DÉVELOPPEMENT FRONTEND

Angular & TypeScript : développement web moderne

Apprenez Angular avec TypeScript : composants autonomes, signaux, contrôle de flux, services, routage, appels HTTP et formulaires, jusqu'à une application reliée à une API.

DURÉE	TARIF	NIVEAU	LANGUE	GROUPE	FORMAT
3j (21h)	**975 € / participant** Inter-entreprise. Intra : sur devis, au plus 975 € / participant	**Débutant**	**Français**	**3-10**	**Formation** À distance

1 PUBLIC VISÉ

- Développeurs web qui connaissent JavaScript et veulent adopter Angular
- Développeurs amenés à faire évoluer une application Angular en équipe

2 PRÉREQUIS

- Écrire du JavaScript moderne : fonctions fléchées, déstructuration, modules, promesses et async / await
- Aucune connaissance d'Angular ni de TypeScript n'est requise

3 OBJECTIFS PÉDAGOGIQUES

- Typer des données en TypeScript : types, interfaces, unions et génériques
- Créer des composants Angular autonomes avec liaison de données, contrôle de flux (@if, @for) et pipes
- Gérer l'état d'un composant avec les signaux et faire communiquer les composants par input() et output()
- Partager la logique dans des services injectés et naviguer entre plusieurs vues avec le routeur (paramètres, garde)
- Consommer une API REST avec HttpClient en gérant le chargement et les erreurs
- Construire un formulaire validé avec les formulaires à signaux

4 PROGRAMME DÉTAILLÉ

Jour 1 (7h) : TypeScript et composants

Module 1 : TypeScript pour Angular (3h30)

Pourquoi TypeScript et types de base (1h30)

- Le problème que TypeScript résout : erreurs silencieuses, modifications risquées
- Compilation et typage statique
- Types de base, any et unknown, inférence ; tableaux, tuples, propriétés optionnelles
- Exercice d'entraînement : typer un catalogue de formations

Interfaces, unions et génériques (2h)

- Interfaces et alias de type

- Unions, types littéraux et réduction des types (narrowing)
- Génériques et contraintes ; les décorateurs : le principe
- TP évalué : modéliser un catalogue de formations et écrire un service générique de collection typée

Module 2 : Composants et templates (3h30)

Créer un projet Angular (1h30)

- Angular CLI : ng new, ng generate
- Structure d'un projet ; composants autonomes, sans NgModule
- Cycle de vie d'un composant ; Angular DevTools
- Exercice d'entraînement : créer le projet et un premier composant

Templates et liaison de données (2h)

- Interpolation, liaison de propriété et d'événement
- Contrôle de flux : @if, @for, @switch
- Pipes : date, currency, uppercase
- TP évalué : composant de fiche formation et liste de formations affichée avec @for et @if

Jour 2 (7h) : Signaux, services et routage

Module 3 : Signaux et communication entre composants (3h30)

Les signaux (1h30)

- signal(), computed(), effect()
- La détection des changements sans zone.js
- Exercice d'entraînement : panier géré par des signaux

Composants parents et enfants (2h)

- input(), output() et model()
- Projection de contenu
- TP évalué : liste de formations filtrée par des signaux, avec un composant enfant de carte qui signale la formation choisie

Module 4 : Services, injection et routage (3h30)

Services et injection de dépendances (1h30)

- Créer un service ; providedIn: root
- Injecter avec inject()
- Un service qui garde son état dans des signaux
- Exercice d'entraînement : service catalogue partagé entre deux composants

Routage (2h)

- Routes, router-outlet, routerLink
- Paramètres d'URL ; navigation par le code
- Gardes ; chargement différé des pages
- TP évalué : application multipage : liste, détail lu depuis un paramètre d'URL, page protégée par une garde

Jour 3 (7h) : HTTP, formulaires et projet de synthèse

Module 5 : Appels HTTP et formulaires (3h)

HttpClient et RxJS (1h30)

- provideHttpClient ; GET et POST typés

- Observable : pipe, map, catchError, switchMap
- Passer d'un Observable à un signal ; intercepteurs
- Exercice d'entraînement : charger les formations depuis une API

Formulaires à signaux (1h30)

- Modèle de formulaire, règles de validation, messages d'erreur
- Les formulaires réactifs, qu'on rencontre dans le code existant
- TP évalué : formulaire de création de formation validé, envoyé à l'API, avec la liste rafraîchie et les états de chargement et d'erreur affichés

Module 6 : Projet de synthèse, évalué (4h)

Réalisation (2h30)

- Projet de synthèse, réalisé individuellement : application Angular reliée à une API publique : liste paginée, vue détail, formulaire de création, routage, service et signaux

Revue et retour (1h30)

- Revue de code collective : composants, signaux, services
- Retour individuel du formateur sur le projet rendu

Le contenu peut être ajusté selon le niveau, les prérequis et les besoins des participants.

5 COMPÉTENCES VISÉES

- Écrire du TypeScript typé
- Construire l'interface d'une application Angular
- Structurer une application Angular : services et routage
- Échanger des données avec une API et des formulaires

6 MODALITÉS PÉDAGOGIQUES

Formation délivrée à distance, en classe virtuelle. Le formateur alterne entre méthode démonstrative, méthode interrogative (analyse et critique de code Angular existant) et méthode active (exercices d'entraînement, TP évalués et projet de synthèse).

7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Support de cours numérique mis à disposition des apprenants
- Dépôt GitHub de démonstration avec exercices et corrections par module
- Environnement de développement : VS Code + Node.js en version LTS + Angular CLI + Angular DevTools, installés sur l'ordinateur du participant
- Classe virtuelle : visioconférence (Microsoft Teams ou équivalent), partage d'écran, chat en direct
- Supports remis par e-mail et sur la plateforme pédagogique LaPolaris (ressources, émargement)

Chaque participant suit la formation depuis son ordinateur, avec un accès à Internet, un micro et des haut-parleurs ou un casque. En formation intra-entreprise, l'entreprise s'assure que chaque participant dispose de ce matériel et des logiciels nécessaires, conformément aux prérequis indiqués dans le programme et aux moyens prévus dans la convention.

8 MODALITÉS D'ÉVALUATION

- Questionnaire de positionnement individuel renseigné par chaque participant avant le démarrage de la formation, analysé par le formateur et donnant lieu, le cas échéant, à un ajustement du déroulé
- Exercices d’entraînement réalisés par chaque participant sur son poste, corrigés en séance
- Évaluation des acquis par les TP évalués, réalisés individuellement sur poste : un par module, puis le projet de synthèse décrit dans le dernier module du programme ; chaque objectif est évalué dans au moins un TP
- Grille de suivi renseignée par le formateur pour chaque participant et pour chacun des objectifs de l’action, au regard des critères d’évaluation ci-dessous
- Questionnaire d’appréciation renseigné par les participants en clôture de formation, et enquête à froid dans un délai de trois à six mois

9 CRITÈRES D'ÉVALUATION

- Modèle de données typé avec interfaces, unions et génériques ; aucun any ; code qui compile sans erreur
- Composants autonomes ; liaisons de propriété et d’événement justes ; listes et conditions écrites avec @for et @if ; pipes utilisés pour l’affichage
- État géré par signal() et computed() ; données transmises par input() et remontées par output()
- Service injecté avec inject() et partagé entre composants ; au moins deux vues reliées par le routeur, avec paramètre et garde
- Appels GET et POST typés ; états de chargement et d’erreur affichés ; erreur HTTP gérée sans casser l’application
- Formulaire à signaux avec règles de validation et messages par champ ; envoi bloqué tant que le formulaire est invalide

10 INDICATEURS DE RÉSULTATS

Les indicateurs de résultats de cette formation (taux de satisfaction et taux de réussite) seront publiés sur sa fiche en ligne à l’issue des premières sessions réalisées.

11 MODALITÉS DE VALIDATION

Une attestation de fin de formation est remise à chaque participant à l’issue du parcours. Elle mentionne **l’intitulé** et la **nature de l’action**, sa **durée** en heures, sa **période de réalisation**, **les objectifs de l’action** et le **niveau d’atteinte** de chacun d’eux, exprimé en trois valeurs : **atteint**, **partiellement atteint**, **non atteint**. **Une attestation d’assiduité**, établie à partir des émargements, est délivrée dans les mêmes conditions.

Formation non certifiante : elle ne délivre pas de certification enregistrée au RNCP ou au Répertoire spécifique.

12 SUIVI ET ACCOMPAGNEMENT

- Émargement électronique signé par chaque participant et par le formateur à chaque demi-journée
- Traçabilité des activités pédagogiques réalisées
- Attestation d’assiduité délivrée en fin de formation

- Suivi individuel via les QCM de fin de journée, les exercices corrigés et les TP évalués

13 CONDITIONS D'ACCÈS

Formation accessible sans prérequis administratif particulier. Cette formation n’est pas éligible aux financements publics ou mutualisés, notamment au compte personnel de formation et à une prise en charge par un opérateur de compétences.

14 DÉLAIS D'ACCÈS

Inscription possible jusqu’à 15 jours ouvrés avant le début de la session pour permettre la préparation logistique et, si besoin, l’adaptation du contenu au contexte client. Pour toute demande spécifique, nous contacter directement.

ACCESSIBILITÉ · HANDICAP

Nos formations peuvent être accessibles aux personnes en situation de handicap. La question d’un besoin d’aménagement (matériel, temporel ou pédagogique) est posée lors de l’analyse du besoin et du questionnaire de positionnement ; vous pouvez aussi nous le signaler à tout moment, y compris pendant la formation, afin d’étudier les adaptations possibles. Référent handicap : contact@lapolaris.fr

LaPolaris

TÉL. +33762584798

EMAIL contact@lapolaris.fr

WEB lapolaris.fr