

Aula 09 – Classe Net.Mail

Disciplina: Programação Web

Prof. Allbert Velleniche de Aquino Almeida

E-mail: allbert.almeida@fatec.sp.gov.br

Site: <http://www.allbert.com.br>

 /allbert.almeida

Objetivo

- O objetivo da aula é realizar o envio de e-mail através da classe NET.MAIL, utilizando credenciais estabelecidas no WebConfig; Com recursos de prioridade da mensagem, cópias, formato da mensagem, etc.

Objetivo

- Também vamos trabalhar com sistemas de alerta de mensagem (SweetAlert);
- E utilizar o editor de texto TinyMCE;

Adicionar as referências

- View >> Shared >> _Layout.cshtml, no final da página abaixo dos "Scripts.Render":

```
<script src="https://cdn.tiny.cloud/1/no-api-key/tinymce/5/tinymce.min.js"
referrerpolicy="origin"></script>
<script>tinymce.init({ selector: 'textarea'
});</script>
```

- Adicionamos o SweetAlert

```
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
```


Inicialização do SweetAlert

- Para exibir as mensagens de alerta usaremos a função "TempData" ainda em "_Layout":

```
@if (TempData["MSG"] != null)
{
    <script>
        $(document).ready(function () {
            var titulo = "";
            switch ("@TempData["MSG"].ToString().Split('|')[0]") {
                case "success":
                    titulo = "Sucesso";
                    break;
                case "warning":
                    titulo = "Atenção";
                    break;
                case "error":
                    titulo = "Erro";
                    break;
                case "info":
                    titulo = "Informação";
                    break;
            }
            swal(titulo, "@Html.Raw(TempData["MSG"].ToString().Split('|')[1])",
"@TempData["MSG"].ToString().Split('|')[0]");
        });
    </script>
}
```

Classe Usuario

- Na classe Usuario iremos adicionar uma propriedade hash para controlar alteração de senha:

```
public string Hash { get; set; }
```


Console Gerenciador de Pacotes

- Confira as credenciais web.config e execute:
Update-Database -Verbose

Web.config

- Credenciais do e-mail no web.config:

```
<system.net>  
  <mailSettings>  
    <smtp>  
      <network host="smtp.gmail.com"  
port="587" userName="fatecgtaads@gmail.com"  
enableSsl="true" password="sgkphybswjbbbthy"  
defaultCredentials="false"/>  
    </smtp>  
  </mailSettings>  
</system.net>
```


Classe Funcoes.cs

```
public static string EnviarEmail(string emailDestinatario, string assunto, string corpomsg)
{
    try
    {
        //Cria o endereço de email do remetente
        MailAddress de = new MailAddress("Fatec ADS <fatecgtaads@gmail.com>");
        //Cria o endereço de email do destinatário -->
        MailAddress para = new MailAddress(emailDestinatario);
        MailMessage mensagem = new MailMessage(de, para);
        mensagem.IsBodyHtml = true;
        //Assunto do email
        mensagem.Subject = assunto;
        //Conteúdo do email
        mensagem.Body = corpomsg;
        //Prioridade E-mail
        mensagem.Priority = MailPriority.Normal;

        //Cria o objeto que envia o e-mail
        SmtpClient cliente = new SmtpClient();
        //Envia o email
        cliente.Send(mensagem);
        return "success|E-mail enviado com sucesso";
    }
    catch { return "error|Erro ao enviar e-mail"; }
}
```

Classe Funcoes.cs

```
public static string Codifica(string texto)
{
    byte[] stringBase64 = new byte[texto.Length];
    stringBase64 = Encoding.UTF8.GetBytes(texto);
    string codifica = Convert.ToBase64String(stringBase64);
    return codifica;
}
public static string Decodifica(string texto)
{
    var encode = new UTF8Encoding();
    var utf8Decode = encode.GetDecoder();

    byte[] stringValor = Convert.FromBase64String(texto);
    int contador = utf8Decode.GetCharCount(stringValor, 0,
stringValor.Length);
    char[] decodeChar = new char[contador];
    utf8Decode.GetChars(stringValor, 0, stringValor.Length, decodeChar, 0);
    string resultado = new String(decodeChar);
    return resultado;
}
```


Criar as ViewModels

- Chegou o momento de criar os formulários para o envio de e-mail, esqueceu senha e redefinir senha. Como apresentado nos conceitos de ViewModel, o “redefinir senha” não será igual ao modelo “Usuario” já criado, pois queremos exibir o campo confirmação de senha (campo não existente no modelo) e regras específicas para a senha (não definida no modelo):

VMUsuario.cs

```
public class Mensagem
{
    [EmailAddress]
    [Required]
    public string Email { get; set; }
    [Required]
    public string Assunto { get; set; }
    [DataType(DataType.MultilineText)]
    [Display(Name = "Mensagem")]
    public string CorpoMsg { get; set; }
}
```


VMUsuario.cs

```
public class EsqueceuSenha
{
    [EmailAddress]
    [Required]
    public string Email { get; set; }
}
```

VMUsuario.cs

```
public class RedefinirSenha
{
    public string Email { get; set; }
    public string Hash { get; set; }
    [DataType(DataType.Password)]
    [RegularExpression("(?=.*\\d)(?=.*[a-z])(?=.*[A-Z]).{6,12})", ErrorMessage = "A senha deve conter aos menos uma letra maiúscula, minúscula e um número. Deve ser no mínimo 6 caracteres")]
    public string Senha { get; set; }
    [DataType(DataType.Password)]
    [Compare("Senha")]
    [Display(Name = "Confirma Senha")]
    public string ConfirmaSenha { get; set; }
}
```


Adicionar Action -> Home

```
public ActionResult Email() {  
    return View();  
}
```

```
[HttpPost]  
[ValidateInput(false)]  
public ActionResult Email(Mensagem msg) {  
    if (ModelState.IsValid) {  
        TempData["MSG"] = Funcoes.EnviaEmail(msg.Email,  
msg.Assunto, msg.CorpoMsg);  
    } else {  
        TempData["MSG"] = "warning|Preencha todos os campos";  
    }  
    return View(msg);  
}
```

Adicionar a View

- Nessa palavra "Email()" do arquivo "HomeController.cs" vamos clicar com o botão direito e selecionar "Adicionar Exibição".

Adicionar modo de exibição

Nome do modo de exibição:

Modelo:

Classe do modelo:

Classe de contexto de dados:

Opções:

Criar como um Modo de exibição parcial

Bibliotecas de scripts de referência

Usar uma página de layout:

...

(deixe em branco se ele estiver definido em um arquivo Razor _viewstart)

Testar Aplicação

- Testar:
 - Acessar ~/Home/Email
 - Utilização de mensagem;
 - TinyMCE;
 - Envio de e-mail;

Adicionar Action -> Home

```
public ActionResult EsqueceuSenha()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult EsqueceuSenha(EsqueceuSenha esq)
{
    if (ModelState.IsValid)
    {
        Contexto db = new Contexto();
        var usu = db.Usuario.Where(x => x.Email == esq.Email).ToList().FirstOrDefault();
        if (usu != null)
        {
            usu.Hash = Funcoes.Codifica(DateTime.Now.AddDays(1).ToString("yyyy-MM-dd HH:mm:ss.ffff"));
            db.Entry(usu).State = EntityState.Modified;
            db.SaveChanges();
            string msg = "<h3>Sistema</h3>";
            msg += "Para alterar sua senha <a href='http://localhost:54305/Home/Redefinir/' + usu.Hash + "
target='_blank'>clique aqui</a>";
            Funcoes.EnviaEmail(usu.Email, "Redefinição de senha", msg);
            TempData["MSG"] = "success|Senha redefinida com sucesso!";
            return RedirectToAction("Index");
        }
        TempData["MSG"] = "error|E-mail não encontrado";
        return View();
    }
    TempData["MSG"] = "warning|Preencha todos os campos";
    return View();
}
```


Adicionar a View

- Nessa palavra “EsqueceuSenha()” do arquivo “HomeController.cs” vamos clicar com o botão direito e selecionar “Adicionar Exibição”.

Adicionar modo de exibição

Nome do modo de exibição:

Modelo:

Classe do modelo:

Classe de contexto de dados:

Opções:

Criar como um Modo de exibição parcial

Bibliotecas de scripts de referência

Usar uma página de layout:

...

(deixe em branco se ele estiver definido em um arquivo Razor _viewstart)

Adicionar Action -> Home

```
public ActionResult Redefinir(string id)
{
    if (!String.IsNullOrEmpty(id))
    {
        Contexto db = new Contexto();
        var usu = db.Usuario.Where(x => x.Hash == id).ToList().FirstOrDefault();
        if (usu != null)
        {
            try
            {
                DateTime dt = Convert.ToDateTime(Funcoes.Decodifica(usu.Hash));
                if (dt > DateTime.Now)
                {
                    RedefinirSenha red = new RedefinirSenha();
                    red.Hash = usu.Hash;
                    red.Email = usu.Email;
                    return View(red);
                }
                TempData["MSG"] = "warning|Esse link já expirou!";
                return RedirectToAction("Index");
            }
            catch
            {
                TempData["MSG"] = "error|Hash inválida!";
                return RedirectToAction("Index");
            }
        }
        TempData["MSG"] = "error|Hash inválida!";
        return RedirectToAction("Index");
    }
    TempData["MSG"] = "error|Acesso inválido!";
    return RedirectToAction("Index");
}
```


Adicionar Action -> Home

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Redefinir(RedefinirSenha red)
{
    if (ModelState.IsValid)
    {
        Contexto db = new Contexto();
        var usu = db.Usuario.Where(x => x.Hash == red.Hash).ToList().FirstOrDefault();
        if (usu != null)
        {
            usu.Hash = null;
            usu.Senha = Funcoes.HashTexto(red.Senha, "SHA512");
            db.Entry(usu).State = EntityState.Modified;
            db.SaveChanges();
            TempData["MSG"] = "success|Senha redefinida com sucesso!";
            return RedirectToAction("Index");
        }
        TempData["MSG"] = "error|E-mail não encontrado";
        return View(red);
    }
    TempData["MSG"] = "warning|Preencha todos os campos";
    return View(red);
}
```

Adicionar a View

- Nessa palavra “RedefinirSenha()” do arquivo “HomeController.cs” vamos clicar com o botão direito e selecionar “Adicionar Exibição”.

Adicionar modo de exibição

Nome do modo de exibição:

Modelo:

Classe do modelo:

Classe de contexto de dados:

Opções:

Criar como um Modo de exibição parcial

Bibliotecas de scripts de referência

Usar uma página de layout:

...

(deixe em branco se ele estiver definido em um arquivo Razor _viewstart)

View RedefinirSenha

- Ocultar o campo Hash e adicionar como HiddenField:

```
@Html.HiddenFor(model => model.Hash);
@*<div class="form-group">
@Html.LabelFor(model => model.Hash, htmlAttributes:
new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.Hash, new {
htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Hash,
"", new { @class = "text-danger" })
  </div>
</div>*@
```

View RedefinirSenha

- Adicionar disabled no campo e-mail:

```
@Html.EditorFor(model => model.Email,  
new { htmlAttributes = new { @class =  
"form-control", @disabled = "disabled" }  
})
```


Atividade prática

- Implementar no sistema do P.I. a funcionalidade do Esqueceu e Redefinir Senha do usuário;