plus poussées

- Les directives de template
 - o Qu'est ce que c'est ?
 - o ng-template
 - o ng-container
 - o ng-content
 - ngTemplateOutlet
 - ngComponentOutlet
- Les annotations de template
 - Qu'est que c'est ?
 - Comment et quand l'utiliser?
 - o @ViewChild et @ViewChildren
 - o @ContentChild et @ContentChildren
- Les annotations de binding
 - Qu'est ce que c'est?
 - o @HostListener
 - @HostBinding
- Les directives structurelles
 - A quoi sert une directives structurelles
 - o La micro syntax d'une directive structurelle
 - o les limites de cette syntaxe

Ateliers:

- Téléporter du template
- Agrémenter une directive d'un listener d'évent et de binding de style
- Ecrire votre première directive structurelle

Module 4: Créer ses propres composant de formulaire

- ControlValueAccessor
 - o Ou'est ce que c'est?
 - o Comment l'utiliser?
 - Les différents Hooks
- NgValidators
 - Qu'est ce que c'est ?
 - o Comment l'utiliser?

Les différents Hooks

Ateliers:

• Créer son propre élément de formulaire

Module 5: Change Detection

- Comment fonction un cycle de changement de détection
 - o en développement
 - en production
- Les différents mode de changement de détection
 - Default
 - o OnPush

Ateliers:

• Démonstration réalisée par le formateur.

Jour 4

Ce quatrième jour est particulièrement accès sur le testing d'Angular

Module 1: State Management

- Le state management
 - Qu'est ce que c'est ?
 - o Pourquoi l'utiliser
 - Quand l'utiliser
- NgRx
 - le store
 - o le reducer
 - les actions
 - les selectors
 - les effets
- NgXs
 - o le store
 - les actions
 - o les sélecteurs

- Mettre en place un state avec NgRx
- Mettre en place un state avec NgXs

Module 2: Les testes unitaires

- Le testing en Angular
- Tester ses composants
- Tester ses services
- Tester ses directives
- Testers ses pipes
- Introduction de la librairie Angular testing

Ateliers:

• Tester une application de bout en bout unitairement

Module 3: PWA

- Qu'est ce qu'une PWA?
- Comment intégrer le package @angular/pwa
- Les stratégies de caches par le package @angular/pwa

Module 4: Server Side Rendering

- Qu'est ce que le server side rendering?
- Qu'est ce que le prerendering
- Comment intégrer le server side rendering?
- Comment hydrater son application Angular

Jour 5

Ce cinquième jour est orienté sur la modernisation du framework et les dernières nouveautés

Module 1: Standalone Components

- Un monde sans modules
 - Les standalone components
 - Le bubbling d'injection
 - Le bootstrapping
- La fonction Inject
 - o la composition api
 - o guards sous forme de class
 - o interceptors sous forme de class

• Transformer une application complète sous format standalone sans oublier d'adapter les testes

Module 2: Signals

- Qu'est ce que Signals
- Interopérabilité Signals et Rxjs
- Input et Output Signals

Ateliers:

• Transformer l'application pour qu'elle n'utilise uniquement des signals

Module 2: Performance et Core Web Vitals

- Qu'est ce que les Core Web Vitals
- Lazy loader des composants au runtime
- Déférer une partie de votre vue

- Lazy loader un composant de modal
- Charger à la demande une partie de votre vue