

درس: إرسال رسائل البريد الإلكتروني في بيئة الدوت نت

المحتويات

- المحتويات
- نظرة خاطفة
- مقدمة
- نظرة على الأصناف
 - الأصناف في System.Net.Mail
 - الأصناف في System.Web.Mail
- سيرفرات SMTP
- التطبيق
- تغيير طريقة إيصال الرسالة
- مثال
- خاتمة

نظرة خاطفة

نقوم في هذا الدرس بإذن الله تعالى بشرح كيفية إرسال رسائل البريد الإلكتروني من خلال بيئة الدوت نت. ويتم ذلك عن طريق خدمة SMTP.

بداية، سوف نقوم بشرح خدمة SMTP ثم سوف نقوم بشرح الأنواع Types التي توفرها لنا بيئة الدوت نت للتعامل مع SMTP. بعد ذلك سوف نقوم بسررد بعض أشهر سيرفرات SMTP ثم سوف نقوم بتطبيق ما تعلمناه من خلال كود مبسط وسهل.

مع الدرس مثال عبارة عن برنامج يقوم بإرسال رسائل البريد الإلكتروني من خلال أكثر من مزود لخدمة البريد الإلكتروني E-Mail Service Provider.

مقدمة

بروتوكول نقل البريد Simple Mail Transfer Protocol أو كما يختصر إلى SMTP هو أحد الطرق المميزة والسهلة التي توفر خدمة إرسال رسائل البريد الإلكتروني عن طريق خادم (سيرفر Server) معين يسمح لهذا البروتوكول بالعمل.

منذ ظهور بيئة الدوت نت وهي تسمح للمبرمج بإرسال رسائل البريد الإلكتروني عن طريق الأصناف Classes الموجودة في الـ Namespace المسمى System.Web.Mail وهو موجود في المكتبة System.Web.dll.

مع ذلك، وبظهور بيئة الدوت نت 2.0، ظهرت Namespace جديدة توفر للمبرمج ميزات أعلى وأكثر وأكبر تمكنه من استغلال كافة إمكانيات بروتوكول SMTP لإرسال الإيميلات، وهذه الـ Namespace هي System.Net.Mail وهي موجودة في المكتبة System.dll لذلك سوف تجدها دائما في كل برامجك لأنه وكما تعلم ليس هناك استغناء عن المكتبة System.dll.

لهذا، فإنه يمكنك استخدام إما الأصناف الموجودة في System.Web.Mail أو الأصناف الموجودة في System.Net.Mail. مع ذلك الثانية أفضل وأقوى. حتى أنه عند استخدامك لـ System.Web.Mail سوف يقوم الـ Compiler بتحذيرك.

في القسم التالي نتكلم عن أشهر الأصناف Classes والأنواع Types التي ستحتاجها عند تعاملك مع بروتوكول SMTP. لاحظ تشابه الأصناف إلى حد كبير في System.Net.Mail إلى System.Web.Mail. في الكود سوف تلاحظ أيضا أن الفرق بسيط في التعامل لذلك إذا كنت تتعامل مع الدوت نت 1.0 أو حتى أي إصدار تستطيع قراءة هذا الدرس.

نظرة على الأصناف

الأصناف في System.Net.Mail

توفر لك الـ Namespace المسمى System.Net.Mail عدد، ليس بالعدد الكبير، بل عدد مناسب من الأصناف والأنواع التي توفر لك التحكم الكامل في رسالتك وفي طريقة إرسالها باستخدام بروتوكول SMTP بل وتوفر لك خيارات مفصلة جدا في رسالتك.

بالطبع لن تحتاج إلى معرفة جميع الأصناف في هذه الـ Namespace ولكن يجب عليك معرفة ما توفره لك بيئة الدوت نت عند التعامل مع بروتوكول SMTP وهذا يجعلك قادر على تطوير برنامجك في جميع الأبعاد.

التالي هو قائمة بأشهر الأنواع -ربما جميعها- الموجودة في هذه الـ Namespace:

• SmtplibClient:

واحدة من الأصناف الهامة جدا والأساسية وهي التي توفر لك الإمكانيات لإرسال الرسائل. بداية، يجب عليك توفير البيانات المناسبة لهذه الكلاس من خلال الخصائص Host، Port، و EnableSsl والتي تمكنك من خلالها تحديد إعدادات السيرفر Server والذي سوف تقوم بإرسال الرسائل من خلاله. فعلى سبيل المثال إذا كنت تريد إرسال رسائل من بريد إلكتروني Hotmail مثلا سوف تقوم بتحديد إعداداته من خلال هذه الخصائص. سوف نشرح كيفية تحديد الإعدادات لاحقا. بجانب هذا، توفر لك هذه الكلاس الدالة Send() لإرسال الرسائل والدالة SendAsync() لإرسال الرسائل بشكل غير متزامن Asynchronously أي باستخدام الـ Threading. سوف نشرح هذا لاحقا.

• MailMessage:

الرسالة التي سوف يتم إرسالها. تحتوي هذه الكلاس على الحقول الخاصة بالرسالة مثل To، CC، Bcc، Subject (عنوان الرسالة)، و Body (نص الرسالة).

- **MailAddress:**

عبارة عن عنوان بريد إلكتروني. تستخدم هذه الكلاس في حقول الرسالة مثل حقل To لتحديد البريد الإلكتروني الخاص بمستلم الرسالة. تحتوي هذه الكلاس على حقلين الأول هو اسم صاحب البريد والثاني هو البريد نفسه.

- **MailAddressCollection:**

عبارة عن مجموعة Collection من عناوين البريد الإلكتروني أي عبارة عن مجموعة من عناصر MailAddress. تستخدم هذه الكلاس في حقول البريد الإلكتروني للرسالة مثل To ، CC ، Bcc وهذا إذا كنت تريد إرسال الرسالة لأكثر من شخص.

- **Attachment:**

عبارة عن ملف مرفق مع الرسالة.

- **AttachmentCollection:**

عبارة عن مجموعة Collection من الملفات المرفقة أي من الكلاس Attachment. تستخدم هذه الكلاس في الحقل Attachments الخاص بالرسالة.

- **SmtpException:**

عند حدوث خطأ أثناء إرسال الرسالة يقوم SmtpClient بإصدار هذا الخطأ. يحتوي هذا الكلاس على الخاصية StatusCode والتي توفر لك طريقة معرفة ما هو الخطأ الذي حدث.

- **SmtpFailedRecipientsException و SmtpFailedRecipientException:**

الاثنتان هما أيضا عناصر خطأ Exception ينحدرا Inherit من SmtpException. إذا لم يستطع SmtpClient إرسال الرسالة إلى شخص معين سوف يقوم بإصدار الخطأ SmtpFailedRecipientException والذي يحتوي على عنوان بريد إلكتروني هذا الشخص. أما إذا فشل SmtpClient في إرسال الرسالة لعدة أشخاص سوف يقوم بإصدار الخطأ SmtpFailedRecipientsException والذي بدوره يحتوي على عناوين الأشخاص الذين لم ترسل إليهم الرسالة.

بالإضافة إلى هذه التصنيفات تحتوي هذه الـ Namespace وهي System.Net.Mail على العديد من الـ Enumerations لأغراض معينة مثل MailPriority وتمثل أهمية الرسالة إما Low ، Normal ، High وتستخدم هذه الـ Enumeration في الحقل Priority الخاص بالرسالة أي في الكلاس MailMessage.

الأصناف في System.Web.Mail

إذا كنت مهتما بالأصناف Classes الموجودة في System.Web.Mail ستجد أنها شبيهة جدا -إن لم تكن مماثلة- للأصناف في System.Net.Mail. ستجد أيضا أنها عبارة عن ثلاثة أصناف Classes وثلاثة Enumerations فقط! لا تنسى أن هذه الأصناف موجودة في المكتبة System.Web.dll ولذلك يتوجب عليك إضافتها إلى مشروعك قبل البدء في التعامل مع هذه الأصناف.

التالي هو قائمة بالأصناف في System.Web.Mail:

- **SmtpMail:** هذا العنصر شبيه جدا -بل مماثل- للعنصر System.Net.Mail.SmtpClient ويعمل نفس العمل وله نفس الحقول.

- **MailMessage:** هو عنصر مماثل للعنصر System.Net.Mail.MailMessage ويمثل الرسالة التي سوف ترسل باستخدام SmtpMail.

- **MailAttachment:** عبارة عن ملف مرفق مع الرسالة. مماثل للعنصر System.Net.Mail.Attachment.

أما الـ Enumerations الموجودة في هذه الـ Namespace فهي MailEncoding ، MailFormat و MailPriority. تمثل الأولى نظام الـ Encoding الخاص بالرسالة وهو في الغالب لن تحتاج إلى تغييره. وتمثل الثانية MailFormat نوع الرسالة هل هي رسالة HTML أم رسالة نصية عادية. سوف نقوم بشرح كيفية إرسال الرسائل HTML لاحقا. والثالثة MailPriority تمثل أهمية الرسالة Low ، Normal ، High.

نتكلم اليوم عن الطريقة الأفضل والأقوى وهي باستخدام العناصر من System.Net.Mail. إذا كنت تفضل System.Web.Mail فالكود سوف يكون شبيها جدا. فقط قم بتغيير أسماء العناصر وعمل اللازم.

سيرفرات SMTP

تعرف بالطبع أنه لكي يمكنك إرسال رسالة بريد إلكتروني باستخدام بروتوكول SMTP يجب عليك أن تقوم بتحديد السيرفر Server المناسب لإرسال هذه الرسالة. ويجب أيضا أن يكون هذا السيرفر يوفر لك خدمة SMTP وإلا لن تستطيع التعامل معه.

بالطبع يمكنك إنشاء السيرفر الخاص بك والذي يوفر خدمة SMTP وبالطبع أيضا يمكنك شراء سيرفر أو على الأقل الاشتراك في سيرفر يوفر لك هذه الخدمة. فتستطيع مثلا من خلال خدمة الاستضافة Hosting الخاصة بموقعك أن تقوم بإنشاء بريد إلكتروني مثلا yourname@yoursite.com وتقوم بإرسال الرسائل من خلاله. ولكن يشترط في هذه الاستضافة أن يكون السيرفر يوفر لك كما قلنا خدمة SMTP.

ولكي تستطيع التعامل مع أي سيرفر SMTP يجب عليك أن تعرف عن هذه السيرفر أربعة أشياء:

- **العنوان Address:** عنوان السيرفر مثلا smtp.example.com
- **المخرج Port:** بالطبع يوفر لك السيرفر الخدمة SMTP عن طريق مخرج معين وهو في الغالب 25 وأحيانا يكون 465 وأحيانا يختلف.
- **هل هذا السيرفر يدعم تأمين SSL ؟** معظم السيرفرات تدعم تأمين SSL وهو Secure Socket Layer. يقوم بتشفير البيانات أثناء نقلها وبالتالي يمنح الرسالة حماية ضد أن يقوم شخص Hacker بالحصول عليها وفهمها أو تغييرها قبل وصولها إلى المستخدم.
- **الاعتمادات Credentials:** أي اسم المستخدم وكلمة المرور.

بالطبع تعرف أن هذه الإعدادات الخاصة بالسيرفر سوف تقوم بتحديدتها عن طريق حقول System.Web.Mail.SmtpMail أو System.Net.Mail.SmtpClient

التالي هي قائمة بأشهر أربعة موفروا لخدمة البريد الإلكتروني E-Mail Service Providers والإعدادات الخاصة بهم:

Live العنوان smtp.live.com المخرج Port 25 يدعم SSL ؟ نعم **GMail** العنوان Address smtp.gmail.com يدعم SSL ؟ 25، 465 نعم **Yahoo!** العنوان plus.smtp.mail.yahoo.com المخرج Port 465، 25، 587 يدعم SSL ؟ نعم/لا هذه الخدمة متاحة فقط لمستخدمي Yahoo Plus! أي غير المشتركين المجانيين **GMX**. العنوان mail.gmx.com المخرج Port 25 يدعم SSL ؟ لا كما تعلم أن **Live** يمثل جميع خدمات البريد الإلكتروني لـ **Live**، **Hotmail**، **MSN**، وغيرها. وأيضا **Gmail** يمثل **Gmail** و **GoogleMail** فهي مجرد أسماء فقط.

أي سيرفر أو موفر للبريد الإلكتروني في الغالب يوفر **SMTP** فقط قم بزيارة صفحات الدعم **Support** الخاصة بهذا السيرفر أو الموفر وعمل بحث عن خدمة **SMTP**.

بعض السيرفرات مثل **Gmail** و **Yahoo!** توفر لك أكثر من مخرج **Port** فقط قم باختيار أي منها ليس هناك فرق . وبعض السيرفرات مثل **Yahoo!** لا تضع شروطا لدعم **SSL** فلماذا يمكنك اختياره أو تركه ولكن يفضل اختيار **SSL**.

بعض السيرفرات تجعل لك حدا يوميا لإرسال الرسائل باستخدام **SMTP** فعلى سبيل المثال **Gmail** يضع حدا أعلى للرسائل من على **SMTP** وهو 100 رسالة يوميا. أما من خلال الموقع أو البرنامج الخاص بـ **Gmail** الحد الأعلى له هو 400 رسالة يوميا.

التطبيق

الكود التالي يقوم بإرسال رسالة من حساب **GMX** إلى حساب آخر باستخدام العناصر من **System.Net.Mail** لاحظ الكود جيدا. لاحظ أن الرسالة هي عبارة عن رسالة **HTML** ولاحظ أن المثال يقوم بإضافة ملف مرفق إلى الرسالة وبالطبع سوف يحدث خطأ إن لم يكن هذا الملف موجود على جهازك. ولاحظ أيضا كيفية تحديد إعدادات السيرفر.

يمكنك زيارة موقع **GMX** وإنشاء حساب خاص عليه.

عناوين البريد الإلكتروني هذه هي عبارة عن عناوين وهمية وليست موجودة وهي فقط للمثال وليست للعمل. قم بتغيير العناوين لتصبح عناوين حقيقية. كما يمكنك تغيير إعدادات السيرفر لتوافق إعدادات أي سيرفر تريده إن لم يكن عندك حساب على **GMX**. وأيضا لا يشترط إرسال رسالة إلى نفس السيرفر. بمعنى أنه لا يشترط بالطبع إرسال رسالة من **Hotmail** إلى **Hotmail** آخر. ولكن يجب أن يكون بريد الرسائل على نفس السيرفر المحدد. أما المستلم فلا يشترط له هذا أبدا.

تأكد أن الملف الذي يريد الكود إضافته موجود على الجهاز. قم بتغيير مساره ليكون موافقا لملف موجود، أو حتى قم بإزالته. بالطبع يمكنك تغيير الكود كما تحب وأيضا تغيير عنوان الرسالة ونصها.

قمت بوضع بعض الملاحظات حول الكود لتوضيحه.

يجب عليك إضافة جملة *using (Imports)* في VB.NET للـ *Namespace* المسمى *System.Net.Mail*.

// C# Code

```
MailMessage msg = new MailMessage();

// Your mail address and display name.
// This what will appear on the From field.
// If you used another credentials to access the SMTP server, the mail
message would be
// sent from the mail specified in the From field on behalf of the real
sender.
msg.From = new MailAddress("example@GMX.com", "Example");

// To addresses
msg.To.Add("friend_a@example.com");
msg.To.Add(new MailAddress("friend_b@example.com", "Friend A"));

// You can specify CC and BCC addresses also

// Set to high priority
msg.Priority = MailPriority.High;

msg.Subject = "Hey, a fabulous site!";

// You can specify a plain text or HTML contents
msg.Body = "Hello everybody,<br /> <br /> I found an interesting site called
" +
    "<a href=\"http://WithDotNet.Wordpress.com\">الدوت نت</a>. Be sure to
visit it soon.";
// In order for the mail client to interpret message body correctly, we mark
the body as HTML
// because we set the body to HTML contents.
msg.IsBodyHtml = true;

// Attaching some data
msg.Attachments.Add(new Attachment("C:\\Site.lnk"));

// Connecting to the server and configuring it
SmtpClient client = new SmtpClient();
client.Host = "mail.gmx.com";
client.Port = 25;
client.EnableSsl = false;
// The server requires user's credentials not the default credentials
client.UseDefaultCredentials = false;
// Provide your credentials
client.Credentials = new System.Net.NetworkCredential("example@GMX.com",
"buzzwrdr");

// Use SendAsync to send the message asynchronously
client.Send(msg);
```

```

Dim msg As New MailMessage()

' Your mail address and display name.
' This what will appear on the From field.
' If you used another credentials to access the SMTP server, the
mail message would be
' sent from the mail specified in the From field on behalf of the
real sender.
msg.From = New MailAddress("example@GMX.com", "Example")

' To addresses
msg.To.Add("friend_a@example.com")
msg.To.Add(New MailAddress("friend_b@example.com", "Friend A"))

' You can specify CC and BCC addresses also

' Set to high priority
msg.Priority = MailPriority.High

msg.Subject = "Hey, a fabulous site!"

' You can specify a plain text or HTML contents
msg.Body = "Hello everybody,<br /> <br /> I found an interesting
site called " & _
"<a href=""http://WithDotNet.Wordpress.com"">مع الدوت نت</a>."
Be sure to visit it soon."
' In order for the mail client to interpret message body correctly,
we mark the body as HTML
' because we set the body to HTML contents.
msg.IsBodyHtml = True

' Attaching some data
msg.Attachments.Add(New Attachment("D:\Site.lnk"))

' Connecting to the server and configuring it
Dim client As New SmtpClient()
client.Host = "mail.gmx.com"
client.Port = 25
client.EnableSsl = False
' The server requires user's credentials not the default
credentials
client.UseDefaultCredentials = False
' Provide your credentials
client.Credentials = New
System.Net.NetworkCredential("example@GMX.com", "buzzwrdr")

' Use SendAsync to send the message asynchronously
client.Send(msg)

```

تغيير طريقة إيصال الرسالة

ماذا إن كنت تريد تجربة برنامجك فقط؟ ماذا إن كنت تريد مشاهدة الرسالة ولكن لا تريد إرسالها؟

يوفر لك الدوت نت طريقة لإرسال الرسالة ليس من سيرفر معين إلى مستلم! بل إنك ترسل الرسالة إلى مسار على جهازك مسار تقوم بتحديدده. وهذه الرسالة لا تشترط أن تكون متصلاً بالإنترنت أو الشبكة أو تكون محدداً لبيانات صحيحة للسيرفر. فهي فقط ترسل الرسالة إلى مسار على جهازك.

لتغيير طريقة إيصال الرسالة Delivery Method يوفر لك العنصر Smtplib.SmtpClient خاصيتان هما DeliveryMethod و PickupDirectoryLocation الخاصية DeliveryMethod تحدد طريقة وصول الرسالة وهي عبارة عن عنصر من الـ Enumeration المسمى Smtplib.SmtpDeliveryMethod وبالتالي تأخذ أحد القيم التالية:

- **Network (الأساسي):** ترسل الرسالة عن طريق الشبكة. أي أن الرسالة ترسل وتصل إلى المستلم الحقيقي.
- **PickupDirectoryFromIis:** ترسل الرسالة إلى مجلد على الجهاز تستطيع تحديده من خلال (Internet Information Services (IIS).
- **SpecifiedPickupDirectory:** ترسل الرسالة إلى مجلد على الجهاز تستطيع تحديده من الخاصية الأخرى وهي PickupDirectoryLocation.

قم بزيارة الدعم **Support** الخاص بـ **IIS** لمعرفة طريقة تغيير مجلد وصول الرسالة في حالة اختيارك **Smtplib.SmtpDeliveryMethod.PickupDirectoryFromIis**.

الكود التالي يوضح كيفية جعل الرسائل ترسل إلى مجلد معين على الجهاز. لاحظ أنه إذا لم يكن المجلد المحدد موجوداً على الجهاز سوف يحدث خطأ ولن يتم إرسال الرسالة. قم بإضافة هذا الكود قبل النداء على دالة Send() أو SendAsync().

// C# Code

```
client.DeliveryMethod =  
SmtpDeliveryMethod.SpecifiedPickupDirectory;  
client.PickupDirectoryLocation = "C:\\mails";
```

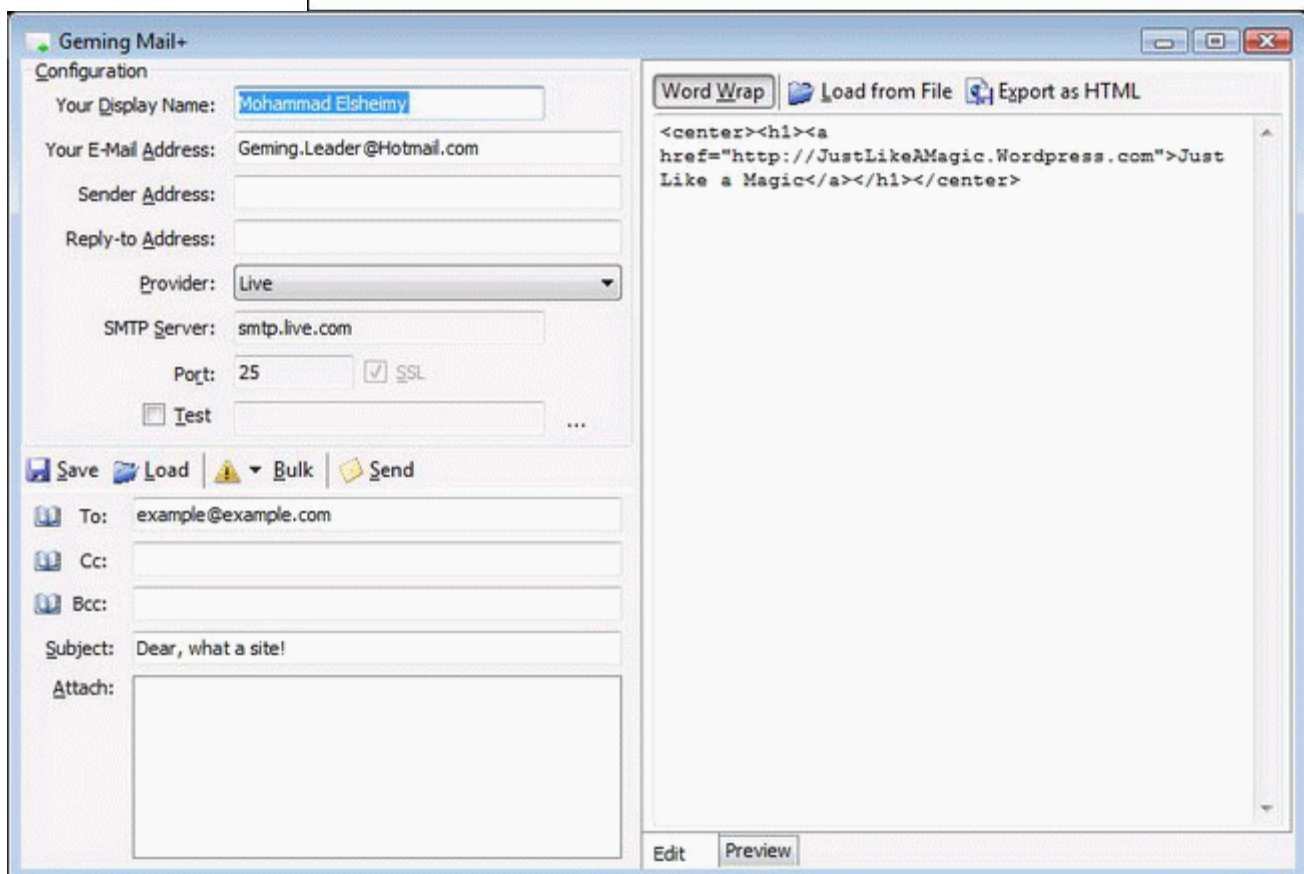
' VB.NET Code

```
client.DeliveryMethod = SmtpDeliveryMethod.SpecifiedPickupDirectory  
client.PickupDirectoryLocation = "C:\\mails"
```

مثال

برنامج Gering Mail+ عبارة عن برنامج يقوم بإرسال رسائل البريد الإلكتروني من خلال أكثر من مزود لخدمة البريد الإلكتروني E-Mail Service Provider والذي يدعم خدمة SMTP.

تم تطوير البرنامج باستخدام C# والدوت نت 2.0. ليس هناك متطلبات لتشغيل الكود فقط Visual Studio 2005 أو إصدار أعلى. التالي هو بعض صور من البرنامج.



[قم بتحميل البرنامج من هنا](#)

[قم بتحميل الكود من هنا](#)

خاتمة

تعلمنا في هذا الدرس كيفية إرسال الرسائل من بيئة الدوت نت. في دروس أخرى بإذن الله تعالى سنتعلم كيفية استقبال الرسائل وعمل إدارة لحساب البريد الإلكتروني بالكامل.