Formation .NET 8 – Développement back-end (ASP.NET Core & MVC)

Informations

Durée : 4 jours (28h.)

Tarif* : 1990 € HT
Intra: Nous consulter

Réf : DTCO

Niveau : Moyen

inter à distance / intra

Mise à jour le 18/12/25

*tarif valable jusqu'au 31/12/2026

Prochaines sessions

30 novembre - 03 décembre
(à distance)

Pré-requis

- Connaissances en HTML/CSS
- Connaissances en JavaScript
- Connaissances fondamentales en C#
- Notions de base sur le Framework .NET
- Bases de la Programmation Orientée Objet (POO)
- Expérience avec Visual Studio

Objectifs

Objectifs pédagogiques :

- Découvrir les fonctionnalités du Framework ASP.NET Core et MVC Core
- Identifier les fonctions essentielles de la plateforme
- Savoir concevoir une application ASP.NET
- Connaître les mécanismes avancés du Framework et s'initier à Entity Framework Core
- Etre en mesure de migrer un projet ASP.NET MVC existant vers ASP.NET MVC Core

Objectifs opérationnels :

- Développer des applications avec le Framework ASP.NET Core et MVC Core

Programme

Introduction à .Net Core

Présentation générale du Framework
Historique des versions d'ASP.NET
Qu'est-ce que ASP.NET Core ?
Principes de base et philosophie d'ASP.NET MVC Core
Avantages et inconvénients de ASP.NET Core

Environnement et outils

Vue d'ensemble
Middleware
Services
Kestrel
Librairie .NET Core
Runtime CoreCLR
Environnement d'exécution .NET

Configuration d'un projet .Net Core

Project.json
Méthode principale
Classe Startup
Méthode Configure
Méthode ConfigureServices
Composants MVC Core
Kestrel Improvements

Contrôleurs MVC Core

Prérequis et conventions
Injection de dépendances
Action et résultats d'action
ViewData / ViewBag et TempData

Modèles MVC Core

Ignorance de persistance

Inversion de dépendance
Mapping objet-relationnel
Validateurs et attributs
Entity Framework (EF) Core
ViewModel

Vues MVC Core

Conventions et ViewStart
Zones (areas), vues partielles et Layouts
Injection de dépendances dans une vue
Syntaxe Razor et C# 8
Helpers et Tag Helpers
L'objet ViewData/TempData
Vues fortement typées
View Component

Entity Framework Core

Configuration des services et middlewares
Modéliser : Modelling
Requêter : Query
Enregistrer avec gestion des transactions
Le Change Tracking

Paramètres d'application

Sources de configuration
Configuration spécifique à l'environnement
Options pattern

Routing de requêtes en MVC Core

API RESTful
Routage Middleware
Templates de routes
Contraintes de routes
Middleware MVC
Attributs de routage
Méthode MapRoute
Fichiers statiques
Filtres de page

Formulaires HTML

Introduction
Annotations de données
Formulaire Tag Helper
Tag Helpers de validation
Soumissions de formulaire

Razor Page

Concept du tout-en-un
Configurer les routes
Architecture orientée contenu

Mixer Razor Page et MVC
Soumissions de formulaire

Web API

Contrôleurs WebAPI REST
AJAX et les mises à jour partielles
Contrôleurs asynchrones
Traitement de données JSON
Gestion du Cross Origin
Mise en place d'un versionning API

Gestion des erreurs

Page d’exceptions pour développeur
Code statut
Filtres d’exception
Gestion des logs / journalisation

Authentification et contrôle d'accès avec .Net Core

Configuration d'ASP Identity
Niveaux de verbosité d’authentification
Personnalisation du User Model
Construire et sécuriser sa propre API
Sécuriser l'AIP avec les tokens JWT
OpenID et OAuth 2 pour les fournisseurs externes (Google, Microsoft…)

Tests sous .Net Core

Introduction
Test unitaire
Test d’intégration

Etat de l’application

Vue d’ensemble
Httpcontext.Items
Etat de session

Déploiement et intégration à partir de .Net Core

Déployer sur un serveur IIS
Déployer vers le cloud Azure
Déployer en utilisant Docker
Azure App Services
Compilation des pages et des vues à la publication

Développement Cross-plateforme

Gestionnaire de version de .NET via .NET CLI (.NET version manager, anciennement DNVM)
Outils de développement : Visual Studio, VS Code, Yeoman…
Support de Windows, Mac OSx et Linux