

Validation Controls

مقدمة

تنقسم المواقع التي تتعامل مع البيانات الى قسمين، القسم الأول هو أن تكون البيانات قادمة من الـ server على سبيل المثال بيانات تم تخزينها في قاعدة بيانات ويتم عرضها للمستخدم، والقسم الثاني هو ما يتم أدخاله من قبل المستخدم الى قاعدة بيانات الموقع بمعنى آخر ما يتم إرساله من قبل المستخدم الى الـ server .

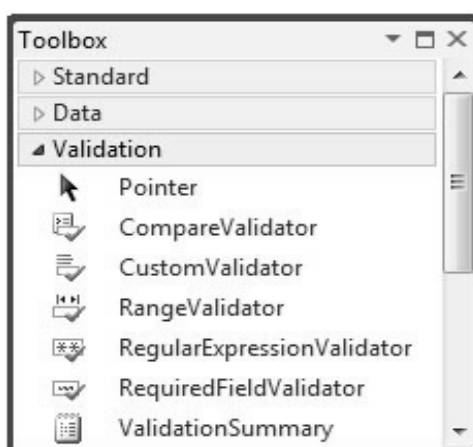
وعلى أساس القسم الثاني من تقسم البيانات نستطيع تسميته بـ المدخلات من قبل المستخدم، حيث تختلف الصفحات في مستوى الأهمية للبيانات المدخلة من قبل المستخدم، على سبيل المثال نموذج contact us حيث البساطة في عمل هذا النموذج وهو أن يقوم المستخدم بملأ البيانات المطلوبة منه وإرسالها.

بالاعتماد على المثال السابق وهو نموذج الـ contact us، لا بد من التأكد أن المستخدم يتعامل مع النموذج بشكل صحيح مثلاً أن المستخدم قام بملأ الحقول المطلوبة منه ولم يترك أحدها، وأيضاً أن يدخل النوع المطلوب منه من البيانات في هذا الحقل على سبيل المثال يوجد حقل للعمر، ف العمر تنطبق عليه شروط مطلوبة مثل أن يكون نوع هذه البيانات هو (integer)، أو يكون بين مجال من الأرقام.

ومن طرف آخر للمستخدم الذي يدخل البيانات لا نستطيع تخمين ماهي طبيعة هذا المستخدم هل هو مستخدم عادي أو مستخدم له هدف على الموقع، أذ بواسطة نموذج الـ contact us يمكن أن يقوم بأرسال معلومات تتعامل مع الخادم (server) لأخذ معلومات مهمة هو يحتاجها، فهذه تعتبر مشكلة وبشكل خاص عندما يتم نشر الموقع على الـ internet .

The ASP.NET Validation Controls

في الإصدار الرابع من الـ .NET ، يوجد هناك ٦ أدوات من أدوات التحقق التي تسهل علينا عمليات التحقق من مدخلات المستخدمين لموقعنا الشخصية، واحدة من الأدوات وهي Validation Summary تستخدم لعرض معلومات شاملة للأخطاء المتواجدة. الشكل التالي يبين القائمة الخاصة بأدوات التحقق.



تستخدم هذه الأدوات الموجودة في القائمة السابقة مع الادوات الموجودة في الصفحة على سبيل مثل TextBox.

Client-Side Validation

كما نعلم بإمكاننا عمل تحقق من المدخلات بواسطة الـ JavaScript، لكن لا يجب عليك الاعتماد عليها بشكل كلي لأن وبكل بساطة يستطيع المستخدم إغلاق تفعيل الـ JavaScript من خلال المتصفح فتصبح هذه التحققات من المدخلات بلا فائدة. لكن بأستطاعتك أستخدم التحقق بواسطة JavaScript وذلك بأعطاء المستخدم رسالة فورية تخبرة على سبيل المثال أن هذا الحقل فارغ وعليه أن يملئه بالبيانات.

Using the Validation Controls

قبل أن يتم شرح أدوات التحقق وعمل مثال بسيط لنتطرق بشكل بسيط للشكل العام لهذه الأدوات، على سبيل المثال RequiredFieldValidator سنلاحظ الشكل العام وهو متشابه بعض الشيء بين الأدوات، ولو نلاحظ وجود رسالة الخطأ التي نريدها تظهر للمستخدم وأيضا أي من الأدوات التي سوف نعمل تحقق منها .

```
<asp:RequiredFieldValidator ID="ReqName" runat="server"
    ErrorMessage="Enter Your Name" ControlToValidate="Name" />
```

في الجدول التالي أهم الخصائص التي نستطيع كتابتها مع أدوات التحقق.

الوصف	الخاصية
توجد ثلاث حالات في هذه الخاصية, عند أستخدم الـ none فإن رسالة الخطأ لن تظهر بينما عند أستخدم الـ static أو الـ Dynamic الفرق الوحيد في طريقة العرض هي أن حالة الـ static تستخدم في طريقة العرض وأخفاء رسالة الخطأ بشكل مشابه للـ CSS وهي خاصية الـ visibility و الـ Dynamic تتعامل مع اخفاء و اظهار رسالة الخطأ بأستخدم خاصية .Display	Display
هذه الخاصية تسمح بأستخدم class المعرف في CSS الخاص بالموقع.	CssClass
هذه الخاصية تسمح تحتوي على رسالة الخطأ التي سوف تظهر للمستخدم.	ErrorMessage
في هذه الخاصية يتم كتابة النص الذي سوف يظهر للمستخدم في الصفحة، ويفضل ان تكون (*) .	Text
هذه الخاصية تستخدم لتحديد أي الادوات التي نريد ان نعمل لها تحقق، مثل المثال السابق تم اختيار الأداة .Name	ControlToValidate
هذه الخاصية تستخدم لجعل التركيز على أول أداة يظهر بها خطأ.	SetFocusOnError
لفرض وجود أكثر من TextBox في الصفحة وكل منهم له وظيفة الخاصة مع زر الإدخال، فعند ملئ أحد الـ TextBox دون الآخر وتم ضغط الزر سوف يتم التحقق من أنه لا يوجد أي خطأ في الصفحة وبعدها تنفيذ العملية، لكن إذا لم يتم عمل group للـ TextBox مع الزر الخاص به للإدخال سوف يتحقق من الادوات الأخرى (القصد هنا أدوات التحقق) سوف يتم ملاحظة أن هناك TextBox فارغ فعلى هذا الأساس سوف يرجع خطأ.	ValidationGroup

بعد التعرف بشكل بسيط على الشكل العام لأدوات التحقق وبعض الخصائص المستخدمة، سوف أبدا بسرد أدوات التحقق مع الخصائص الهامة المتعلقة بهذه الاداة.

RequiredFieldValidator

تستخدم هذه الأداة لمساعدة المستخدم واعلامه أن هذا الحقل مطلوب وعليك أن تملئه بالبيانات المطلوبة منك، والقائمة التالية تبين أهم الخصائص التي يمكن أن نطبقها على هذه الأداة:

- ControlToValidate
- ErrorMessage
- Text
- CssClass
- Display

RegularExpressionValidator

تستخدم هذه الأداة للتحقق من المدخلات بواسطة الـ Regular Expression وذلك بكتابة الكود في الخاصية ValidationExpression، أيضا توجد مجموعة ممتازة وجاهزة للاستخدام في هذه الخاصية، والقائمة التالية تبين أهم الخصائص التي يمكن أن نطبقها على هذه الأداة:

- ControlToValidate
- ErrorMessage
- Text
- CssClass
- Display
- ValidationExpression

CompareValidator

هذه الاداة تستخدم لعمل مقارنة بين عنصرين والتحقق إذا ما كانوا متطابقين في القيم على سبيل المثال التحقق من قيمة password بأن تكون متطابقة في الحقلين، والقائمة التالية تبين أهم الخصائص التي يمكن أن نطبقها على هذه الاداة:

- ControlToValidate
- ErrorMessage
- Text
- CssClass
- Display
- ControlToCompare
- Operator
- Type

هنا الأمر مختلف قليلا حيث توجد خاصيتان لتحديد الأدوات التي سوف نعمل لهم تحقق من المدخلات، عندما نستخدم ControlToValidate سوف نحدد الأداة التي سوف نتحقق منها و ControlToCompare هي لتحديد ماهي الأداة التي سوف نقارن بها هذه الأداة.

وأيضا الـ Operator نستطيع تحديد نوع المقارنه هي Equal والخ...، ولابد من استخدام نوع محدد لهذه المقارنه فيأتي دور الـ Type.

أذاً كأستنتاج نستطيع أيضاً استخدام هذه الأداة لنقارن على أداة واحدة فقط، على سبيل المثال يوجد لدينا TextBox يستطيع المستخدم إدخال العمر، اذاً نستطيع استخدام ControlToValidate واختيار هذا الـ TextBox وبعدها تحديد الـ Operator الى DataTypeCheck وبعدها نختار من الـ Type النوع الذي سوف نتحقق منه وفي هذه الحالة نريد Integer.

RangeValidator

نستخدم هذه الأداة للتحقق من قيمة مدخلة ما بين مجال لأكبر قيمة وأصغر قيمة، والقائمة التالية تبين أهم الخصائص التي يمكن أن نطبقها على هذه الأداة:

- ControlToValidate
- ErrorMessage
- Text
- CssClass
- Display
- MaximumValue
- MinimumValue

ValidationSummary

هذه الأداة تستخدم لعرض ملخص لجميع الأخطاء التي ظهرت على شكل قائمة، والقائمة التالية تبين أهم الخصائص التي يمكن أن نطبقها على هذه الأداة:

- CssClass
- DisplayMode

CustomValidator

إذا اردت أن تعمل تحقق على طريقتك الخاصة فهذه الأداة تسمح لك بكتابة كود لعمل هذا التحقق، والقائمة التالية تبين أهم الخصائص التي يمكن أن نطبقها على هذه الأداة:

- ErrorMessage
- Text
- CssClass
- Display
- ClientValidationFunction

نستطيع التحقق من المدخلات بأسلوبين باستخدام هذه الأداة وهما server validation و client validation وتعتبر الـ client validation عبارة عن استخدام الـ Javascript و ربط أسم الدالة بخاصية

ClientValidationFunction

ولتفعيل الـ server validation فقط نحتاج على تفعيل الـ Event الخاص به وهو ServerValidate الكود التالي يبين طريقة كتابة الكود:

Server Validation

```
protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args)
{
    if (!string.IsNullOrEmpty(TextBoxAge.Text))
    {
        args.IsValid = true;
    }
    else
    {
        args.IsValid = false;
    }
}
```

ببساطة يتم التحقق في الكود أن الـ TextBox الخاص للعمر ليس فارغ، أذاً لا يوجد أي خطأ.

Client Validation

كما ذكرت سابقاً يمكننا كتابة كود تحقق JavaScript ، لنأخذ مثال على ذلك ونجعل أسم الدالة (AgeFunctionValidate) في خاصية ClientValidationFunction:

```
<script type="text/javascript">

function AgeFunctionValidate(source, args) {

    var AgeTextBox = document.getElementById('TextBoxAge');

    if (AgeTextBox.value != "") {

        args.IsValid = true;

    } else {

        args.IsValid = false;

    }

}
```

```
</script>
```

في الختام المدخلات مهمة للتفاعل مع الموقع الخاص بنا، وهذه المدخلات تأتي من أكثر من مصدر وأكثر من شخص، لدى وبهذه الأدوات التي تم التحدث عنها في الأعلى نستطيع أن نحدد مانريد من البيانات المدخلة الى موقعنا وأيضا تأمين جزء من الحماية من قبل مدخلات المستخدم.