

Formation MEAN Stack : MongoDB, Express.js, Angular et Node.js

■ Durée :	10 jours (70 heures)
■ Tarifs inter-entreprise :	5 475,00 € HT (standard) 4 380,00 € HT (remisé)
■ Public :	Développeurs Javascript
■ Pré-requis :	Bonnes connaissances de Node.js
■ Objectifs :	Construire une application web fullstack avec la MEAN Stack : MongoDB, Express.js, Angular et Node.js
■ Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
■ Modalités d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
■ Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
■ Référence :	SIT102245-F
■ Note de satisfaction des participants:	4,93 / 5
■ Contacts :	commercial@dawan.fr - 09 72 37 73 73

■ **Modalités d'accès :**

Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.

■ **Délais d'accès :**

Variable selon le type de financement.

■ **Accessibilité :**

Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

Découvrir Angular

Présentation de l'environnement de développement NodeJS

Présentation de la CLI Angular

Rappel des normes EcmaScript

Créer un projet Angular à l'aide d'angular CLI

Atelier : Configurer un projet Angular

TypeScript

Présentation de TypeScript

Les types de variables de TS

Le typage fort

Les interfaces de classe

Présentation des décorateurs

Présentation

La Syntaxe de base d'Angular

L'organisation en Module et Composant

Les décorateurs Angular

Le MVVM / MVW

Le One Way et Two Way Data Binding

L'opérateur "moustache"

La gestion des Events et l'objet \$event

Les NgModel

Les variables de template

Atelier : Réaliser une TodoList

Gérer les routes

Le Module de Routage

Déclarer les routes d'un projet

Gérer les redirections

La route par défaut

Récupérer les paramètres d'url, les query-strings, etc

Limiter l'accès au route: Les guards

Routes imbriquées

Atelier : Mettre en place une application multi-vues

Utiliser les pipes

Présentation des pipes «|»

Utiliser le safeOperator «?.»

Importer des locales dans notre projet

Définir une locales par default

Utiliser les locales avec les filtres

Créer des pipes personnalisés

Les pipes pures et impures

Atelier: Créer un pipe personnalisé

Les Directives

Différence entre directives d'attributs et structurelles

Les directives ngFor, ngIf, ngSwitch, ngClass, ngStyle

Utiliser les attributs ARIA (directive attr.*)

Créer une directive personnalisée

Transmettre des valeurs à une directive (@input)

Gérer les événements dans une directive

Atelier: Créer une directive personnalisé

Construire des Formulaires

Présentation des formulaires

Les Templates Driven Form et Reactive Form

Gérer la validations des champs et du formulaire

Présentation des états d'un formulaire (ngControl)

Valider les formulaires (ngSubmit)

Créer des validateurs personnalisés

Atelier : Gestion des entrées utilisateurs

Gérer l'injection de dépendances

Principes de DI

Configurer l'injecteur

Enregistrer des services dans un composant

Création de l'injecteur

Différents types de service: classe, valeur, fabrique...

Atelier : Architecturer notre application sous forme d'une organisation de services

Utiliser le client HTTP

Principe de communication asynchrone

Fonctionnement des promesses et async/await

Présentation des Observables RxJS

Le Module HttpClientModule

Communiquer avec une API REST (http.get, http.post, etc)

Appliquer des fonctions de callback

Gestion des erreurs

Typage les données retournées

Atelier : Mettre en place un client d'API REST

Déployer une application Angular

Les commandes de compilation

Les modes de compilation

Les compilations AOT et JIT

Gestion du base href

Les règles de réécriture

Atelier : Compiler un projet pour la publication

Comprendre l'architecture microservices

Architecture en couches : du monolythe au microservices

Contraintes d'architecture des microservices

Gestion de l'authentification centralisée dans une architecture microservices

Intérêt d'une passerelle d'API
Gestion centralisée des traces

Développement d'API REST avec Express.js

Express.js : principe, fonctionnalités, pré-requis
Configuration du projet : packages et modules additionnels : Morgan, Helmet, Cors,...
Configuration de morgan pour la gestion des logs de requêtes HTTP
Organisation des couches du projet : routes, services, repositories
Intérêt d'une couche DTO, utilisation d'un mapper d'objets : automapper ou autre
Implémentation de contrôleurs REST : gestion des routes, traitement asynchrone
Gestion des paramètres de méthodes et du mapping
Gestion du download
Gestion de l'upload, configuration
Gestion des services et des transactions associées
Gestion du cross origin avec Cors et restriction des domaines appelants
Test de l'api REST avec Postman
Ecriture de tâches asynchrones, planification
Cache web

Atelier : Écriture de micro-services avec Express.js - Test des méthodes de services avec Postman ou autre

Documenter une API REST

Open API Specification (Swagger) : présentation, outil
Utilisation de express-openapi
Visualisation avec Swagger Editor
Documentation du code Java, génération de javadoc

Atelier : Documentation de l'api

Intercepter des requêtes et gérer les erreurs

Gestion globale des exceptions
Capture d'exceptions personnalisées
Intercepteurs de requêtes/réponses

Atelier : Gestion des exceptions et implémentation d'intercepteurs

Appeler d'autres API REST (écriture de clients)

http_request : méthodes et paramètres
Ecriture de requêtes GET, POST, PUT, DELETE
Gestion des paramètres et du corps de la requête
Gestion des headers
Gestion des réponses et utilisation de mappers

Atelier : Implémentation de clients pour un service REST

Sécuriser l'API REST

Configuration de Helmet
Gestion des données d'entête
Gestion des utilisateurs et des rôles : mise en place de JWT (passport-jwt)
Hachage des mots de passe avec bcryptjs

Atelier : Intégration de Helmet, jsonwebtoken, dotenv et bcryptjs

Tester une application Express.js avec Jest et SuperTest

Stratégies de tests, types supportés
Installation de jest, supertest et cross-env, configuration de l'application
Mocking des couches de l'application
Tests auto-configurés
Exécution et reporting

Atelier : implémentation et exécution de tests avec Jest et supertest

Configurer un projet Mongoose

Mongoose : Présentation, fonctionnalités
MongoDB : Installation, structure d'un schéma (collections, documents)
Configuration d'un projet Node.js utilisant Mongoose : pool de connexion, contraintes
Propriétés par défaut et paramétrage
Gestion des options de logging

Atelier : Intégration de Mongoose dans un projet Node.js, configuration de la Bdd NoSQL MongoDB et des traces

Construire un modèle et écrire des requêtes

Modèle de base
Documents et sous-dodcuments

Gestion des types
Requêter des documents
Mise à jour et suppression de documents
Gestion des validations
Mongoose migrations (ts-migrate-mongoose)

Atelier : Réalisation d'un schéma global de BDD MongoDB, gestion des migrations et écriture de requêtes

Maîtriser des concepts avancées

Cache : fonctionnement, mise en cache de requêtes (mongoose-cache)
Hooks
Mise en place d'une solution d'audit de tables (historique de modifications)

Atelier : Implémentation d'une couche complète de persistance - mise en place d'un cache