

## Formation ASP.NET Core MVC : Initiation + Approfondissement

<b>Durée :</b>	5 jours
<b>Public :</b>	Développeurs .Net (C# ou VB.Net)
<b>Pré-requis :</b>	Maîtrise de l'objet (C# ou VB.Net), connaissances en HTML/CSS/JS
<b>Objectifs :</b>	Maîtriser le développement d'applications ASP.Net MVC Core avec le moteur de vue Razor et une couche d'accès aux données pilotée par Entity Framework
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	.NE100926-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Découvrir le framework ASP.NET MVC Core

Pattern MVC : architecture, apports  
Création d'applications web .Net  
.NET Core vs .NET Framework  
ASP.Net MVC : présentation, versions, technologies utilisées  
Structure d'un projet ASP.Net MVC, choix du moteur de vue  
Configuration globale : dépendances, bundles, filtres, classe de configuration globale  
Ressources disponibles (documentation)

#### Atelier : Création d'un projet ASP.NET MVC Core, détail de la structure

### Implémenter des contrôleurs et maîtriser le routage

Contrôleur ASP.NET : implémentation, nommage  
Route : configuration par défaut, syntaxe  
Passage et récupération des paramètres dans les actions du contrôleur  
Mapping des paramètres et modification du routage dans le RouteConfig  
Choix du type de résultat de l'action : ActionResult, FileContentResult, ...  
Gestion du download

#### Atelier : Création de contrôleurs, routage et récupération de paramètres

### Créer des vues

Gestion des redirections depuis le contrôleur : vers une vue, vers une action, vers un flux  
Moteur de vue Razor : présentation, syntaxe  
Templating de vues : \_ViewStart (template par défaut), \_Layout personnalisé  
Accès aux bundles : styles, scripts

Création de vues : association à un template, structure de la page  
Bootstrap : grille responsive design, principales classes, composants disponibles  
Gestion des CSS de la vue  
Création de classes modèles, envoi dans le contrôleur et récupération dans la vue  
Création de formulaires : contrôles de base, restauration de valeurs après Postback  
Validation des entrées : gestion dans le modèle et affichage des erreurs  
Génération de vues associées à un modèle avec l'éditeur  
Internationalisation de la vue  
Gestion des sessions HTTP, accès dans la vue  
Manipulation de variables globales  
Création et récupération de cookies  
Gestion de l'upload  
Gestion de requêtes asynchrones AJAX  
Debug de vues : helpers (ServerInfo, ObjectInfo)

**Atelier : Implémentation d'un CRUD complet MVC, validation d'entrées et imports/exports**

### **Mettre en place une stratégie de sécurité**

Gestion de l'authentification : méthodes disponibles, critères de choix  
Implémentation d'une stratégie d'authentification, gestion des rôles  
Filtres d'authentification  
Sécurité des formulaires contre les injections  
AntiForgeryToken : intérêt, mise en place

**Atelier : Implémentation d'une stratégie d'authentification, sécuriser des formulaires contre les injections**

### **Optimiser l'application et déploiement**

Mise en place d'un cache  
Analyse du trafic  
Gestion des ressources (Minify, Bundle) dans une vue Razor  
Publication de l'application

**Atelier : Mise en place d'un cache, multiples optimisations de l'application**

### **Implémenter une couche d'accès aux données avec Entity Framework**

Architecture et technologies d'accès aux données  
Mapping relationnel objet  
Présentation du framework , outils  
Entity framework vs autres ORM (NHibernate)  
Entity Data Model vs Code First  
Contexte et chaine de connexion

**Atelier : Mise en place d'Entity Framework dans un projet sous Visual Studio**

### **Construire le modèle et gérer le contexte dans un contrôleur MVC**

Définition de la classe du contexte (DbContext), cycle de vie  
Gestion des logs  
Définition des classes du modèle et ajout dans le contexte  
Initialisateurs de Bdd

Conventions disponibles, ajouts, gestion des datetimes  
Mapping des tables : Attributs et API Fluent  
Gestion des clés primaires : simples, composées  
Gestion de la concurrence : Concurrency Check, Timestamp  
Mapping des colonnes calculées  
Relations : OneToMany/ManyToOne, OneToOne, ManyToMany  
Stratégies de mapping de l'héritage : TPH, TPC, TPS  
Génération d'actions CRUD dans un contrôleur  
Gestion de la sauvegarde en mode asynchrone

**Atelier : Création, modification d'un modèle de données complet**

### **Gérer les évolutions de la Bdd (Migrations)**

Migrations : définition, intérêts  
Activation des migrations, migration initial  
Ajout d'une migration et modification du code : requêtes SQL  
Application d'une migration : mise à jour ou régression de version  
Mise en production à une migration spécifiée

**Atelier : Création de manipulation de Bdd avec des migrations**

### **Manipuler les données du modèle**

Comprendre l'exécution des requêtes  
Gestion des connexions  
Maîtrise des transactions  
Passage d'une connexion existante  
Méthodes disponibles du DbContext : Création, mise à jour et suppression  
Gestion des chargements des associations : Lazy vs Eager  
Génération de procédures stockées pour les entités du modèle et utilisation  
Optimiser le cache du contexte, erreurs à éviter

**Atelier : Mise à jour des données et modification du modèle**

### **Ecrire des requêtes (LINQ, SQL)**

Panorama des méthodes de récupération de données disponibles  
Utilisation de LINQ to Entities, erreurs à éviter  
Requêtes avec Entity SQL  
Procédures stockées et gestion des résultats multiples  
Interrogations et modifications de données en utilisant les commandes ADO.Net  
Utilisation des DataSets  
Utilisation de LINQ to SQL  
Gestion des performances

**Atelier : Ecriture de multiples requêtes et traitement des résultats - gestion de la pagination / création des vues**

### **Notions avancées**

Construire des solutions optimisées en utilisant les services d'objet  
Utilisation des vues Entity Framework générées à la conception  
Surveillance des performances

Exécuter des modifications de données asynchrones  
Mise en cache de données hors-ligne en utilisant XML

**Atelier : Optimisations de requêtes, monitoring et mise en cache de données hors-ligne**