

# AngularJS, développement avancé de la conception à l'analyse des performances

**NEW**

<b>Objectifs</b>	Appréhender AngularJS en profondeur Devancer les besoins conceptuels Savoir développer et gérer des composants métiers Optimiser une application AngularJS
<b>Participants</b>	Architectes, développeurs désireux de se spécialiser et d'optimiser leur conception d'application avec d'AngularJS.
<b>Prérequis</b>	Connaissances pratiques du framework AngularJS et des outils modernes du développement Front-End (Bower, Grunt, Framework CSS...). Bonnes connaissances JavaScript et des Patterns Objets.
<b>Moyens pédagogiques</b>	1 poste par participant - 1 Vidéo projecteur - Support de cours fourni à chaque participant – Formation présentielle
<b>Durée</b>	2 jours

**Code : ANGUJS-DEAVANCE**

## Programme.

### "Best Pratices" et rappels

Ecosystème Front-End autour d'AngularJS. Rappels des composants essentiels.

AngularJS Coding Style Guide. Quels plugins pour votre IDE ?

Mise en oeuvre du workflow AngularJs avec nodejs, Yo, Grunt.

Bonnes pratiques. Production : RequireJS, Browserify ou concaténation. (ngAnnotate).

"Test Driven Development" avec Jasmine et Karma.

Services natifs : interception et décoration.

### Travaux pratiques

Intégration d'une structure d'application, rédiger des services en conformité avec les tests.

### Maîtriser les directives et la "compilation" AngularJS

Le "Dirty Checking" AngularJS détection des changements : \$watch.

\$digest, \$compile, comprendre le cycle de digestion.

Expressions AngularJS : \$eval et \$parse.

Options avancées : require, controllerAs, bindToController, terminal.

Ensembles de directives inter-dépendantes.

Gérer le cache des templates : \$templateCache.

### Travaux pratiques

Organiser vos directives en ensembles de composants réutilisables et distribuables.

### Utiliser et écrire des composants avancés.

De la directive au composant distribuable.

AngularUI. Principaux modules complémentaires.

UI-Utils, les outils indispensables. Intégrer ngTranslate. UI-Bootstrap.

UI-Router, gestion de routes avancées.

Famille de composants métiers personnalisés.

### Travaux pratiques

Développer des composants dépendants de services asynchrones.

### AngularJS : optimisation des développements

Nomenclature et évaluation du code. Minification et gestion des scripts.

Astuces programmatiques. Utilisation de \$log et \$exceptionHandler.

Performances : bindOnce, préfixe "::", \$scope.\$apply versus to \$scope.\$digest.

Guide des bonnes pratiques. Configuration dynamique des routes.

Chargement des contrôleurs et modules à l'exécution : "On Demand".

Déboguer une application AngularJS.

### Démonstration

Analyser et améliorer les performances de votre application AngularJS