



MailDefinition - skabeloner til email-udsendelse

En smart men ofte overset klasse i ASP.NET er muligheden for at benytte skabeloner i forbindelse med udsendelse af emails. Derfor tager vi et hurtigt kig på MailMessage og specielt MailDefinition her.

Skrevet den **28. Jul 2009** af **keysersoze** | kategorien **Programmering / ASP.NET** | ★★★★★

Er der noget udviklere kan lide så er det (blandt andet) at kunne benytte hurtige genveje til at løse ofte mødte problemer, let overskuelige løsninger og mulighed for at ændre design uden at skulle pille direkte i kode - og alle disse ting er MailDefinition med til at løse når udfordringen er at sende emails fra vores ASP.NET applikation.

Det er en ret simpel operation at sende emails i .NET da vi har adgang MailMessage i System.Net.Mail - hvordan MailMessage nærmere fungerer og hvilke muligheder vi har i den vil jeg ikke komme ind på da den er forholdsvis lige til samt rimelig velbeskrevet på nettet i forvejen, men jeg vil lige give et hurtigt eksempel på hvordan jeg så godt som altid ser MailMessage blive brugt - altså uden MailDefinition - så alle er med på hvad vi taler om.

Web.Config

Først sætter vi vores SMTP konfiguration op i Web.Config så vi kan slippe for at have fx mail og standard fra-mailadresse indtastet flere steder direkte i vores applikation;

```
<system.net>
  <mailSettings>
    <smtp deliveryMethod="Network" from="noreply@domain.dk">
      <network defaultCredentials="true" host="smtpservername" />
    </smtp>
  </mailSettings>
</system.net>
```

Send mail uden MailDefinition

Herefter kan vi sende emails på denne måde;

```
string oVariabel = "skabelon";

MailMessage oMailMessage = new MailMessage();
oMailMessage.From = new MailAddress("webmaster@domain.dk");
oMailMessage.To.Add(new MailAddress("modtager@otherdomain.dk"));
oMailMessage.Subject = "Mail emne";
oMailMessage.Body = "<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
\"http://www.w3.org/TR/html4/strict.dtd\"><html><meta http-equiv=\"Content-Type\"
content=\"text/html; charset=UTF-8\" /></head><body>Mail indhold - din super hemmelige variabel
er: " + oVariabel + ".</html></body>";
oMailMessage.IsBodyHtml = true;

SmtpClient oSmtpClient = new SmtpClient();
```

```
oSmtpClient.Send(oMailMessage);
```

Ovenstående kode fejler for så vidt intet, vores email bliver sendt ud med det ønskede indhold og udseende til de ønskede modtagere. Udfordringerne i ovenstående metode er bare dels, at opsætningen af indholdet - i dette tilfælde vores HTML - kan være lidt besværlig at overskue da den bare består af en lang tekst-streng og dels, at ændringer i mailens opsætning eller indhold kræver ændringer direkte i vores kode hvilket vil kræve ny build eller publish af vores kode.

Disse ting kan vi selvfølgelig vælge at leve med - men hvorfor ikke vælge en lettere mulighed, nemlig den ofte oversete MailDefinition, som netop gør, at vi fx kan have vores mailskabeloner liggende adskilt fra vores kode, hvilket giver let mulighed for at opsætte vores HTML, og som ikke kræver ændring i vores kode ved opdatering af skabelonen? Det ville jeg i hvert fald vælge, så her kommer et eksempel.

Skabelonen

Det første vi gør er at oprette vores skabelon og det gør vi i form af fx en ganske almindelig HTML-fil - de "magiske" ting i skabelonen er de to variabler EMAIL og MESSAGE omgivet af << >> da det er disse vi igennem vores MailDefinition kan erstatte til at være den dynamiske del af indholdet;

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Kontakt</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<h1>Henvendelse fra domain.dk</h1>
<h2>Fra email</h2>
<<EMAIL>>
<br />
<h2>Besked</h2>
<<MESSAGE>>
</body>
</html>
```

Send mail med MailDefinition

Herefter benytter vi selvfølgelig stadig MailMessage, men indholdet styrer vi nu ud fra vores MailDefinition og skabelon;

```
MailDefinition oMailDefinition = new MailDefinition();
oMailDefinition.BodyFileName = "~/skabeloner/skabelon.html";
oMailDefinition.From = "webmaster@domain.dk";

Dictionary<string, string> oReplacements = new Dictionary<string, string>();
oReplacements.Add("<<EMAIL>>", "modtager@rotherdomain.dk");
oReplacements.Add("<<MESSAGE>>", "Her er en besked");

MailMessage oMailMessage = oMailDefinition.CreateMailMessage("dinmail@domain.dk",
oReplacements, new LiteralControl());
oMailMessage.Subject = "Ny mail fra domain.dk";
oMailMessage.IsBodyHtml = true;
```

```
SmtClient oSmtClient = new SmtClient();  
oSmtClient.Send(oMailMessage);
```

Det der sker i ovenstående kode er, at vi selvfølgelig først importerer skabelonen, dernæst laves en Dictionary med det formål at erstatte variablerne i skabelonen med vores reelle ønskede værdier, så laver vi et MailMessage-objekt ud fra MailDefinition- og Dictionary-objektet og til sidst sender vi mailen afsted. Dette er selvfølgelig lidt mere kode end det første eksempel - men ofte vil dette være lettere både at opbygge og vedligeholde sådan en løsning.

Ovenstående eksempler er kun simple eksempler men det er tilstrækkeligt for at komme i gang med denne ofte oversete MailDefinition. Ved siden af disse eksempler, uanset om du benytter skabeloner gennem MailDefinition eller ej, er det også vigtigt at overveje korrekt håndtering af adresser hvis man sender ud til mere end én email-adresse (så ikke alle modtagere kan se andre modtageres adresser) ligesom det, hvis man vælger at sende HTML-mails ud, bestemt også er værd at overveje at sende en plaintext-version ud sammen med HTML-versionen da mailprotokollen netop understøtter dette.

Nyttige links

<http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.maildefinition.aspx> - MailDefinition Class

<http://msdn.microsoft.com/en-us/library/system.net.mail.mailmessage.aspx> - MailMessage Class