الفصل السابع
مفهوم خدمات الويب

تعتبر خدمات الويب **Web Services** من السمات الأساسية داخل نطاق **.NET**. والتى حاولت مايكروسوفت من خلالها التمهيد لنقل التركيز من برامج الحاسب المثبتة على كل حاسب شخصى 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 Studio2012**.

obeikandl.com

مفهوم خدمات الويب

خدمة الويب **Web Service** عبارة عن مجموعة من الدوال التى يتم الوصول إليها من خلال عنوان معين باستخدام بعض المقاييس المعرفة من قِبَل مايكروسوفت. كما أنها إحدى التقنيات الأساسية المعرفة داخل نطاق **.NET**. والتى تساعد على التطوير السريع للتطبيقات باستخدام العديد من الأدوات التى تم تصميمها خصيصاً لتطوير هذه الخدمات وكذلك المعالجات التى 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 الذى يقوم بنقل هذه البيانات من الخادم إلى العميل الذى يقوم بدوره باستدعاء هذه الخدمة أو استخدامها. وباستخدام كلٍ من **HTTP** و**XML** يستطيع أى نظام تشغيل متصل بالإنترنت استخدام خدمات الويب التى يوفرها **.NET**..

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

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

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

الاستخدام من خلال الإنترنت ومن ثمَّ 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**. وعامةً يفضل استخدام **ASP.NET** فى إنشاء خدمات الويب والوصول إليها للأسباب الآتية:

- تدعم **ASP.NET** استخدام البروتوكول **SAOP** ومن ثمَّ 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 يلى:

- لا يوجد جديد حقيقةً فى الكود السابق غير السطر رقم ١ الذى يحتوى على العنصر <%@ WebService...%> الذى يقوم بدوره بإخبار ASP.NET أن الصفحة الحالية عبارة عن خدمة ويب من خلال كلمة WebService. كما يحتوى هذا العنصر أيضاً على معلومات إضافية عن هذه الخدمة مثل لغة البرمجة المستخدمة فى إنشاء الخدمة (VB)وكذلك اسم تصنيف الخدمة الذى تختاره بنفسك.
- فى السطر رقم ٤ يتم إنشاء التصنيف TimeUtilities المشتق من التصنيف System.Web.Services.WebService الموجود بدوره داخل المسمى System.Web.Services الذى يحتوى على التصنيفات اللازمة لإنشاء وبناء خدمات الويب وخاصةً التصنيف WebService الذى يعد التصنيف الأساسى لجميع خدمات الويب المنشأة باستخدام ASP.NET (السطر رقم ٥). لاحظ أننا قمنا بتعريف الصفة الآتية قبل تعريف التصنيف:

  Namespace:="http://tempuri.org/MyServices"

  والذى يحتوى على المسمى الذى تريد وضع تصنيف خدمة الويب بداخله.
- تحتوى السطور من ٦ إلى ١٠ على كود الدالة ()GetTime التى تقوم بإرجاع الوقت الموجود على الخادم. وتم استخدام الصفة <WebMethod> مع هذه الدالة دلالةً على إمكانية الوصول إلى الدالة ليس فقط من داخل الحاسب الحالى وإنما من خلال أى حاسب آخر متصل بالإنترنت.

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

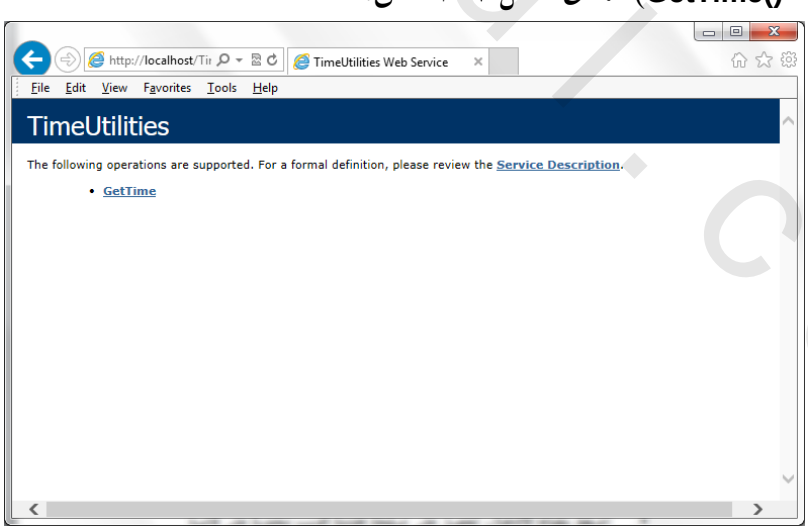
اختبار خدمة الويب

بعد الانتهاء من إنشاء ملف خدمة الويب، يمكنك حينئذٍ عرضها داخل نافذة مستعرض الويب بكتابة العنوان كاملاً، حيث يقوم المستعرض بالبحث عن الملف الذى يتم تفسيره من قِبَل 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 التالى 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 وهى الدالة GetTime()) كما فى شكل ٧-١ التالى.

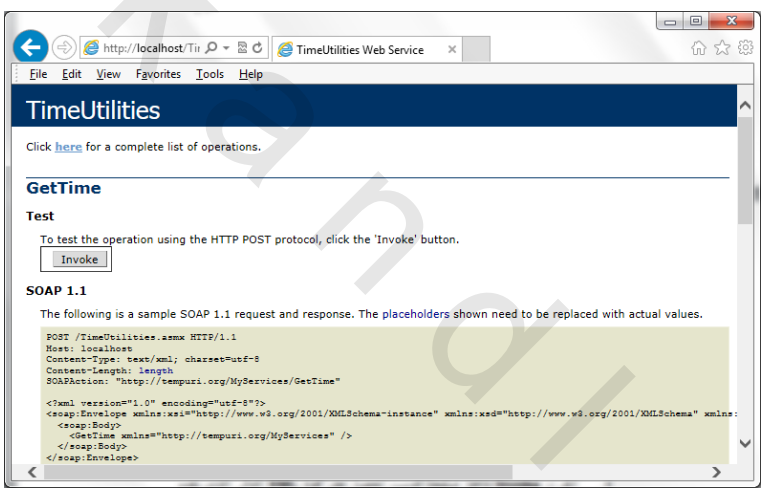


شكل ٧-١ عرض خدمة الويب داخل نافذة المستعرض

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>
```

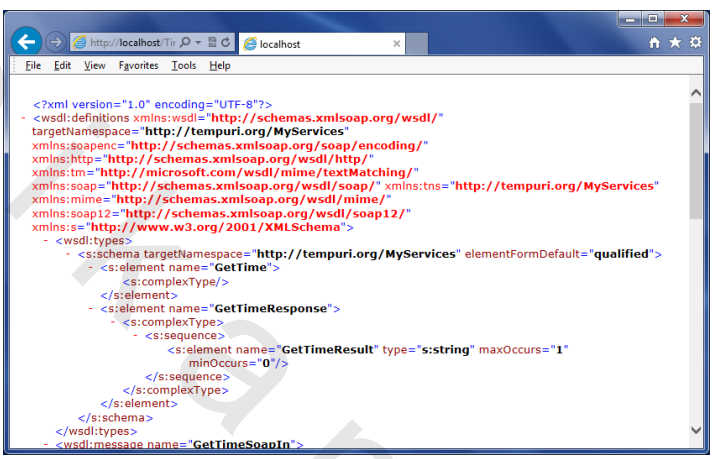


شكل ٧-٢ تقوم **ASP.NET** بإنشاء قالب توثيقى للدالة **GetTime()**

وصف خدمة الويب

ذكرنا منذ قليل استخدام ملف التوصيف **WSDL** المصاحب لخدمة الويب والذى يحتوى على التفصيلات 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

XMLوالذى يحتوى على ما يقرب من ١٢٠ سطر على الرغم من البساطة الشديدة لخدمة الويب المصاحبة والتى تقوم فقط بإرجاع الوقت الحالى من خلال دالة واحدة (انظر شكل ٧-٣).



شكل ٧-٣ محتويات ملف التوصيف المصاحب لخدمة الويب

وفيما يلى نقوم إتماماً للفائدة بالتعرف على السمات العامة لمحتويات ملف التوصيف حيث يحتوى 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 Types وMessage وPortType وBinding وService حيث تقوم العناصر Types بوصف أسماء الوظائف والدوال الموجودة بخدمة الويب وكذلك المعاملات وأنواع البيانات التى 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 فى هذه الحالة) كما يقوم بتعريف مجموعة من المنافذ Ports حيث يعبر كل منفذ من هذه المنافذ عن أحد البروتوكولات التى تدعمها خدمة الويب. فخدمة الويب التى بين أيدينا على سبيل المثال تدعم البروتوكولات SOAP وHTTP Post وHTTP Get كما يلى:

```
<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 أى نوع من أنواع البيانات.

العنصر PortType

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

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

العنصر Message

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

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

وكما ترى فإن التعريف بسيط للغاية لأن الوظيفة **()GetTime** لا تحتوى 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 فى 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 Studio2012**. وعامةً يتكون عميل الويب من جزأين، الأول هو تفويض الويب **Web Proxy** والآخر عبارة عن برنامج العميل **Client Program**.

إنشاء تفويض الويب Web Proxy

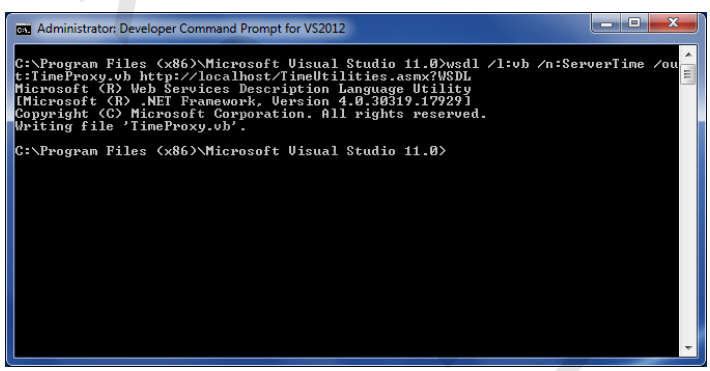
تفويض الويب عبارة عن تصنيف قادر على استدعاء الدوال الموجودة داخل خدمة الويب دون الحاجة إلى الاهتمام بكيفية الاتصال أو تمرير المعاملات من وإلى الوظائف الموجودة بالخدمة، فهو يقوم بإرسال البيانات والمعاملات من العميل إلى خدمة الويب، لذا يتم تثبيته على جهاز العميل.

يمكنك إنشاء تفويض الويب باستخدام محث الأوامر أو من خلال بيئة تطوير **Visual Studio2012** كما سنرى فيما بعد. لإنشاء تفويض ويب من محث الأوامر، افتح قائمة **Start** بشريط المهام ثم اختر **All Programs >> Microsoft Visual Studio 2012 >> Visual Studio Tools>>Developer Command Prompt for VS2012** ثم قم بتنفيذ الأمر **Wsdl**التالى من موضع محث النافذة **Developer 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 **/l:vb** بإخبار البرنامج أنه يستخدم لغة **Visual Basic** فى إنشاء التفويض.
- يقوم الجزء **/n:ServerTime** بتعيين مسمى **Namespace** لأدوات **WSDL** يتم استخدامه عند إنشاء تصنيف التفويض. يمكنك استخدام الاسم الذى 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 فى هذا المثال.



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.

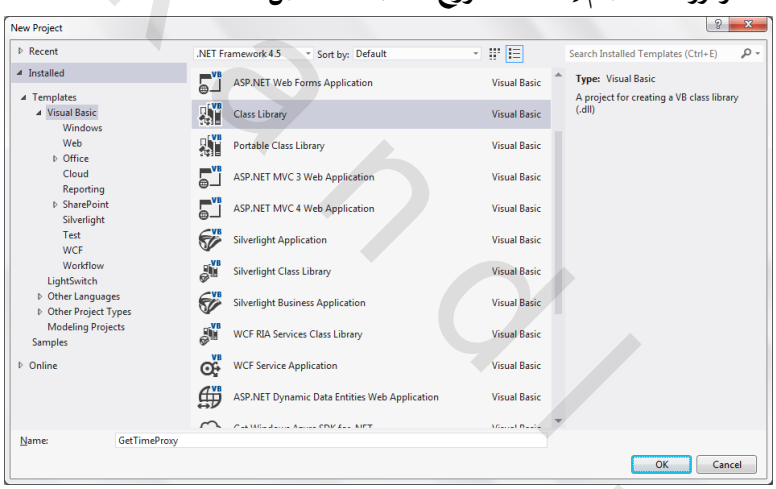
الخطوة التالية هى ترجمة تفويض الويب الذى قمنا بإنشائه. يمكنك ترجمة تفويض الويب من داخل بيئة تطوير **Visual Studio2012**. لأداء ذلك، تابع معنا الخطوات الآتية:

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 القائمةالمنسدلة 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 الموضعية، 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 ثمَّ 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
namespaceserverTime
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
```

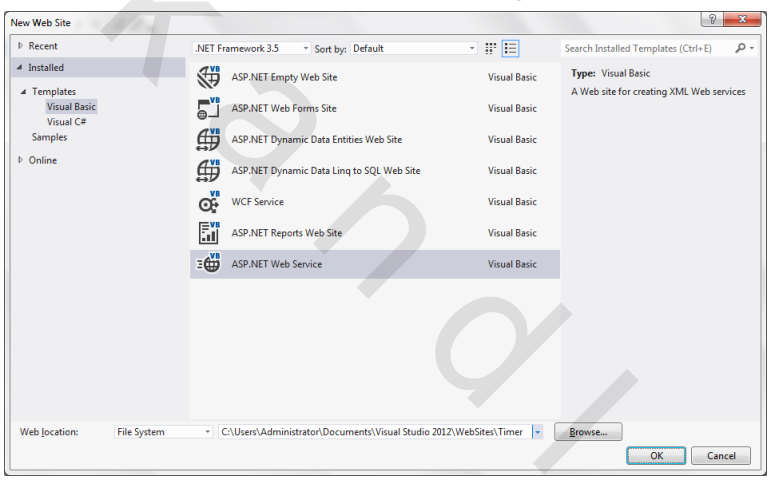
قم بتشغيل التطبيق **CallService** تحصل على النتيجة التالية:

```
Calling web service…
The Current time on the server is: 15/09/201310:55:14 PM
```

إنشاء خدمة الويب باستخدام بيئة تطوير Visual Studio2012

يمكنك استخدام بيئة التطوير Visual Studio2012 لإنشاء خدمات ويب قوية ومعقدة بسهولة تامة. سنقوم فيما يلى بإنشاء نفس خدمة الويب السابقة من خلال بيئة التطوير. لأداء ذلك، تابع معنا الخطوات الآتية:

١. تأكد أنك داخل بيئة تطوير Visual Studio2012 ثم افتح قائمة File من شريط القوائم واختر NewWeb Site من القائمة المنسدلة، يظهر المربع الحوارى New Web Site (انظر شكل ٧-٦).



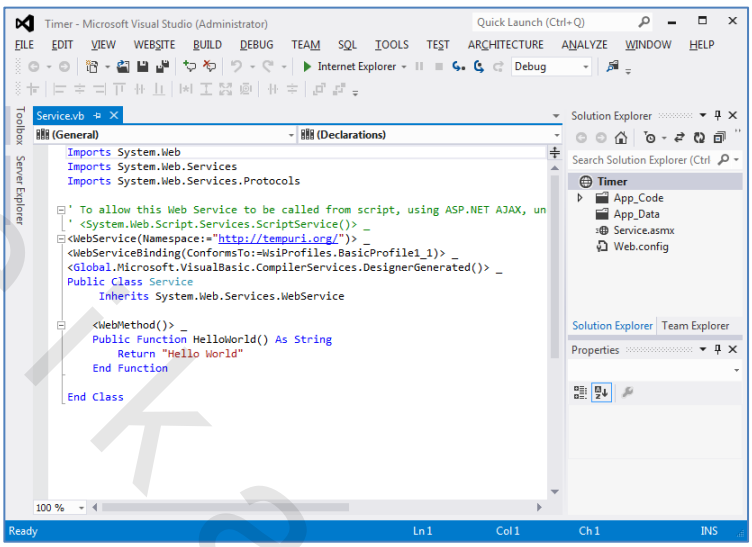
شكل ٧-٦ المربع الحوارى NewWeb Site

٢. اختر .NET Framework 3.5 بدلاً من .NET Framework 4.5 بمربع السرد الموجود بالجزء العلوى من المربع الحوارى.

٣. اختر الرمز ASP.NET Web Service من العمود الأوسط بالمربع الحوارى.

٤. قم بتعيين اسم خدمة الويب ومكانها داخل مربع النص المجاور للقائمة Web Locationوليكن Timer.

٥. انقر زر Ok، يتم بعد قليل إنشاء خدمة الويب (انظر شكل ٧-٧).



شكل ٧–٧يتم إنشاء خدمة الويب تلقائياً

بالنظر إلى نافذة المستكشف فى الشكل السابق، تلاحظ وجود ملفينهامين وهما:

- الملف **Service.asmx** وهو عبارة عن صفحة **ASP.NET** يتم عرضها داخل نافذة مستعرض الويب ويشبه إلى حد كبير الملف **aspx.** المستخدم فى صفحات **ASP.NET** العادية.
- الملف **Service.vb** ويحتوى على كود **Visual Basic** الذى يقوم بتنفيذ مهام خدمة الويب مثل إرجاع الوقت والتاريخ الحاليين كما فى 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 هى إضافة الكود الذى يعبر عن مهام خدمة الويب إلى الملف **Service.vb**. لأداء ذلك، تابع معنا الخطوات الآتية:

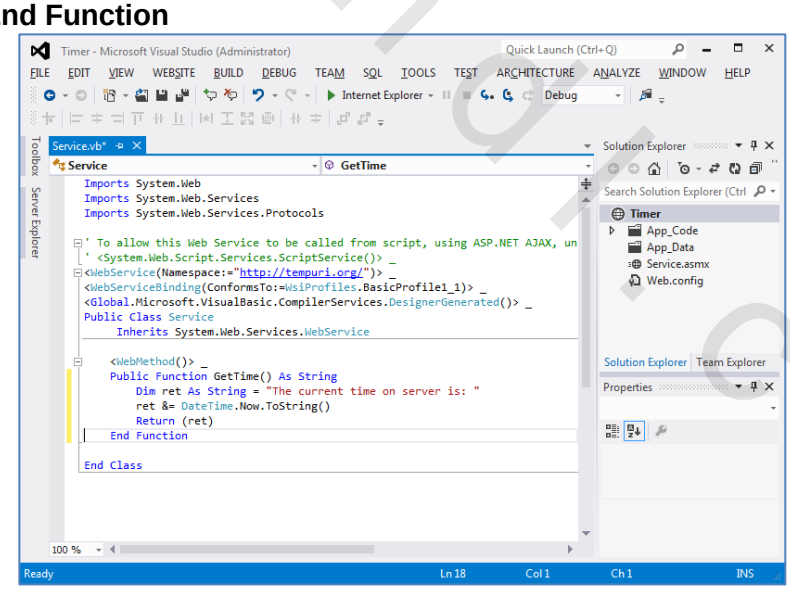
١. من نافذة المستكشف، انقر الملف **Service.vb** نقراً مزدوجاً إذا لم يكن مفتوحاً بالفعل، يتم فتح الملف الذى يحتوى على الكود المنشأ من قِبَل بيئة تطوير **Visual Studio2012**.

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 التالى:

```
<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 Studio2012** 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 مباشرةً 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:

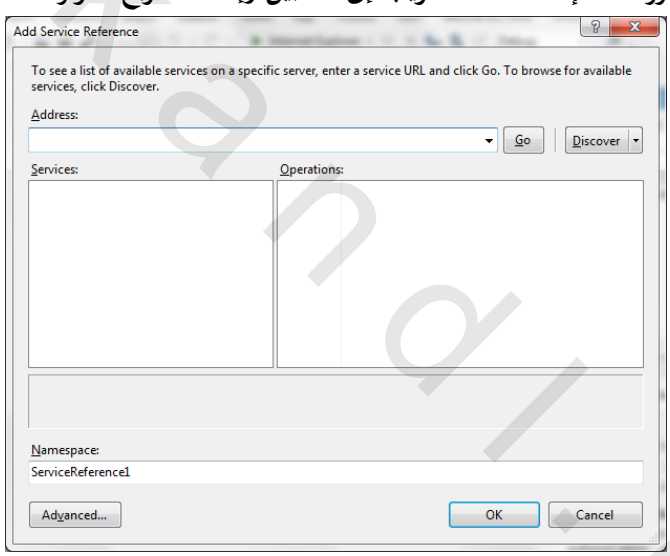
١. قم بإنشاء تطبيق ويب جديد باسم مناسب وليكن **GetTime** مع إضافة النموذج **Default**.

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).

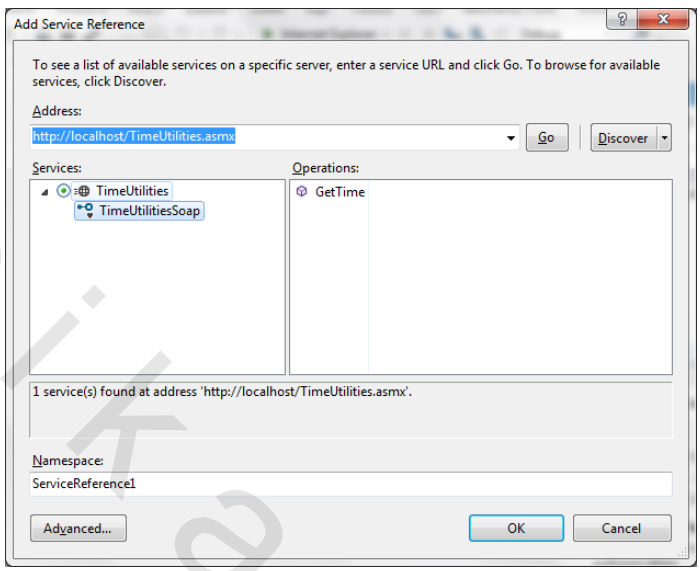
٣. قم بإدخال عنوان خدمة الويب داخل مربع النص **Address** كما يلى:
http://localhost/TimeUtilities.asmx

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**

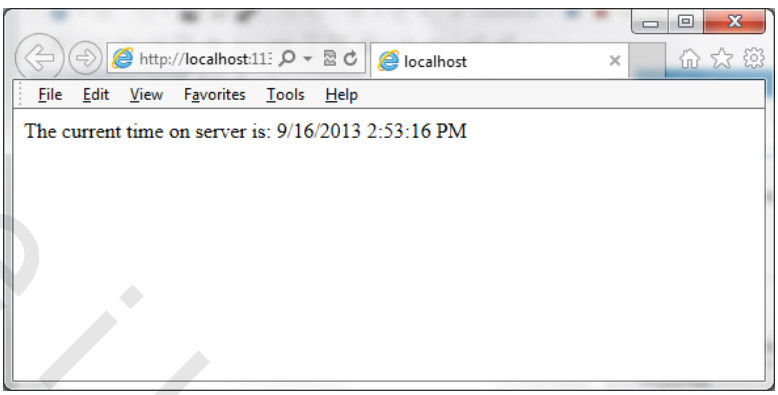


شكل ٧-١٠ تحميل خدمة الويب كمرجع من الخادم

٦. انقر النموذج نقراً مزدوجاً ثم قم بإدخال الكود التالى 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

obeikandl.com

الباب 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.

obeikandl.com