

الفصل السابع مفهوم خدمات الويب

تعتبر خدمات الويب **Web Services** من السمات الأساسية داخل نطاق **.NET**. والتي حاولت مايكروسوفت من خلالها التمهيد لنقل التركيز من برامج الحاسب المثبتة على كل حاسب شخصي على حده إلى البرامج المتاحة من خلال الإنترنت.

بانتهاء هذا الفصل ستتعرف على:

- مفهوم خدمات الويب.
- إنشاء خدمة ويب باستخدام محرر نصوص.
- اختبار خدمة الويب.
- وصف خدمة الويب.
- استخدام خدمة الويب.
- إنشاء خدمة الويب باستخدام بيئة تطوير **Visual Studio 2012**.

مفهوم خدمات الويب

خدمة الويب **Web Service** عبارة عن مجموعة من الدوال التي يتم الوصول إليها من خلال عنوان معين باستخدام بعض المقاييس المعروفة من قبل مايكروسوفت. كما أنها إحدى التقنيات الأساسية المعروفة داخل نطاق **NET**. والتي تساعد على التطوير السريع للتطبيقات باستخدام العديد من الأدوات التي تم تصميمها خصيصاً لتطوير هذه الخدمات وكذلك المعالجات التي تقوم تلقائياً بإنشاء الكود نيابةً عنك.

تتصل خدمات الويب مع بعضها البعض ومع أجهزة العميل باستخدام بعض المقاييس والبروتوكولات مثل **XML** و **HTTP**، حيث يمثل **XML** البيانات الموجودة داخل خدمات الويب، بينما يمثل **HTTP** البروتوكول الذي يقوم بنقل هذه البيانات من الخادم إلى العميل الذي يقوم بدوره باستدعاء هذه الخدمة أو استخدامها. وباستخدام كلٍ من **HTTP** و **XML** يستطيع أى نظام تشغيل متصل بالإنترنت استخدام خدمات الويب التي يوفرها **NET**.

وعلى الجانب الآخر فإن خدمات الويب تقوم أساساً بتمثيل تكامل التطبيقات، فهي لا تعتمد على البيئة التي تستخدمها أو اللغة المستخدمة في إنشائها أو نظام التشغيل المستخدم في الوصول إليها، وهذا نتيجةً للمقاييس والبروتوكولات التي تدعم أساساً كل هذه السمات. لذا تساعد خدمات الويب على تكامل التطبيقات بتجميع مكونات من بيئات وأنظمة تشغيل ولغات برمجة مختلفة.

تشابه خدمات الويب مع المكونات **COM** في احتوائها على مجموعة من الوظائف المجمعة في عنصر واحد حيث تعمل كل خدمة من خدمات الويب كوحدة مستقلة يتم إخفاء وظائفها الداخلية عن العميل، ومن ثمّ يمكنك تضمينها داخل أى تطبيق ويب دون وجود أى مشاكل في عملية التمثيل.

وعامةً من الممكن أن تحتوى البرامج على مراجع للبرامج والدوال الأخرى المخزّنة على حاسب آخر بالشبكة. أما خدمات الويب فتعمل بنفس الطريقة إلا أنها تتميز بإمكانية

الاستخدام من خلال الإنترنت ومن ثمّ يستطيع أى حاسب متصل بالإنترنت استخدام هذه الخدمات.

إنشاء خدمة ويب باستخدام محرر نصوص

يحتوى نطاق NET. على الأدوات اللازمة لإنشاء خدمات الويب. سنقوم فيما يلى بإنشاء خدمة ويب بسيطة باستخدام ASP.NET التى تعد جزءاً هاماً داخل نطاق NET.. وعمامة يفضل استخدام ASP.NET فى إنشاء خدمات الويب والوصول إليها للأسباب الآتية:

- تدعم ASP.NET استخدام البروتوكول SOAP ومن ثمّ يمكن استخدامها فى إرسال واستقبال الطلبات باستخدام العناصر <HTTP GET> و<HTTP POST> على الترتيب.

- تقوم ASP.NET تلقائياً بإنشاء ملفات WSDL المصاحبة لخدمة الويب.

- من الممكن استخدام ASP.NET فى إنشاء ملف الاستكشاف الذى يحتوى على قائمة بخدمات الويب الموجودة داخل URL معين.

- تقوم ASP.NET تلقائياً بإنشاء الكود اللازم لتنفيذ الإجراءات المختلفة مثل إرسال الرسائل أو استقبالها وكذلك ترميز البيانات باستخدام XML.

- تقوم ASP.NET بالتأكد من إخفاء التمثيل الخاص بالخادم عن العميل.

سنقوم فيما يلى بإنشاء خدمة ويب دون استخدام بيئة تطوير Visual Studio2012

وذلك باستخدام أحد برامج تحرير النصوص وليكن برنامج Notepad.

الخطوة الأولى فى إنشاء خدمة الويب هى كتابة الكود. افتح برنامج Notepad أو أى محرر

نصوص آخر ثم قم بإدخال الكود التالى:

1. <%@WebService Language = "vb" class = "TimeUtilities %>"
2. Imports System
3. Imports System.Web.Services
4. <WebService(Namespace:="http://tempuri.org/MyServices")>public class TimeUtilities
5. Inherits WebService

```
6. <WebMethod> public Function GetTime() As string
7. Dim ret As string = "The current time on server is: "
8. ret &= DateTime.Now.ToString()
9. return ret
10. End Function
11. End Class
```

وعن هذا الكود، نوضح ما يلي:

- لا يوجد جديد حقيقةً في الكود السابق غير السطر رقم ١ الذي يحتوي على العنصر `<%@ WebService...%>` الذي يقوم بدوره بإخبار ASP.NET أن الصفحة الحالية عبارة عن خدمة ويب من خلال كلمة `WebService`. كما يحتوي هذا العنصر أيضاً على معلومات إضافية عن هذه الخدمة مثل لغة البرمجة المستخدمة في إنشاء الخدمة (VB) وكذلك اسم تصنيف الخدمة الذي تختاره بنفسك.
- في السطر رقم ٤ يتم إنشاء التصنيف `TimeUtilities` المشتق من التصنيف `System.Web.Services.WebService` الموجود بدوره داخل المسمى `System.Web.Services` الذي يحتوي على التصنيفات اللازمة لإنشاء وبناء خدمات الويب وخاصةً التصنيف `WebService` الذي يعد التصنيف الأساسي لجميع خدمات الويب المنشأة باستخدام ASP.NET (السطر رقم ٥). لاحظ أننا قمنا بتعريف الصفة الآتية قبل تعريف التصنيف:
`Namespace:="http://tempuri.org/MyServices"`
والذي يحتوي على المسمى الذي تريد وضع تصنيف خدمة الويب بداخله.
- تحتوي السطور من ٦ إلى ١٠ على كود الدالة `GetTime()` التي تقوم بإرجاع الوقت الموجود على الخادم. وتم استخدام الصفة `<WebMethod>` مع هذه الدالة دلالةً على إمكانية الوصول إلى الدالة ليس فقط من داخل الحاسب الحالي وإنما من خلال أي حاسب آخر متصل بالإنترنت.

قم بحفظ ملف الكود السابق باسم مناسب مع استخدام الامتداد **.asmx**. وليكن **TimeUtilities.asmx**، مع تخزين الملف داخل المجلد المرجعي لخدم بيانات الإنترنت **Internet Information Server (IIS)** وهو **wwwroot** في هذه الحالة.

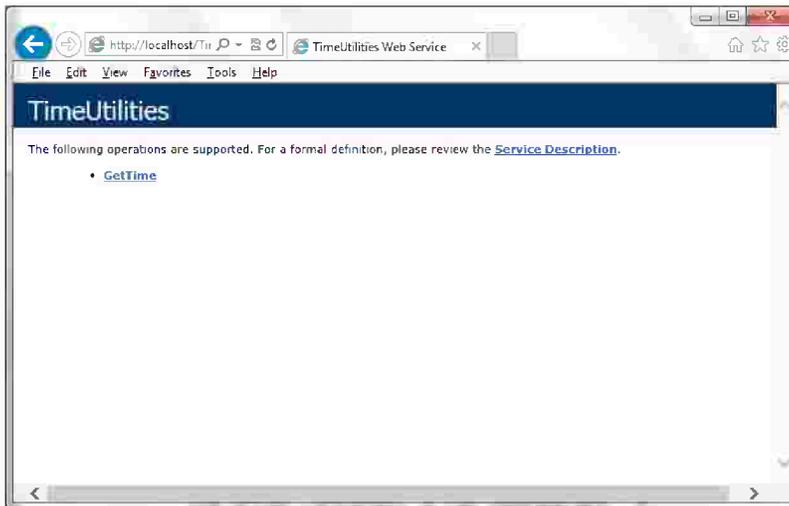
اختبار خدمة الويب

بعد الانتهاء من إنشاء ملف خدمة الويب، يمكنك حينئذٍ عرضها داخل نافذة مستعرض الويب بكتابة العنوان كاملاً، حيث يقوم المستعرض بالبحث عن الملف الذي يتم تفسيره من قبل **ASP.NET** كخدمة ويب ويتم استدعاء التصنيف المصاحب وعرض خدمة الويب داخل نافذة المستعرض. ويقوم **ASP.NET** تلقائياً بإضافة بعض بيانات التوثيق بالإضافة إلى إمكانية الوصول إلى وظائف ودوال الخدمة. تابع معنا الخطوات الآتية:

١. قم بكتابة العنوان التالي داخل مستعرض الويب:

http://localhost/TimeUtilities.asmx

٢. اضغط مفتاح الإدخال، تحصل على صفحة خدمة الويب محتويةً على قائمة بالوظائف (الدوال) الموجودة بالخدمة (لاحظ أن لدينا دالة واحدة فقط وهي الدالة **(GetTime)**) كما في شكل ٧-١ التالي.

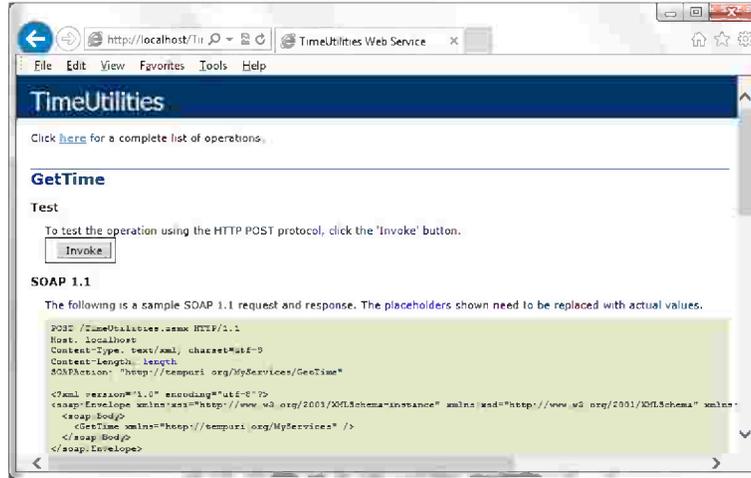


شكل ٧-١ عرض خدمة الويب داخل نافذة المستعرض

٣. انقر الارتباط **GetTime**، تظهر صفحة ويب أخرى تحتوى على قالب توثيق للدالة **GetTime()** (انظر شكل ٧-٢).

٤. انقر زر **Invoke** داخل صفحة الويب، تحصل على كود **XML** التالى الذى يقوم الخادم **IIS** بإرساله إلى المستعرض استجابةً لطلبك حيث يقوم الخادم بتنفيذ الدالة وإرجاع الوقت الحالى طبقاً لساعة الحاسب:

```
<?xml version="1.0" encoding="UTF-8"?>
<string xmlns="http://tempuri.org/MyServices">The current time
on server is: 9/15/2013 9:47:18 PM</string>
```

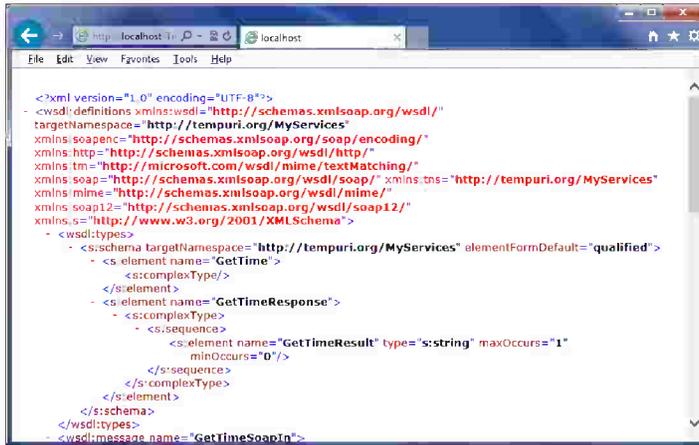


شكل ٧-٢ تقوم **ASP.NET** بإنشاء قالب توثيقى للدالة **GetTime()**

وصف خدمة الويب

ذكرنا منذ قليل استخدام ملف التوصيف **WSDL** المصاحب لخدمة الويب والذى يحتوى على التفاصيل الضرورية للاتصال بهذه الخدمة واستخدامها. لمشاهدة محتويات ملف **WSDL** المصاحب لخدمة الويب التى أنشأناها، انقر الارتباط **Service Description** من الصفحة الرئيسية للخدمة أو قم بإضافة الحروف **?WSDL** إلى عنوان الصفحة (راجع شكل ٧-١)، تحصل على صفحة ويب تحتوى على ملف التوصيف المكتوب بلغة

XML والذي يحتوي على ما يقرب من ١٢٠ سطر على الرغم من البساطة الشديدة لخدمة الويب المصاحبة والتي تقوم فقط بإرجاع الوقت الحالي من خلال دالة واحدة (انظر شكل ٣-٧).



شكل ٣-٧ محتويات ملف التوصيف المصاحب لخدمة الويب

وفيما يلي نقوم إتماماً للفائدة بالتعرف على السمات العامة لمحتويات ملف التوصيف حيث يحتوي الملف على العناصر الأساسية الآتية:

```
<definitions>
  <types>
    <message name="GetTimeSoapIn">
    <message name="GetTimeSoapOut">
    <message name="GetTimeHttpGetOut">
    <message name="GetTimeHttpPostOut">
    <portType name="TimeUtilitiesSoap">
    <portType name="TimeUtilitiesHttpGet">
    <portType name="TimeUtilitiesHttpPost">
    <binding name="TimeUtilitiesSoap">
    <binding name="TimeUtilitiesHttpGet">
    <operation name="GetTime">
    <binding name="TimeUtilitiesHttpPost">
    <service name="TimeUtilities">
  </definitions>
```

يحتوى ملف التوصيف على أنواع متعددة من العناصر مثل **Message** و **Types** و **PortType** و **Binding** و **Service** حيث تقوم العناصر **Types** بوصف أسماء الوظائف والدوال الموجودة بخدمة الويب وكذلك المعاملات وأنواع البيانات التي تقوم هذه الدوال بإرجاعها بينما تصف العناصر الأخرى كيفية الاتصال بين العميل وخدمة الويب. والآن نقوم بإلقاء نظرة سريعة على كل عنصر من هذه العناصر.

العنصر *Service*

يقوم العنصر **Service** بتعريف اسم خدمة الويب (**TimeUtilities** في هذه الحالة) كما يقوم بتعريف مجموعة من المنافذ **Ports** حيث يعبر كل منفذ من هذه المنافذ عن أحد البروتوكولات التي تدعمها خدمة الويب. فخدمة الويب التي بين أيدينا على سبيل المثال تدعم البروتوكولات **SOAP** و **HTTP Post** و **HTTP Get** كما يلي:

```
<service name="TimeUtilities">
  <port name="TimeUtilitiesSoap"
    binding="s0:TimeUtilitiesSoap">
    <soap:address location="http://localhost/timeutilities.asmx" />
  </port>
  <port name="TimeUtilitiesHttpGet"
    binding="s0:TimeUtilitiesHttpGet">
    <http:address location="http://localhost/timeutilities.asmx" />
  </port>
  <port name="TimeUtilitiesHttpPost"
    binding="s0:TimeUtilitiesHttpPost">
    <http:address location="http://localhost/timeutilities.asmx" />
  </port>
</service>
```

لا تدعم جميع خدمات الويب استخدام البروتوكول **HTTP** وذلك لصعوبة تمرير بعض المعاملات المركبة من وإلى خدمة الويب من خلال **HTTP**. أما البروتوكول **SOAP** على الجانب الآخر فيدعم أي نوع من أنواع البيانات.



العنصر *PortType*

يدعم كل منفذ من المنافذ عملية معينة **Operation** تكون مصاحبة لإحدى الوظائف الموجودة داخل خدمة الويب. ففي الكود التالي تجد أن نوع البيانات **SOAP** يدعم تنفيذ الدالة **GetTime()**. كما تحتوي كل عملية على نوعين من الرسائل **Messages**، الأول عبارة عن رسالة الإدخال **Input Message** التي يتم من خلالها استدعاء الوظيفة (الدالة) والثاني عبارة عن رسالة الإخراج **Output Message** لاسترجاع البيانات من الوظيفة وذلك كما في الكود التالي:

```
<portType name="TimeUtilitiesSoap">
  <operation name="GetTime">
    <input message="s0:GetTimeSoapIn" />
    <output message="s0:GetTimeSoapOut" />
  </operation>
</portType>
```

العنصر *Message*

ويستخدم هذا العنصر في وصف كيفية تمرير المعاملات من وإلى خدمة الويب. يوضح الكود التالي تعريف الرسالة **GetTimeSoapIn**:

```
<message name="GetTimeSoapIn">
  <part name="parameters" element="s0:GetTime" />
</message>
```

وكما ترى فإن التعريف بسيط للغاية لأن الوظيفة **GetTime()** لا تحتوي على أى معاملات.

استخدام خدمة الويب

على الرغم من إمكانية الوصول إلى خدمة الويب من خلال مستعرض الويب، إلا أنه من الأفضل نداء هذه الخدمة من داخل تطبيق أو صفحة ويب أخرى أو ما نطلق عليه "عميل الويب" **Web Client** هو التصنيف القادر على إرسال واستقبال الطلبات والرسائل إلى ومن خدمة الويب. وكما ذكرنا من قبل فإن خدمة الويب تتكون أساساً من جزأين أحدهما خاص بالخادم والآخر خاص بالعميل. وقد قمنا في الجزء المنقضى من هذا الفصل بالتركيز على إنشاء خدمات الويب وهو الجزء المتعلق بالخادم **Server-side Web Services**. إلا أنك تستطيع إنشاء عملاء **Clients** لخدمة الويب للوصول إلى بياناتها وذلك من خلال بيئة تطوير **Visual Studio 2012**. وعامةً يتكون عميل الويب من جزأين، الأول هو تفويض الويب **Web Proxy** والآخر عبارة عن برنامج العميل **Client Program**.

إنشاء تفويض الويب **Web Proxy**

تفويض الويب عبارة عن تصنيف قادر على استدعاء الدوال الموجودة داخل خدمة الويب دون الحاجة إلى الاهتمام بكيفية الاتصال أو تمرير المعاملات من وإلى الوظائف الموجودة بالخدمة، فهو يقوم بإرسال البيانات والمعاملات من العميل إلى خدمة الويب، لذا يتم تثبيته على جهاز العميل.

يمكنك إنشاء تفويض الويب باستخدام محث الأوامر أو من خلال بيئة تطوير **Visual Studio 2012** كما سنرى فيما بعد. لإنشاء تفويض ويب من محث الأوامر، افتح قائمة **All Programs >> Microsoft Visual Studio Start** بشريط المهام ثم اختر **2012 >> Visual Studio Tools>>Developer Command Prompt for Developer VS2012** ثم قم بتنفيذ الأمر **Wsdll** التالي من موضع محث النافذة **Command Prompt for VS2012** (انظر شكل ٧-٤):

```
wsdl /l:vb /n:ServerTime /out:TimeProxy.vb  
http://localhost/TimeUtilities.asmx?WSDL
```

يتكون هذا الأمر من العناصر الآتية:

- يقوم الجزء //I:vb بإخبار البرنامج أنه يستخدم لغة **Visual Basic** في إنشاء التفويض.
- يقوم الجزء /n:ServerTime بتعيين مسمى **Namespace** لأدوات **WSDL** يتم استخدامه عند إنشاء تصنيف التفويض. يمكنك استخدام الاسم الذي يروق لك.
- يقوم الجزء /out:TimeProxy.vb بتعريف ملف الخرج الذى يحتوى على تصنيف التفويض.
- يحتوى الجزء الأخير بالأمر على ملف التوصيف **WSDL** حيث يمكنك تعيين اسم الملف مباشرةً أو تعيين عنوانه كما فعلنا في هذا المثال.



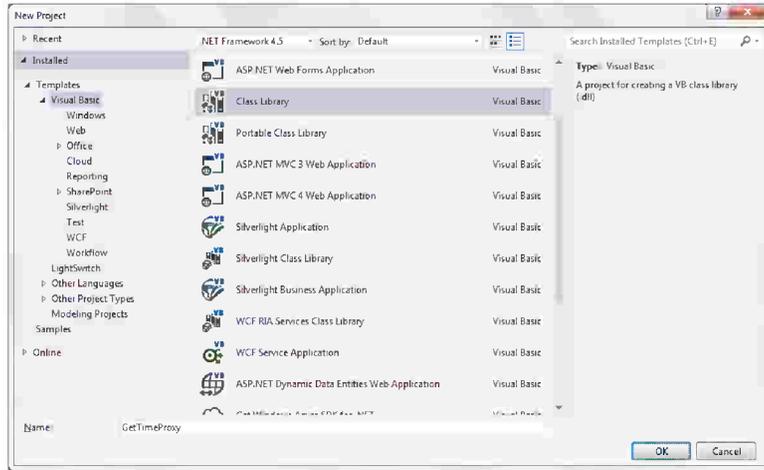
```
Administrator: Developer Command Prompt for VS2012
C:\Program Files (x86)\Microsoft Visual Studio 11.0>wsdl /I:vb /n:ServerTime /out:TimeProxy.vb http://localhost/TimeUtilities.asmx?WSDL
Microsoft (R) Web Services Description Language Utility
[Microsoft (R) .NET Framework, Version 4.0.30319.17929]
Copyright (C) Microsoft Corporation. All rights reserved.
Writing File 'TimeProxy.vb'.
C:\Program Files (x86)\Microsoft Visual Studio 11.0>
```

شكل ٧-٤ إنشاء تفويض ويب من محث الأوامر

وبذلك نحصل على الملف **TimeProxy.vb** الذى يحتوى على عدد من التصنيفات والدوال التى تتحكم فى طريقة إجراء الاتصالات بين خدمة الويب والعميل حيث يلعب البروتوكول **SOAP** دوراً هاماً فى هذه الحالة. وأهم تصنيف بين تصنيفات هذا الملف هو التصنيف **Service1** الذى يحتوى على دالة بنفس توقيع الوظيفة **(GetTime)** التى أنشأناها من قبل.

الخطوة التالية هى ترجمة تفويض الويب الذى قمنا بإنشائه. يمكنك ترجمة تفويض الويب من داخل بيئة تطوير **Visual Studio 2012**. لأداء ذلك، تابع معنا الخطوات الآتية:

١. قم بإنشاء مشروع جديد. لأداء ذلك، افتح قائمة **File** واختر **NewProject** من القائمة المنسدلة، يظهر المربع الحوارى المعتاد **New Project**.
٢. تأكد من اختيار **Visual Basic** من العمود الأيسر بالمربع الحوارى ثم اختر الرمز **Class Library** من العمود الأوسط بالمربع لأننا نرغب فى إنشاء مكتبة تصنيفات جديدة (انظر شكل ٧-٥).
٣. قم بتعيين اسم مناسب للمشروع داخل مربع النص **Name** وليكن **GetTimeProxy**.
٤. انقر زر **Ok**، يتم إنشاء المشروع الجديد بعد قليل.



شكل ٧-٥ إنشاء مكتبة تصنيفات جديدة

٥. افتح قائمة **Project** من شريط القوائم ثم اختر **Add Existing Item** من القائمة المنسدلة وقم بإضافة ملف تفويض الويب **TimeProxy.vb** الذى أنشأناه منذ قليل والموجود بالمجلد **C:\Program Files (x86)\Microsoft Visual Studio 11.0**.

٦. من نافذة المستكشف، انقر الخيار **References** بزر الفأرة الأيمن ثم اختر **Add Reference** من القائمة الموضوعية، يظهر المربع الحوارى **Reference Manager**.

٧. قم باستعراض المسميات الموجودة بمربع السرد ومنه اختر المسمى **System.Web.Services** من الجزء **Framework**، يتم إضافة المرجع إلى المراجع الموجودة بالمشروع.

٨. افتح قائمة **Build** من شريط القوائم ثم اختر **BuildGetTimeProxy** من القائمة المنسدلة لترجمة المشروع بالكامل ومن ثم ترجمة تفويض الويب الموجود بالمشروع حيث يتم إنشاء مكتبة تصنيفات تحتوى على التفويض الذى قمنا بإنشائه وذلك باسم **GetTimeProxy.dll** أسفل المجلد **bin** بمجلد التطبيق.

لتوضيح استخدام خدمة الويب داخل برنامج صغير، قم بإنشاء ملف باسم **CallService.vb** داخل برنامج **Notepad** مثلاً ثم قم بإدخال الكود التالى إلى الملف:

```
Imports System
Imports ServerTime
namespace serverTime
public class callService
public static sub Main()
Dim remoteService As new TimeUtilities()
Console.WriteLine(remoteService.GetTime())
End Sub
End class
End Namespace
```

والآن قم بترجمة هذا الملف من خلال نافذة المحث باستخدام الأمر التالى:

vbcCallService.vb /r:TimeProxy.dll

قم بتشغيل التطبيق **CallService** تحصل على النتيجة التالية:

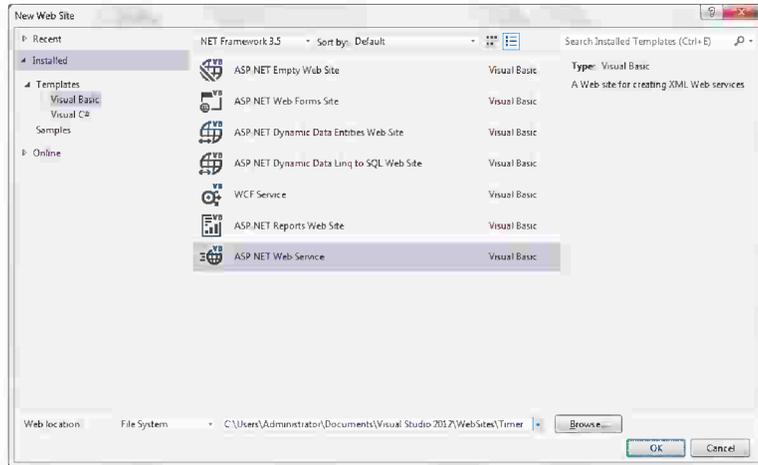
Calling web service...

The Current time on the server is: 15/09/2013 10:55:14 PM

إنشاء خدمة الويب باستخدام بيئة تطوير Visual Studio 2012

يمكنك استخدام بيئة التطوير Visual Studio 2012 لإنشاء خدمات ويب قوية ومعقدة بسهولة تامة. سنقوم فيما يلي بإنشاء نفس خدمة الويب السابقة من خلال بيئة التطوير. لأداء ذلك، تابع معنا الخطوات الآتية:

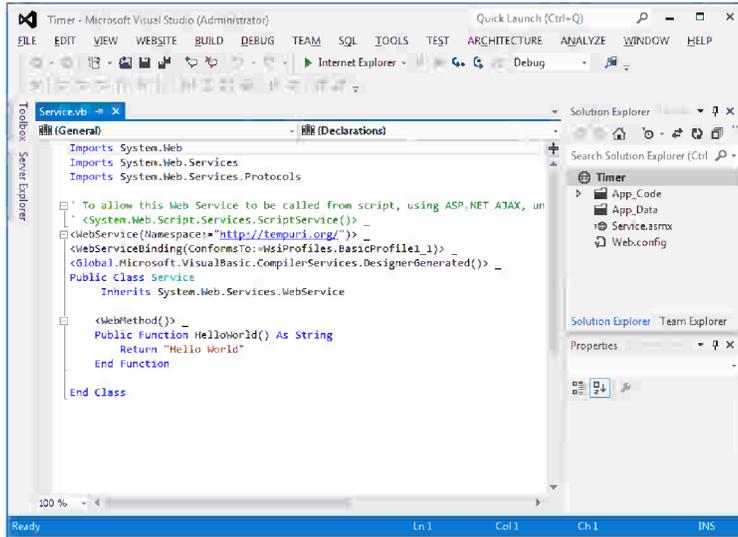
1. تأكد أنك داخل بيئة تطوير Visual Studio 2012 ثم افتح قائمة File من شريط القوائم واختر New Web Site من القائمة المنسدلة، يظهر المربع الحوارى New Web Site (انظر شكل ٧-٦).



شكل ٧-٦ المربع الحوارى New Web Site

2. اختر .NET Framework 3.5 بدلاً من .NET Framework 4.5. بمربع السرد الموجود بالجزء العلوى من المربع الحوارى.
3. اختر الرمز ASP.NET Web Service من العمود الأوسط بالمربع الحوارى.
4. قم بتعيين اسم خدمة الويب ومكانها داخل مربع النص المجاور للقائمة Web Location وليكن Timer.
5. انقر زر Ok، يتم بعد قليل إنشاء خدمة الويب (انظر شكل ٧-٧).

الفصل السابع: مفهوم خدمات الويب



شكل ٧-٧ يتم إنشاء خدمة الويب تلقائياً

بالنظر إلى نافذة المستكشف في الشكل السابق، تلاحظ وجود ملفين هما:

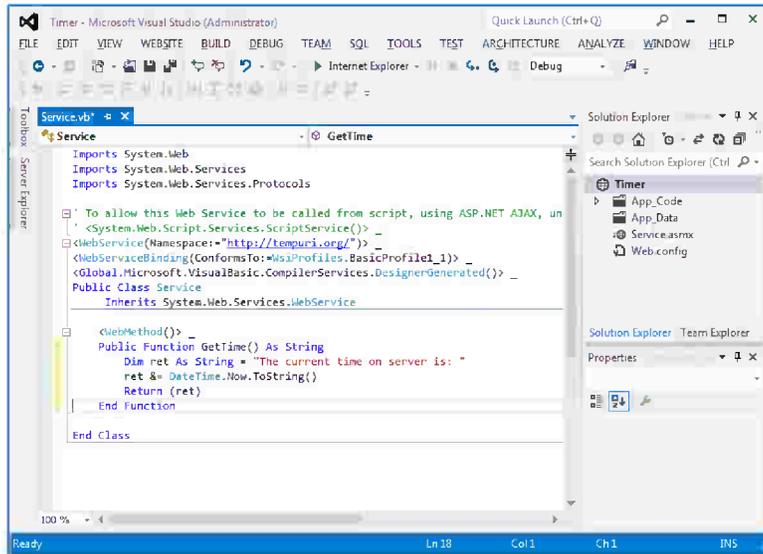
- الملف **Service.asmx** وهو عبارة عن صفحة **ASP.NET** يتم عرضها داخل نافذة مستعرض الويب ويشبه إلى حد كبير الملف **.aspx**. المستخدم في صفحات **ASP.NET** العادية.
 - الملف **Service.vb** ويحتوي على كود **Visual Basic** الذى يقوم بتنفيذ مهام خدمة الويب مثل إرجاع الوقت والتاريخ الحاليين كما في هذا المثال. يطلق على هذا الملف اسم **Code behind**. لعرض محتويات هذا الملف من داخل النموذج، انقر النموذج بزر الفأرة الأيمن ثم اختر **View Code** من القائمة الموضعية، حيث يتم إظهار هذا الملف تلقائياً بمجرد إنشاء خدمة الويب.
- الخطوة التالية هي إضافة الكود الذى يعبر عن مهام خدمة الويب إلى الملف **Service.vb**.
لأداء ذلك، تابع معنا الخطوات الآتية:

١. من نافذة المستكشف، انقر الملف **Service.vb** نقرأ مزدوجاً إذا لم يكن مفتوحاً بالفعل، يتم فتح الملف الذي يحتوي على الكود المنشأ من قِبَل بيئة تطوير **Visual Studio2012**.
٢. قم بكتابة كود الوظيفة **GetTime()** كما سبق داخل برنامج **Notepad** إلى الملف **Service.vb** (انظر شكل ٧-٨) هكذا:

```
<WebMethod()> _
    Public Function GetTime() As String
        Dim ret As String = "The current time on server is: "
        ret &= DateTime.Now.ToString()
        Return (ret)
    End Function
```

مع حذف الكود الموجود والخاص بالدالة **HelloWorld()** وهو الكود التالي:

```
<WebMethod()> _
    Public Function HelloWorld() As String
        Return "Hello World"
    End Function
```



شكل ٧-٨ كود خدمة الويب بعد كتابة الوظيفة **GetTime()**

وكما ترى فإن هناك تشابهاً كبيراً بين كود الملف **vb**. والملف السابق الذى قمنا بإنشائه داخل برنامج **Notepad**. الفرق الوحيد هو تنظيم الكود فى حالة ملف **vb**. وكذلك وجود مثال صغير لخدمة ويب تقوم بعرض العبارة "**Hello World**" على الشاشة (راجع شكل ٧-٧).

لا يتبقى لنا الآن إلا بناء المشروع وتنفيذه. لبناء المشروع، افتح قائمة **Build** من شريط القوائم ثم اختر **Build Web Site** من القائمة المنسدلة. ولتنفيذه، افتح قائمة **Debug** ثم اختر **Start Debugging** من القائمة المنسدلة (أو اضغط مفتاح **F5** من لوحة المفاتيح)، تحصل على نفس النتيجة السابقة التى حصلنا عليها من خارج **Visual Studio** (راجع شكل ٧-١).

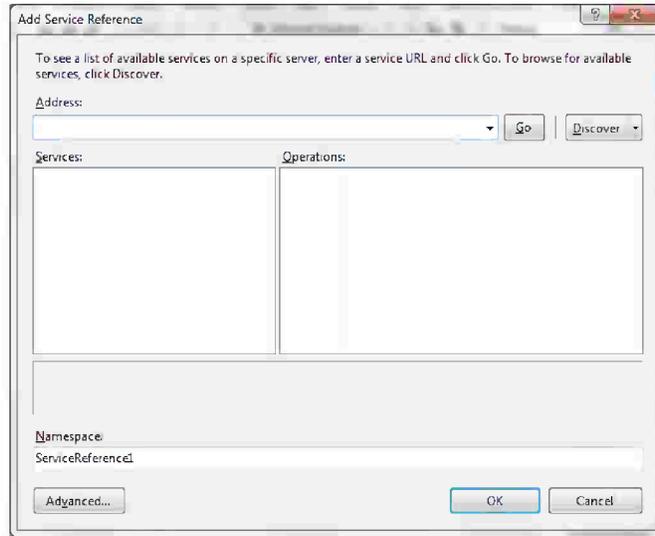
وهكذا يتم إنشاء خدمة الويب باستخدام بيئة تطوير **Visual Studio 2012** كما يتم استدعائها والحصول على خرج البرنامج بنفس الطريقة السابقة فى حالة استخدام **Notepad**. وهنا ربما تسأل، لماذا استخدام بيئة تطوير **Visual Studio**؟. والحقيقة أن قوة بيئة التطوير لا تكمن فى إنشاء خدمات الويب البسيطة التى يمكنك إنشاؤها مباشرة من خلال برنامج **Notepad**، وإنما تظهر قوتها عند بناء خدمات الويب الكبيرة المعقدة حيث تقوم بيئة التطوير تلقائياً بإنشاء الكود نيابةً عنك ومساعدتك على كتابة التصنيفات والدوال الموجودة بالمسميات التى تقوم باستخدامها.

استخدام خدمة الويب داخل صفحة ويب

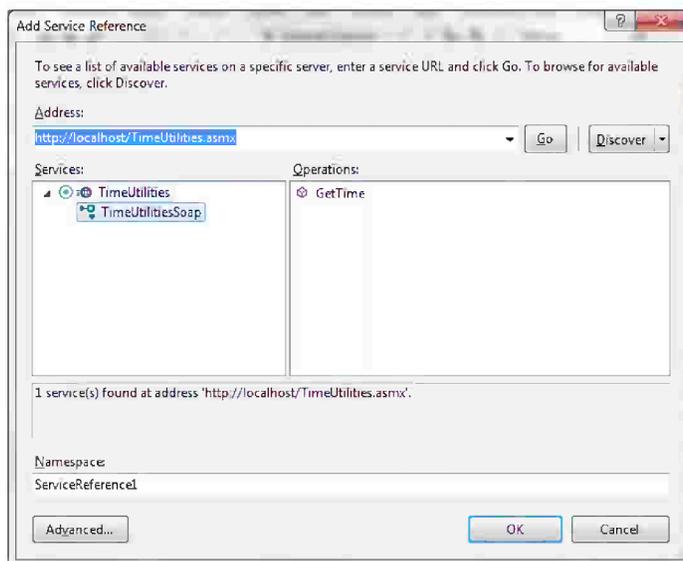
لاستخدام خدمة الويب التى أنشأناها منذ قليل داخل صفحة ويب، تابع معنا الخطوات الآتية:

١. قم بإنشاء تطبيق ويب جديد باسم مناسب وليكن **GetTime** مع إضافة النموذج **.Default**.

٢. افتح قائمة **Website** من شريط القوائم ثم اختر **Add Service Reference** من القائمة المنسدلة، يظهر المربع الحوارى **Add Service Reference** (انظر شكل ٧-٩).
٣. قم بإدخال عنوان خدمة الويب داخل مربع النص **Address** كما يلي:
http://localhost/TimeUtilities.asmx
٤. انقر زر الإدخال، يتم تحميل خدمة الويب كمرجع من على الخادم (انظر شكل ٧-١٠).
٥. انقر زر **Ok** لإضافة خدمة الويب إلى التطبيق وإغلاق المربع الحوارى.



شكل ٧-٩ المربع الحوارى **Add Service Reference**



شكل ٧-١٠ تحميل خدمة الويب كمرجع من الخادم

٦. انقر النموذج نقرًا مزدوجاً ثم قم بإدخال الكود التالي داخل إجراء الحدث

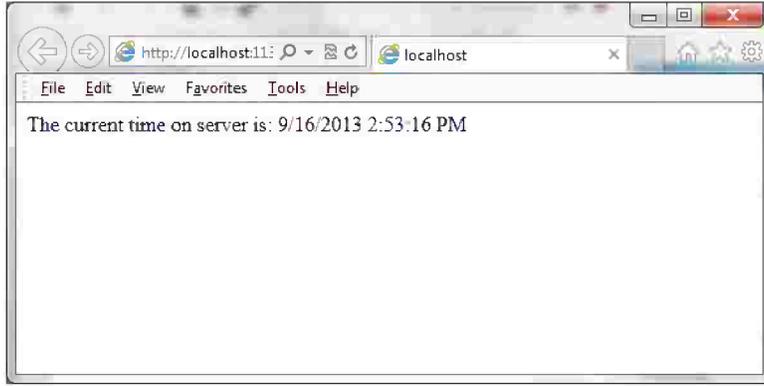
:Page_Load

**Protected Sub Page_Load(sender As Object, e As EventArgs)
Handles MyBase.Load**

```
Dim remoteService As Newlocalhost.TimeUtilities  
Response.Write(remoteService.GetTime())  
End Sub
```

٧. قم بتشغيل التطبيق، تحصل على نتيجة تنفيذ الوظيفة (`GetTime()`) حيث يظهر

التاريخ والوقت الحاليان بالخادم داخل مستعرض الويب (انظر ٧-١١).



شكل ٧-١١ استخدام خدمة الويب داخل أحد التطبيقات



الواجب الثالث قواعد البيانات

- ٨ . أساسيات قواعد البيانات .
- ٩ . استخدام SQL .
- ١٠ . استخدام كائنات ADO .
- ١١ . استخدام كائنات ADO.NET .
- ١٢ . استخدام مجموعات البيانات DataSets .
- ١٣ . العمل مع الطباعة .