

# ASP.NET MVC : deuxième partie

Nouhaila Bensalah

Chercheuse en IA/NLP

nouhaila.bensalah@etu.fstm.ac.ma



## 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

## 4 Création d'une page web dynamique avec ASP.NET

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

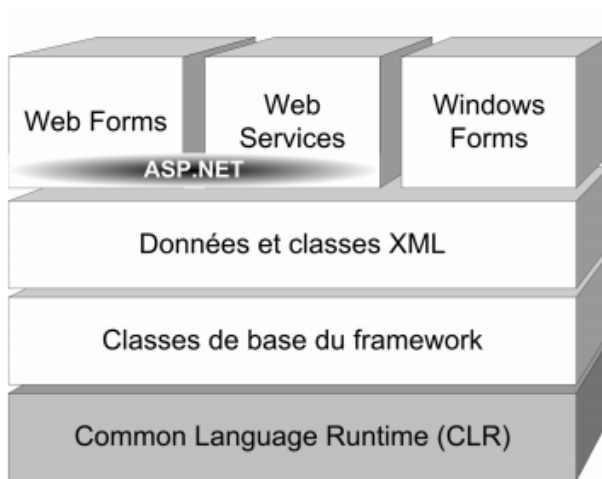
5 Création d'une page web avec ASP.NET: Contrôleur suite

6 Création d'une page web avec ASP.NET: Modèle suite

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.

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

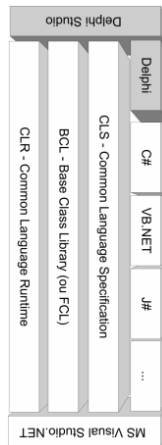
5 Création d'une page web avec ASP.NET: Contrôleur suite

6 Création d'une page web avec ASP.NET: Modèle suite

## 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:

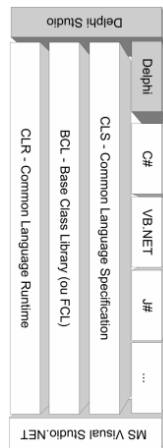
- La compilation en code natif et l'exécution de code IL.
- La gestion de la sécurité.
- La gestion de la mémoire.
- La gestion des processus.
- La gestion des threads (tâches).





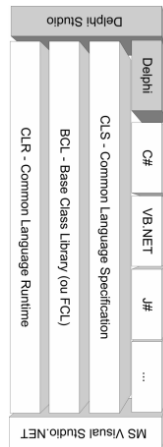
## La BCL ou FCL

- 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.



# 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

## 4 Création d'une page web dynamique avec ASP.NET

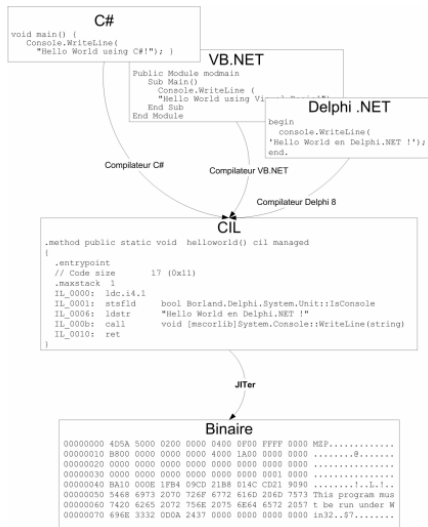
# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

# 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

## 4 Création d'une page web dynamique avec ASP.NET



# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

- 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

- 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

- 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

## 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

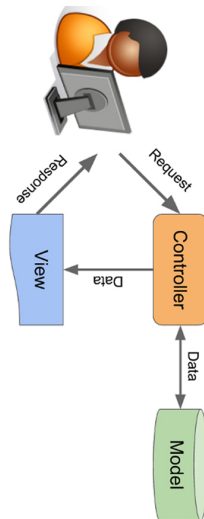
- Création d'une page web dynamique avec ASP.NET: Modèle
- Création d'une page web dynamique avec ASP.NET: Contrôleur
- Création d'une page web dynamique avec ASP.NET: Vue

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

6 Création d'une page web avec ASP.NET: Modèle suite



- 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

## 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

## 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. //Home et Index sont les valeurs par défaut du {controlleur} et {action}, respectivement
- [/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.

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

## 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**)

## 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

# 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>
```

# 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

## 4 Création d'une page web dynamique avec ASP.NET



# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

## Objectif

- Réalisation d'une petite application web afin d'intégrer la notion de Modèle-Vue-Contrôleur.
- Dans cette application, le visiteur rentrera son nom dans un champ de texte modifiable et lorsqu'il va cliquer sur le bouton **Valider**, un message va s'afficher: "Bonjour" suivi du texte saisi.

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

## Réalisation d'une petite application web afin d'intégrer la notion de MVC

### Décrire les informations: création du modèle

- Faire un clic droit sur **Models** .
- Aller dans Ajouter — > Class.
- Choisir Class.
- Saisir Visiteur dans Nom : et valider.

```
class Visiteur
{
    public string Prenom { get; set; }
}
```

## Modèle

On distingue trois types :

- *Modèle de données* : constitué des classes qui interagissent avec une base de données pour l'objectif de stocker ou récupérer des données.
- *Modèle Business* : des classes qui implémentent des fonctionnalités représentant des règles de traitements.
- *Modèle de vue* : des classes utilisées pour passer des informations d'un contrôleur à une 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite



## Réalisation d'une petite application web afin d'intégrer la notion de MVC

### Création du contrôleur:

#### Création du contrôleur

- Faire un clic droit sur **Controllers** et aller dans **Ajouter** → **Contrôleur**.
- Sélectionner **MVC 5 Controller - Empty** et cliquer sur **Ajouter**.
- Saisir un nom pour le contrôleur (**SalutationController**) et cliquer sur **Add**.

# Création d'une page web dynamique avec ASP.NET

## Contenu de: SalutationController.cs

```
public class SalutationController : Controller
{
    public ActionResult Index()
    {
        Visiteur client = new Visiteur();

        return View(client);
    }

    [AcceptVerbs(HttpVerbs.Post)]
    public ActionResult Index(Visiteur visiteur)
    {
        Visiteur client = new Visiteur();

        string prenom = "";

        prenom = Request.Form["prenom_visiteur"];

        client.Prenom = prenom; // Model binding: allows you to map request information (example: forms) to named parameters

        ViewData["message"] = "Bonjour, " + prenom;

        return View("Index", client);
    }
}
```

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite

Réalisation d'une petite application web afin d'intégrer la notion de MVC

## Création de la vue:

### 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** pour le *modèle*, **Visiteur** pour la *classe du modèle* et cliquer sur **Ajouter**.

# Création d'une page web dynamique avec ASP.NET

## Contenu Index.cshtml

```
<html>
<head>

    <meta name="viewport" content="width=device-width" />

    <title>Formulaire</title>
</head>
<body>

    <div>

        @using (Html.BeginForm()){

            <h1>Bonjour MVC Learner </h1>

            <p>Veuillez entrez votre nom?</p>

            @Html.TextBox("prenom_visiteur") //c'est la syntaxe d'un moteur de vue appelé Razor, On peut utiliser <input
type="text" name="prenom_visiteur" id="prenom_visiteur" />

            <input type="submit" value="Valider" /> }

            <p>@ViewData["message"]</p>

        </div>

    </body>
</html>
```

# Création d'une page web dynamique avec ASP.NET

## Transfert des données: Vue

- **ViewBag, ViewData:**  
permet de stocker des données dans une action de contrôleur qui sera utilisée dans la vue correspondante.
- **TempData:**  
permet de stocker des données pendant la durée de la requête HTTP dans une action du contrôleur qui sera utilisée dans une autre action du contrôleur ou à une vue. Une fois la redirection effectuée, les données sont automatiquement supprimées.

## TempData: Exemple (transfert des données: d'une action à une autre)

```
public ActionResult Test() {  
    TempData["test"] = DateTime.Now;  
  
    return RedirectToAction("action", "controleur"); // a controller action that will consume this data (string str=  
    TempData["test"].ToString();}
```

# Création d'une page web dynamique avec ASP.NET

## TempData: Exemple

```
public ActionResult Test() {  
    TempData["test"] = "object";  
  
    return View(); // et dans la vue: @TempData["test"] }
```

## ViewBag: Exemple

```
public ActionResult Test() {  
    ViewBag.Test = "object";  
  
    return View(); } // et dans la vue: @ViewBag.Test
```

## ViewData: Exemple

```
public ActionResult Test() {  
    ViewData["Test"] = "object";  
  
    return View(); } // et dans la vue: @ViewData["Test"]
```



# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

## 6 Création d'une page web avec ASP.NET: Modèle suite

Pour rediriger vers une autre action, on utilise la méthode

## **RedirectToAction**

```
RedirectToAction("action", "controller", new { paramName = value});  
//Ce qui génère la route suivante controller/action/value
```

Pour récupérer la route courante dans le contrôleur

```
string host = HttpContext.Request.RawUrl; //En allant à  
https://localhost:Numéro_de_port/home/index/9, la variable host récupère  
la valeur /home/index/9
```

## Autres informations à récupérer

- **HttpContext.Request.HttpMethod** : pour récupérer le verbe HTTP utilise pour executer l'action
- **HttpContext.Request.Url.AbsolutePath** : pour récupérer le chemin absolu demande (comme /home/index/9)
- **HttpContext.Request.Url.AbsoluteUri** : pour récupérer l'URL complète (comme https://localhost:Numéro\_de\_port/home/index/2)
- **HttpContext.Request.Url.Host** : pour récupérer l'adresse IP du visiteur
- ...

Pour modifier le nom d'une action, on utilise le décorateur **ActionName**

```
public class HomeController : Controller
{
    [ActionName("first")]
    public string Index()
    {
        return $"Hello first";
    }
}
```

En allant à `/home/first`, un Hello first sera affiché.

En allant à `/home/index`, une erreur 404 sera affichée.

Pour ne pas considérer une méthode du contrôleur comme une action, on utilise le décorateur **NonAction**

```
public class HomeController : Controller
{
    [NonAction]
    public string DoSomething()
    {
        //...
    }
}
```

# 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

## 4 Création d'une page web dynamique avec ASP.NET

# Outline II

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

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

6 Création d'une page web avec ASP.NET: Modèle suite



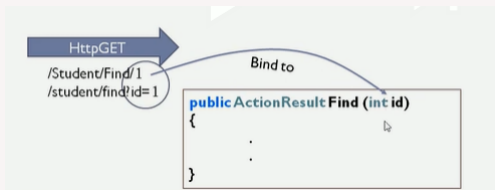
## Model binding

With the Model binding, MVC framework converts the http request values to action method parameters.

Le processus du **model binding** s'exécute automatiquement si la l'action demandée porte des paramètres.

Ces paramètres peuvent être de:

- Primitive type



- Complex type (class, structure)

## Model binding: Include, Exclude and IsPropertyAllowed

- **Exclude** : La propriété Exclude est utilisée pour obtenir ou définir une liste délimitée par des virgules de noms de propriétés pour lesquelles la liaison n'est pas autorisée.
- **Include** : La propriété Include est utilisée pour obtenir ou définir une liste délimitée par des virgules de noms de propriétés pour lesquels la liaison est autorisée.
- **IsPropertyAllowed** : La méthode IsPropertyAllowed est utilisée pour déterminer si la propriété spécifiée est autorisée. Elle renvoie True si la propriété spécifiée est autorisée ; sinon, False.

Supposons qu'on a: Classe Produit

- ProductId
- ProductName
- ProductType

## Include

```
public ViewResult Test([Bind(Include = "ProductName", ProductType")]  
Produit product) // Dans le contrôleur créé
```

## Exclude

```
public ViewResult Test([Bind(Exclude = "ProductId")] Produit product)
```

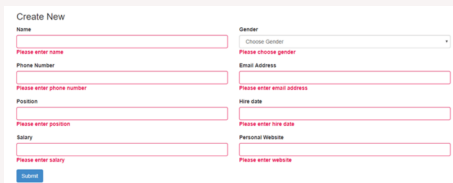
## Validation des données

- Avant qu'une application ne stocke des données dans une base de données, il faut vérifier que les données ne présentent pas de menaces potentielles pour la sécurité (exemple: formatées par type et par taille) en utilisant des annotations.
- Le model binder intercepte toutes les erreurs, et les mettent dans le ModelState : une propriété du Controller.
- Exemple:
  - `ModelState.IsValid == True` // vérification au niveau de tout le modèle (dans le contrôleur crée)
  - `ModelState.IsValidField("FirstName") == True` // vérification au niveau de chaque propriété (dans le contrôleur crée)

## Validation client vs Validation serveur

Dans MVC, la validation s'effectue à la fois sur le client et sur le serveur.

- La saisie de l'utilisateur qui est validée dans le navigateur avant d'être soumise au serveur est appelée **validation côté client**.



The screenshot shows a web form titled "Create New" with two columns of input fields. The left column contains fields for Name, Phone Number, Position, and Salary. The right column contains fields for Gender (a dropdown menu), Email Address, Hire date, and Personal Website. Each field has a red border and a red error message below it: "Please enter name", "Please enter phone number", "Please enter position", "Please enter salary", "Please choose gender", "Please enter email address", "Please enter hire date", and "Please enter website". A blue "Submit" button is located at the bottom left of the form.

- La validation côté serveur** entre en jeu lorsqu'un utilisateur malveillant peut soumettre des données par le biais de différents canaux ou encore soumettre des entrées dangereuses au serveur.

## Annotations: Exemple

```
public class WebUser
{
    [Required]

    [StringLength(25)]

    public string FirstName { get; set; }
    [Required]

    [StringLength(50, MinimumLength = 3)]

    public string LastName { get; set; }

    [EmailAddress]

    public string MailAddress { get; set; }
}
```

See [This](#) for the other annotations