



# Angular, développement avancé

Angular, devenu la référence des infrastructures JavaScript côté client, propose une refonte du cœur du framework en s'appuyant sur les nouveaux standards du Web. Cette formation vous permettra de maîtriser en profondeur les bonnes pratiques de développement avec le framework Angular 5 et ses versions suivantes.

## Objectifs

- Savoir utiliser les décorateurs Angular
- Architecturer les applications Web complexes
- Intégrer les outils de documentation et les tests unitaires
- Développer et intégrer des bibliothèques de composants

## Moyens pédagogiques

- Présentation du formateur et du programme.
- Présentation et écoute de chacun de stagiaires.
- Apports didactiques pour apporter des connaissances communes.
- Mises en situation de réflexion sur le thème du stage et des cas concrets.
- Méthodologie d'apprentissage interactive et participative.
- Exercices et études de cas concrets.
- Temps d'échanges.
- Accompagnement pédagogique individualisé.

## Formateur

Les formateurs de CROSTHINK sont des experts de leur domaine, disposant d'une expérience terrain qu'ils enrichissent continuellement. Leurs connaissances techniques et pédagogiques sont rigoureusement validées en amont en interne.

## Suivi de l'exécution et évaluation des résultats

- Accueil des stagiaires dans une salle dédiée à la formation / A distance
- Feuilles de présence.
- Documents supports de formation projetés.
- Mise à disposition du stagiaire des documents et supports de formation.
- Tout au long et/ou à l'issue de la formation : Evaluation des acquis des stagiaires via des exercices, des QCM, des QUIZZ, des mises en situation et/ou des cas pratiques.
- Enquête de satisfaction.
- Attestation de fin de formation.

## EN BREF

Durée : 2 jours

Tarif : 990€

Public et pré requis

Cette formation ne nécessite pas de prérequis.

## FORMATIONS A DISTANCE

Contactez-nous  
[contact@crossthink.fr](mailto:contact@crossthink.fr)

## PROCHAINES DATES

Nous consulter



# Angular, développement avancé

## Programme

### Angular, mise en œuvre des bonnes pratiques

- Injection de dépendances.
- Types applicatifs partagés.
- PWA : les Service Workers.
- Requêtes HTTP avancées.
- Lazy Loading.

#### Travaux pratiques

*Analyser et optimiser une application.*

### Fonctionnement interne d'Angular

- ZoneJS : le concept.
- Optimisation des cycles de rendu, exécution hors ZoneJS.
- Choisir RxJS.
- Utilisation des observables.
- Création, combinaison, opérateurs clés.
- Compilation Ahead Of Time.
- Webpack Bundle Analyzer.

#### Travaux pratiques

*Créer une application utilisant RxJS.*

### Création de composants distribuables

- Les Web Components.
- Méthodologie : Interactive Component Sheet.
- Les décorateurs.
- Le Change Detection Mode.
- Composants neutres versus à état. Communication entre composants, optimisation ES6.
- Projection de contenu, pilotage de composants enfants.
- Préparer les composants pour la distribution.
- Documentation : génération dynamique.

#### Travaux pratiques

*Développer et packager des composants distribuables.*

### Composants riches et bibliothèques externes

- Découverte de l'écosystème Angular pour la gestion des données, AngularFire, NgRx.
- Les bibliothèques UI : Angular Material, Prime NG...
- Les animations.

#### Travaux pratiques

*Réutilisation de composants, intégration de bibliothèques externes.*

### Formulaire dynamique : le FormBuilder

- Création de formulaire dynamique : ReactiveFormsModule.
- FormControl et FormGroup, AbstractControl, FormArray.
- Validation et gestion d'erreur personnalisée.



CROSSTHINK

- Création de modèles de données.
- Utilisation du FormBuilder.
- Création dynamique de template.
- Abstraction de composant métier de formulaire.

#### **Travaux pratiques**

*Mise en œuvre de la génération et les cycles de validation avancée de formulaire.*

### **Tests unitaires. Bonnes pratiques et outils.**

- Karma et Jasmine.
- Tests d'intégration avec Protractor.
- Le Code-Coverage.
- Behaviour Driven Development, Test Driven Development.
- Cas de test : Pipe, Component, Service, etc.

#### **Travaux pratiques**

*TDD : développer une application à partir de tests unitaires.*