**خدمات الإنترنت XML Web Services:**

* [**ما هي:**](file:///C:\Users\Administrator\Desktop\vb.net\f-%20%20برمجة%20الإنترت\21-%20خدمات%20الإنترنت\3-%20خدمات%20الإنترنت\1-%20ما%20هي.htm)
* [**بناء خدمة انترنت:**](file:///C:\Users\Administrator\Desktop\vb.net\f-%20%20برمجة%20الإنترت\21-%20خدمات%20الإنترنت\3-%20خدمات%20الإنترنت\2-%20بناء%20خدمة%20انترنت.htm)
* [**استخدام خدمات الإنترنت Consuming the Web Service:**](file:///C:\Users\Administrator\Desktop\vb.net\f-%20%20برمجة%20الإنترت\21-%20خدمات%20الإنترنت\3-%20خدمات%20الإنترنت\3-%20استخدام%20خدمات%20الإنترنت.htm)
* [**بناء الخدمة بواسطة VS.Net:**](file:///C:\Users\Administrator\Desktop\vb.net\f-%20%20برمجة%20الإنترت\21-%20خدمات%20الإنترنت\3-%20خدمات%20الإنترنت\4-%20بناء%20الخدمة%20بواسطة.htm)
* [**إبقاء الحالة Maintaining State:**](file:///C:\Users\Administrator\Desktop\vb.net\f-%20%20برمجة%20الإنترت\21-%20خدمات%20الإنترنت\3-%20خدمات%20الإنترنت\5-%20إبقاء%20الحالة.htm)
* [**التعامل مع البيانات عبر خدمات الإنترنت:**](file:///C:\Users\Administrator\Desktop\vb.net\f-%20%20برمجة%20الإنترت\21-%20خدمات%20الإنترنت\3-%20خدمات%20الإنترنت\6-%20التعامل%20مع%20البيانات%20عبر%20خدمات%20الإنترنت.htm)

**ما هي:**

خدمة الإنترنت هي خليّة Class تعمل على الخادم ـ مثلها مثل تطبيق الإنترنت.. ولكنّها على خلافه لا تعرض أيّ صفحات.. بل تقدّم دوالَّ، تستطيع تطبيقات الإنترنت الأخرى استدعاءها.

ولكي تتصل بخدمة إنترنت، يجب عليك أن تنشئ متغيرا (كائنا) مرجعيا يشير لهذه الخدمة (تماما كما تفعل مع أيّ خليّة عاديّة).

قد لا تتخيّل فائدة مثل هذه الخدمات.. ولكنّها قد تقدّم لك حلولا تعجز أنت عن إنجازها.. مثل الحصول على بعض الإحصائيات من مواقع البورصة.. أو تشكيل نصّ عربيّ بالكامل عن طريق فصيح (لو قامت شركة RDI بمدّ خدمة منه) أو ترجمة نصّ إنجليزي إلى العربية عن طريق عجيب (لو قامت صخر بمدّ خدمة منه)... إلخ.

ولا تقلق بخصوص اختلاف نظم التشغيل.. فإرسال البيانات بتنسيق XML أو SQAP القياسيين، لن يجعل هناك أيّ مشكلة مع حوائط النار firewalls أو نظم التشغيل.

**بناء خدمة انترنت:**

لإظهار مدى سهولة الأمر، سنستخدم NotePad لإنشاء أوّل خدمة إنترنت بالنسبة لنا!

انظر لهذه الخليّة البسيطة:

**Public Class MasteringServices**

**Public Function Add(a As Decimal, b As Decimal) As Decimal**

**Return(a + b)**

**End Function**

**Public Function Multiply(a As Decimal, b As Decimal) As Decimal**

**Return(a \* b)**

**End Function**

**End Class**

لتحويل هذه الخليّة إلى خدمة إنترنت، أضف هذين السطرين لبداية الملفّ:

**<%@ webservice class="MasteringServices" language="vb" %>**

**Imports System.Web.Services**

أعتقد أنّهما واضحان وبسيطان.

بعد ذلك، أضف هذه البادئة أمام كلّ دالة في الخليّة، لإخبار المترجم Compiler أنّ الدالة يمكن استدعاؤها عبر خدمة الإنترنت (وليست داخليّة فقط):

<WebMethod()>

ويمكن أن يحتوي هذا التنسيق على بعض السمات، مثل سمة الوصف Description، كالتالي:

<WebMethod("Description", "هذه الوسيلة تجمع رقمين")>

والآن هذا هو كود خدمة الإنترنت كاملا.. ضعه في ملفٍّ نصّيّ، وسمّه Mastering.asmx.. حيث ASMX هو امتداد ملفات خدمات الإنترنت.. وضع الملفّ في المجلّد Inetpub\wwwroot على المحرّك Drive الذي يحتوي نظام التشغيل:

**<%@ webservice class="MasteringServices" language="vb" %>**

**Imports System.Web.Services**

**Public Class MasteringServices**

**<WebMethod()> Public Function Add(a As Decimal, \_**

**b As Decimal) As Decimal**

**Return(a + b)**

**End Function**

**<WebMethod()> Public Function Multiply(a As Decimal, \_**

**b As Decimal) As Decimal**

**Return(a \* b)**

**End Function**

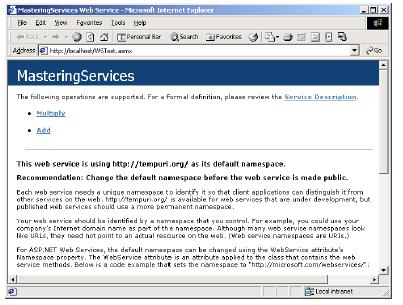
**End Class**

ولاختبار هذه الخدمة، افتح متصفّح الإنترنت، وضع العنوان التالي في شريط العناوين، واضغط زرّ الإدخال:

<http://localhost/Mastering.asmx>

(يمكنك أيضا أن تضغط الوصلة السابقة من صفحتنا هذه (طبعا يشترط إعداد IIS ووجود إطار العمل على جهازك أولا، قبل أن تجرّب هذا.. راجع [مقدمة الكتاب](file:///C:\\Users\\Administrator\\Desktop\\vb.net\\مقدمة\\5-%20إعداد%20اللغة.htm)).

لحظات وستظهر لك هذه الصفحة:



وهي تخبرك باسمي الدالتين المتاحتين عبر هذه الخدمة (Add، Multiply).. واضح طبعا أنّ هذه الصفحة للاختبار فقط، فنحن لم ننشئها سابقا.

ستجد كذلك في الصفحة تنبيها حول استخدام فضاء الاسم الافتراضيّ default namespace، وهو:

<http://tempuri.org/>

هذا موقع خاص بميكروسوفت.. وعليك إذا أردت تشغيل الخدمة من على موقعك، أن تضيف الخاصية التالية للسمة WebService:

namespace="http://yourserver.com/yourservices"

طبعا مع وضع عنوان موقعك واسم الخدمة بدلا من:

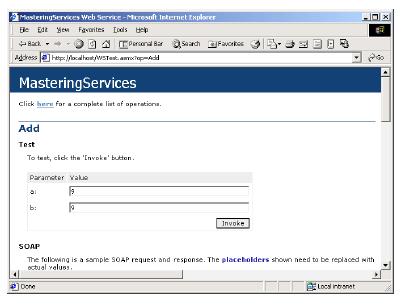
yourserver.com/yourservices

مثال:

**<%@ webservice namespace="http://your.server.com/yourservices" \_**

**class="MasteringServices" language="vb" %>**

والآن اضغط اسم أيّ دالة بالفأرة ولتكن دالة الجمع.. ستظهر لك الصفحة التالية:



حيث تمّ وضع مربعي نصّ لاستقبال قيمتي معاملي الدالة، وزر للتنفيذ.. اكتب الرقمين 2، 16، واضغط الزرّ.. ستظهر لك صفحة ثالثة، تحتوي على ما يلي:

<?xml version="1.0" encoding="utf-8" ?>

<decimal xmlns="http://tempuri.org/">18</decimal>

تلاحظ أنّ الناتج في صيغة XML.. هكذا يمكن عبور نظم التشغيل المختلفة.

هاه: منتهى البساطة.. صح؟

بإمكانك أن تطوّر هذه الخليّة كما تريد، وأن تستخدم بها أيّ مكون من مكونات إطار العمل، وأي ّدالة من دوال VB.. الفضاء رحب أمامك، فافرد جناحيك.

**استخدام خدمات الإنترنت Consuming the Web Service:**

حان الآن وقت بناء تطبيق يستخدم خدمة الإنترنت التي أنشأناها..

ابدأ تطبيق ويندوز جديدا Windows Application، وسمّه WSTest.. ضع زرّا على النموذج.. ثم أضف مرجعا للخدمة المطلوبة، وذلك كالتالي:

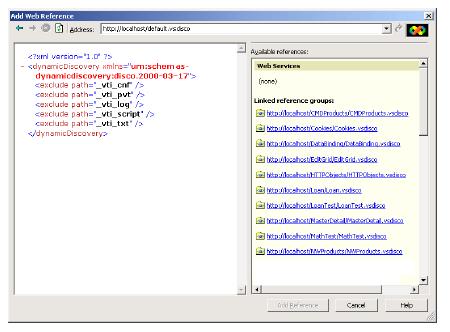
في متصفح المشاريع Solution Explorer، حدّد عنصر المراجع References، ومن القائمة الموضعيّة اضغط الأمر Add Web Reference..

في مربّع الحوار الذي سيظهر لك، ستجد وصلتين إلى المجلدين UDDI و UDDI test.. لا تهتمّ بهما الآن!

اكتب في شريط العناوين عنوان الخدمة:

<http://localhost/mastering.asmx>

واضغط زرّ الإدخال.. لا تقلق لو طال انتظارك.. سترى وصف الخدمة في نفس مربّع الحوار كما في الصورة التالية:



عندئذ يمكنك أن تضغط زرّ Add Reference في الجزء السفليّ من مربّع الحوار.

الآن يمكنك أن تضيف هذا الكود لحدث ضغط الزرّ:

**Dim WS As New localhost.MasteringServices()**

**MsgBox(WS.Add(43.32, 22.1))**

جرّب البرنامج، ولا تقلق من بطء التنفيذ، فخليّة الخدمة ستتمّ ترجمتها في أوّل مرّة للتنفيذ.. ولو ضغطت الزرّ بعد ذلك، فلن يكون هناك أيّ تأخير.

**ملحوظة:1**

نفس هذه الطريقة يمكن اتّباعها لاستخدام خدمات الإنترنت من تطبيقات الإنترنت.

**ملحوظة2:**

يمكنك استخدام الخدمة من متصفّح الإنترنت Internet Explorer، وذلك بكتابة عنوان الخدمة متبوعا باسم الدالة التي تريد تنفيذها ومعاملاتها، كالتالي:

[**http://localhost/mastering.asmx/Add?a=5&b=7**](http://localhost/mastering.asmx/Add?a=5&b=7)

حيث ستظهر لك صفحة جديدة تعرض النتيجة (12) في صورة كود XML التالي:

**<?xml version="1.0" encoding="utf-8" ?>**

**<decimal xmlns="http://tempuri.org/">12</decimal>**

**بناء الخدمة بواسطة VS.Net:**

ابدأ مشروعا جديدا، واختر النوع ASP.NET Web Service.

سمّ المشروع WSLoan.

ستجد أنّ المشروع يحتوي تلقائيّا على ملف بالامتداد ASMX يحمل الاسم Service1، وملفّ بالامتداد vsdisco، وهو يحتوي على المعلومات اللازمة للتطبيقات الأخرى، لكي تتعرّف على أسماء وصيغ الدوال التي تقدّمها الخدمة.

ولو نقرت ملفّ الخدمة مرّتين بالفأرة، فسيظهر لك سطح تصميم في مصمّم النماذج، ليسمح لك بإضافة أيّ مكوّنات أو أدوات إليه.

انتقل إلى محرّر الكود، وانقل الكود التالي داخل مقطع الخليّة Service1:

**<WebMethod()> Public Function Loan(Amount As Decimal, \_**

**Duration As Integer, Rate As Decimal, \_**

**PayEarly As Boolean) As Decimal**

**If Rate < 0 Then**

**Throw New Exception("Interest rate can't be negative!")**

**End If**

**If Rate < 1 Then**

**Rate = Rate / 12**

**Else**

**Rate = Rate / (12 \* 100)**

**End If**

**Return Pmt(Rate, Duration, Amount, 0, PayEarly)**

**End Function**

لاحظت طبعا أنّه ذلك الكود الذي يحسب معامل الفائدة (الربا) على القرض ([راجع الفصل الثاني](file:///C:\\Users\\Administrator\\Desktop\\vb.net\\f-%20%20برمجة%20الإنترت\\01-%20أساسيّات%20اللغة\\02-%20المشاريع\\1-%20برنامج.محاسب%20القروض.htm))..

اضغط F5 لاختبار التطبيق.

اختبر الدالة، ثمّ أنهِ التشغيل.

ابدأ مشروع ويندوز جديد وأسمه TestLoan.. صمّم النموذج كما في الشكل:



ضع مرجعا للخدمة التي أنشأناها، واكتب الكود التالي في حدث ضغط الزرّ:

**Dim ws As New localhost.Service1()**

**Dim amount, duration, rate As Decimal**

**Dim payEarly As Boolean**

**Try**

**amount = CDec(txtAmount.Text)**

**duration = CInt(txtDuration.Text)**

**rate = CSng(txtRate.Text)**

**payEarly = False**

**payEarly = chkPayEarly.Checked**

**Catch DataException As SystemException**

**MsgBox(DataException.ToString)**

**Exit Sub**

**End Try**

**Try**

**txtPayment.Text = -Math.Round(ws.Loan(amount, duration, \_**

**rate, payEarly), 2)**

**Catch calcException As Exception**

**MsgBox(calcException.Message)**

**End Try**

والآن جرّب التطبيق.. ضع قيمة سالبة لمعدّل الفائدة Interest rate.. ستجد أنّ الاستثناء الذي انطلق في خدمة الإنترنت، تمّ التقاطه في تطبيق الويندوز!

**إبقاء الحالة Maintaining State:**

مثلها كصفحات الإنترنت، يمكن لخدمات الإنترنت أن تحتفظ بحالة المتغيّرات عن طريق الكائنين [Application و Session](file:///C:\\Users\\Administrator\\Desktop\\vb.net\\f-%20%20برمجة%20الإنترت\\21-%20خدمات%20الإنترنت\\1-%20مقدّمة%20لبرمجة%20الإنترنت\\7-%20إبقاء%20الحالة.htm).. ولكي تفعّل ذلك، عليك باستخدام السمة EnableSession مع العنصر WebMethod.. وستجد في تطبيق WSLoan في مجلد برامج هذا الفصل، الدالة Get-MyID، وهي تعيد معرّف العميل الحاليّ ID:

**<WebMethod(EnableSession:=True)> \_**

**Public Function GetMyID() As Guid**

**If Session("MyID") Is Nothing Then**

**Session("MyID") = Guid.NewGuid**

**End If**

**Return Session("MyID")**

**End Function**

مع ملاحظة أنّ الوسيلة Guid.NewGuid تعيد معرّفا متفرّدا عامّا GUID (تحدّثنا عنه سابقا في [الفصل الرابع من فصول قواعد البيانات](file:///C:\\Users\\Administrator\\Desktop\\vb.net\\f-%20%20برمجة%20الإنترت\\04-%20Databases%20قواعد%20البيانات\\18-%20ADO.NET%20برمجة%20كائنات\\7-%20تحديث%20مجموعات%20البيانات\\1-%20التحدّيات.htm)).

وطبعا سيظلّ الكائن Session محتفظا بقيمته طالما ظلّ المتغيّر (الكائن) الذي تتصل عبره بالخدمة له قيمة (ليس Nothing).

**التعامل مع البيانات عبر خدمات الإنترنت:**

في هذا المثال، سنقوم بنقل مجموعة بيانات للعميل.. وستحتوي هذه الخدمة على وسيلتين: GetAuthors (وتعيد مجموعة بيانات بها جدول المؤلفين)، و GetBooks (وتعيد مجموعة بيانات بها الكتب ومؤلّف كلّ منها).. أعتقد أنّك تستطيع تكوين موصلي البيانات المطلوبين.

ها هما الدالتان:

**<WebMethod()> Public Function GetAuthors() As DataSet**

**DAAuthors.Fill(DsAuthors1)**

**Return DsAuthors1**

**End Function**

**<WebMethod()> Public Function GetBooks() As DataSet**

**DABooks.Fill(DsBooks1)**

**Return DsBooks1**

**End Function**

شغّل التطبيق واختبر هذه الخدمة.. ولاحظ كيف سيتمّ عرض بيانات مجموعة البيانات في صورة XML.

والآن تعال نبنِ تطبيق اختبار.

سنبني تطبيق الإنترنت DSTest، وهو يعرض صفحة عليها جدول بيانات وزران.. الأوّل لعرض جدول المؤلفين في جدول البيانات، والثاني لعرض الكتب ومؤلفيها.



لا تنس أن تضيف للتطبيق مرجعا للخدمة:

<http://localhost/WSBooks/Books.asmx>

والفكرة كلّها تتمثّل في وضع مجموعة البيانات التي تعيدها الخدمة في خاصية DataGrid1.DataSource.

ها هو ذا كود أحد الزرّين:

**Dim WS As New localhost.Books()**

**DataGrid1.DataSource = WS.GetAuthors**

**DataGrid1.DataBind()**

أعتقد أنّ بإمكانك إنشاء تطبيق ويندوز يستخدم هذه الخدمة.. مع ملاحظة أنّ جدول البيانات في هذه الحالة قادر على عرض العلاقات بين الجداول.. ولا تنسَ وضع قيم خصائص ربط البيانات (مثل DataMeber وغيرها) في الكود ([راجع فصول قواعد البيانات](file:///C:\Users\Administrator\Desktop\vb.net\f-%20%20برمجة%20الإنترت\04-%20Databases%20قواعد%20البيانات\17-%20تطبيقات%20قواعد%20البيانات\6-%20ربط%20البيانات.htm)).

أعتقد أنّ هذا الفصل كان بسيطا وشيّقا.. وقصيرا.

أتمنّى أن تكون قد استمتعت به.