

ASP.NET MVC

Nouhaila Bensalah

Chercheuse en IA/NLP

nouhaila.bensalah@etu.fstm.ac.ma



ASP.NET | MVC | Web API

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

C'est quoi .NET

- Un standard proposé par la société Microsoft en 2002.
- Une plateforme de développement, gratuite et open source qui supporte un grand nombre de langages de programmation.
- Un cadre de travail formé d'outils, de modules actifs au runtime et de classes formant une API très étendue.
- Un label marketing.
- Un environnement d'exécution sécurisé.
- Le remplaçant de l'architecture 3-tiers DNA (Distribued interNet Architecture).

Outline I

1 Dot Net

- C'est quoi .NET
- **L'architecture .NET**
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

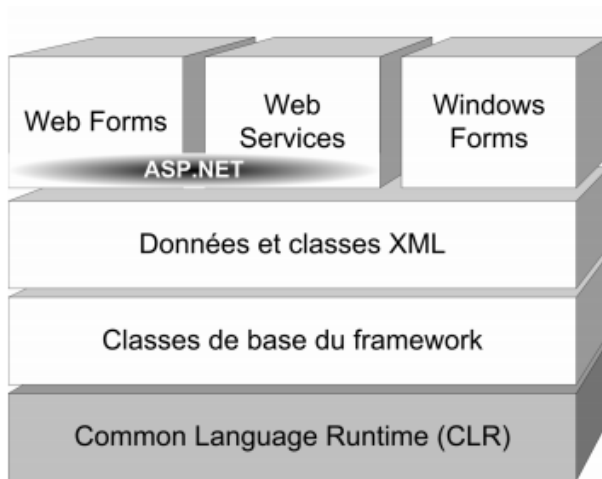
3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

L'ensemble .NET peut être divisé en quatre pôles:

- ❶ Des outils de développement comme Visual Studio .NET ou Web Matrix ...
- ❷ Les techniques Web incluant les services Web et les sites dynamiques avec ASP.NET.
- ❸ La gestion des données avec ADO.NET et les serveurs Microsoft spécialisés pour les e-mails, le stockage des données ...
- ❹ Le support de plate-formes techniques telles que les téléphones cellulaires, les consoles de jeux ...

L'architecture .NET



- **Common Language Runtime (CLR):** La machine virtuelle de .NET, qui exécute le code IL appelé aussi CIL (Common Intermediate Language).
- **Bibliothèque de classes:** Une très riche collection de types réutilisables pour développer des applications.
- **Données et classes XML:**
 - Les classes XML permettent de manipuler les données XML.
 - Les classes d'accès aux données (ADO.NET) sous format XML.
- **Web Services, Web Forms, Windows Forms:** Création de Web Services, de pages Web, et d'applications Windows.

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- **Les bases .NET**
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

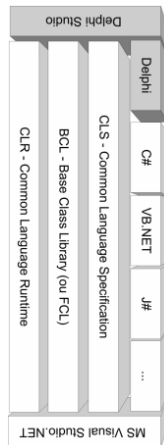
3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

Le CLR

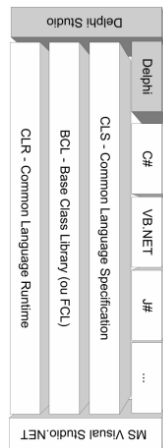
Ce runtime est le moteur de .NET, c'est lui qui est en charge de l'exécution des logiciels écrits pour .NET compilés en CIL. Il utilise JITer (Just in Time enhanced runtime). Le CLR gère:

- La compilation en code natif et l'exécution de code IL.
- La gestion de la sécurité.
- La gestion de la mémoire.
- La gestion des processus.
- La gestion des threads (tâches).



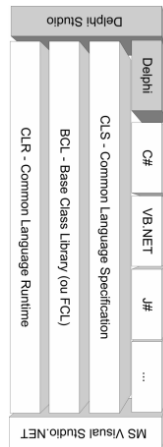
La BCL ou FCL

- Se plaçant directement au dessus du CLR
- Offre de nombreuses classes, interfaces et types qui forment le socle des développements sous .NET
- Permet d'unifier les développement puisqu'elle fournit l'ensemble des outils de base aux applications



Les espaces de nom

- Une sorte de "boîte" dans laquelle des classes ayant (plus ou moins) rapport entre elles sont regroupées. Par exemple:
 - System.Data.SqlClient
 - System.Web
- Une Assembly (assemblée en français) est un fichier DLL dans lequel les classes sont stockées.



1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

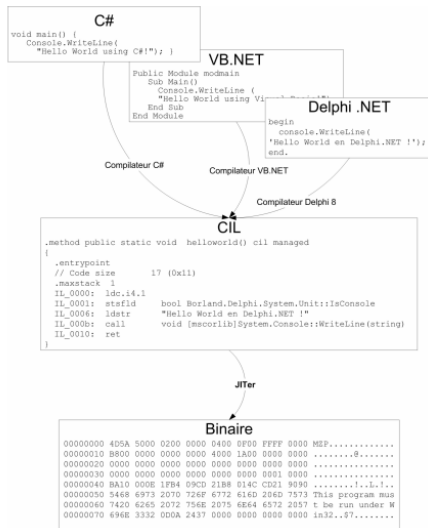
2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

Du code source au binaire exécuté



Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

C'est quoi ASP.NET

- Plateforme de développement d'applications web sous Windows
- Utilise, par défaut, le serveur Web de référence de Microsoft: IIS (pour Internet Information Services)
- Repose sur le .NET Framework
- Plateforme Web unifiée fournissant les services nécessaires à la création d'applications d'entreprise
- Ne dépend ni du langage de programmation ni du navigateur

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- **MVC**
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

- Introduit par Trygve Reenskaug en 1978
- Permettant de bien organiser le code source
- Conçu, initialement, pour les applications client-lourd et ensuite généralisé aux applications client-léger (Web)
- Une approche consistant à séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données
- Chacun de ces composants est construit pour manipuler un aspect particulier de développement de l'application
- Indispensable pour des applications dynamiques et de taille importante

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- **MVC: Les trois composants**
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

- ➊ **Modèle:** La partie qui concerne les données et l'état de notre application. Pour le cas d'une base de données, on peut utiliser un ORM (Object-Relational Mapping) comme Entity Framework pour ASP.NET MVC
- ➋ **Vue:** La partie qui concerne l'affichage : l'interface avec laquelle l'utilisateur interagit (HTML + CSS...)
- ➌ **Contrôleur:** c'est l'intermédiaire entre le modèle et la vue. Il reçoit la requête de l'utilisateur. Il demande les données au modèle, les analyse et renvoie le résultat à afficher à la vue

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- **ASP.NET MVC**
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

- Créé en 2007 par Scott Guthrie et intégré dans ASP.NET depuis 2009
- Framework de développement d'applications web selon le design pattern: MVC
- Par rapport à ASP.NET:
 - ASP.NET MVC permet de structurer d'avantage l'application, en créant des composants avec des rôles bien identifiés
 - ASP.NET MVC permet de structurer d'avantage l'application, en créant des composants avec des rôles bien identifiés
 - Indispensable pour des applications dynamiques et de taille importante

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- **ASP.NET vs ASP.NET Core**
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

ASP.NET

- Pour Windows.
- Utilise le runtime .NET Framework.
- Bonnes performances.
- Supporte C#, F# et VB.
- Disponible sous Visual Studio.

ASP.NET Core

- Pour Windows, Mac et Linux.
- Utilise le runtime .NET Core.
- Plus performant que ASP.NET.
- Supporte C# et F#.
- Disponible sous VS, VSC.

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

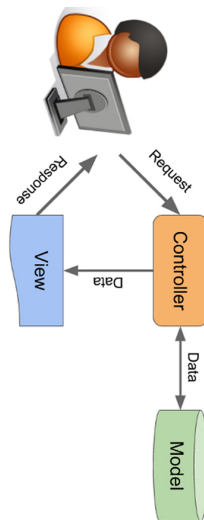
2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- **ASP.NET MVC: Déroulement**

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

- 1 Le client envoie une requête depuis la **Vue**
- 2 Le **Contrôleur** intercepte et analyse la requête du client
- 3 Le **Contrôleur** détermine quelle partie du **Modèle** est concernée afin d'effectuer les traitements nécessaires
- 4 Le **Modèle** s'occupe de l'interaction avec les données, applique les règles métier et renvoie les données au **Contrôleur**
- 5 Le **Contrôleur** sélectionne la **Vue** correspondante et lui injecte les données
- 6 La **Vue** affiche les données au client



- MVC est un patron de conception particulièrement adapté pour réaliser une application web
- MVC est l'acronyme de Modèle-Vue-Contrôleur :
 - ❶ Le Modèle contient les données de l'application
 - ❷ la Vue contient le code pour afficher les pages de l'application
 - ❸ le Contrôleur gère les interactions de l'utilisateur en faisant le lien avec le Modèle et la Vue
- MVC permet une séparation claire des intentions et optimise la création d'une application web ou d'un site, sa maintenance et ses tests automatisés.

Exemple d'applications web:

- Stack OverFlow (forum de discussion entre développeurs)
- CodePlex (dépôt de projets)

IDE pour ASP.NET MVC:

- Web Matrix (pour les petites applications)
- MonoDevelop (pour Windows et Mac)
- **Visual Studio**

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

Étapes à suivre

- Créer un nouveau projet **Fichier** → **Nouveau** → **Projet**.
- Cliquer sur **Installé** et choisir **C#**.
- Étendre la rubrique **Web** et sélectionner **Application web ASP.NET (.NET Framework)**.
- Remplir le champs **Nom** par: **BonjourMVC**.

Étapes à suivre: Plusieurs types de projets

- **Vide** sans cocher la case **MVC**: projet complètement vide.
- **Vide** en cochant la case **MVC**: projet avec des répertoires vides (**Models**, **Views**, **Controllers...**).
- **MVC**: projet avec des répertoires (**Models**, **Views**, **Controllers...**) non vides + inclusion de Bootstrap et jQuery.
- **WEB API**: projet non vide pour la création de services RESTful.
- ...

- Sélectionner un modèle **Vide** et cocher la case **MVC**.
- Valider et attendre la fin de création du projet.

Structure du projet

- Une application MVC \Rightarrow
 - **Models**: où on va placer les classes qui vont interagir avec les données.
 - **Views**: où on va placer les pages (HTML/CSS, etc) qui serviront de réponse à la requête utilisateur.
 - **Controllers**: où on va mettre les classes qui contiendront les actions à exécuter des réception d'une requête.
- **packages.config**: fichier de configuration des packages **NuGet**.
- **App Start**: où on va placer les fichiers de configuration. Il contient **RouteConfig.cs** qui servira de mapping route/contrôleur/action.
- **Global.asax**: point d'entrée vers l'application. Il fait appel à **RouteConfig** présent dans **App_Start**.

Outline I

- 1 Dot Net
 - C'est quoi .NET
 - L'architecture .NET
 - Les bases .NET
 - Du code source au binaire exécuté

- 2 ASP.NET
 - C'est quoi ASP.NET
 - MVC
 - MVC: Les trois composants
 - ASP.NET MVC
 - ASP.NET vs ASP.NET Core
 - ASP.NET MVC: Déroulement

- 3 Création d'une page web avec ASP.NET
 - Création d'une page web avec ASP.NET: Contrôleur
 - Création d'une page web avec ASP.NET: Vue

Erreur HTTP 404

- Si on exécute, on obtient une erreur HTTP 404.
- Aucune action associée à la route /.
- D'ailleurs, le répertoire **Controllers** est vide \equiv pas de contrôleur \Rightarrow pas d'actions.
- **Solution:** Créer un contrôleur (et une action) et l'associer à cette route.

Traiter les informations: création du contrôleur

- Faire un clic droit sur **Controllers** et aller dans **Ajouter** \rightarrow **Contrôleur**.
- Sélectionner **MVC 5 Controller - Empty** et cliquer sur **Ajouter**.
- Saisir un nom pour le contrôleur (par exemple : **HomeController**) et cliquer sur **Add**.
- Une classe **HomeController** qui hérite de **Controller** (du namespace **Web.Mvc**) est générée.

Traiter les informations: création du contrôleur

```
public class HomeController : Controller
```

```
{
```

```
    // GET: Home
```

```
    public ActionResult Index()
```

```
    {
```

```
        return View();
```

```
    }
```

```
}
```

C'est quoi?

- Classe C# héritant de la classe **Controller**.
- Ayant des méthodes **public** appelées **Action**.
- Nom suffixe par **Controller**.

Controller: Afficher Hello World !!

```
public class HomeController : Controller
{
    // GET: Home

    public string Index()
    {
        return "Hello World !!";
    }
}
```

Contenu de RouteConfig

```
routes.MapRoute(  
    name: "Default",  
    url: "{controller}/{action}/{id}",  
    defaults: new { controller = "Home",  
        action = "Index", id = UrlParameter.Optional }  
);
```

Les routes par défaut

Une URL est composée de:

- **controller**: mot-clé obligatoire permettant de préciser le nom du contrôleur à exécuter.
- **action**: mot-clé obligatoire permettant de préciser le nom de la méthode à exécuter.
- **id**: correspond aux paramètres de la méthode.

Testez les URL suivantes

- [/index](#)
- [/Home](#)
- [/home](#)
- [/HOME](#)
- [/home/index](#)
- [/home/index/2](#)
- [/home/index2](#)

Explication du code

```
defaults: new { controller = "Home", action = "Index", id =  
UrlParameter.Optional }
```

Dans ce cas :

- Le contrôleur HomeController sera instancié.
- La méthode Index() sera exécutée.
- Saisir cette URL `http://localhost:Numéro_de_port/` engendrera l'exécution de la méthode (action) Index() définie dans HomeController.

RouteConfig avec clés

```
routes.MapRoute(  
    name: "Default",  
    url: "{controller}/{action}/{id}",  
    defaults: new { controller = "Home",  
        action = "Index", id = UrlParameter.Optional }  
);
```

RouteConfig sans clés

```
routes.MapRoute(  
    "Default",  
    "{controller}/{action}/{id}",  
    new { controller = "Home",  
        action = "Index", id = UrlParameter.Optional }  
);
```

Création de deux contrôleurs: SecondController

```
public class SecondController : Controller
```

```
{
```

```
    // GET: Home
```

```
    public string Index()
```

```
    {
```

```
        return "Bonjour Index";
```

```
    }
```

```
    public string Action()
```

```
    {
```

```
        return "Bonjour";
```

```
    }
```

```
}
```

Création de deux contrôleurs: SecondController

Testez les URL suivantes:

- `/second` : affiche Bonjour Index.
- `/second/action` : affiche Bonjour.
- `/` : affiche Hello world.
- `/second/second` : génère une erreur 404.

Routage par des attributs

```
public class HomeController : Controller  
{  
    public string Index(string id)  
    {  
        return $"Hello {id}";  
    }  
}
```

Routage par des attributs

Testez les URL suivantes:

- </home/index?id=Nouhaila> : un Hello Nouhaila sera affiché.
- </home/index/Nouhaila> : un Hello Nouhaila sera affiché.
- </> : affiche Hello.

Routage par des attributs

Le nom choisi "**id**" n'est significatif:

Comment utiliser le nom "**name**" en obtenant les mêmes résultats?

Solution : modifier le RouteConfig

```
routes.MapRoute(  
    name: "Default",  
    url: "{controller}/{action}/{nom}",  
    defaults: new { controller = "Home", action = "Index", nom = UrlParameter.Optional }  
);
```

Solution : modifier aussi le contrôleur

```
public class HomeController : Controller  
{  
  
    public string Index(string nom)  
    {  
        return $"Hello {nom}";  
    }  
}
```

Routage par des attributs : paramètre de type numérique

Quand le paramètre est de type numérique

modifier le RouteConfig

```
routes.MapRoute(  
    name: "Default",  
    url: "{controller}/{action}/{id}",  
    defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }  
);
```

modifier aussi le contrôleur

```
public class HomeController : Controller  
{  
  
    public string Index(int id)  
  
    {  
  
        return $"Hello {id}";  
  
    }  
}
```


Routage par des attributs: paramètre de type numérique

Testez les URL suivantes:

- / : génère une erreur.
- /home : génère une erreur.
- /home/index : génère une erreur.
- /home/index/2 : affiche Hello 2.

Routage par des attributs: paramètre de type numérique

Testez les URL suivantes:

- / : génère une erreur.
- /home : génère une erreur.
- /home/index : génère une erreur.
- /home/index/2 : affiche Hello 2.

Remarque

Le paramètre **id** est obligatoire car un entier ne prend pas la valeur null (pas comme **string**)

Création d'une page web avec ASP.NET: Contrôleur

Routage par des attributs

NAME	EXAMPLE USAGE	DESCRIPTION
bool	{n:bool}	A Boolean value
datetime	{n:datetime}	A DateTime value
decimal	{n:decimal}	A Decimal value
double	{n:double}	A Double value
float	{n:float}	A Single value
guid	{n:guid}	A Guid value
int	{n:int}	An Int32 value
long	{n:long}	An Int64 value
minlength	{n:minlength(2)}	A String value with at least two characters
maxlength	{n:maxlength(2)}	A String value with no more than two characters
length	{n:length(2)} {n:length(2,4)}	A String value with exactly two characters A String value with two, three, or four characters
min	{n:min(1)}	An Int64 value that is greater than or equal to 1
max	{n:max(3)}	An Int64 value that is less than or equal to 3
range	{n:range(1,3)}	The Int64 value 1, 2, or 3
alpha	{n:alpha}	A String value containing only the A-Z and a-z characters
regex	{n:regex (^a+\$)}	A String value containing only one or more 'a' characters (a Regex match for the ^a+\$ pattern)

Exercice

- Modifier le programme pour qu'il affiche **n** fois le message **Hello** + **nom** + **prénom**.
- **nom**, **prénom** et **n** étant des paramètres récupérés de la requête.
- Si **n** n'a pas été précisé par l'utilisateur ou s'il est inférieur ou égal à zéro, le message sera affiché une seule fois.

Outline I

1 Dot Net

- C'est quoi .NET
- L'architecture .NET
- Les bases .NET
- Du code source au binaire exécuté

2 ASP.NET

- C'est quoi ASP.NET
- MVC
- MVC: Les trois composants
- ASP.NET MVC
- ASP.NET vs ASP.NET Core
- ASP.NET MVC: Déroulement

3 Création d'une page web avec ASP.NET

- Création d'une page web avec ASP.NET: Contrôleur
- Création d'une page web avec ASP.NET: Vue

Création d'une page web avec ASP.NET: Vue

Si on veut mettre en forme le message ?

Il faut utiliser la vue :)

Afficher les informations : création de la vue

- Faire un clic droit sur la méthode **Index** et cliquer sur **Ajouter une vue**.
- Sélectionner **Vue MVC 5** et cliquer sur **Ajouter**.
- Sélectionner : **Empty (sans modèle)** pour le *modèle* et cliquer sur **Ajouter**.

Remarque

- Pour passer des valeurs entre le contrôleur et la vue: On utilise un dictionnaire de clé/valeur appelé **ViewData**
- ⇒: On positionne la valeur du paramètre **id** dans le dictionnaire et cette valeur sera accessible via la clé ["id"].

Création d'une page web avec ASP.NET: Vue

Modifier HomeController.cs

```
public ActionResult Index(int? id)
{
    ViewData["id"] = id;

    return View();
}
```

Modifier Index.cshtml

```
<html>
<head>
<title> Hello World </title>
</head>
<body>

    <p> <b> Hello @ViewData["id"] </b></p>

</body>
</html>
```