

Building a Contact Form and Sending Email

This guide is applicable when you've converted a template/site from HTML and are ready to begin with MVC. The example site below is a multipage template and not a single page application.

For all contact forms you should begin with the ContactViewModel – (Add a class to the Models folder of the UI Project and call it ContactViewModel.cs) – You will also embed your Data Annotations here see example below:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;

namespace CWExamples.UI.Models
{
    public class ContactViewModel
    {
        [Required(ErrorMessage = "* Your name is required.")]
        public string Name { get; set; }
        [Required(ErrorMessage = "* Your email is required.")]
        [EmailAddress]
        public string Email { get; set; }
        //if used
        [Required(ErrorMessage = "* A subject is required.")]
        public string Subject { get; set; }

        //optionally, you could hard code the subject line
        //by using a read only property - This can help you discern
        //from where an email was sent. This should typically be used
        //in subsites. Use the standard subject for Personal Site.
        //You should NOT have both Subject and HardCodedSubject - it was
        //done here because I cannot have 2 properties with the same name.
        //This would NEVER be required, the value cannot be set.
        //public string HardCodedSubject
        //{
        //    get { return "Email Sent from S.A.T."; }
        //}

        [UIHint("MultilineText")] //replaces textbox with a textarea
        //in the UI wherever an editorFor() is located
        [Required(ErrorMessage = "* A message is required.")]
        public string Message { get; set; }
    }
}
```

Once the Model has been built, you will want to move to the controller and clean up the Contact Actions and write the HttpPost Contact Action. See example below: You will need a Using Statement for your UI.Models namespace to gain access to the ContactViewModel – (the Using Statement is not pictured).

BEFORE

```
public ActionResult Contact()
{
    ViewBag.Message = "Your contact page.";

    return View();
}
```

AFTER

```
public ActionResult Contact()
{
    return View();
}
[HttpPost]
public ActionResult Contact(ContactViewModel cvm)
{
    //If it did not pass Model validation, return the view with cvm object to
    //repopulate the form
    return View(cvm);
}
```

This is the shell of your Get and your Post. The next piece of code will go in the Post action before the note and return View(): You will need to add Using Statements for System.Net.Mail; and System.NET;

```
[HttpPost]
public ActionResult Contact(ContactViewModel cvm)
{
    if (ModelState.IsValid)
    {
        //You can make this whatever you want, it will be the body of the message sent
        string body = $"{cvm.Name} has sent you the following message: <br />" +
            $"{cvm.Message} <strong>from the email address:</strong> {cvm.Email}";
        //Message object from your website to your yahoo/outlook/gmail
        MailMessage m = new MailMessage("you@yourDomain.com", "ToYourPersonalEmail.com",
            //subject of the email, then the message body
            cvm.Subject, body);

        //allow HTML in email (that is our formatting with br and strong tags above
        m.IsBodyHtml = true;

        //you can make it high priority
        m.Priority = MailPriority.High;

        //reply to the Person who filled out the form, not your domain email
        m.ReplyToList.Add(cvm.Email);

        //configure the mail client
        SmtpClient client = new SmtpClient("mail.yourDomain.ext");
        //Configure your email credentials
        client.Credentials = new NetworkCredential("YourEmailUserName - Web host",
            "Your Email Password - Webhost");

        try
        {
            //send mail
            client.Send(m);
        }
        catch (Exception e)
        {
            //log error in viewbag to be seen by admins
            ViewBag.Message = e.StackTrace;
        }
        return View("EmailConfirmation");
    }
}
```

There are 2 steps that need to be done. We need generate the view to go with the contact form (and delete the existing one) and generate a confirmation view to display after the email is sent.

1. Right click and delete the default contact.cshtml that is found in Views/Home/Contact.cshtml
2. Right click the Home folder and add a view
 - a. View name: Contact

- b. Select Create as the template
- c. Select your ContactViewModelClass as the Model Class
- d. No Data Context is supplied
- e. Check the other options as shown below
- f. Click Add

- g. The view gets generated for you, you can do regular view clean up
 - i. ViewBag.Title
 - ii. The h2
 - iii. The h4 and hr in the form Horizontal
 - iv. Hold off on adding in any styling from your template at this point

2. Right click the Home folder and add a view

- a. View name: EmailConfirmation
- b. Select Details as the template
- c. Select your ContactViewModelClass as the Model Class
- d. No Data Context is supplied
- e. Check the other options as shown below
- f. Click Add

- g. You can clean this template up and do your own formatting or leave it as is. You will want to remove the Edit and back to list links at the bottom.

3. At this point you are ready to test your contact form.

