

## Sujet du stage

### « Recherche et développement » autour de la technologie ASP.NET / RAZOR

#### Introduction

RAZOR a été conçu pour faciliter la conception des pages ASP.NET. Il introduit une syntaxe de programmation assez facilement compréhensible, qui vous permet d'insérer du code serveur dans une page Web qui peut également contenir du HTML, du CSS et des scripts JavaScript. RAZOR est un MOTEUR DE VUE pour ASP.NET MVC 3. C'est une alternative au moteur de vue « WebForm » d'ASP.NET. Les objectifs de ce moteur sont :

- Rendre le design des vues plus facile et plus clair
- Obtenir une syntaxe compacte et lisible
- Intégrer les blocs serveurs plus facilement au HTML.
- Etc...

#### Sujet du stage

Il s'agit d'une mission de RECHERCHE & DEVELOPPEMENT autour de la technologie RAZOR / ASP.NET de Microsoft. Les tâches à produire sont les suivantes :

#### **1) Développer des exemples de code, dont l'objectif est de montrer le principe de fonctionnement de RAZOR (=éléments de SYNTAXE de base, autour d'une ou plusieurs « Pages Web Razor »):**

- Variables et constantes, structures conditionnelles & itératives, tableaux....
- Balises HTML, formulaires & objets HTML tels que « label », « input », « text » etc...
- Utilisation de classes et d'objets C#, telles que « System.DateTime », par exemple.
- Etc...

#### **2) Développer des exemples autour de l'architecture MVC (« Modèle – Vue – Contrôleur »)**

Il s'agira tout d'abord de bien étudier le principe de création du MODELE, des VUES et du CONTROLEUR adopté par la technologie RAZOR.

En effet, il s'agira d'utiliser les « templates » (=modèles ou patrons) permettant de générer automatiquement le code nécessaire à chacune des parties de l'architecture. Ce code pourra, par la suite être modifié et/ou complété.

- Schéma et descriptif de l'adaptation de l'architecture MVC par la technologie RAZOR.
- Principe de création du MODELE, de la VUE et du CONTROLEUR.
- Plusieurs exemples de niveau de complexité croissants, dont certains peuvent reproduire des applications MVC précédemment étudiées en cours (=non obligatoire).
- Etc...

**Contraintes à respecter dans l'ensemble des applications MVC (=chaque contrainte devra se retrouver dans au moins une application :**

- MODELE : au moins une classe METIER avec des propriétés ayant des attributs *[DisplayName]*, *[Required]*, *[StringLength]*, *[DataType]*, *[Range]* etc... (=liste non exhaustive). Ces attributs vont permettre de générer du code de VALIDATION des données, dans la VUE associée au MODELE.
- MODELE : au moins une propriété gérée par un attribut *[RegularExpression]* (=par exemple un email ou un mot de passe...)
- MODELE : au moins deux classes TECHNIQUES d'accès aux données (=en « dur » puis une source au choix : XML, BD...)
- VUES déduites à partir du MODELE, avec affichage d'une LISTE d'objets METIER (=une ligne avec les valeurs des propriétés, par objet métier), puis la possibilité d'AJOUTER, MODIFIER ou SUPPRIMER des éléments de la liste.

Enfin, le code devra être commenté, d'autant plus que le MODELE, puis les VUES et le CONTROLEUR générés sont intimement liés au MODELE.

**3) Une application permettant d'illustrer le principe de VUES PARTIELLES (« Partial View »)**

**4) Une application permettant d'illustrer le principe des « dictionnaires » au sein du CONTROLEUR :** propriétés « ViewData » & « ViewBag » qui permettent de passer des données depuis un CONTROLEUR à une VUE. Montrer les différences d'utilisation pour ces deux propriétés qui remplissent le même rôle.

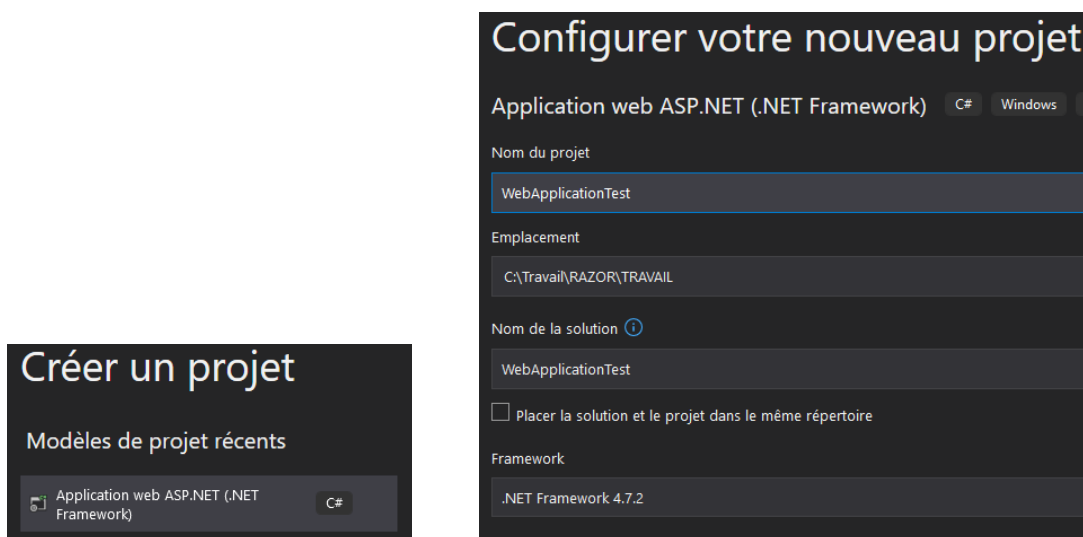
**5) Enfin, une ou plusieurs applications, pas nécessairement MVC, illustrant et mettant en œuvre au moins TROIS éléments de CYBERSECURITE (=au choix).**

**6) Rédaction d'un MANUEL TECHNIQUE / UTILISATEUR :**

- Création d'une application RAZOR « simple ».
- Création d'une application RAZOR MVC.
- Mode de fonctionnement de l'architecture MVC dans RAZOR.
- Etc...

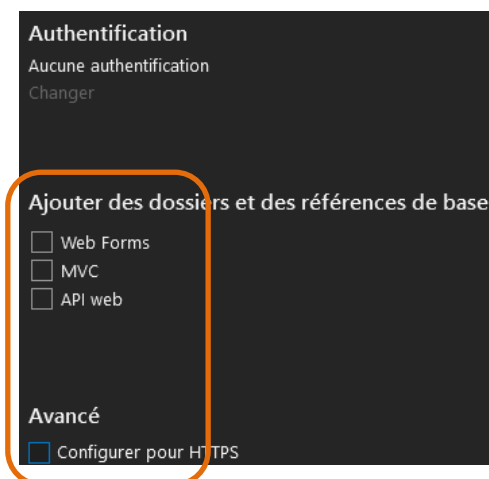
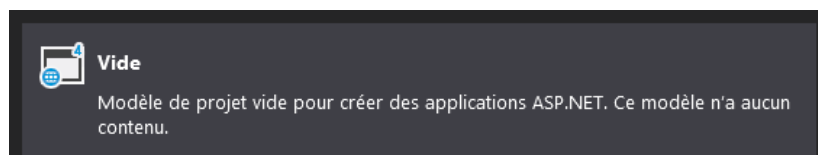
**ANNEXE - Principes de création d'une application ASP.NET / RAZOR (Visual Studio 2019)**

**Application Web VIDE, puis ajout d'éléments (ex : une ou plusieurs Pages Web RAZOR)**



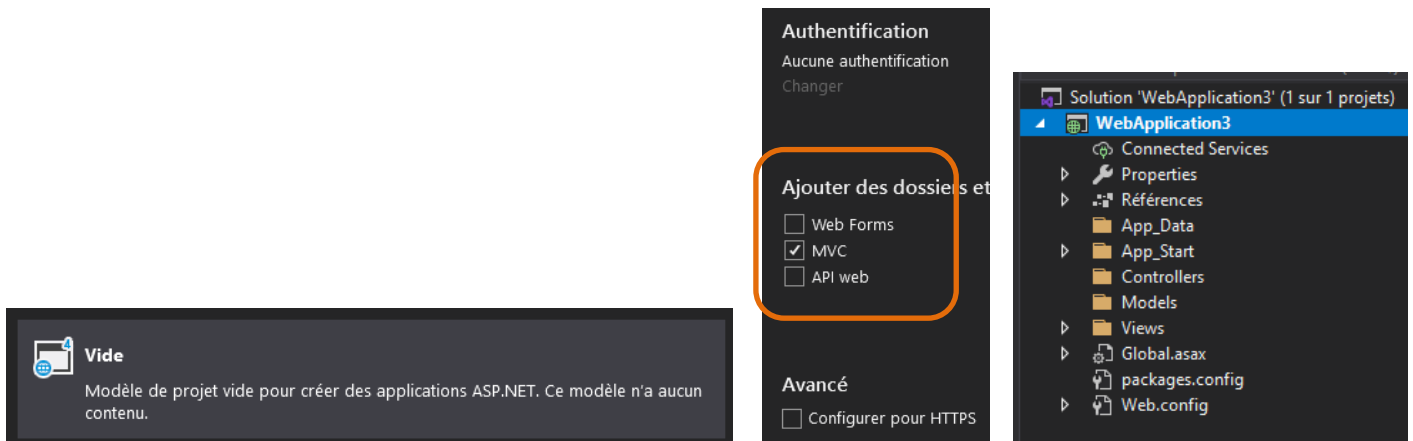
Deux possibilités :

- 1) Sans la mise en œuvre de l'architecture MVC. Il suffira d'ajouter des pages web (=VUES)

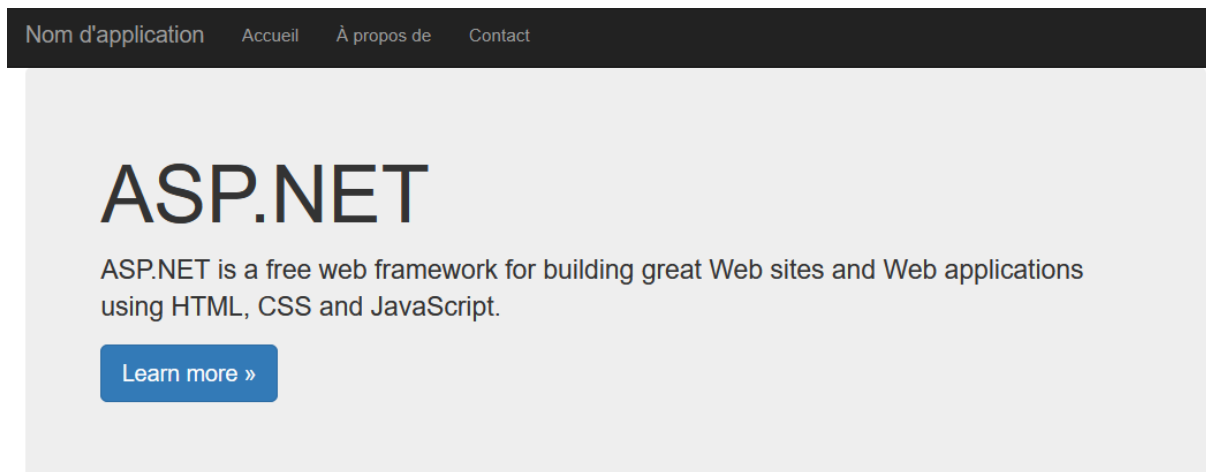
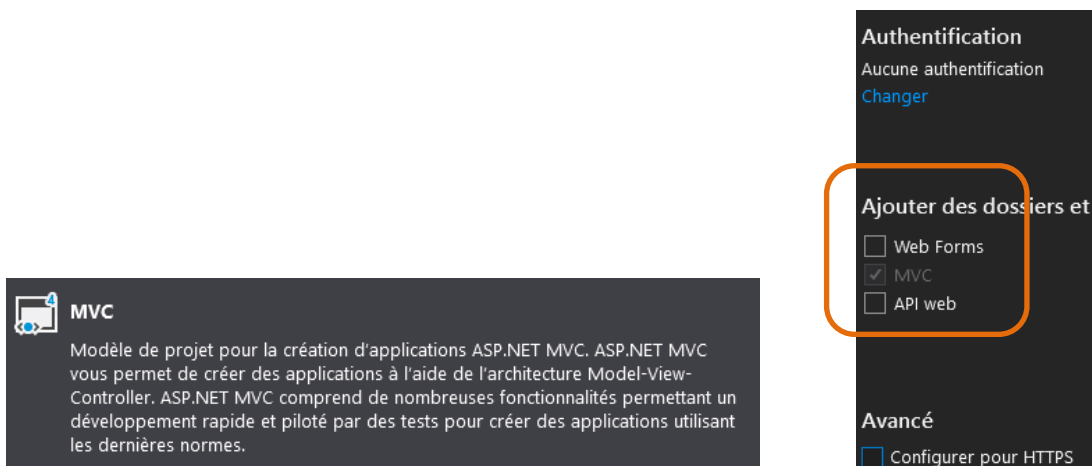


## 2) Avec mise en œuvre de MVC

- Organisation « VIDE » (les dossiers uniquement ; le reste est à ajouter et à placer au « bon » endroit)



## Application MVC (cf. « template ») à ADAPTER ou COMPLETER



### Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

### Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

### Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)