

الفصل السادس أدوات التحقق من صحة المدخلات

تعرفنا في الفصلين السابقين على مفهوم أدوات تحكم الويب كما تعرفنا على عدد من الأدوات الهامة. وفي هذا الفصل نتعرف على كيفية التحقق من صحة البيانات التي يقوم المستخدم بإدخالها من خلال هذه الأدوات.

بانتهاء هذا الفصل، ستتعرف على:

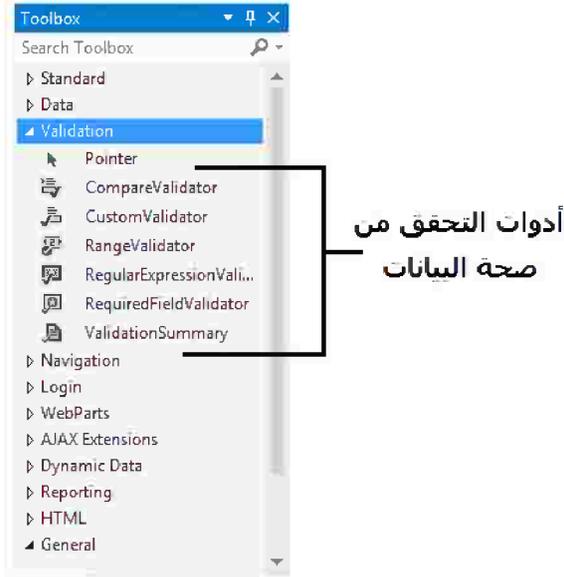
- استخدام أدوات التحقق من صحة البيانات.
- اختبار الحقول المطلوبة.
- التحقق على مستوى مدى من البيانات.
- مقارنة قيم الحقول.
- التحقق من صحة الصفحة بالكامل.
- التحكم في رسائل الخطأ.
- التحقق من صحة التعبيرات.
- إنشاء دالة تحقق مخصصة.

بالإضافة إلى استقبال المدخلات من المستخدم، من الضروري التحقق من صحة هذه المدخلات، حيث يجب أن يحتوي نموذج إدخال البيانات المثالي على كود كافٍ للتحقق من صحة هذه البيانات. فقبل أن يتم إرسال بيانات المستخدم إلى قاعدة البيانات، يتم أولاً اختبار هذه البيانات وإظهار رسالة الخطأ التي تحث المستخدم على إدخالها مرةً أخرى إذا لم تكن صحيحة.

أدوات التحقق من صحة البيانات

لا تحتاج بعض أدوات التحكم إلى المزيد من التحقق من صحة بياناتها لأنها تكون أساساً مقيدة بمجموعة معينة من البيانات. فمثلاً، تحتوي أداة مربع السرد دائماً على مجموعة من الخيارات التي يقوم المستخدم بالاختيار من بينها وبالتالي لا يوجد مجال لعملية الخطأ. أما مربع النص `TextBox` فهو على العكس من ذلك يتيح للمستخدم إدخال ما شاء من البيانات الحرفية والرقمية، لذا فمن المنطقي التأكد من صحة بياناته قبل نقلها إلى قاعدة البيانات الموجودة على الخادم. ولمساعدتك في التحقق من صحة بيانات المستخدم على الويب، أتاح لك نطاق `NET` مجموعة من التصنيفات التي يمكنك استخدامها في نماذج الويب في صورة عدد من أدوات التحكم التي يمكن إجمالها فيما يلي (انظر شكل ٦-١):

- **الأداة `RequiredFieldValidator`** والتي تقوم باختبار إدخال المستخدم لأي قيمة داخل أحد الحقول.
- **الأداة `CompareValidator`** والتي تقوم بالمقارنة بين حقلين أو مقارنة أحد الحقول مع قيمة معينة.
- **الأداة `RangeValidator`** وتقوم بمعرفة مدى وقوع إحدى القيم داخل مدى معرف من القيم.
- **الأداة `RegularExpressionValidator`** وتقوم بالتأكد من توافق قيمة أحد الحقول مع تعبير منتظم.

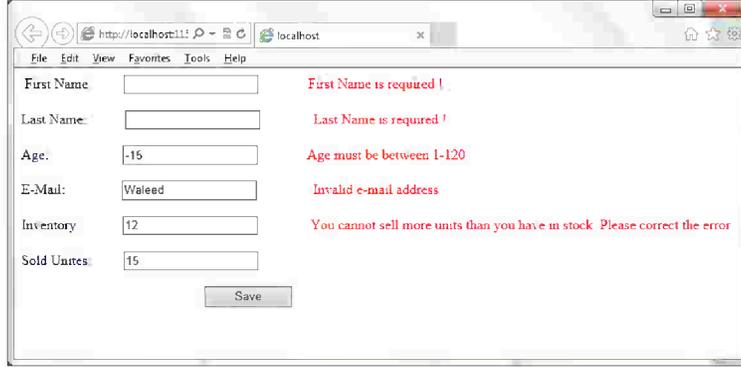


شكل ٦-١ عناصر تحكم التحقق من صحة البيانات

- الأداة **CustomValidator** وتتيح لك إضافة كود مخصص لتحديد مدى صحة أحد حقول الإدخال.
 - الأداة **ValidationSummary** وتقوم بجمع رسائل الخطأ من عدد من أدوات تحكم التحقق من الصحة.
- ولعل الهدف الأساسي من جميع هذه الأدوات هو تبسيط عملية اختبار البيانات التي يقوم المستخدم بإدخالها. وعند تصميم صفحة الويب، تظهر هذه الأدوات كأدوات عنوان تحتوي على الرسائل المناسبة لعدم التحقق من الصحة.

استخدام أدوات التحكم من صحة البيانات

لتوضيح مدى سهولة استخدام أدوات التحقق من الصحة السابقة داخل تطبيقات الويب، سنقوم بإنشاء مثال بسيط لصفحة ويب تحتوي على مجموعة من مربعات النصوص (انظر شكل ٦-٢).



شكل ٦-٢ نموذج الويب المزعم إنشاؤه

تابع معنا الخطوات الآتية:

١. قم بإنشاء تطبيق ويب جديد باسم مناسب وليكن Validation مع اختيار .NET Framework 3.5 بدلاً من .NET Framework 4.5 .
 ٢. قم بإضافة نموذج ويب جديد إلى التطبيق كما سبق باسم Default .
 ٣. قم بإضافة ستة مربعات نصوص إلى النموذج بالأسماء الآتية: txtFirstName و txtLastName و txtAge و txtEMail و txtInventory و txtSoldUnits .
 ٤. قم بحذف النص الافتراضي الموجود في مربعات النص إن وجد حتى لا تظهر أي نصوص داخل هذه المربعات عند فتح النموذج داخل المستعرض .
 ٥. قم بوضع أداة عنوان يسار كل مربع من مربعات النصوص الستة لشرح محتوياته، كما يمكنك أيضاً كتابة العناوين مباشرةً قبل كل مربع .
 ٦. قم بوضع زر أمر Button بالجزء السفلي من النموذج مع تخصيص القيمة Save للخاصية Text والقيمة btnSave للخاصية ID .
- وبذلك قد أضفنا الحقول الأساسية للنموذج. الخطوة التالية هي إضافة أداة تحقق من الصحة لكل حقل من هذه الحقول حيث تختلف الأداة المستخدمة باختلاف نوع العملية والهدف منها.

اختبار الحقول المطلوبة

ربما صادفتك العديد من الأمثلة على الحقول المطلوبة أثناء عملك في الحياة العملية. فلا يمكن مثلاً أن تقوم بسحب مبلغ من المال من حسابك بالبنك دون أن تحدد كمية هذا المبلغ. كذلك الحال في المثال الذى بين أيدينا، من الضروري عدم ترك مربعات النصوص التى تعبر عن الاسم الأول والاسم الأخير للمستخدم خالية. للتحقق من صحة هذه البيانات وإظهار الرسالة المناسبة للمستخدم فى حالة خطئها، تابع معنا الخطوات الآتية:

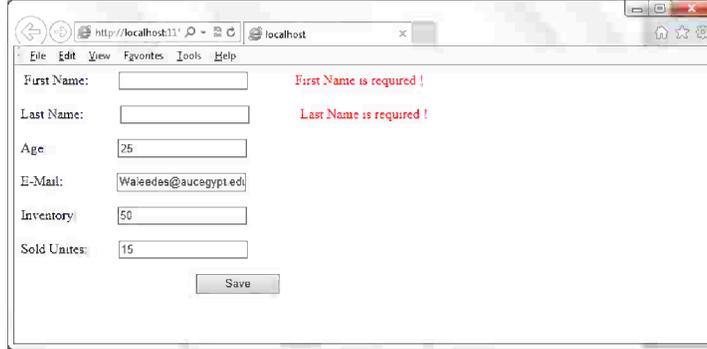
١. من مربع الأدوات، قم بإضافة أدواتي تحكم `RequiredFieldValidator` إلى النموذج، إحدهما يمين مربع النص `First Name` والأخرى يمين مربع النص `Last Name` (راجع شكل ٦-٢).

٢. من نافذة الخصائص المصاحبة لكل أداة من أدواتي التحقق من الصحة، قم بتخصيص القيمة `valFirstName` للخاصية `ID` لأداة التحكم الأولى والقيمة `valLastName` لأداة التحكم الثانية.

٣. قم بتخصيص القيمة `First Name is required !` للخاصية `ErrorMessage` للأداة الأولى والقيمة `Last Name is required !` للأداة الثانية.

٤. قم باختيار أداة التحكم المصاحبة لكل من أدواتي التحقق من الصحة التى قمت بإضافتها وذلك من خلال الخاصية `ControltoValidate` التى تحتوى على مربع سرد يمكنك من خلاله اختيار الأداة المصاحبة.

قم الآن بتشغيل البرنامج. وحينما يظهر النموذج داخل مستعرض الويب، اضغط زر `Save` دون أن تقوم بإدخال أية بيانات بمربعي النص `First Name` و `Last Name`. ماذا تلاحظ؟ يتم إظهار رسائل الخطأ التى قمت بتحديددها بجوار مربع النص بمجرد تحديث الصفحة (انظر شكل ٦-٣).



شكل ٦-٣ تظهر رسائل الخطأ لعدم إدخال بيانات في مربعي النص

التحقق على مستوى مدى من البيانات

نريد أحياناً أن نتأكد من وقوع القيمة المدخلة من المستخدم ضمن مدى معين من القيم. وحقل العمر **Age** في النموذج الذي بين أيدينا على سبيل المثال خير مثال على ذلك. فمن الطبيعي أن تقع جميع أعمار المستخدمين بين ١ و ١٢٠ ومن غير الطبيعي إدخال قيمة سالبة أو قيمة رقمية مثل العدد 2,000. لأداء ذلك، يمكنك استخدام الأداة **RangeValidator** وتخصيص إعداداتها المختلفة بنفس طريقة الأداة السابقة، الفرق الوحيد هو إضافة بعض الخصائص التي تتحكم في مدى القيم المحدد. لإجراء التحقق من الصحة على مدى معين من البيانات (عمر المستخدم) في المثال الذي بين أيدينا، تابع معنا الخطوات الآتية:

١. قم بإدراج الأداة **RangeValidator** من مربع الأدوات إلى النموذج يمين مربع النص **Age**.
٢. قم بتغيير اسم الأداة من خلال الخاصية **ID** إلى اسم مناسب وليكن **valAge**.
٣. انقر النموذج نقرأ مزدوجاً ثم قم بإدخال الكود التالي داخل إجراء الحدث **:Page_Load**

```
valAge.ControlToValidate = "txtAge"
valAge.ErrorMessage = "Age must be between 1-120"
valAge.Type = ValidationDataType.Integer
valAge.MinimumValue = 1
```

valAge.MaxValue = 120

وعلى الرغم من إمكانية استخدام مربع الخصائص في تعيين الخصائص المختلفة لهذه الأداة، إلا أننا آثرنا هنا استخدام الكود إتماماً للفائدة وللتعرف على كيفية التحكم في هذه الخصائص أثناء تشغيل التطبيق.

من الضروري تحديد نوع البيانات التي يتم اختبارها بواسطة الأداة **RangeValidator** وذلك لأن معالجة الأرقام الصحيحة مثلاً تختلف كثيراً عن معالجة الحروف وسلاسل البيانات. فالحرف 2 مثلاً لا يقع في المدى من 1 إلى 120، لذا نحري دائماً اختيار نوع البيانات المناسب.



قم الآن بتشغيل التطبيق مع إدخال عمر لا يقع في المدى 1-120، تلاحظ ظهور رسالة الخطأ يمين المربع **Age** (انظر شكل ٤-٦).

The screenshot shows a web browser window displaying a form with the following fields and values:

First Name:	<input type="text" value="Raneem"/>
Last Name:	<input type="text" value="Waleed"/>
Age:	<input type="text" value="12"/>
E-Mail:	<input type="text" value="Waleedes@aucegypt.edu"/>
Inventory:	<input type="text" value="50"/>
Sold Units:	<input type="text" value="15"/>

A red error message is displayed next to the Age field: "Age must be between 1-120". A "Save" button is visible at the bottom of the form.

شكل ٤-٦ إدخال رقم لا يقع في المدى 1-120

مقارنة قيم الحقول

تعتبر الأداة **CompareValidator** إحدى الأدوات الهامة للتحقق من الصحة حيث تستخدم لإجراء مقارنة حسابية بين القيمة الموجودة بمربع النص وقيمة مقارنة أخرى كما يمكن استخدامها أيضاً للتحقق من صحة نوع بيانات الحقل. ولا تختلف خطوات إضافة هذه الأداة إلى النموذج وتعيين خصائصها كثيراً عن الأدوات السابقة، وإنما الجديد هنا هو تحديد نوع المقارنة من خلال الخصائص **Operator** و **Type** ثم تحديد قيمة المقارنة تبعاً للقواعد العامة التالية:

- عند إجراء المقارنة بين قيمتين في مربعي نصوص، قم بتخصيص اسم المربع الثاني للخاصية **ControlToCompare** ولا تقم بتعيين الخاصية **ValueToCompare**.
- عند إجراء مقارنة بين محتويات الحقل وقيمة أخرى ثابتة، قم بتخصيص هذه القيمة للخاصية **ValueToCompare** ولا تقم بتعيين الخاصية **ControlToCompare**.
- عند التأكد من صحة بيانات مربع النص، فلا تقم بتعيين أي من الخاصيتين **ControlToCompare** و **ValueToCompare** ولكن قم بتعيين نوع البيانات داخل الخاصية **Type**.

يمكنك حقيقةً تصور عملية المقارنة كما لو كانت عبارة **If** مع وجود قيمة الخاصية **ControlToValidate** باليسار ثم معامل المقارنة ثم قيمة المقارنة. فإذا كانت النتيجة **True**، تكون العملية ناجحة وإلا يتم إظهار رسالة الخطأ المحددة كما يلي:

```
If ControlToValidate.operand value Then  
    continue  
Else  
    ErrorMessage  
End If
```

سنقوم فيما يلي بالتأكد من أن القيمة المباعة أقل من أو تساوى القيمة الموجودة بالمخزن.

تابع معنا الخطوات الآتية:

١. قم بإضافة الأداة **CompareValidator** من مربع الأدوات إلى النموذج يمين مربعي النص **Sold Units** و **Inventory**.

٢. أعد تسمية الأداة إلى **valCompare** من خلال الخاصية **ID**.

٣. قم بتخصيص عبارة الخطأ التالية للخاصية **ErrorMessage**:

You cannot sell more units than you have in stock. Please correct the error

٤. قم بتخصيص مربع النص الثاني **txtSoldUnits** إلى الخاصية **ControlToValidate**.

٥. قم بتخصيص مربع النص الأول **txtInventory** إلى الخاصية **ControlToCompare**.

٦. قم باختيار القيمة **LessThanEqual** داخل الخاصية **Operator**.

تعتبر الأداة **CompareValidator** أن القيمة الحالية لمربع النص صحيحة. فإذا أردت التأكد من وجود قيمة داخل المربع، يمكنك إضافة الأداة **RequiredFieldValidator** واستخدامها مع نفس المربع.



والآن قم بتشغيل التطبيق مع إدخال قيمة داخل مربع النص **Sold Units** أقل من القيمة الموجودة بمربع النص **Inventory**، تلاحظ ظهور رسالة الخطأ التي قمنا بإدخالها في الخطوات السابقة (انظر شكل ٦-٥).

The screenshot shows a web browser window with a form. The form has the following fields: First Name (Raneem), Last Name (Waleed), Age (3), E-Mail (Waleedes@aucegypt.edu), Inventory (15), and Sold Unites (50). A red error message is displayed next to the Inventory field: "You cannot sell more units than you have in stock. Please correct the error". A "Save" button is located at the bottom of the form.

شكل ٦-٥ يجب أن تكون الكمية المباعة أقل من أو تساوى الكمية الموجودة بالمخزن

التحقق من صحة الصفحة بالكامل

لعلك الآن تعلم علم اليقين أن استخدام أدوات تحكم التحقق من الصحة يمكنك من إشعار المستخدم بالبيانات الخاطئة التي يقوم بإدخالها. ولكن يبقى القرار النهائي لإرسال البيانات إلى قاعدة البيانات من عدمه في يدك من خلال الكود الذي تقوم بكتابته.

تحتوى كل أداة من أدوات التحقق من الصحة على الخاصية **IsValid** والتي تكون قيمتها **True** إذا نجحت عملية الاختبار بينما تكون هذه القيمة **False** في حالة العكس لا قدر الله. يمكنك استخدام الخاصية **IsValid** الخاصة بالكائن **Page** الذي يعبر عن الصفحة بالكامل لاختبار جميع أدوات التحقق من الصحة الموجودة بالصفحة وذلك كما في الكود التالي الخاص بحدث نقر الزر **Save**:

```
Private Sub btnSave_Click(sender As Object, e As EventArgs)
Handles btnSave.Click
If Page.IsValid Then
    Call YourSaveDataFunction()
Response.Redirect("http://www.compuscience.com.eg")
End If
End Sub
```

يتم في هذا الكود استدعاء دالة مخصصة باسم **YourSaveDataFunction()** لتخزين بيانات المستخدم ثم الذهاب إلى صفحة أخرى، هذا فقط في حالة صحة هذه البيانات وذلك

من خلال الخاصية `IsValid`.

التحكم في رسائل الخطأ

لعلك لاحظت في المثال الذى بين أيدينا أننا نقوم بوضع أدوات التحقق من الصحة يمين الحقول التي يتم التحقق من صحتها وبالتالي تظهر الرسائل دائماً يمين هذه الحقول وهذا هو النمط النموذجي لصفحات الويب، وهذا في حالة وجود متسع داخل الصفحة لعرض هذه الرسائل. أما في حالة عدم توفر هذه المساحة، فيمكنك استخدام الأداة `ValidationSummary` والتي لا ترتبط بجقل معين داخل الصفحة وإنما تقوم بعرض رسائل الخطأ الموجودة بأدوات التحقق من الصحة الموجودة بالصفحة إذا كانت قيمة الخاصية `IsValid` بها `False` وهو ما يعنى خطأ البيانات التي قام المستخدم بإدخالها.

ولأن نص أداة التحكم `ValidationSummary` مبنى أساساً على رسائل الخطأ الموجودة بالأدوات الأخرى، فمن الضروري تعيين هذه الرسائل بتلك الأدوات إلى جانب إمكانية إضافة عنوان مخصص للأداة باستخدام الخاصية `HeaderText` وتحديد نمط عرض النص المختصر من خلال الخاصية `DisplayMode` كما في الكود التالي:

```
valSum.HeaderText = "Your request cannot be processed because: "  
valSum.DisplayMode = ValidationSummaryDisplayMode.  
BulletList
```

تدعم بعض مستعرضات الويب إظهار نص الأداة `ValidationSummary` داخل مربع رسالة وذلك بتخصيص القيمة `True` للخاصية `ShowMessageBox` كما يمكنك تعطيل هذه الأداة وعدم إظهار محتوياتها داخل المستعرض عن طريق تخصيص القيمة `False` للخاصية `ShowSummary` الخاصة بالأداة.



التحقق من صحة التعبيرات

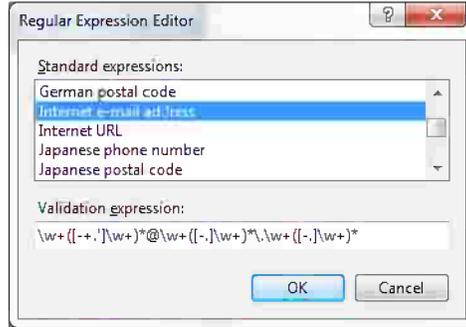
تقوم معظم أدوات التحقق من الصحة الموجودة داخل نطاق ASP.NET بأداء العمليات البسيطة، إلا أن النموذج البرمجي في جعبته العديد من الاختبارات المركبة من خلال إمكانية التحقق من صحة التعبيرات المعتادة **Regular Expressions** والتي تأخذ تنسيقات معينة، هذا بالإضافة إلى إمكانية برمجة الكود الخاص بك لأداء المزيد من المهام. يمكنك استخدام الأداة **RegularExpressionValidator** لمقارنة قيمة مربع النص بأحد التعبيرات المعتادة. ففي المثال الذي بين أيدينا، لدينا حقل لإدخال البريد الإلكتروني الخاص بالمستخدم. وكما نعرف فإن البريد الإلكتروني يحتوى دائماً على الاسم متبوعاً بالعلامة **@** ثم اسم مجال الإنترنت المستخدم هكذا مثلاً:

Waleedes@Aucegypt.edu

استخدام التعبيرات المعتادة

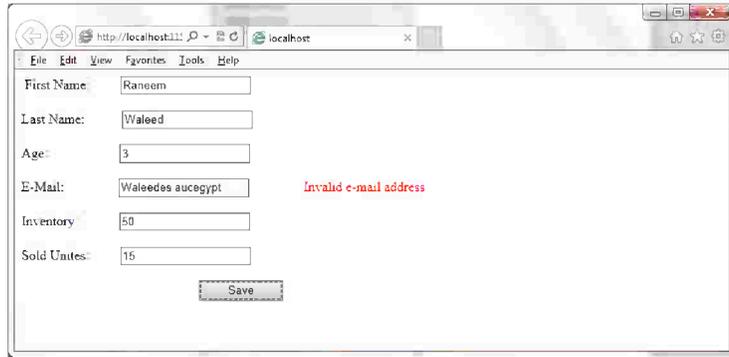
سنقوم فيما يلي بالتعرف على كيفية استخدام هذه الأداة للتحقق من صحة البريد الإلكتروني الذي يقوم المستخدم بإدخاله. تابع معنا الخطوات الآتية:

1. قم بإضافة الأداة **RegularExpressionValidator** من مربع الأدوات إلى النموذج يمين مربع النص **Email**.
2. قم بتغيير اسم الأداة إلى اسم معبر وليكن **valRegular** من خلال الخاصية **ID** المصاحبة للأداة.
3. قم بتخصيص القيمة **txtEmail** للخاصية **ControlToValidate** المصاحبة للأداة.
4. قم بتخصيص العبارة **Invalid e-mail address** للخاصية **ErrorMessage**.
5. انقر الزر المجاور للخاصية **ValidationExpression**، يظهر المربع الحوارى **Regular Expression Editor** (انظر شكل ٦-٦).



شكل ٦-٦ اختيار التعبير المناسب

٦. اختر **Internet e-mail address** من قائمة التعبيرات القياسية ثم انقر زر **Ok**.
والآن قم بتشغيل التطبيق مع إدخال بريد إلكتروني غير صحيح، وفي هذه الحالة تظهر رسالة الخطأ التي قمنا بتعيينها منذ قليل (انظر شكل ٦-٧).



شكل ٦-٧ تظهر رسالة الخطأ في حالة إدخال بريد إلكتروني غير صحيح

إنشاء دالة تحقق مخصصة

إذا لم تلبي أي من أدوات التحقق من الصحة الموجودة داخل **ASP.NET** متطلباتك، يمكنك إنشاء دالة مخصصة للتحقق من الصحة من خلال الأداة **CustomValidator** والتي تقوم تلقائياً باستدعاء هذه الدالة عند الضرورة والتي تقوم بدورها هي الأخرى بتخصيص القيمة المناسبة للخاصية **IsValid**.
يمكنك حقيقةً كتابة كل ما تريد من وظائف داخل الدالة المخصصة بما في ذلك البحث في

قاعدة البيانات. يوضح الكود التالي دالة مخصصة تقوم بالتأكد من أن القيمة المدخلة في مربع النص عبارة عن يوم من أيام الأسبوع.

```
Private Sub CheckWeekDay(source As Object, args As ServerValidateEventArgs)
    Dim userInput As String = args.Value
    If IsDate(userInput) Then
        Dim theDate As Date = Convert.ToDateTime(userInput)
        If theDate.DayOfWeek = DayOfWeek.Saturday Or
        theDate.DayOfWeek = DayOfWeek.Sunday Then
            args.IsValid = False
        Else
            args.IsValid = True
        End If
    Else
        args.IsValid = False
    End If
End Sub
```

يحتوي المعامل الثاني الممرر للدالة المخصصة على خاصيتين هامتين، الأولى هي الخاصية Value والتي تتضمن القيمة التي قام المستخدم بإدخالها، والثانية عبارة عن الخاصية IsValid والتي يقوم كود الدالة المخصصة بتعيينها بناءً على مدى صحة البيانات من عدمه. وكى تعمل هذه الدالة بشكل صحيح، يجب إضافة معالج للحدث الموجود على الخادم كما يلي:

```
AddHandler valMeetingDate.ServerValidate,
AddressOf CheckWeekDay
```

حيث valMeetingDate عبارة عن اسم الأداة CustomValidator.

