

Stage pratique de 4 jour(s)
Réf : AGU

Participants

Architectes, développeurs et chefs de projets Web.

Pré-requis

Bonnes connaissances des technologies du Web et des outils modernes de développement Front-End. Connaissances de JavaScript.

Prix 2020 : 2490€ HT

Dates des sessions

AIX

21 avr. 2020, 22 sep. 2020
15 déc. 2020

ANGERS

07 avr. 2020, 08 sep. 2020
01 déc. 2020

BORDEAUX

14 avr. 2020, 15 sep. 2020
08 déc. 2020

BRUXELLES

03 mar. 2020, 09 juin 2020
08 sep. 2020, 01 déc. 2020

DIJON

27 avr. 2020, 29 sep. 2020
01 déc. 2020

GENEVE

03 mar. 2020, 09 juin 2020
08 sep. 2020, 01 déc. 2020

GRENOBLE

07 avr. 2020, 08 sep. 2020
01 déc. 2020

LILLE

07 avr. 2020, 08 sep. 2020
01 déc. 2020

LIMOGES

14 avr. 2020, 15 sep. 2020
08 déc. 2020

LUXEMBOURG

03 mar. 2020, 09 juin 2020
08 sep. 2020, 01 déc. 2020

LYON

27 avr. 2020, 29 sep. 2020
01 déc. 2020

MONTPELLIER

28 jan. 2020, 27 avr. 2020
29 sep. 2020, 01 déc. 2020

NANCY

14 avr. 2020, 15 sep. 2020
08 déc. 2020

NANTES

07 avr. 2020, 08 sep. 2020
01 déc. 2020

NIORT

07 avr. 2020, 08 sep. 2020
01 déc. 2020

ORLEANS

12 mai 2020, 08 sep. 2020
01 déc. 2020

PARIS

18 fév. 2020, 03&17 mar. 2020
07&21 avr. 2020, 12&26 mai 2020

Angular, maîtriser le framework Front-End de Google migration et développement

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 d'en maîtriser les concepts et d'améliorer les performances de vos applications notamment mobiles. Ce programme est compatible avec les dernières versions du framework Angular 5 et suivantes.

OBJECTIFS PEDAGOGIQUES

Organiser, modulariser et tester ses développements JavaScript
Maîtriser les fondamentaux du Framework Angular
Créer rapidement des applications Web complexes
Savoir intégrer les tests unitaires au développement
Connaître les bonnes pratiques de développement et de mise en production

1) Développement JavaScript : rappels

2) Migrer d'AngularJS 1.x à Angular

3) L'utilitaire ng ou @angular/cli

4) Définition de composants

5) Classifications des composants applicatifs

6) Gestion des formulaires, "Routing" et requête HTTP

7) Tests unitaires. Bonnes pratiques et outils

Méthodes pédagogiques

Chaque nouveau concept théorique sera appliqué immédiatement de façon pratique.

Travaux pratiques

Composition modulaire d'une application avec Angular.

1) Développement JavaScript : rappels

- Bonnes pratiques ECMAScript 5.
- ES7/ES6/ES2015, présentation générale.
- Nouveautés syntaxiques : portée, "template string", "arrow functions", les promesses...
- Le développement JavaScript Objet avec la syntaxe de class.
- Le pattern observer/observable (ES7).
- La librairie RX.js : opérateur clés pour la manipulation d'observable.
- Outils indispensables. Babel, Traceur et Typescript.
- Typescript en détail, configuration.
- ES6/2015 approche modulaire.
- ES7 gestion de l'asynchronicité : await async.
- "Modules Loaders" : Webpack, "import/export".

Travaux pratiques

Migrer un script ES5/ES6. Mise en œuvre de l'environnement avec TypeScript.

2) Migrer d'AngularJS 1.x à Angular

- Comparaison et "topographie" des concepts.
- Préparer la migration. Structure d'une application Angular.
- Les modules Angular, "core" et principaux modules.
- Principe de l'injection de dépendance.
- Classification des directives : composant, attribut, structure.
- Les décorateurs : définition des hiérarchies.

Démonstration

Migrer une application AngularJS 1.x vers Angular.

3) L'utilitaire ng ou @angular/cli

- Utilisation de l'utilitaire en ligne de commande.
- Scaffold une structure de projet : anatomie et dépendance.
- Configuration des utilitaires de tâches.
- Configuration et commandes clés.
- Lancer un serveur de développement/production (build).
- Compilation Ahead of Time. La notion de "Tree Shaking".
- Gestion des modules : bonnes pratiques.
- BootStrap d'application.

Travaux pratiques

Structurer, "scaffold" un projet d'application.

4) Définition de composants

09&23 juin 2020, 07&21 juil. 2020
18&25 août. 2020, 08&22 sep. 2020
06&20 oct. 2020, 03&17 nov. 2020
01&15 déc. 2020

REIMS

03 mar. 2020, 09 juin 2020
08 sep. 2020, 01 déc. 2020

RENNES

21 avr. 2020, 22 sep. 2020
15 déc. 2020

ROUEN

03 mar. 2020, 09 juin 2020
08 sep. 2020, 01 déc. 2020

SOPHIA-ANTIPOLIS

07 avr. 2020, 08 sep. 2020
01 déc. 2020

STRASBOURG

21 avr. 2020, 22 sep. 2020
15 déc. 2020

TOULON

21 avr. 2020, 22 sep. 2020
15 déc. 2020

TOULOUSE

14 avr. 2020, 15 sep. 2020
08 déc. 2020

TOURS

14 avr. 2020, 15 sep. 2020
08 déc. 2020

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers des multiples exercices à réaliser (50 à 70% du temps).

Compétences du formateur

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

Moyens pédagogiques et techniques

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour

- Comprendre les Web Components. (standard, concepts, shadow DOM, scoped CSS...).
- Cycle de vie dans l'application.
- Angular Compiler : Change Detection.
- Syntaxe des templates : interpolation/expression, "Binding" et filtres.
- Directives de transformation : ngIf, ngFor, ngSwitch...
- Définition syntaxique, le symbole (*).
- Variables locales et variables de Template.
- Classe de composants. Directives de configuration : selector, provider.
- Événements utilisateur et événements logiques personnalisés : EventEmitter.

Travaux pratiques

Création de composants.

5) Classifications des composants applicatifs

- Module, RouterModule, SharedModule.
- Component, Directive, Pipe, Service, Guard.
- Principe de l'injection de dépendances.
- Création de services injectables. Classification des services.
- Configuration de l'injecteur.
- Les décorateurs et leurs propriétés en détails.
- @Host, @ContentChild, @ViewChild.
- JavaScript Pure function, PurePipe.

Travaux pratiques

Création de composants et de directives personnalisées.

6) Gestion des formulaires, "Routing" et requête HTTP

- FormControl et FormGroup.
- TDF versus DDF : Template Driven Form et Data Driven Form
- Validation et gestion d'erreur personnalisée.
- Liaison de données via HTTP.
- Gestion et configuration des échanges HTTP au niveau applicatif.
- Création de routes.
- Interceptor les paramètres de routage et wildcard.
- Ciblage, "router-outlet" événements de routage.
- Gestion de routes dans l'arbre des composants.
- Configuration des "Guard" pour le l'initialisation des routes.

Travaux pratiques

Mise en œuvre des cycles de validation de formulaire. Consommation d'une API REST.

7) Tests unitaires. Bonnes pratiques et outils

- Configurer l'environnement de test.
- Présentation de Karma et de Jasmine.
- Ecrire les tests avec Jasmine.
- Ecrire des tests d'intégration avec protractor.
- Couverture du code. Indices du code-coverage.
- BDD Behaviour Driven Development, TDD Test Driven Development.
- Cas de test : pipe, composant, application.
- Angular "Coding guide Style".

Travaux pratiques

Développement d'une application à partir de tests unitaires. Mise en œuvre du Test Driven Development.

les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.

- A l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.

- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.