ASB.NET CORE

Razor

- من اجل تسهيل عملية بناء الصفحات ولدمج أوامر السيرفر مع HTML سمحت مايكروسوفت للمطورين باستعمال أسلوب Razor
- يمثل هذا الأسلوب إضافة قوية حيث أصبح بإمكاننا ان نكتب أوامر بلغة c# بصفحات HTML
- عند استدعاء هذه الصفحات يتم تنفيذ أوامر السيرفر وادراج النتيجة بصفحة HTML ومن ثم ارجاعها للمستخدم على شكل HTML
- لتمييز أوامر Rezor عن HTML نستعمل الرمز @ قبل التعليمات

مثال ١ :

```
Razor Snippets:

@{
    var name = "Khalid ESSAADANI";
}

<p>Hello, @name</p>

@{
    name = "Said HAMRI";
}

<p>@name</p>
```

مثال ٢:

في المثال التالي لاحظ معي كيف سمح لنا Razor بأن نخلط بين كود برمجي بلغة سي شارب مع Markup HTML بطريقة سلسة:

Razor Snippets:

```
<h1>Current Date: @DateTime.Now.ToLongDateString()</h1>
```

لو قمنا باستعراض الصفحة التي تحتوي على الأمر أعلاه، فسوف نحصل على النتيجة التالية:

Current Date: 05 August 2019

مثال ٣:

```
@{
    var a = 3;
    var b = 5;

    var sum = a + b;

    <h3>Sum of a + b is: @(a+b)</h3>
}
```

عند التنفيذ سيتم جمع قيمتي المتغيرين a و b وتتم طباعة النتيجة كما يلي:

Sum of a + b is: 8

مثال ٤:

يمكننا أيضا التحقق من عبارة معينة باستعمال if statement كما يلي:

Razor Snippets:

```
@{
    var userName = "Khalid";
    var password = "123456";

    if (userName == "Khalid" && password == "123456")
    {
        <h2>Hello, @userName!</h2>
    }
    else
    {
        <h2>Invalid login attempt.</h2>
    }
}
```

في المثال أعلاه سيتم التحقق من قيمتي المتغيرين، فإن تحقق الشرط تم عرض رسالة ترحيبية، وإن لم يتحقق تم عرض رسالة مفادها أن محاولة تسجيل الدخول لم تنجح.

لأن الشرط في حالتنا متحقق، فعتد التنفيذ سنحصل على النتيجة التالية:

Hello, Khalid!

مثال ٥:

يمكننا أيضا القيام بعمليات تكرارية كما في لغة سي شارب باستعمال الحلقات، وهذا مثال يوضح ذلك:

Razor Snippets:

```
@{
    string[] names = {"Ahmed", "Kamal", "Rachid", "Yassine"};

    foreach (var name in names)
    {
        <h2>@name</h2>
    }
}
```

عند تنفيذ الصفحة أعلاه، سنلاحظ أن الآلية التكرارية تقوم بعرض جميع عناصر المصفوفة النصية names، وفيما يلي نتيجة البلوك:

Ahmed

Kamal

Rachid

Yassine

مثال ٦:

```
@{
    string[] names = { "Ahmed", "Kamal", "Rachid", "Yassine" };

    // Using for
    <h2>Using for loop</h2>
    for (int i = 0; i < names.Length; i++)
    {
        <h3>@names[i]</h3>
    }

    // Using do
    <h2>Using do loop</h2>
    var counter = 0;
    do
    {
        <h3>@names[counter]</h3>
        counter++;
    } while (counter < names.Length);

    // Using while
    <h2>Using while loop</h2>
    counter = 0;
    while (counter < names.Length)
    {
        <h3>@names[counter]</h3>
        counter++;
    }
}
```

مثال ٧:

```
@functions
{
    DateTime GetCurrentDateTime()
    {
        return DateTime.Now;
    }

    string GetName()
    {
        return "Khalid";
    }

    int Sum(int a, int b)
    {
        return a + b;
    }
}
```

تتمة مثال ٧:

يمكننا استدعاء هذه الدوال بالطريقة الاعتيادية كما يلي:

```
Razor Snippets:

<h3>Current Date: @GetCurrentDateTime()</h3>
<h3>Name: @GetName()</h3>
<h3>The sum of 5 + 6 is: @Sum(5,6)</h3>
```

وفيما يلي النتيجة التي سنحصل عليها عند التنفيذ:

Current Date: 05/08/2019 11:15:29

Name: Khalid

The sum of 5 + 6 is: 11

تنشيط ndows

من اهم مزايا ASP.net core:

- متعدد البيئة cross-platform : بحيث يمكن العمل على مشاريع من هذا النوع على مختلف أنظمة التشغيل(Windows, Mac, Linux)
- مفتوح المصدر Open source: من اجل سهولة تطويره
- Modular: بحيث ان يتضمن المشروع فقط المكتبات التي يستعملها والتي تكون على شكل حزم.
- لا يفرض قيود بين الاصدارات: يمكن للتطبيق ان يعمل على اصدار احدث او اقدم دون مشاكل.

من اهم مزايا ASP.net core:

- يحتوي على نظام إدارة الارتباطات : مما يسمح باضعاف الترابط بين مكونات المشروع وجعلها غير مرتبطة ببعضها البعض مما يضمن سهولة الاختبارات والصيانة والتوسيع.
- يدمج بين ASP.NET MVC و ASP.NET Web API
- الدعم الكامل للحوسبة السحابية Cloud-Optimized : حيث يستطيع المطورون بناء تطبيقات Cloud-Based Web Applicationsوتوفير مكتبات API لاستخدامها من قبل تطبيقات أخرى.