



Formation MEVN Stack : MongoDB, Express.js, Vue.js et Node.js

■ Durée :	10 jours (70 heures)
■ Tarifs inter-entreprise :	5 475,00 CHF HT (standard) 4 380,00 CHF HT (remisé)
■ Public :	Développeurs Javascript
■ Pré-requis :	Bonnes connaissances de Node.js
■ Objectifs :	Construire une application web fullstack avec la MEVN Stack : MongoDB, Express.js, Vue.js et Node.js
■ Modalités pédagogiques, techniques et d'encadrement :	<ul style="list-style-type: none">• 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 :	<ul style="list-style-type: none">• 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 :	SIT102246-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 Vue.js

Vue.js : présentation, principes

Comparaison avec d'autres frameworks JS : Angular, React,...

Initialisation d'un projet et interface en lignes de commande

Architecture d'une application Vue.js ; objet Vue

Interface de debug

Remise à niveau : EcmaScript5, NodeJS

Notions de composants, cycle de vie

Atelier : Installation de l'environnement, création d'une application

Créer des composants

Structure et implémentation d'un composant

Gestion des événements

Imbrication de composants

Slots

Événements personnalisés

Atelier : Organisation de l'application en composants

Peupler les vues

Méthodes d'écritures des templates : render methods, jsx

Composants fonctionnels

Gestion des animations et des des transitions

Atelier : Utilisation de différentes méthodes pour enrichir les vues

Gérer la navigation dans l'application

Configuration des routes
Sécurité de l'application (filtres d'interceptions)
Gestion des transitions

Atelier : Utilisation de Vue router pour naviguer entre les pages.

Gérer l'état et stocker des données

Vuex : présentation, concepts de base.
Gestion des plugins
Stockage des données

Atelier : Utilisation de Vuex dans l'application

Interagir avec une application back-end

Chargement et envoi de données de manière asynchrone
Interrogation d'un service web REST et traitement d'un flux JSON
Gestion des formulaires et validation des saisies
Introduction du data binding

Atelier : Interrogation de services web et exploitation du résultat.

Améliorer son code

Mixins et plugins
Utilisation des Single File Components .vue.
Directives personnalisées
Optimisation des performances
Server Side Rendering
Implémenter des tests unitaires

Atelier : optimisation du code applicatif et tests unitaires.

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-doduments

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