



Expert
EXPO
EX

ASP.NET

avec **C#** sous

Visual Studio 2017

Conception et développement
d'applications Web

Fichiers complémentaires
à télécharger



Brice-Arnaud GUÉRIN

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **EI17CASP** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1 Visual Studio 2017 et .NET

- 1. Nouveautés de Visual Studio 2017 15
 - 1.1 Installation 17
 - 1.2 Interface du logiciel 18
 - 1.2.1 La page de démarrage 18
 - 1.2.2 Les fenêtres de Visual Studio 20
 - 1.2.3 Les activités liées au développement 26
 - 1.2.4 Les packages NuGet 35
 - 1.2.5 Les tests codés d'interface utilisateur 37
 - 1.3 Gestion du code 42
 - 1.3.1 Le mode plan et les régions 42
 - 1.3.2 La refabrication (refactoring) 43
 - 1.3.3 Les extraits de code (code snippets) 44
 - 1.4 Documentation 46
 - 1.5 Contrôle du code source avec Visual Studio Online 46
 - 1.6 L'outil MS Build 56
- 2. C#5 en bref 58
 - 2.1 Classes partielles 59
 - 2.2 Méthodes anonymes 60
 - 2.2.1 Les événements internes 60
 - 2.2.2 Les fonctions auxiliaires 63
 - 2.2.3 Simplifier l'écriture du code 65
 - 2.3 L'inférence de type 67

2.4	Les expressions lambdas	67
2.5	Classes dynamiques et types anonymes	68
2.6	Extension de classes sans héritage	69
2.7	Types nullable	70
2.8	Itérateurs	71
2.8.1	Itérateur en C#1	72
2.8.2	Itérateur à partir de C#3	73
2.9	Généricité	74
2.9.1	Définir un type générique	75
2.9.2	Spécialisation partielle	77
2.9.3	Utilisation d'un type générique	77
2.9.4	L'espace de noms System.Collections.Generic	78
2.9.5	L'interpolation	78
3.	Les variantes de .NET	79
3.1	.NET Core	79
3.2	.NET Standard	81

Chapitre 2 **Les sites web ASP.NET**

1.	Le modèle de compilation	83
1.1	Du CGI au modèle ASP.NET 1.X	83
1.1.1	L'interface CGI	84
1.1.2	Les pages dynamiques ASP	87
1.2	Des classes partielles pour les pages	88
1.2.1	Structure d'une page ASPX	88
1.2.2	Modifications d'une page ASPX	90
1.3	Le code partagé dans App_Code	91
1.4	Les assemblages référencés	94
1.4.1	Références dynamiques	94
1.4.2	Références explicites dans le fichier Web.config	95
1.5	Le cache de construction	96
1.6	Les applications web de Visual Studio	96

- 2. Le rôle du serveur web 99
 - 2.1 Le serveur IIS. 99
 - 2.1.1 Le filtre ISAPI pour ASP.NET 99
 - 2.1.2 Création d'un site web ASP.NET avec IIS 100
 - 2.2 Le serveur de développement ASP.NET 102
- 3. Le pipeline HTTP de IIS 103
 - 3.1 Fonctionnement de IIS. 103
 - 3.1.1 Premiers pas sous HTTP avec Telnet 104
 - 3.1.2 Détail du traitement IIS. 105
 - 3.2 La classe HttpContext 107
 - 3.3 La classe HttpApplication 108
 - 3.3.1 Cycle de vie de l'application. 108
 - 3.3.2 Ajouter un fichier Global.asax 109
 - 3.3.3 Créer un module HTTP 113
 - 3.4 Les gestionnaires (handlers) HTTP 116
 - 3.4.1 Créer un handler ASHX 117
 - 3.4.2 Créer un handler dans une DLL 120

Chapitre 3
Les Web Forms

- 1. Présentation des Web Forms 123
 - 1.1 Structure d'une page ASPX 124
 - 1.1.1 Style imbriqué, en ligne et séparé 128
 - 1.1.2 Les scriptlets 131
 - 1.1.3 Hiérarchie des contrôles 133
 - 1.1.4 Ajouter dynamiquement des contrôles 136
 - 1.1.5 Objets intrinsèques 137
 - 1.2 Cycle de vie d'une page 139
 - 1.2.1 Le cycle nominal 139
 - 1.2.2 Identifier les requêtes de type postback 143

1.3	Les contrôles web	143
1.3.1	Les balises HTML	144
1.3.2	L'attribut runat="server"	145
1.3.3	Les contrôles HTML	147
1.3.4	Les contrôles web	148
1.3.5	Les contrôles à base de modèles (template)	152
1.3.6	Les contrôles utilisateurs et les contrôles personnalisés	152
1.4	Navigation entre les pages	153
1.4.1	Les liens hypertextes	153
1.4.2	Redirections par serveur	154
1.5	Postback et cross postback	155
1.6	Les callback	157
1.7	Validation des entrées utilisateur	163
1.7.1	Principe de la validation	163
1.7.2	Les contrôles de validation	166
1.7.3	La validation personnalisée	170
1.7.4	La validation discrète	171
2.	Organiser la présentation	175
2.1	Thèmes et skins	175
2.1.1	Les feuilles de style CSS	175
2.1.2	D'autres approches des CSS	176
2.1.3	Les thèmes	178
2.1.4	Les skins	181
2.2	Les contrôles utilisateurs .ascx	183
2.2.1	Créer un contrôle utilisateur	184
2.2.2	Utiliser un contrôle utilisateur	185
2.2.3	Ajouter des propriétés et des événements	186
2.3	Les pages maîtres (master pages)	190
2.3.1	Créer une page maître	191
2.3.2	Créer une page de contenu	194
2.3.3	Programmer les pages maîtres et les pages de contenu	197
2.3.4	Appliquer dynamiquement une page maître	199

- 3. Les composants personnalisés 200
 - 3.1 Fonctionnement des composants personnalisés. 200
 - 3.1.1 Les types de composants personnalisés (custom controls) 200
 - 3.1.2 Création d'une bibliothèque de composants. 201
 - 3.1.3 Mise au point du composant ColoredPad. 202
 - 3.1.4 Enregistrement et tests 210
 - 3.2 NumericTextBox, un composant dérivé de TextBox. 212
 - 3.2.1 Création du contrôle 212
 - 3.2.2 Propriétés et événements 212
 - 3.2.3 Rendu 214
 - 3.3 ChartControl, un composant graphique utilisant GDI+ 215
 - 3.3.1 Fonctionnement 215
 - 3.3.2 Rendu 217
 - 3.3.3 Intégration et tests 217
 - 3.4 PictureBox, un composant basé sur un modèle. 218
 - 3.4.1 Fonctionnement 219
 - 3.4.2 Implémentation du composant 221
 - 3.4.3 Les modèles 222
 - 3.4.4 Le rendu. 223
 - 3.4.5 Les événements. 226
 - 3.4.6 Informations relatives à la conception dans Visual Studio 227
 - 3.4.7 Utilisation du composant 228
 - 3.5 Des ressources incorporées aux DLL 230
- 4. AJAX 232
 - 4.1 Du callback à AJAX 232
 - 4.2 Le gestionnaire de script ScriptManager 233
 - 4.3 Le composant UpdatePanel 237
 - 4.3.1 Fonctionnement 237
 - 4.3.2 Mise en œuvre 237
 - 4.3.3 Gestion des erreurs 239
 - 4.3.4 Les triggers 241

4.4	Le composant UpdateProgress	242
4.5	Le Timer	243
4.6	La programmation objet avec JavaScript	244
4.6.1	Insertion de code JavaScript dans une page	244
4.6.2	Créer des objets et des classes JavaScript	246
4.6.3	Le style AJAX	249
4.6.4	Des classes dérivées	250
4.6.5	Implémenter des interfaces	251
4.7	Le modèle d'extension AJAX	252
4.7.1	Structure du framework	252
4.7.2	La classe d'application	254
4.7.3	Les contrôles AJAX du toolkit	257
4.7.4	Définir des contrôles personnalisés en JavaScript	261
4.8	Introduction à jQuery	267
4.8.1	Installation	267
4.8.2	Parcourir le DOM	268
4.8.3	Intervenir sur la page	269
4.8.4	Les plug-ins	272

Chapitre 4

Les sites web MVC

1.	L'approche MVC	275
1.1	Le design pattern MVC	275
1.2	Les évolutions de MVC	277
2.	Les sites ASP.NET MVC	277
2.1	Création d'un site	277
2.2	Organisation des répertoires	279
2.3	Création du modèle	280
2.4	Définition du contrôleur	283
2.5	Ajout des vues	285
3.	Définition des routes	289

- 4. Aller plus loin 290
 - 4.1 D'une action à l'autre 290
 - 4.2 Mise à jour du modèle et redirection 296
 - 4.3 Validation 296
- 5. Le moteur de rendu Razor et les vues 298
 - 5.1 La syntaxe C# dans les vues CSHTML 298
 - 5.1.1 Principes de base 298
 - 5.1.2 Les balises Action 300
 - 5.1.3 Les méthodes de formulaires 302
 - 5.1.4 Créer ses propres extensions HTML 303
 - 5.2 Structure et organisation des vues 305
 - 5.2.1 Les gabarits Layout 305
 - 5.2.2 Les vues partielles 307
 - 5.2.3 Rendu des scripts et des bundles 308
- 6. Sécurisation des sites MVC 308
 - 6.1 Authentification 308
 - 6.2 Autorisations 310
- 7. Définir des zones (areas) 312
- 8. Les Single Page Applications (SPA) 313
 - 8.1 Utiliser les Web API 314
 - 8.1.1 Créer un projet Web API 314
 - 8.1.2 Établir un modèle et un contrôleur 315
 - 8.1.3 La page unique 316
 - 8.2 Utiliser KnockOut pour la liaison de données 318

Chapitre 5
ASP.NET Core

- 1. Un site web ASP.NET Core 321
 - 1.1 Création du projet 321
 - 1.2 Contenu du projet 323

2.	Configuration	325
2.1	Les fichiers Program et Startup	325
2.1.1	Program	325
2.1.2	La classe Startup	325
2.2	La configuration JSON	327
2.2.1	appSettings.json	327
2.2.2	launchSettings.json	327
2.2.3	bundleConfig.json	328
2.3	Gestion des packages avec NuGet et Bower	329
2.3.1	Les packages NuGet	329
2.3.2	Les packages Bower	330
2.4	Application de thèmes avec Bootstrap	331
3.	Développement MVC	333
3.1	Les contrôleurs web	333
3.2	Les vues	334
4.	Définir des environnements d'exécution	335
4.1	Détection de l'environnement d'exécution	335
4.2	Définition d'environnements	336

Chapitre 6

L'accès aux données avec ADO.NET

1.	Les bases d'ADO.NET	339
1.1	Le mode connecté	339
1.1.1	La connexion	340
1.1.2	La commande	342
1.1.3	Le DataReader	344
1.1.4	Les paramètres	347
1.1.5	Les transactions	348
1.2	Les bases de données SQL Server	352
1.2.1	Les déclinaisons du logiciel SQL Server	352
1.2.2	Création de bases	354
1.2.3	Création de tables	357

1.2.4	Les vues	358
1.2.5	Les procédures stockées	359
1.3	Rendre l'accès aux bases transparent	360
1.3.1	Le mode déconnecté	361
1.3.2	DataAdapter et TableAdapter	363
1.3.3	Le mapping objet-relationnel et les frameworks spécialisés	370
1.3.4	Les fabriques ADO.NET	370
2.	Accès aux données à base de fournisseurs	374
2.1	Introduction au développement par fournisseurs	374
2.1.1	Contrôles sources de données en mode fournisseur	375
2.1.2	Contrôles de présentation des données	376
2.2	Les sources SqlDataSource et AccessDataSource	377
2.2.1	La requête de sélection	377
2.2.2	Les requêtes de mises à jour	380
2.2.3	Les paramètres	381
2.2.4	Le cache	384
2.3	Le fournisseur ObjectDataSource	385
2.3.1	Le principe	385
2.3.2	Mise en œuvre	386
2.3.3	Paramètres de création	390
2.3.4	Gestion du cache	391
2.3.5	Une version avancée	391
2.4	Le fournisseur XmlDataSource	398
2.5	LinqDataSource	402
2.5.1	Un DAO pour LinqDataSource	402
2.5.2	Le contexte de données .dbml	404
2.5.3	Les événements de LinqDataSource	408
2.6	EntityDataSource	409
2.6.1	Le framework Entity	409
2.6.2	Créer le modèle conceptuel	411
2.6.3	Requêter avec LINQ to Entities	416
2.6.4	Mise en œuvre du composant EntityDataSource	417

3.	Les composants graphiques de présentation des données	420
3.1	Le composant GridView	420
3.1.1	Présentation tabulaire des données	420
3.1.2	Les opérations de sélection et de navigation	423
3.1.3	Les clés et les opérations de mise à jour	424
3.1.4	Les formatages et les tris	426
3.1.5	Les colonnes modèles	428
3.1.6	La liaison bidirectionnelle	429
3.1.7	Gérer les jointures	430
3.2	Le composant DetailsView	434
3.2.1	Présentation du DetailsView	434
3.2.2	Les événements	435
3.3	Le composant FormView	436

Chapitre 7

Gestion de l'état

1.	Les différents moyens pour maintenir l'état	437
1.1	Les champs cachés	437
1.2	Le ViewState	438
1.2.1	Utiliser le ViewState dans un Web Form	439
1.2.2	Contrôler l'application du ViewState	440
1.3	La chaîne de requêtes (Query String) et les URI	441
1.4	Les cookies	442
2.	Les sessions	443
2.1	Utilisation de l'objet Session	443
2.1.1	Mémorisation d'un objet et recherche	444
2.1.2	Initialisation de l'objet Session	444
2.1.3	Sécurisation du jeton de session	445
2.2	Sessions sans cookie et délai d'abandon de session	445
2.2.1	Sessions sans cookie	445
2.2.2	Timeout	446

- 2.3 Services de conservation des données en session 446
 - 2.3.1 Le processus en mémoire InProc 446
 - 2.3.2 Le service Windows ASP.NET State Service 448
 - 2.3.3 Le service SQL Server 449
 - 2.3.4 Services personnalisés 450
- 3. Les objets Application et Cache 451
 - 3.1 L'objet Application 451
 - 3.1.1 Utilisation 451
 - 3.1.2 Verrouillage 451
 - 3.2 Le cache de données d'applications Cache 452
 - 3.2.1 Les dépendances de temps 452
 - 3.2.2 Le callback 454
 - 3.2.3 Les dépendances fichiers 455
 - 3.2.4 Les dépendances SQL sous SQL Server 456
 - 3.3 Le cache HTML 458
 - 3.3.1 Cache de sortie 458
 - 3.3.2 Fragments de pages en cache 460
 - 3.3.3 Les substitutions 461
 - 3.3.4 Les profils de cache 462

Chapitre 8
Personnalisation et sécurisation

- 1. Sécurisation des sites ASP.NET 463
 - 1.1 Le modèle de sécurisation du site 463
 - 1.1.1 Les objets de la sécurité 463
 - 1.1.2 L'authentification 464
 - 1.1.3 Les autorisations 465
 - 1.2 Sécurisation en mode Windows 466
 - 1.2.1 Activation du mode d'authentification 466
 - 1.2.2 Configuration de IIS 467
 - 1.2.3 Autorisations 468

1.3	Sécurisation en mode Forms	469
1.3.1	Activation du mode Forms et création d'une page de connexion	470
1.3.2	Endossements de rôles	472
1.3.3	Le mode Forms sans cookie	475
1.3.4	Autorisations	475
1.4	Le fournisseur MemberShip	475
1.4.1	Fonctionnement du fournisseur	475
1.4.2	Utiliser AspNetSqlMembershipProvider	478
1.5	Sécurisation en comptes d'utilisateurs individuels	481
1.6	Le répertoire Account	483
1.7	Le référentiel local d'utilisateurs	485
1.8	Activer un référentiel externe	487
1.9	Le fournisseur de rôles	490
1.9.1	AspNetSqlRoleProvider	490
1.9.2	WindowsRoleTokenProvider	491
1.10	Les contrôles intégrés	492
2.	Présentation personnalisée	493
2.1	Les profils utilisateur	493
2.1.1	Formation du profil	493
2.1.2	Utilisation du profil	494
2.1.3	Groupage et types complexes	495
2.2	Navigation au sein du site	497
2.2.1	Le fichier de définition du site	498
2.2.2	Le fournisseur SitemapProvider, l'API Sitemap et le SitemapDataSource	499
2.2.3	Les contrôles associés à la navigation	499
2.2.4	Filtrer l'affichage selon le rôle de l'utilisateur	500
2.3	Internationalisation	501
2.3.1	Les ressources globales	501
2.3.2	Les ressources locales	503
2.3.3	Le composant Localize	504
2.3.4	Localisation des validations	505

- 3. Les WebParts 506
 - 3.1 Du site web au portail 506
 - 3.2 Créer un portail 506
 - 3.2.1 Le gestionnaire WebPartManager 507
 - 3.2.2 Les zones WebPartZone 508
 - 3.2.3 Les éléments WebPart 509
 - 3.3 Les contrôles catalogues CatalogZone et PageCatalogPart . . . 510
 - 3.3.1 Le catalogue de zones 510
 - 3.3.2 Un menu pour changer de mode 512
 - 3.3.3 Donner des noms aux éléments 513
 - 3.3.4 Les éditeurs 513
 - 3.4 Créer des éléments personnalisés 515
 - 3.4.1 Créer un WebPart à partir d'un composant utilisateur . 515
 - 3.4.2 Créer un WebPart personnalisé 516
 - 3.4.3 Connecter les éléments 519

Chapitre 9
Les services web WCF et REST

- 1. Les services web WCF 523
 - 1.1 Le dialecte commun SOAP 524
 - 1.2 Créer un service web WCF 526
 - 1.2.1 Implémentation du service 526
 - 1.2.2 Test du service 530
 - 1.3 Consommer un service web 531
 - 1.3.1 Génération du proxy 531
 - 1.3.2 Appel synchrone 534
 - 1.3.3 Appel asynchrone 535
- 2. Les services web REST 538
 - 2.1 Implémentation d'un service REST 539
 - 2.2 Utilisation d'un service REST 540

Chapitre 10**Configuration, déploiement et administration**

1. Configuration	543
1.1 Héritage de la configuration	543
1.2 Configuration de test et de production.	545
1.2.1 Le gestionnaire de configuration de Visual Studio	545
1.2.2 Plusieurs fichiers de configuration Web.config.	546
1.2.3 Les pages d'erreurs du fichier Web.config	547
2. Déploiement des applications ASP.NET	547
2.1 Déploiement manuel	547
2.1.1 Création d'un répertoire virtuel	547
2.1.2 Sélection des fichiers à copier	549
2.1.3 La page par défaut	550
2.2 Déploiement par le système de copie	552
2.3 Déploiement avec Microsoft Azure.	556
2.3.1 Création d'un compte Azure	556
2.3.2 Vue d'ensemble de l'interface de gestion des services.	557
2.3.3 Création d'un projet associé à un compte Azure	558
2.3.4 Développement de l'application	560
3. Supervision des applications ASP.NET	561
3.1 L'infrastructure de supervision Health Monitoring	561
3.1.1 La hiérarchie des événements web	561
3.1.2 La hiérarchie des fournisseurs	562
3.2 Mise en œuvre dans ASP.NET.	562
3.2.1 Déclarer des événements	563
3.2.2 Déclarer des fournisseurs d'écoute.	563
3.2.3 Ajouter des règles d'abonnement.	563
Index	565

Chapitre 4

Les sites web MVC

1. L'approche MVC

Passé l'époque du tâtonnement quant à la structure d'une application web, l'univers Java a popularisé l'utilisation de frameworks tels que Struts ou Spring. Ceux-ci, et Struts en premier lieu, ont jeté les bases d'une séparation des responsabilités entre les différents niveaux d'une application web. Il est vrai que les premières technologies web n'incitaient pas les programmeurs à organiser leurs applications ; la maintenance s'en trouvait très délicate, en même temps que les performances restaient en berne.

1.1 Le design pattern MVC

L'expression MVC désigne une approche de conception généralisée, ou design pattern. Le but est de ne pas réinventer la roue à chaque application. Comme nous le verrons, MVC est un pattern assez simple. Ne pas l'adopter, c'est vraisemblablement aboutir à une application compliquée donc sans doute mal ficelée, ce qui nous ramènerait au cas évoqué ci-dessus.

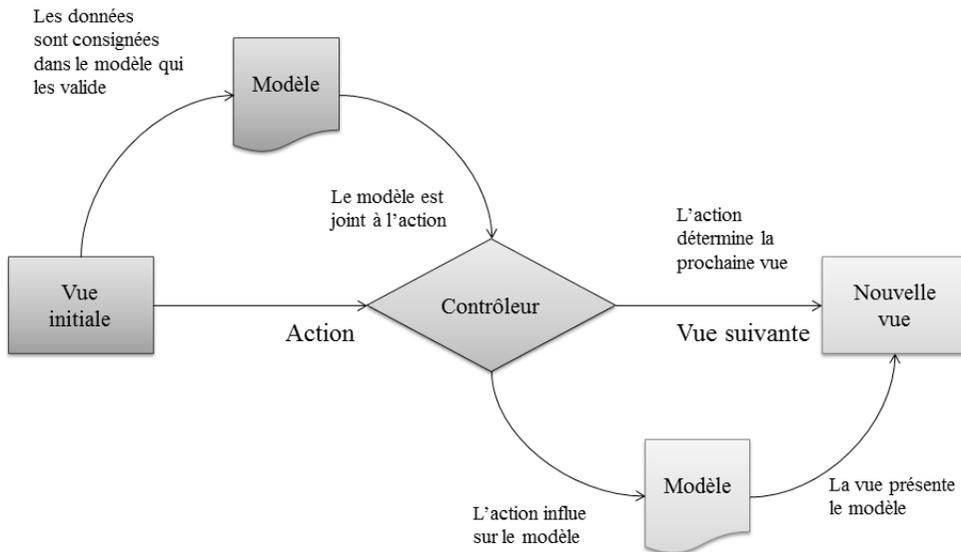
À chaque lettre de l'acronyme MVC correspond un rôle bien défini ; le modèle, la vue et le contrôleur.

Le modèle est un objet "métier", regroupant ses données, son comportement (ses méthodes) et ses règles de validation. Il ne contient normalement pas de logique technique (présentation, navigation). Il peut se voir attribuer des aspects (injection de services tels que la persistance fichier ou SQL, les transactions, la sécurité...). Dans les approches moins complètes, l'objet métier est associé à une classe de services qui sert d'interface (API).

La vue est chargée de restituer le modèle au sein d'une interface graphique (web dans notre cas), et de permettre à l'utilisateur d'interagir avec le modèle.

Le contrôleur définit les règles de navigation (on dit la cinématique). Le passage d'une vue à l'autre se fait au moyen d'actions conduites par le contrôleur. Le modèle est alors interrogé ou enrichi pour conditionner le déroulement des actions.

L'illustration suivante décrit la séquence d'interactions entre ces objets :



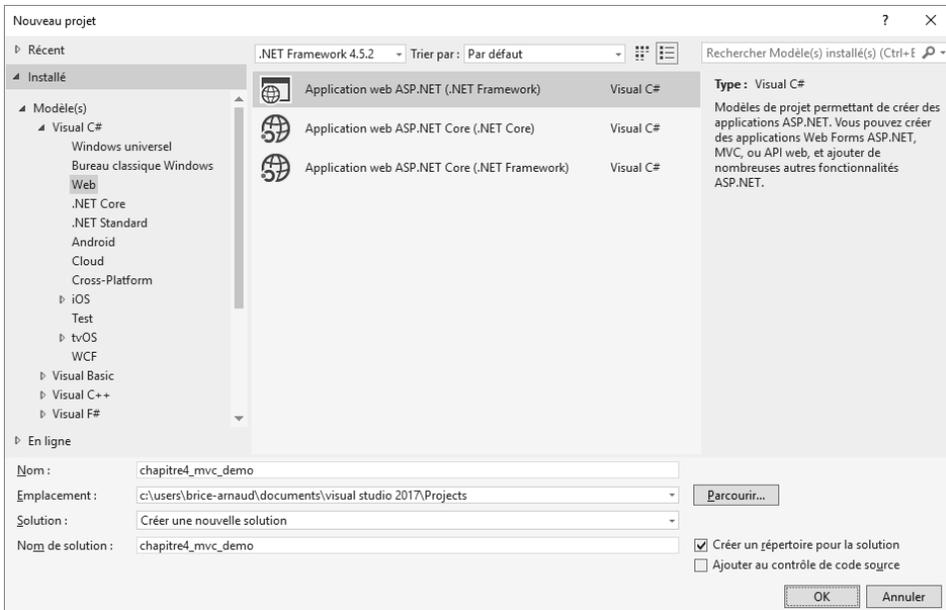
1.2 Les évolutions de MVC

L'approche MVC 2 est surtout une évolution du framework ; elle consiste à n'utiliser qu'un seul contrôleur pour plusieurs actions. Cette évolution réduit considérablement l'effort en programmation et en configuration. Par chance, le framework ASP.NET supporte d'ores et déjà le niveau MVC 2 sous les appellations ASP.NET MVC 3/4/5.

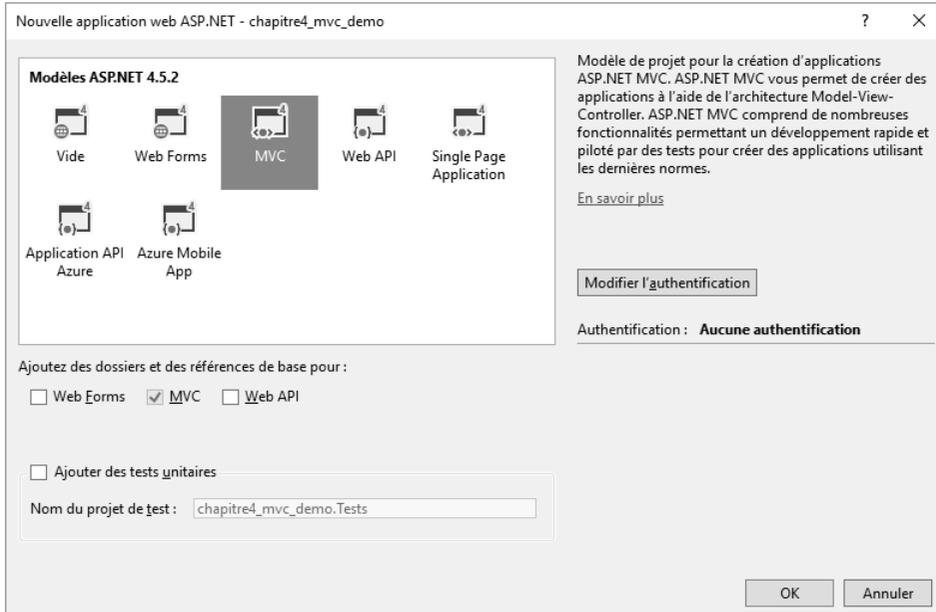
2. Les sites ASP.NET MVC

2.1 Création d'un site

La création d'un site web MVC s'effectue à l'aide de la commande **Nouveau projet** :



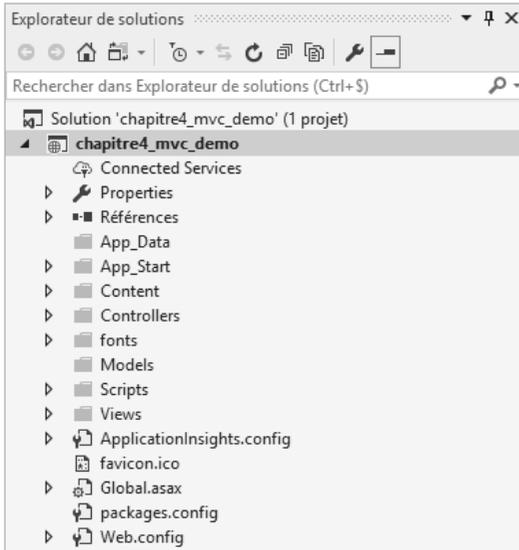
Puis nous sélectionnons le modèle MVC :



Comme l'application MVC nécessite l'emploi de classes qui ne sont pas dans le code sous-jacent (comme pour les Web Forms), le modèle Visual Studio n'est disponible qu'en projet web et non comme site web.

2.2 Organisation des répertoires

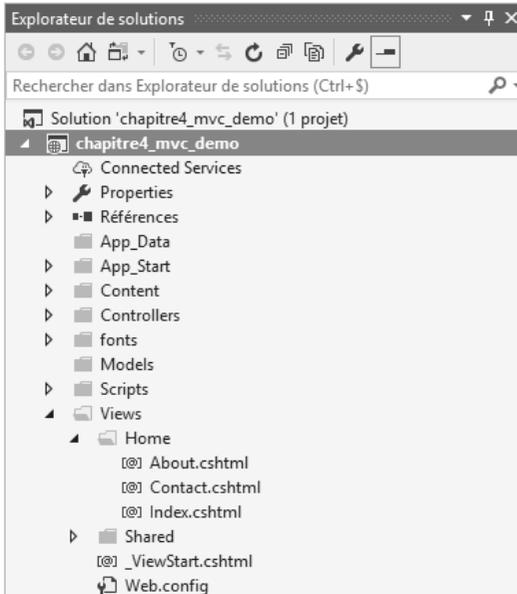
La solution du projet web contient beaucoup plus de répertoires qu'un projet Web Forms.



Ces répertoires ont pour but de guider le programmeur :

App_Start	Instructions de configuration exécutées au démarrage du site.
Content	Contient les feuilles de style CSS et autres ressources partagées.
Controllers	Regroupe les contrôleurs destinés à traiter les actions.
fonts	Polices de caractères téléchargées par le navigateur.
Models	Rassemble les modèles qui sont des entités métier.
Scripts	Ensemble de modules JavaScript, jQuery et AJAX.
Views	Vue .cshtml.

Les vues sont des pages web mais qui n'ont pas de code sous-jacent (voir ci-dessous). Elles sont en principe regroupées dans des dossiers appelés zones, lesquels correspondent à des contrôleurs. Cette règle n'a aucun caractère obligatoire d'un point de vue technique, même si l'emploi du framework s'en trouve facilité dans le cas où elle est appliquée.



2.3 Création du modèle

Un modèle est une classe dont les instances sont appelées "objets métier". Cela signifie qu'aucune logique technique n'y figure, et que le framework est chargé de gérer le cycle de vie du composant métier, et de lui apporter des services techniques de haut niveau tels que la sécurité, les transactions, la validation, la persistance...