



ASP.NET MVC

Introduction



Technologies Web

- Technologies
 - Clients & Serveurs
 - HTTP
 - HTML 5 + CSS 3
 - Frameworks
 - Javascript
 - Frameworks
- Utiliser des frameworks répandus
 - Éprouvés
 - Documentés
 - Aide de la communauté

Différents acteurs, technologies propriétaires ou non

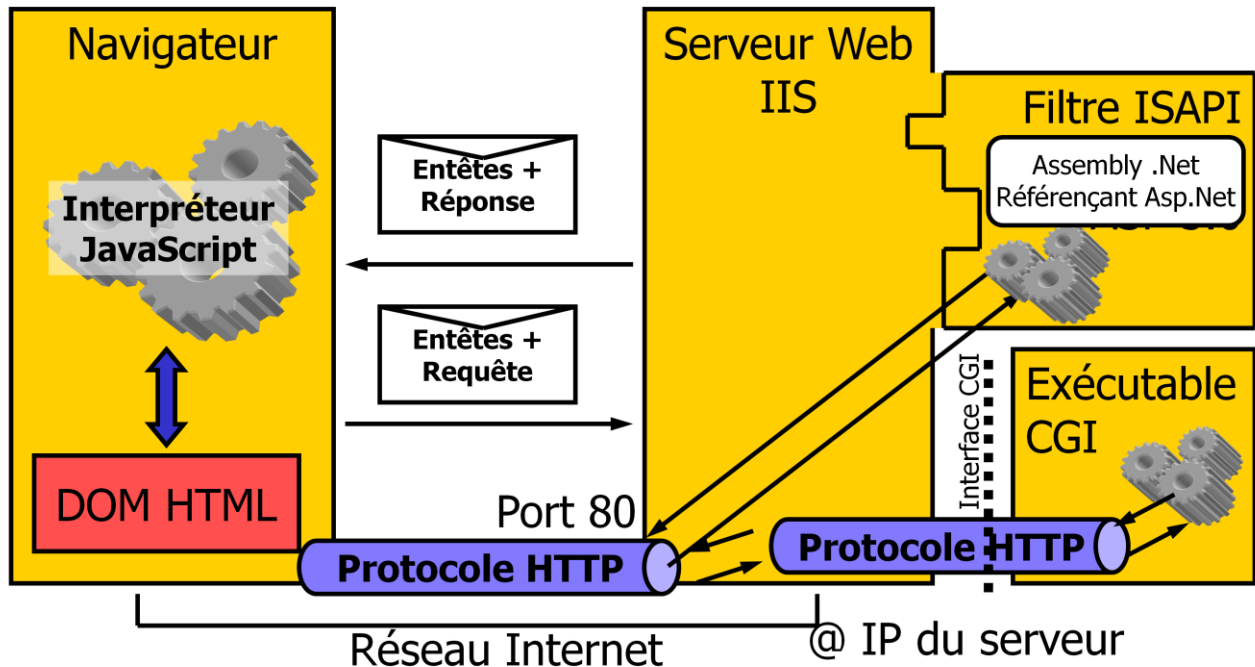
Capitaliser ses connaissances => réutiliser les technologies maîtrisées



Technologies Web

- Navigateurs ≈ No problemo
- Hébergement
 - IIS Express
 - Installé avec Visual Studio, pour développer
 - IIS
 - Hébergement personnel
 - Azure...
 - Cloud
 - €, \$, £, ¥, ₣, ...
 - Alternatives
 - Mod_Mono : plugin Mono pour Apache
 - OWIN (Open Web Interface for .NET, Microsoft.Owin.Hosting)

Architecture Web avec IIS : historique



DOM : Document Object Model : hiérarchie d'objet décrivant le document HTML

HTTP : Hyper Text Transfer Protocole. Protocole réseau gérant la connexion entre un navigateur et un serveur Web. La connexion HTTP est susceptible d'être rompue après l'échange d'une requête et de la réponse correspondante.

Filtre ISAPI : Bibliothèque utilisant ISAPI pour communiquer avec IIS

CGI : Common Gateway Interface.

Un exécutable CGI est lancé par un serveur Web.

L'entrée standard du processus CGI lit en fait dans la requête HTTP envoyée par le navigateur.

La sortie standard du processus CGI écrit en fait dans la connexion HTTP vers le navigateur.

ISAPI : Internet Server API, interface de programmation de IIS



Propositions historiques de MS - Asp

■ Asp Classique

```
<%@ Language=JavaScript %>
<%
  var i = 0;
  for( i = 1; i < 5; i++ ) { %>
    <p> 2 x <%= i %> = <%= 2*i %> </p>
  }
  %>

var cnx = Server.CreateObject( "ADODB.Connection" );
var rs = Server.CreateObject( "ADODB.Recordset" );
cnx.Open( "bdPersonnes" );
rs.Open( "select * from personnes", cnx );
var champ = 0, rec = 0;
while( !rs.EOF ) {
  rec++;
  Response.Write( "<HR />Enregistrement " + rec + "<BR />\n" );
  for( champ = 0; champ < rs.Fields.Count; champ++ ) {
    Response.Write( rs.Fields.Item( champ ).Name + " = " );
    Response.Write( rs.Fields.Item( champ ).Value + "<BR />\n" );
  }
  rs.MoveNext();
}
rs.Close();
cnx.Close(); %>
```



Simple



Mélange code et présentation



Propositions historiques de MS - WebForms

■ Asp.Net Web Forms : composants RAD

■ Code

```
public partial class Test : System.Web.UI.Page {
    public IEnumerable ListerMesTrucs() {
        return new object[] {
            new { Chaîne = "Un", Date = new DateTime(1832, 12, 25) },
            new { Chaîne = "Deux", Date = new DateTime( 2589, 12, 26 ) },
            new { Chaîne = "Trois", Date = new DateTime( 1777, 12, 27 ) }
        };
    }
}
```

■ Présentation

```
<%@ Page Language="C#" CodeBehind="Test.aspx.cs" Inherits="WebApplication1.Test" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
    <form id="form1" runat="server">
        <h1>Mes trucs</h1>
        <asp:GridView runat="server" SelectMethod="ListerMesTrucs" GridLines="Both">
            <HeaderStyle BackColor="Black" ForeColor="White" />
        </asp:GridView>
    </form>
</body>
</html>
```

Mes trucs

Chaîne	Date
Un	25/12/1832 00:00:00
Deux	26/12/2589 00:00:00
Trois	27/12/1777 00:00:00



Propositions historiques de MS - WebForms

■ Asp.Net Web Forms

😊 Avantages

- Capitalisation du savoir faire .Net + langage C# / VB.Net
- Compilation
- RAD
- Séparation code / présentation
- Puissance
- ViewState

😞 Inconvénients

- L'approche RAD
 - Éloigne des fondamentaux HTTP / HTML / CSS
 - Permet de trop mélanger les couches
- La puissance implique d'acquérir une masse critique avant d'être opérationnel
- ViewState 😊



Propositions historiques de MS Web Pages

■ Asp.Net Web Pages

```
<html>
<body>
  <div>
    Il est @DateTime.Now.ToShortTimeString()
    @for( int i = 10; i <= 20; i++ ) {
      <p style="font-size: @(i.ToString()+"pt")">
        Coucou (taille de fonte : @i)
      </p>
    }
  </div>
</body>
</html>
```

Il est 06:31

Coucou (taille de fonte : 10)

Coucou (taille de fonte : 11)

Coucou (taille de fonte : 12)

Coucou (taille de fonte : 13)

Coucou (taille de fonte : 14)

Coucou (taille de fonte : 15)

Coucou (taille de fonte : 16)

Coucou (taille de fonte : 17)

Coucou (taille de fonte : 18)

Coucou (taille de fonte : 19)

Coucou (taille de fonte : 20)

■ Fichier ".cshtml"

- Html + Razor

■ Architecture simpliste

😊 Peu d'investissement

☹ Pour les pages simples

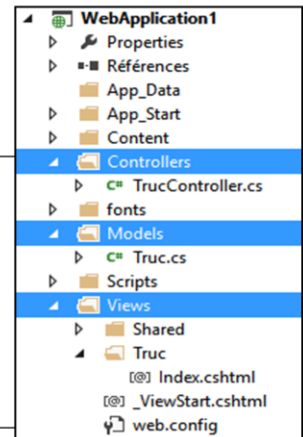


Propositions historiques de MS - MVC

■ Asp.Net MVC

■ Modèle : Models/Truc.cs

```
public class Truc {  
    public string Chaîne { get; set; }  
    public DateTime Date { get; set; }  
    public Truc( string c, string d ) {  
        Chaîne = c; Date = DateTime.Parse( d );  
    }  
}
```



■ Contrôleur : Controllers/TrucControler.cs

```
public class TrucController : Controller {  
    private static Truc[] Trucs = { new Truc( "Un", "25/12/1832" ), ... };  
    public ActionResult Index() {  
        ViewBag.Trucs = TrucController.Trucs;  
        return View();  
    }  
}
```



Propositions historiques de MS - MVC

■ Asp.Net MVC

■ Vue : Views/Index.cshtml

Mes trucs :

- Un (25/12/1832)
- Deux (26/12/2589)
- Trois (27/12/1777)

Y a pas de *Boum* dans mes trucs

```
<html>
<head> <title>Index</title> </head>
<body>
  <div> Mes trucs :
    <ul>
      @foreach( var t in ViewBag.Trucs ) {
        <li>@String.Format( "{0} ({1:d})", t.Chaine, t.Date )</li>
      }
    </ul>
    @try {
      <p>@ViewBag.Trucs.Boum</p>
    }
    catch {
      <p>Y a pas de <em>Boum</em> dans mes trucs</p>
    }
  </div>
</body>
</html>
```



Propositions historiques de MS - MVC

■ Asp.Net MVC

- Plus proche de HTTP, HTML, CSS
- Séparation forte du modèle, de la vue et du contrôleur

😊 Plus bas niveau que Web Forms

- Facilite la réutilisation des frameworks classiques
 - Bootstrap, JQuery, Angular JS, ...
- Plus facile d'en faire le tour
- Les contrôleurs supportent les tests unitaires

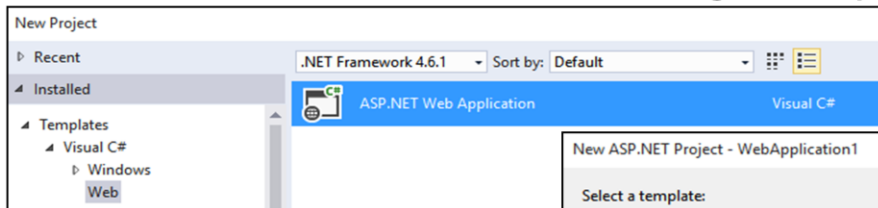
😞 Plus bas niveau que Web Forms

- Nécessite la maîtrise de HTML + CSS + JS
 - Multiplication des frameworks utilisables
- Moindre vitesse de développement
- M+V+C → multiplication des fichiers

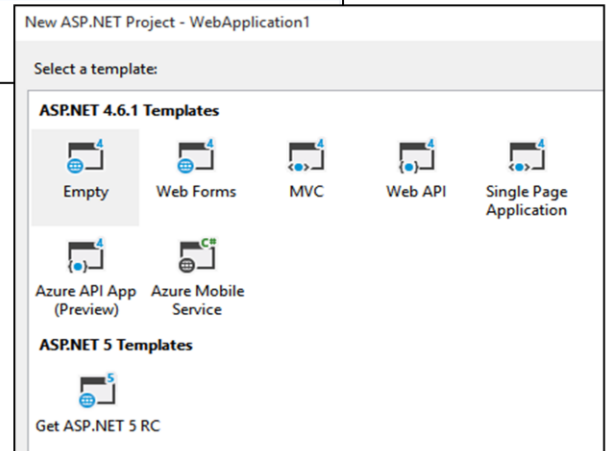


We stand as one... undivided

- One Asp – à partir de Web Forms 4.5 & MVC 5
 - Web Forms / MVC : même technologie : Asp.Net



- Capitalisation du savoir pour les fonctions de base
- Choix de l'architecture pour chaque page, en fonction des besoins



- Fonctionnalités réutilisables

- Tout .Net
- Fichier Web.config
- Authentification
 - Identité Windows / applicative
 - Rôles
 - Autorisations
- Session
- Cache
 - Asp.Net
 - IIS