



Giáo trình

ASP.NET

nâng cao



Giáo trình ASP.NET nâng cao

phần 1: Các điều khiển nâng cao trong asp.net

Chương 1: Sử dụng điều khiển Login

- 1.1 Tổng quan về các điều khiển Login
- 1.2 Sử dụng điều khiển Login
- 1.3 Sử dụng điều khiển LoginStatus
- 1.4 Sử dụng điều khiển ChangePassword
- 1.5 Sử dụng điều khiển PasswordRecovery

Chương 2: ASP.NET Membership

- 2.1 Cấu hình Authentication
- 2.2 Cấu hình Authorization
- 2.3 Sử dụng ASP.NET Membership
- 2.4 Sử dụng Role Manager

Chương 3: Global Resource và Local Resource

- 3.1 Thiết lập Current Culture
- 3.2 Sử dụng lớp CulterInfo
- 3.3 Tạo Local Resources
- 3.4 Tạo Global Resources
- 3.5 Sử dụng điều khiển Localize

Chương 4: Sử dụng các điều khiển điều hướng

- 4.1 Tổng quan về Site Map
- 4.2 Sử dụng điều khiển SiteMapPath
- 4.3 Sử dụng điều khiển Menu
- 4.4 Sử dụng điều khiển TreeView

Chương 5: Xây dựng và sử dụng các Điều khiển do người dùng tạo ra

- 5.1 Tổng quan về xây dựng các điều khiển
- 5.2 ViewState và ControlState
- 5.3 Xử lý sự kiện và Dữ liệu trả về

Chương 6: ASP.NET và AJAX

Phần 2: Xây dựng ứng dụng Với ASP.NET

(Mục tiêu: Xây dựng website thương mại điện tử, bán máy tính trực tuyến.)

Chương 8: Giới thiệu về hệ thống thương mại điện tử

Chương 9: Mô hình kinh doanh

- 9.1 Thu thập yêu cầu
- 9.2 Phân tích hiệu quả
- 9.3 Nghiên cứu và quản lý rủi do

Chương 10: Mô hình hóa và xây dựng giao diện ứng dụng

- 10.1 Mô hình hóa hệ thống
- 10.2 Thiết kế dữ liệu với sqlServer 2005

Chương 11: Thiết kế kiến trúc

- 11.1 Xây dựng kiến trúc hệ thống

- 11.2 Tạo các đối tượng dùng chung
- 11.3 Tạo tầng truy cập dữ liệu
- 11.4 Tạo tầng xử lý nghiệp vụ
- 11.5 Lựa chọn tích hợp cho ứng dụng
- 11.6 Tạo tầng trình bày

Chương 12: Xây dựng ứng dụng

12.1 Phát triển danh mục sản phẩm

- 12.2 Xây dựng Giỏ hàng
- 12.3 Tích hợp thanh toán trực tuyến
- 12.4 Cài đặt Xử lý kiểm tra
- 12.5 Xây dựng các điều khiển quản trị.
- 12.6 Xây dựng tài khoản khách hàng

Chương 13: triển khai và bảo trì ứng dụng

Nội Dung

Chương 1. Sử dụng các điều khiển login

Bạn có thể sử dụng các điều khiển Login của ASP.NET để xây dựng các hệ thống đăng ký người sử dụng cho website của mình, Bạn có thể sử dụng các Login Control để tạo form đăng nhập, đăng ký, thay đổi mật khẩu hay ghi nhớ mật khẩu trên Form.

Trong chương này chúng ta sẽ học chi tiết các điều khiển

- Login: Cho phép hiển thị Form đăng nhập người sử dụng.
- CreateUserWizard: Cho phép hiển thị Form đăng ký người sử dụng
- LoginStatus: Hiển thị trạng thái Login hay Logout phụ thuộc vào trạng thái kiểm chứng người sử dụng
- LoginName: Hiển thị tên người đăng ký hiện tại
- ChangePassword: Hiển thị Form cho phép người sử dụng thay đổi mật khẩu
- PasswordRecovery: Cho phép người sử dụng khôi phục password, password này sẽ được gửi vào mail cho người sử dụng.
- LoginView: hiển thị các nội dung khác nhau tới mỗi người sử dụng thuộc thuộc vào authentication hoặc role.

1.1 Tổng quan về các điều khiển login

Giả sử bạn có một trang web như sau:

Listing 1.1 Baomat/Secret.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Secret.aspx.cs"
Inherits="_Default" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Bao mat</title>
</head>
<body>
<form id="form1" runat="server">
```

```
<div>
Đây là trang web được bảo mật
</div>
</form>
</body>
</html>
```

Khi chạy, trang sẽ hiển thị dòng chữ “Đây là trang web được bảo mật”
Để bảo mật cho trang web này bạn cần thiết lập hai cấu hình trên ứng dụng của bạn, cần
cấu hình cả authentication và authorization.

Đầu tiên bạn cần thiết lập authentication về mode=”Forms” trong file web.config trong
thư mục gốc.

Listing 1.2 web.config

```
<system.web>
<authentication mode="Forms" />
</system.web>
```

Bởi mặc định tất cả mọi người đều có thể truy cập vào website, nếu bạn ngăn cản người sử
dụng truy cập vào thư mục nào bạn cấu hình authorization cho thư mục đó, trong Listing
1.3 sau sẽ ngăn cản người sử dụng truy cập vào thư mục “Baomat”.

Listing 1.3 Baomat/web.config

```
<configuration>
<system.web>
<authorization>
<deny users="?">
</authorization>
</system.web>
</configuration>
```

Khi bạn thiết lập hai cấu hình trên thì khi website của bạn yêu cầu tới trang Secret.aspx
thì ứng website sẽ tự động chuyển về trang login.aspx

Listing 1.4 Login.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Login.aspx.cs"
Inherits="Login" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Trang đăng nhập</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Login ID="loginVidu" CreateUserUrl="~/Register.aspx" CreateUserText="Đăng
ký" runat="server">
```

```
</asp:Login>
</div>
</form>
</body>
</html>
```

Giao diện trang Login.aspx

(Hình 1)

Trong Listing 1.4 bạn để ý điều khiển Login có hai thuộc tính CreateUserText, CreateUserUrl. Nó sẽ đưa ra một link với Text là “Đăng ký” và một địa chỉ URL dẫn tới trang đăng ký trong trường hợp này là trang Register.aspx.

Tập tin đính kèm:

The screenshot shows a login interface. At the top right is a 'Log In' button. Below it are two input fields: 'User Name:' and 'Password:', both enclosed in light gray boxes. Underneath the password field is a checkbox labeled 'Remember me next time.' which is checked. At the bottom right is another 'Log In' button, and to its left is a blue underlined link labeled 'Đăng ký'.

1.2 Sử dụng điều khiển Login

Điều khiển Login đưa ra một form đăng nhập tiêu chuẩn. Mặc định điều khiển Login sử dụng ASP.NET Membership để kiểm chứng người sử dụng, tuy nhiên bạn có thể tùy chỉnh kiểu kiểm chứng người sử dụng với điều khiển Login.

Điều khiển Login hỗ trợ rất nhiều thuộc tính cho phép bạn tùy chỉnh cách hiển thị và ứng xử của điều khiển như Listing 1.5 sau:

Listing 1.5 showLogin.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="showLogin.aspx.cs"
Inherits="showLogin" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Hiển thị Form đăng nhập</title>
    <style type="text/css">
```

```

.login
{
    width:250px;
    font:14px Verdana,Sans-Serif;
    background-color:lightblue;
    border:solid 3px black;
    padding:4px;
}
.login_title
{
    background-color:darkblue;
    color:white;
    font-weight:bold;
}
.login_instructions
{
    font-size:12px;
    text-align:left;
    padding:10px;
}
.login_button
{
    border:solid 1px black;
    padding:3px;
}

```

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:Login ID="Login1" InstructionText="Bạn cần nhập tên đăng nhập và mật khẩu để đăng nhập" TitleText="Đăng nhập" TextLayout="TextOnTop" LoginButtonText="Đăng nhập" CssClass="login" TitleTextStyle-CssClass="login_title" InstructionTextStyle-CssClass="login_instructions" LoginButtonStyle-CssClass="login_button" runat="server">

</asp:Login>

</div>

</form>

</body>

</html>

Kết xuất của chương trình trên

Hình 2

1.2.1 Tự động chuyển trang tới một trang chỉ định

Nếu bạn yêu cầu 1 trang mà bạn chưa được kiểm chứng, ASP.NET sẽ tự động chuyển bạn tới trang Login.aspx. Sau khi bạn đăng nhập thành công, nó sẽ chuyển bạn ngược lại tới trang yêu cầu.

Khi bạn bị chuyển sang trang Login.aspx, một chuỗi truy vấn tham số đặt tên là returnUrl được tự động thêm vào trang yêu cầu. Chuỗi truy vấn này sẽ chứa đường dẫn của trang yêu cầu. Điều khiển Login sẽ sử dụng tham số returnUrl này chuyển trang trở lại trang nguồn.

1.2.2 Tự động ẩn điểu khiển Login khi kiểm chứng người sử dụng

Một vài website hiển thị điểu khiển trên đỉnh tất cả các trang, khi người sử dụng đăng nhập thành công thì điểu khiển này tự động ẩn đi. Để làm điều này thật đơn giản trên ASP.NET, bạn chỉ cần thêm vào một điểu khiển Login trên MasterPage, và điểu khiển login này có thể hiển thị trên tất cả các trang có sử dụng MasterPage. Bạn có thể sử dụng thuộc tính Orientation để hiển thị điểu khiển Login này theo chiều ngang hoặc chiều dọc như ví dụ sau:

Listing 1.6 Main.master

Mã:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Main.master.cs"
Inherits="Main" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Điều khiển Login</title>
    <style type="text/css">
        body
        {
            background-color:#e5e5e5;
        }
        .content
        {
            margin:auto;
            width:650px;
            border:solid 1px black;
            background-color:white;
            padding:10px;
        }
        .login
        {
            font:10px Arial,Sans-Serif;
            margin-left:auto;
```

```

        }
    .login input
    {
        }

    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div id="content">
            <asp:Login
                id="loginVidu"
                Orientation="Horizontal"
                VisibleWhenLoggedIn="false"
                DisplayRememberMe="false"
                TitleText=""
                CssClass="login"
                Runat="server" />
            <hr />
            <asp:contentplaceholder id="ContentPlaceHolder1" runat="server">
                </asp:contentplaceholder>
            </div>
        </form>
    </body>
</html>

```

Trang loginMaster.aspx

```

<%@ Page Language="C#" MasterPageFile="~/Main.master" AutoEventWireup="true"
CodeFile="loginMaster.aspx.cs" Inherits="loginMaster" Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <h1>Chào mừng bạn đến với website của chúng tôi</h1>
</asp:Content>

```

Kết xuất của chương trình

Hình 3

1.2.3 Sử dụng Template

Điều khiển Login bao gồm thuộc tính LayoutTemplate cho phép bạn tùy chỉnh cách thể hiện của điều khiển Login.

Khi bạn thêm vào một mẫu hiển thị, bạn cần thêm vào điều khiển và Temple các ID sau:

- UserName
- Password
- RememberMe
- FailureText

Và bạn cần thêm vào một thuộc tính CommandName với giá trị Login
Listing 1.7 LoginTemplate.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="LoginTemplate.aspx.cs" Inherits="LoginTemplate" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Trang đăng nhập</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Login ID="Login1" runat="server">
                <LayoutTemplate>
                    <asp:Label ID="lblusername" runat="server" Text="Tên đăng
nhập"></asp:Label><br />
                    <asp:TextBox ID="UserName" runat="server"></asp:TextBox><br />
                    <asp:Label ID="lblpass" runat="server" Text="Mật khẩu"></asp:Label><br />
                    <asp:TextBox ID="Password" runat="server"></asp:TextBox><br /><br />
                    <asp:Button ID="btnLogin" CommandName="Login" Text="Đăng nhập"
runat="server" />
                </LayoutTemplate>
            </asp:Login>
        </div>
    </form>
</body>
</html>
```

Kết xuất của chương trình

Hình 4

1.2.4 Thực hiện tùy chỉnh kiểm chứng với điều khiển Login

Mặc định, điều khiển Login sử dụng ASP.NET MemberShip để kiểm chứng tên sử dụng và mật khẩu. Nếu bạn cần thay đổi ứng xử mặc định bạn có thể điều khiển sự kiện Authenticate của điều khiển Login như ví dụ dưới đây.

Listing 1.8 Web.config

Mã:

```
<configuration>
    <appSettings/>
    <connectionStrings/>
    <system.web>
```

```

<authentication mode="Forms">
  <forms>
    <credentials passwordFormat="Clear">
      <user name="Thietke" password="itechpro"/>
      <user name="Daotao" password="itechpro"/>
    </credentials>
  </forms>
</authentication>
</system.web>
</configuration>

```

Trang LoginCustom.aspx

Mã:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="LoginCustom.aspx.cs"
Inherits="LoginCustom" %>

<script runat="server">
  protected void loginCustom_Authenticate(object sender, AuthenticateEventArgs e)
  {
    string userName = loginCustom.UserName;
    string Password = loginCustom.Password;
    e.Authenticated = FormsAuthentication.Authenticate(userName, Password);
  }
</script>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Login Custom</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Login ID="loginCustom" OnAuthenticate="loginCustom_Authenticate"
runat="server">
        </asp:Login>
    </div>
  </form>
</body>
</html>

```

1.3 Sử dụng điều khiển CreateUserWizard

Điều khiển CreateUserWizard đưa ra một Form đăng ký người sử dụng. Một người đăng ký thành công, người đăng ký mới đó sẽ được thêm vào website của bạn. Điều khiển

CreateUserWizard sử dụng ASP.NET MemberShip để tạo một người sử dụng mới. Điều khiển CreateUserWizard hỗ trợ rất nhiều thuộc tính cho phép bạn tùy chỉnh cách hiển thị và ứng xử như ví dụ sau:

Listing 1.9 showCreateUserWizard.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="showCreateUserWizard.aspx.cs" Inherits="showCreateUserWizard" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Hiển thị Form đăng ký người sử dụng</title>
    <style type="text/css">
        .createUser
        {
            width:350px;
            font:14px Verdana,Sans-Serif;
            background-color:lightblue;
            border:solid 3px black;
            padding:4px;
        }
        .createUser_title
        {
            background-color:darkblue;
            color:white;
            font-weight:bold;
        }
        .createUser_instructions
        {
            font-size:12px;
            text-align:left;
            padding:10px;
        }
        .createUser_button
        {
            border:solid 1px black;
            padding:3px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:CreateUserWizard
```

```

        id="CreateUserWizard1"
        ContinueDestinationPageUrl="~/Default.aspx"
        InstructionText="Please complete the following form to register at this Website."
        CompleteSuccessText="Your new account has been created. Thank you for
registering."
        CssClass="createUser"
        TitleTextStyle-CssClass="createUser_title"
        InstructionTextStyle-CssClass="createUser_instructions"
        CreateUserButtonStyle-CssClass="createUser_button"
        ContinueButtonStyle-CssClass="createUser_button"
        Runat="server" />
    </div>
</form>
</body>
</html>

```

Kết xuất của chương trình

Hình 5

1.3.1 Gửi Email thông báo tới người sử dụng

Bạn có thể thiết lập cho phép điều khiển CreateUserWizard gửi thư tự động đến người sử dụng khi đăng ký thành công một tài khoản mới trên website của mình
Ví dụ bạn có thể gửi một mail chứa đựng thông tin về tài khoản và mật khẩu của người sử dụng về tài khoản email của người này.

Listing 1.10 CreateUserWizardEmail.aspx

Mã:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="CreateUserWizardEmail.aspx.cs" Inherits="CreateUserWizardEmail" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Send Email</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:CreateUserWizard
                id="CreateUserWizard1"
                Runat="server">
                <MailDefinition
                    BodyFileName="Register.txt"
                    Subject="Xác nhận đăng ký"

```

```
        From="Admin@YourSite.com" />
    </asp:CreateUserWizard>
</div>
</form>
</body>
</html>
```

Lớp MailDefinition hỗ trợ các thuộc tính sau:

BodyFileName: chỉ định đường dẫn chứa nội dung thư

CC: Cho phép gửi một bản copy tới hòm thư khác

EmbeddedObjects: Cho phép gửi kèm các file khác như là ảnh, doc...

From: Địa chỉ hòm thư gửi.

IsBodyHtml: Cho phép gửi định dạng Html

Priority: Cho phép bạn chỉ định độ ưu tiên của thư, nó có thể có các giá trị sau: High, Low, và Normal

Subject: Chỉ định tiêu đề của thư.

Lớp MailDefinition sử dụng mail server được cấu hình bởi thành phần SMTP trong file Web.config như ví dụ sau:

Listing 1.11 Web.config

Mã:

```
<configuration>
  <system.net>
    <mailSettings>
      <smtp deliveryMethod="PickupDirectoryFromIis"></smtp>
    </mailSettings>
  </system.net>
</configuration>
```

Với ví dụ trên là bạn dùng mail server từ máy cục bộ, bạn cũng có thể thiết lập mail server từ một máy chủ khác bằng việc chỉ định các mail host, username và Passsword

Listing 1.12 Web.config

Mã:

```
<configuration>
  <system.net>
    <mailSettings>
      <smtp>
        <network host="mail.yourdomain.com" userName="admin" password="secret"/>
      </smtp>
    </mailSettings>
  </system.net>
</configuration>
```

1.3.2 Chuyển người sử dụng sang một trang khác tự động

Khi bạn đăng nhập thành công trên trang Login.aspx, trang tự động chuyển lại trang người sử dụng vừa yêu cầu. Với điều khiển CreateUserWizard không tự động làm việc

này cho chúng ta, để nó có thể làm việc giống với điều khiển Login chúng ta cần viết thêm một ít code.

Điều khiển login trong Listing 1.12 sau bao gồm một đường để người sử dụng chuyển đến trang đăng ký sử dụng được đặt tên là CreateUserWizardReturn.aspx. Trong sự kiện Page_Load(). Giá trị của chuỗi truy vấn được thêm vào trang đăng ký.

Listing 1.13 LoginReturn.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="LoginReturn.aspx.cs"
Inherits="LoginReturn" %>

<script runat="server">
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            string dest = Request.QueryString["ReturnUrl"];
            Login1.CreateUserUrl = "~/CreateUserWizardReturn.aspx?ReturnUrl=" +
Server.UrlEncode(dest);
        }
    }
</script>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Đăng nhập</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Login
                id="Login1"
                CreateUserText="Register"
                CreateUserUrl="~/CreateUserWizardReturn.aspx"
                Runat="server" />
        </div>
    </form>
</body>
</html>
```

Trước khi sử dụng Listing 1.13 bạn cần thay đổi tên trang LoginReturn.aspx thành trang Login.aspx. Nếu người sử dụng yêu cầu đến một trang yêu cầu phải kiểm chứng, người sử dụng sẽ tự động chuyển tới trang Login.aspx. Tham số ReturnUrl sẽ được gắn vào

trang Login này.

Trong Listing 1.14 sau chưa đựng một điều khiển CreateUserWizard. Trang này chỉ chứa đựng một điều khiển sự kiện Page_Load(). Giá trị của tham số ReturnUrl được sử dụng để đưa người sử dụng trở về trang mà người sử dụng yêu cầu.

Listing 1.14 CreateUserWizardReturn.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="CreateUserWizardReturn.aspx.cs" Inherits="CreateUserWizardReturn" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
    void Page_Load()
    {
        if (!Page.IsPostBack)
        {
            string dest = "~/Default.aspx";
            if (!String.IsNullOrEmpty(Request.QueryString["ReturnURL"]))
                dest = Request.QueryString["ReturnURL"];
            CreateUserWizard1.ContinueDestinationPageUrl = dest;
        }
    }
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Gọi lại trang CreateUserWizardReturn.aspx</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:CreateUserWizard ID="CreateUserWizard1" runat="server" />
        </div>
    </form>
</body>
</html>
```

1.3.3 Sinh Password tự động

Một vài website khi bạn đăng ký sử dụng, nó chỉ yêu cầu bạn nhập các thông tin cá nhân còn mật khẩu website sẽ tự sinh ra và gửi về hộp thư của bạn. sau khi đăng nhập lần đầu bạn có thể thay đổi mật khẩu này.

Nếu bạn cần sử dụng kịch bản này cho website của bạn khi người sử dụng đăng ký sử dụng, bạn cần lầm được 3 thuộc tính sau của điều khiển CreateUserWizard

AutoGeneratePassword: Cho phép tự sinh ra một password tự động

DisableCreatedUser: Cho phép vô hiệu hóa tạo tài khoản mới từ điều khiển CreateUserWizard.

LoginCreatedUser: Cho phép bạn ngăn cản người sử dụng mới sẽ được đăng nhập tự

động

Bạn có thể gửi hai kiểu mail xác nhận. Đầu tiên bạn có thể sinh password tự động và gửi Password vào hòm thư của người sử dụng. Trong trường hợp này bạn sẽ cho phép thuộc tính AutoGeneratePassword và vô hiệu hóa thuộc tính LoginCreatedUser. Trường hợp thứ 2, bạn có thể cho phép một người sử dụng nhập mật khẩu và gửi mã xác nhận đến hòm thư xác nhận. Trong trường hợp này bạn sẽ cho phép thuộc tính DisableCreatedUser và vô hiệu hóa thuộc tính LoginCreatedUser.

Listing 1.15 chúa đựng một điều khiển CreateUserWizard mà không yêu cầu nhập mật khẩu. Điều khiển này cho phép thuộc tính AutoGeneratePassword và vô hiệu hóa thuộc tính LoginCreatedUser

Listing 1.15 CreateUserWizardPasswordConfirmation.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="CreateUserWizardPasswordConfirmation.aspx.cs"
Inherits="CreateUserWizardPasswordConfirmation" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Tự động sinh Password</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:CreateUserWizard
                id="CreateUserWizard1"
                CompleteSuccessText="A confirmation email
                containing your new password has been
                sent to your email address."
                AutoGeneratePassword="true"
                LoginCreatedUser="false"
                ContinueDestinationPageUrl="~/Login.aspx"
                Runat="server">
                <MailDefinition
                    From="Admin@YourSite.com"
                    BodyFileName="PasswordConfirmation.htm"
                    IsBodyHtml="true"
                    Subject="Registration Confirmation" />
            </asp:CreateUserWizard>
        </div>
    </form>
</body>
</html>
```

Điều khiển CreateUserWizard gửi thư chứa đựng như trong Listing 1.16
Listing 1.16 PasswordConfirmation.htm

Mã:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" >  
<head>  
    <title>Nội dung xác nhận mật khẩu</title>  
</head>  
<body>  
    Nội dung xác nhận mật khẩu  
    Your new password is <% Password %>.  
</body>  
</html>
```

Khi tạo tài khoản thành công trong mail chứa đựng mật khẩu tự sinh ra. Người sử dụng sử dụng mật khẩu này để đăng nhập trên website.

Trong kịch bản thứ 2, người sử dụng có thể sử dụng password mà người đó chọn. Tuy nhiên tài khoản này sẽ bị vô hiệu hóa cho đến khi người này nhập mã xác nhận.

Điều khiển CreateUserWizard trong Listing 1.17 cho phép thuộc tính DisableCreateUser và vô hiệu hóa thuộc tính LoginCreatedUser.

Listing 1.17 CreateUserWizardCodeConfirmation.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeFile="CreateUserWizardCodeConfirmation.aspx.cs"  
Inherits="CreateUserWizardCodeConfirmation" %>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
    <script runat="server">  
        protected void CreateUserWizard1_SendingMail(object sender,  
MailMessageEventArgs e)  
        {  
            MembershipUser user = Membership.GetUser(CreateUserWizard1.UserName);  
            string code = user.ProviderUserKey.ToString();  
            e.Message.Body = e.Message.Body.Replace("<%ConfirmationCode%>", code);  
        }  
    </script>  
<html xmlns="http://www.w3.org/1999/xhtml" >  
<head runat="server">  
    <title>Tạo form đăng ký yêu cầu xác nhận mã</title>  
</head>  
<body>  
    <form id="form1" runat="server">
```

```

<div>
    <asp:CreateUserWizard id="CreateUserWizard1" CompleteSuccessText="A
confirmation email
        containing your new password has been sent to your email address."
        DisableCreatedUser="true"
        ContinueDestinationPageUrl="~/ConfirmCode.aspx"
        OnSendingMail="CreateUserWizard1_SendingMail"
        Runat="server">

        <MailDefinition
            From="Admin@YourSite.com"
            BodyFileName="CodeConfirmation.htm"
            IsBodyHtml="true"
            Subject="Registration Confirmation" />
    </asp:CreateUserWizard>
</div>
</form>
</body>
</html>

```

Trong Listing 1.17 gồm một điều khiển sự kiện SendingMail. Mã sử dụng là một khóa duy nhất gửi tới người sử dụng bởi MemberShip Provider. Mã xác nhận được đê trình trong mail trước khi mail được gửi. Mail này được chua đựng trong Listing 1.18 Listing 1.18

Mã:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>Untitled Page</title>
</head>
<body>
    <div>
        <%UserName%>,
        Mã xác nhận của bạn là <%ConfirmationCode%>
    </div>
</body>
</html>

```

Sau khi hoàn thành Form đưa ra bởi CreateUserWizard bạn nhấn vào nút Continue để mở trang ConfirmCode.aspx.

Listing 1.19 ConfirmCode.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ConfirmCode.aspx.cs"
Inherits="ConfirmCode" %>
```

```

<script runat="server">
    protected void btnConfirm_Click(object sender, EventArgs e)
    {
        MembershipUser user = Membership.GetUser(txtUserName.Text);
        if (user == null)
        {
            lblError.Text = "Tên sử dụng không đúng";
        }
        else
        {
            string providerCode = user.ProviderUserKey.ToString();
            string userCode = txtConfirmationCode.Text.Trim();
            if (providerCode != userCode)
            {
                lblError.Text = "Sai mã xác nhận";
            }
            else
            {
                user.IsApproved = true;
                Membership.UpdateUser(user);
                Response.Redirect("~/Baomat/Secret.aspx");
            }
        }
    }
</script>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>kiểm tra xác nhận</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <p>
                Nhập mã xác nhận mà bạn được gửi từ mail.
            </p>
            <asp:Label id="lblError" EnableViewState="false" ForeColor="Red"
Runat="server" />
            <br /><br />
            <asp:Label id="lblUserName" Text="tên sử dụng:"
AssociatedControlID="txtUserName" Runat="server" />
            <br />
            <asp:TextBox id="txtUserName" Runat="server" />

```

```

<br /><br />
<asp:Label id="lblConfirmationCode" Text="Mã xác nhận:" AssociatedControlID="txtConfirmationCode" Runat="server" />
<br />
<asp:TextBox id="txtConfirmationCode" Columns="50" Runat="server" />
<asp:Button id="btnConfirm" Text="Xác nhận" OnClick="btnConfirm_Click" Runat="server" />
</div>
</form>
</body>
</html>

```

1.4 Sử dụng điều khiển LoginStatus

Điều khiển LoginStatus hiển thị trạng thái liên kết Login hoặc Logout, phụ thuộc vào trạng thái kiểm chứng của bạn. Khi bạn nhấp vào liên kết Link, bạn được chuyển đến trang Login.aspx. Khi nhấp vào liên kết Logout bạn sẽ đăng xuất khỏi website.

Listing 1.20 ShowLoginStatus.aspx

Mã:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ShowLoginStatus.aspx.cs" Inherits="ShowLoginStatus" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>hiển thị LoginStatus</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:loginstatus ID="Loginstatus1" runat="server"></asp:loginstatus>
        </div>
    </form>
</body>
</html>

```

LoginStatus hỗ trợ các thuộc tính sau:

- LoginImageUrl: định nghĩa một ảnh cho Login Link.
- LoginText: định nghĩa Text cho Login Link.
- LogoutAction: cho phép bạn điều khiển việc gì sẽ xảy ra khi bạn nhấp vào Logout Link. Có thể là các giá trị sau: Redirect, RedirectToLoginPage, Refresh.
- LogoutImageUrl: cho phép bạn định nghĩa ảnh cho Logout Link.
- LogoutPageUrl: Định nghĩa trang mà người sử dụng sẽ chuyển đến khi họ đăng xuất. Thuộc tính này mặc định sẽ bị bỏ qua trừ khi bạn thiết lập thuộc tính LogoutAction có giá trị là Redirect.

- LogoutText: Định nghĩa nội dung cho Logout Link.

LoginStatus hỗ trợ hai sự kiện sau:

- LoggingOut: Xảy ra trước khi người sử dụng đăng xuất
- LoggedOut: Xảy ra sau khi người sử dụng đăng xuất

1.5 Sử dụng điều khiển LoginName

Sử dụng điều khiển LoginName bạn có thể cho phép hiển thị tên người sử dụng đã được đăng ký. Nếu người sử dụng hiện tại không được kiểm chứng điều khiển LoginName sẽ đưa ra giá trị rỗng.

Listing 1.21 ShowLoginName.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ShowLoginName.aspx.cs" Inherits="ShowLoginName" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>hiển thị LoginName</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:LoginName id="LoginName1" FormatString="{0} /" Runat="server" />
            <asp:LoginStatus id="LoginStatus1" Runat="server" />
        </div>
    </form>
</body>
</html>
```

Trong Listing 1.21 ở trên bạn thấy rằng điều khiển LoginName có chứa đựng thuộc tính FormatString. Nó cho phép bạn định dạng tên người sử dụng khi tên người sử dụng được đưa ra.

1.6 Sử dụng điều khiển ChangePassword

Điều khiển ChangePassword cho phép người sử dụng hay người quản trị có thể thay đổi mật khẩu của mình

Listing 1.22 ShowChangePassword.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ShowChangePassword.aspx.cs" Inherits="ShowChangePassword" %>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>hiển thị thay đổi mật khẩu</title>
    <style type="text/css">
        .changePassword
        {
            font:14px Verdana,Sans-Serif;
            background-color:lightblue;
            border:solid 3px black;
            padding:4px;
        }
        .changePassword_title
        {
            background-color:darkblue;
            color:white;
            font-weight:bold;
        }
        .changePassword_instructions
        {
            font-size:12px;
            text-align:left;
            padding:10px;
        }
        .changePassword_button
        {
            border:solid 1px black;
            padding:3px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:LoginName ID="LoginName1" runat="server" />
            <asp:ChangePassword
                id="ChangePassword1"
                InstructionText="Complete this form to create a new password."
                DisplayUserName="true"
                ContinueDestinationPageUrl="~/Default.aspx"
                CancelDestinationPageUrl="~/Default.aspx"
                CssClass="changePassword"
                TitleTextStyle-CssClass="changePassword_title"
                InstructionTextStyle-CssClass="changePassword_instructions">
        </div>
    </form>
</body>

```

```

    ChangePasswordButtonStyle-CssClass="changePassword_button"
    CancelButtonStyle-CssClass="changePassword_button"
    ContinueButtonStyle-CssClass="changePassword_button"
    Runat="server" />
</div>
</form>
</body>
</html>

```

Kết xuất của chương trình

Hình 6

1.6.1 Gửi một Email thay đổi Password

Sau khi người sử dụng thay đổi mật khẩu thành công, bạn có thể sử dụng điều khiển ChangePassword để tự động gửi một email tới người sử dụng với nội dung chứa kèm mật khẩu mới của người đó.

Listing 1.23 ChangePasswordEmail.aspx

Mã:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ChangePasswordEmail.aspx.cs" Inherits="ChangePasswordEmail" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Thay đổi mật khẩu</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ChangePassword
                id="ChangePassword1"
                DisplayUserName="true"
                Runat="server">
                <MailDefinition
                    From="Admin@YourSite.com"
                    BodyFileName="ChangePassword.txt"
                    Subject="Your New Password" />
            </asp:ChangePassword>
        </div>
    </form>
</body>
</html>

```

Trong đó nội dung file ChangePassword.txt có nội dung như sau
<%UserName%>,
your new password is <%Password%>.

1.6.2 Sử dụng Temples với điều khiển ChangePassword

Nếu bạn cần tùy chỉnh cách xuất hiện của điều khiển ChangePassword bạn có thể sử dụng templates để định dạng điều khiển. Điều khiển ChangePassword hỗ trợ cả hai ChangePasswordTemplate và SuccessTemplate. Ví dụ sau sẽ hướng dẫn bạn sử dụng cả hai temples này.

Listing 1.24 ChangePasswordTemplate.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ChangePasswordTemplate.aspx.cs" Inherits="ChangePasswordTemplate" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Sử dụng template với ChangePassword</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ChangePassword id="ChangePassword1" DisplayUserName="true"
Runat="server">
                <ChangePasswordTemplate>
                    <h1>Thay đổi mật khẩu</h1>
                    <asp:Label id="FailureText" EnableViewState="false" ForeColor="Red"
Runat="server" />
                    <br />
                    <asp:Label id="lblUserName" Text="Tên sử dụng:" 
AssociatedControlID="UserName" Runat="server" />
                    <br />
                    <asp:TextBox id="UserName" Runat="server" />
                    <br /><br />
                    <asp:Label id="lblCurrentPassword" Text="Mật khẩu hiện tại:"
AssociatedControlID="CurrentPassword" Runat="server" />
                    <br />
                    <asp:TextBox id="CurrentPassword" TextMode="Password" Runat="server" />
                    <br /><br />
                    <asp:Label id="lblNewPassword" Text="Mật khẩu mới:" 
AssociatedControlID="NewPassword" Runat="server" />
                    <br />
```

```

<asp:TextBox id="NewPassword" TextMode="Password" Runat="server" />
<br /><br />
<asp:Button id="btnChangePassword" Text="Change Password"
CommandName="ChangePassword" Runat="server" />
</ChangePasswordTemplate>
<SuccessTemplate>
    Your password has been changed!
</SuccessTemplate>
</asp:ChangePassword>
</div>
</form>
</body>
</html>

```

Kết xuất của chương trình

Hình 7

Khi sử dụng Templates bạn với điều khiển ChangePassword bạn cần thêm vào các ID kèm theo

- UserName
- CurrentPassword
- ConfirmPassword
- NewPassword
- FailureText

Bạn chỉ có thể thêm vào các điều khiển Button theo các giá trị cho thuộc tính CommandName:

- ChangePassword
- Cancel
- Continue

1.7 Sử dụng điều khiển PasswordRecovery

Nếu người sử dụng quên mật khẩu của mình, người đó có thể sử dụng điều khiển PasswordRecovery để khôi phục mật khẩu, điều khiển PasswordRecovery sẽ gửi lại mật khẩu ban đầu hoặc tạo một mật khẩu mới và gửi tới cho người sử dụng.

Listing 1.25 showPasswordRecovery.aspx

Mã:
<%@ Page Language="C#" AutoEventWireup="true" %>

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">

```

```

<title>show Password Recovery</title>
<style type="text/css">
    .passwordRecovery
    {
        font:14px Verdana,Sans-Serif;
        background-color:lightblue;
        border:solid 3px black;
        padding:4px;
    }
    .passwordRecovery_title
    {
        background-color:darkblue;
        color:white;
        font-weight:bold;
    }
    .passwordRecovery_instructions
    {
        font-size:12px;
        text-align:left;
        padding:10px;
    }
    .passwordRecovery_button
    {
        border:solid 1px black;
        padding:3px;
    }
</style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:PasswordRecovery
                id="PasswordRecovery1"
                CssClass="passwordRecovery"
                TitleTextStyle-CssClass="passwordRecovery_title"
                InstructionTextStyle-CssClass="passwordRecovery_instructions"
                SubmitButtonStyle-CssClass="passwordRecovery_button"
                Runat="server">
                <MailDefinition
                    From="Admin@YourSite.com"
                    Subject="Password Reminder" />
            </asp:PasswordRecovery>
        </div>
    </form>
</body>
</html>

```

Khi chạy chương trình trên đầu tiên bạn sẽ bị yêu cầu nhập tên sử dụng, tiếp theo bạn phải nhập câu trả lời bí mật mà bạn sử dụng trong lúc đăng ký, cuối cùng mật khẩu sẽ được gửi đến tài khoản email của người đăng ký.

Bởi mặc định điều khiển PasswordRecovery sẽ khởi tạo mật khẩu và gửi đến người sử dụng.

1.7.1 Gửi mật khẩu ban đầu

Bởi mặc định điều khiển PasswordRecovery không gửi mật khẩu cũ của người sử dụng, nếu bạn không muốn điều khiển PasswordRecovery khởi tọa lại password của người sử dụng trước khi gửi nó bạn phải thay đổi cấu hình MemberShip Provider, Ba thiết lập cấu hình chính passwordFormat, enablePasswordRetrieval, và enablePasswordReset.

Bởi mặc định thuộc tính passwordFormat có giá trị là Hashed. Khi mật khẩu được Hashed, Điều khiển PasswordRecovery không thể gửi mật khẩu gốc ban đầu của người sử dụng, Nếu bạn muốn người sử dụng nhận được password cũ của mình bạn cần thiết lập thuộc tính passwordFormat về các giá trị Clear hoặc Encrypted.

Mặc định thuộc tính enablePasswordRetrieval có giá trị là false, nếu bạn muốn người sử dụng nhận được mật khẩu cũ bạn phải thiết lập cho phép thuộc tính này trong file web.config.

Cuối cùng, mặc định thuộc tính enablePasswordReset có giá trị là true, Nó không chú ý đến giá trị của PasswordFormat hay enablePasswordRetrieval, bạn có thể luôn luôn khởi tạo lại mật khẩu của người sử dụng

Listing 1.26 sau chứa đựng những cấu hình cần thiết cho phép gửi mật khẩu cũ đến người sử dụng.

Listing 1.26 Web.config

Mã:

```
<?xml version="1.0"?>

<configuration>
  <appSettings/>
  <connectionStrings/>
  <system.web>
    <authentication mode="Windows" />
    <membership defaultProvider="MyMemberShip">
      <providers>
        <add name="MyMembership"
          type="System.Web.Security.SqlMembershipProvider"
          connectionStringName="LocalSqlServer"
          passwordFormat="Clear"
          enablePasswordRetrieval="true" />
      </providers>
    </membership>
  </system.web>
</configuration>
```

1.7.2 Yêu cầu câu hỏi bảo mật và trả lời

Khi bạn sử dụng CreateUserWizard để tạo form đăng ký, bạn bị yêu cầu nhập câu hỏi bảo

mật và câu trả lời cho câu hỏi đó, điều khiển PasswordRecovery hiển thị một form chứa đựng câu hỏi bảo mật, nếu bạn không nhập vào đúng câu trả lời bảo mật của bạn, mật khẩu của bạn sẽ không được gửi.

Nếu bạn không muốn người sử dụng phải nhập câu hỏi bảo mật khi khôi phục mật khẩu, bạn có thể chỉnh sửa cấu hình của membership, listing 1.27 sau sẽ gán giá trị là false cho thuộc tính requiresQuestionAndAnswer.

Listing 1.27

Mã:

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <authentication mode="Forms" />
    <membership defaultProvider="MyMembership">
      <providers>
        <add
          name="MyMembership"
          type="System.Web.Security.SqlMembershipProvider"
          connectionStringName="LocalSqlServer"
          requiresQuestionAndAnswer="false" />
      </providers>
    </membership>
  </system.web>
</configuration>
```

Chương 2. Sử dụng ASP.NET MemberShip

Trong chương trước, bạn đã được học cách sử dụng điều khiển login để tạo form đăng ký người sử dụng với hệ thống. Trong chương này chúng ta cùng khám phá và giải nghĩa bảo mật framework trên các điều khiển Login.

ASP.NET Framework bao gồm 4 khung quan hệ bảo mật:

- ASP.NET Authentication: Cho phép định nghĩa người sử dụng.
- ASP.NET Authorization: Cho phép bạn ủy nhiệm quyền truy xuất dữ liệu cho người sử dụng.
- ASP.NET Membership: cho phép bạn diễn tả người sử dụng và chỉnh sửa các thuộc tính nó.
- Role Manager: Đưa ra vai trò của người sử dụng và chỉnh sửa các thuộc tính của nó.

2.1 Cấu hình Authentication

Ứng dụng chỉ dẫn xử lý xác định bạn là ai. ASP.NET Framework hỗ trợ 3 kiểu của xác thực.

- Windows Authentication
- .NET Passport Authentication
- Forms Authentication

Một ứng dụng riêng chỉ có thể áp dụng một kiểu xác thực. bạn không thể áp dụng đồng thời nhiều kiểu.

Mặc định Windows authentication được cho phép, Khi windows authentication được cho

phép các tên tài khoản Microsoft Windows của họ. Vai trò phù hợp với nhóm Microsoft Windows.

Windows authentication ủy quyền chịu trách nhiệm định danh người sử dụng trên IIS. IIS có thể sử dụng cấu hình Basic, Integrated Windows, hoặc Digest authentication.

Kiểm chứng .NET Passport giống với kiểm chứng ở website của Microsofts như là MSN hay Hotmail. Nếu bạn muốn người sử dụng đăng nhập trong ứng dụng của bạn bởi các tài khoản Hotmail đã tồn tại, bạn có thể cho phép kiểm chứng .Net PassPort.

Cuối cùng là kiểu kiểm chứng Form Authentication. Khi Form Authentication được cho phép, Các người sử dụng được định nghĩa bởi một cookie. Khi người sử dụng được kiểm chứng. Một cookie mã hóa được thêm vào trình duyệt của người sử dụng.

Khi Form Authentication được cho phép, người sử dụng và và thông tin vai trò được lưu trữ trong một kho dữ liệu tùy biến. Ví dụ bạn có thể lưu trữ tên người sử dụng và mật khẩu trong một file XML, database, hay một file Text cơ bản.

2.1.1 Cấu hình Form Authentication

Các lựa chọn cấu hình riêng được chỉ định đến Form Authentication

- Cookieless: Cho phép bạn sử dụng sự kiện Form authentication khi trình duyệt không hỗ trợ Cookie, có thể là các giá trị: UseCookies, UseUri, AutoDetect, và UseDeviceProfile. Mặc định giá trị là UseDeviceProfile.
- defaultUrl: Cho phép bạn chỉ định trang mà sau khi người sử dụng được kiểm chứng chuyển tới. mặc định là giá trị Default.aspx.
- domain: cho phép bạn chỉ định domain được kết hợp mới kiểm chứng Cookie, giá trị mặc định là rỗng.
- enableCrossAppRedirects: Cho phép người sử dụng kiểm chứng qua ứng dụng bằng cách thẻ xác thực trong một chuỗi truy vấn.. Giá trị mặc định là fasle.
- loginUrl: Cho phép bạn chỉ định đường dẫn tới trang Login. Giá trị mặc định là Login.aspx
- name: Cho phép bạn chỉ định tên của cookie kiểm chứng. giá trị mặc định là .ASPXAUTH.
- path: Cho phép bạn chỉ định đường dẫn kết hợp với cookie kiểm chứng mặc định giá trị là /.
- Protection: cho phép bạn chỉ định cookie kiểm chứng được mã hóa như thế nào. Giá trị có thể là All, Encryption, None và Validation, giá trị mặc định là All.
- requiresSSL: Cho phép bạn yêu cầu một SSL(Secure Sockets Layer) kết nối khi truyền cookie kiểm chứng. mặc định giá trị là false.
- slidingExpiration: Cho phép bạn ngăn chặn cookie xác thực hết hạn như là người sử dụng tiếp tục tạo một yêu cầu trong một khoảng thời gian, có thể có giá trị là False hoặc True, mặc định là Fasle.
- timeout: Cho phép bạn chỉ định một lượng thời gian hết hạn của cookie xác thực tính bởi phút. Giá trị mặc định là 30.

Ví dụ sau sẽ thay đổi tên của cookie authentication.

Listing 2.1 web1.config

Mã:

```
<?xml version="1.0"?>
<configuration>
    <appSettings/>
```

```

<connectionStrings/>
<system.web>
    <authentication mode="Forms" >
        <forms name="MyApp" />
    </authentication>
</system.web>
</configuration>

```

2.1.2 Sử dụng kiểm chứng Cookieless Forms.

Bình thường, kiểm chứng Form sử dụng một cookie để xác định người sử dụng, tuy nhiên Forms authentication hỗ trợ một thuộc tính đặt tên là cookieless authentication. Khi cookieless authentication được cho phép, một người sử dụng có thể được định danh ngoài cookie của trình duyệt.

Bởi việc thêm vào kiểm chứng cookieless, bạn có thể sử dụng Forms Authentication và ASP.NET Membership để kiểm chứng người sử dụng, một người sử dụng có thể được định nghĩa bởi một thẻ duy nhất được thêm vào địa chỉ URL. Nếu người sử dụng sử dụng các URL quan hệ tới đường dẫn từ trang này tới trang khác, sau đó thẻ này được truyền qua giữa các trang tự động và người sử dụng có thể được định danh trên nhiều trang.

Khi bạn gọi một trang mà yêu cầu xác thực và xác thực cookieless được cho phép, địa chỉ URL trên trình duyệt nhìn giống như sau:

[http://localhost:2500/Original/\(F\(WfAne ... VlIOKdQkRk tOqV7cfcrgrUJ2NKxNhH9dTA7fgzZ-cZwyr4ojyU6EnarC-bbf8g4sl6m4k5kk6Nmcsq1\)\) /SecretFiles/Secret2.aspx](http://localhost:2500/Original/(F(WfAne ... VlIOKdQkRk tOqV7cfcrgrUJ2NKxNhH9dTA7fgzZ-cZwyr4ojyU6EnarC-bbf8g4sl6m4k5kk6Nmcsq1)) /SecretFiles/Secret2.aspx)

Bạn cấu hình kiểm chứng cookieless bởi việc gán một giá trị của thành phần form trong file web.config. thuộc tính cookieless chấp nhận một vài thuộc tính sau:

- UseCookies: luôn luôn sử dụng cookie xác thực.
- UseUri: Không bao giờ sử dụng cookie xác thực.
- AutoDetect: tự động phát hiện để sử dụng cookie xác thực.
- UseDeviceProfile: sử dụng profile để định rõ khi nào để sử dụng cookie xác thực.

Mặc định là giá trị UseDeviceProfile. Bởi mặc định ASP.NET Framework là một cookie chỉ khi nào một kiểu riêng của thiết bị hỗ trợ cookie. ASP.NET Framework duy trì một cơ sở dữ liệu tùy thuộc khả năng thiết bị trong thiết lập của các file chakra đựng theo đường dẫn sau:

\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CONFIG\Browsers

Bởi mặc định ASP.NET Framework không bao giờ sử dụng xác thực cookieless với một trình duyệt như là IE. Nếu bạn muốn ASP.NET Framework tự động phát hiện trình duyệt có hỗ trợ cookie hay không thì bạn thiết lập thuộc tính cookieless có giá trị là AutoDetect.

Listing 2.2

Mã:

```

<configuration>
    <system.web>
        <authentication mode="Forms" >
            <forms cookieless="AutoDetect"></forms>

```

```
</authentication>
</system.web>
</configuration>
```

2.1.3 Sử dụng sự hết hạn trượt với Forms Authentication

Bởi mặc định Forms Authentication sử dụng các giải quyết trượt hết hạn, Người sử dụng không yêu cầu trang trong vòng 30 phút, trang sẽ tự động đăng xuất.

Nếu bạn có yêu cầu bảo mật đúng đắn, bạn có thể sử dụng giải pháp thời hạn tuyệt đối thay cho trượt thời hạn. Trong trường hợp này, bạn có thể hiệu lực một người sử dụng đăng nhập lại sau một khoảng thời gian riêng.

Listing 2.3

Mã:

```
<?xml version="1.0"?>
<configuration>
  <appSettings/>
  <connectionStrings/>
  <system.web>
    <authentication mode="Forms" >
      <forms slidingExpiration="false" timeout="1" />
    </authentication>
  </system.web>
</configuration>
```

2.1.4 Sử dụng Forms authentication ngang qua ứng dụng

Trong phần trước bạn được học chia sẻ xác thực cookie qua các ứng dụng khác nhau trong cùng server hoặc server khác nhau, trong phần này bạn sẽ học cách chia sẻ xác thực cookie qua như domain.

Một cookie trình duyệt luôn quan hệ với domain, Ví dụ website Amazon không thể đọc cookie thiết lập bởi website itechpro hoặc vietnamnet. Tuy nhiên bạn có thể khám phá rằng bạn cần chia sẻ thông tin xác thực qua nhiều website với nhiều domain khác nhau. Bạn có thể làm việc quanh vấn đề này bởi việc truyền thẻ xác thực trong một chuỗi truy vấn hơn là trong một cookie Không có gì ngăn cản bạn truyền tham số qua các domain khác nhau.

Để có thể cho phép trong ngữ cảnh này, bạn phải cấu hình ứng dụng của bạn chấp nhận thẻ xác thực được truyền trong chuỗi truy vấn, như ví dụ sau:

Listing 2.4 web4.config

Mã:

```
<configuration>
  <system.web>
    <authentication mode="Forms" >
      <forms enableCrossAppRedirects="true" />
    </authentication>
    <machineKey decryption="AES" validation="SHA1"
decryptionKey="306C1FA852AB3B0115150DD8BA30821CDFD125538A0C606DAC
```

```

A53DBB3C3E0AD2"
validationKey="61A8E04A146AFFAB81B6AD19654F99EA7370807F18F5002725DA
B98B8EFD19C711337E26948E26D1D174B159973EA0BE8CC9CAA6AAF513BF84E
44B2247792265" />
</system.web>
</configuration>

```

Nếu bạn cấu hình như Listing 2.4 cho phép hai ứng dụng khác nhau định vị trên các domain khác nhau, hai ứng dụng khác nhau có thể chia sẻ token kiểm chứng khác nhau. Khi bạn liên kết hoặc chuyển sang trang từ một trang khác bạn phải truyền token kiểm chứng này trong chuỗi tham số truy vấn

Listing 2.5 QueryStringAuthenticate.aspx

Mã:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="QueryStringAuthenticate.aspx.cs" Inherits="QueryStringAuthenticate" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
    void Page_Load()
    {
        string cookieName = FormsAuthentication.FormsCookieName;
        string cookieValue = FormsAuthentication.GetAuthCookie(User.Identity.Name,
false).Value;
        lnkOtherDomain.NavigateUrl += String.Format("?{0}={1}", cookieName,
cookieValue);
    }
</script>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:HyperLink id="lnkOtherDomain" Text="liên kết tới Domain khác"
NavigateUrl="http://www.OtherDomain.com/Secret.aspx" Runat="server" />
        </div>
    </form>
</body>
</html>

```

2.1.5 Sử dụng Lớp FormsAuthentication

- Giao tiếp lập trình ứng dụng chính cho tương tác với kiểm chứng Forms là lớp FormsAuthentication Lớp này hỗ trợ các thuộc tính sau:
- CookieDomain: Trả về domain kết hợp với cookie xác thực
 - CookieMode: Trả về kiểu xác thực cookieless. Có thể là các giá trị: AutoDetect, UseCookies, UseDeviceProfile, and UseUri.
 - CookiesSupported: Trả về đúng khi trình duyệt hỗ trợ cookie và xác thực Forms được cấu hình để sử dụng cookies.
 - DefaultUrl: Trả về URL của trang mà người sử dụng được chuyển tới sau khi được kiểm chứng.
 - EnableCrossAppRedirects: Trả về true khi thẻ kiểm chứng có thể gỡ bỏ từ chuỗi truy vấn
 - FormsCookieName: trả về tên của cookie xác thực
 - FormsCookiePath: Trả về đường dẫn kết hợp với cookie kiểm chứng.
 - LoginUrl: trả về URL của trang mà người sử dụng được chuyển tới khi sẽ được kiểm chứng.
 - RequireSSL: Trả về đúng khi cookie kiểm chứng phải được truyền thông qua SSL.
 - SlidingExpiration: Trả về True khi cookie kiểm chứng sử dụng chính sách trượt quá hạn.

Các thuộc tính này trả về các thiết lập cấu hình trong file web.config
Lớp FormsAuthentication hỗ trợ các phương thức sau:

- Authenticate: Cho phép bạn kiểm tra lại UserName và Password dựa vào một danh sách UserName và Password được lưu trữ trong file web.config.
- Decrypt: cho phép bạn giải mã một cookie xác thực
- GetAuthCookie: Cho phép bạn lấy thông tin cookie xác thực.
- GetRedirectUrl: Cho phép bạn lấy thông tin đường dẫn trang ban đầu gây ra chuyển tới trang Login.aspx.
- HashPasswordForStoringInConfigFile: Cho phép bạn lưu trữ một mật khẩu mà nó có thể được lưu trữ trong file web.config.
- RedirectFromLoginPage: Cho phép bạn chuyển người sử dụng quay trở lại trang ban đầu được yêu cầu trước khi người sử dụng được chuyển tới trang Login.aspx.
- RedirectToLoginPage: cho phép chuyển người sử dụng tới trang Login.aspx
- RenewTicketIfOld: Cho phép bạn cập nhật thời gian hết hạn của cookie kiểm chứng.
- SetAuthCookie: Cho phép bạn tạo và đưa ra một cookie kiểm chứng.
- SignOut: Cho phép bạn gỡ bỏ một cookie kiểm chứng và đăng xuất người sử dụng.

Bạn có thể sử dụng các phương thức và thuộc tính của lớp FormsAuthentication để xây dựng người đăng ký sử dụng và hệ thống kiểm chứng ngoài việc sử dụng ASP.NET Membership. Ví dụ Listing 2.6 chứa đựng một danh sách tên sử dụng và mật khẩu

Listing 2.6 web6.config

Mã:

```
<configuration>
  <system.web>
    <authentication mode="Forms">
      <forms>
        <credentials passwordFormat="Clear">
          <user name="Bill" password="secret" />
```

```

<user name="Jane" password="secret" />
<user name="Fred" password="secret" />
</credentials>
</forms>
</authentication>
</system.web>
</configuration>

```

Listing 2.6 chứa đựng thành phần forms mà chưa đựng thành phần credentials. credentials bao gồm một danh sách UserName và Password.

Chú ý rằng thành phần credentials chứa một thuộc tính PasswordFormat mà được thiết lập với giá trị Clear, Nếu bạn thích lưu trữ mật khẩu trong Text hơn bạn có thể lưu trữ mật khẩu trong các giá trị hash, Với con đường đó thì bất cứ ai trên webserver không thể nhìn thấy mật khẩu của người khác. Trường hợp 2 giá trị của PasswordFormat có thể là MD5 và SHA1.

Listing 2.7 FormsLogin.aspx

Mã:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="FormsLogin.aspx.cs"
Inherits="FormsLogin" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
    protected void btnLogin_Click(object sender, EventArgs e)
    {
        if (FormsAuthentication.Authenticate(txtUserName.Text,txtPassword.Text))
            FormsAuthentication.RedirectFromLoginPage(txtUserName.Text,
chkRememberMe.Checked);
        else
            lblError.Text = "Invalid user name/password";
    }
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Đăng nhập hệ thống</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label id="lblError" EnableViewState="false" ForeColor="Red"
Runat="server" />
            <br /><br />
            <asp:Label id="lblUserName" Text="User Name:" 
AssociatedControlID="txtUserName" Runat="server" />
            <br />
        </div>
    </form>
</body>
</html>

```

```

<asp:TextBox id="txtUserName" Runat="server" />
<br /><br />
<asp:Label id="lblPassword" Text="Password:" 
AssociatedControlID="txtPassword" Runat="server" />
<br />
<asp:TextBox id="txtPassword" TextMode="Password" Runat="server" />
<br /><br />
<asp:CheckBox id="chkRememberMe" Text="Remember Me" Runat="server" />
<br /><br />
<asp:Button id="btnLogin" Text="Login" OnClick="btnLogin_Click"
Runat="server" />
</div>
</form>
</body>
</html>

```

Khi bạn nhấn vào nút Button Login, hàm btnLogin_Click() được thực thi và phương thức FormsAuthentication.Authenticate() được sử dụng để kiểm tra tên sử dụng và mật khẩu nhập trong Textbox có trong file web.config không. Nếu người sử dụng xác thực thành công thì phương thức FormsAuthentication.RedirectFromLoginPage() được gọi.

Phương thức RedirectFromLoginPage() làm hai việc: thêm một cookie xác thực vào trình duyệt của người sử dụng và chuyển người sử dụng tới trang đầu tiên bị chuyển sang trang Login.aspx. Nếu người sử dụng yêu cầu trực tiếp trang Login.aspx thì nó sẽ chuyển về trang Default.aspx.

Tham số thứ 2 truyền tới phương thức RedirectFromLoginPage() cho biết có bạn có muốn sử dụng một session hay một persistent cookie hay không. Nếu bạn tạo một persistent cookie thì bạn không cần phải đăng nhập trang web khi bạn trở lại trong một thời gian sau đó.

2.1.6 Sử dụng lớp User

Bạn có thể sử dụng thuộc tính Page.User hoặc HttpContext.User để lấy thông tin về người sử dụng hiện tại. Thuộc tính Page.User đưa ra một đối tượng Principal mà hỗ trợ phương thức sau:

IsInRole: Cho phép kiểm tra người sử dụng có phải là một thành viên của Role riêng hay không.

Ví dụ Khi Windows Authentication được cho phép, bạn có thể sử dụng phương thức IsInRole để kiểm tra người sử dụng có phải là thành viên của nhóm riêng trong MS Windows như là nhóm BUILTIN\Administrators hay không?

```
if(User.IsInRole("BUILTIN\Administrators"))
```

```
{
// thực hiện công việc của quản trị viên hệ điều hành
}
```

Đối tượng Principal chỉ bao gồm một thuộc tính Identity cho phép bạn lấy thông tin về đặc tính của người sử dụng hiện tại. Đối tượng Identity hỗ trợ ba thuộc tính sau:

AuthenticationType: cho phép bạn xác định người sử dụng được kiểm chứng như thế nào có thể là các giá trị: Forms, Basic, và NTLM.

IsAuthenticated: cho phép bạn xác định người sử dụng có được kiểm chứng hay không.
Name: cho phép lấy thông tin tên của người sử dụng.

Chương 2. Sử dụng ASP.NET Membership

Trong chương trước, bạn đã được học cách sử dụng điều khiển login để tạo form đăng ký người sử dụng với hệ thống. Trong chương này chúng ta cùng khám phá và giải nghĩa bảo mật framework trên các điều khiển Login.

ASP.NET Framework bao gồm 4 khung quan hệ bảo mật:

- ASP.NET Authentication: Cho phép định nghĩa người sử dụng.
- ASP.NET Authorization: Cho phép bạn ủy nhiệm quyền truy xuất dữ liệu cho người sử dụng.
- ASP.NET Membership: cho phép bạn diễn tả người sử dụng và chỉnh sửa các thuộc tính nó.
- Role Manager: Đưa ra vai trò của người sử dụng và chỉnh sửa các thuộc tính của nó.

2.1 Cấu hình Authentication

Ứng dụng chỉ dẫn xử lý xác định bạn là ai. ASP.NET Framework hỗ trợ 3 kiểu của xác thực.

- Windows Authentication
- .NET Passport Authentication
- Forms Authentication

Một ứng dụng riêng chỉ có thể áp dụng một kiểu xác thực. bạn không thể áp dụng đồng thời nhiều kiểu.

Mặc định Windows authentication được cho phép, Khi windows authentication được cho phép các tên tài khoản Microsoft Windows của họ. Vai trò phù hợp với nhóm Microsoft Windows.

Windows authentication ủy quyền chịu trách nhiệm định danh người sử dụng trên IIS. IIS có thể sử dụng cấu hình Basic, Integrated Windows, hoặc Digest authentication.

Kiểm chứng .NET Passport giống với kiểm chứng ở website của Microsofts như là MSN hay Hotmail. Nếu bạn muốn người sử dụng đăng nhập trong ứng dụng của bạn bởi các tài khoản Hotmail đã tồn tại, bạn có thể cho phép kiểm chứng .Net PassPort.

Cuối cùng là kiểu kiểm chứng Form Authentication. Khi Form Authentication được cho phép, Các người sử dụng được định nghĩa bởi một cookie. Khi người sử dụng được kiểm chứng. Một cookie mã hóa được thêm vào trình duyệt của người sử dụng.

Khi Form Authentication được cho phép, người sử dụng và và thông tin vai trò được lưu trữ trong một kho dữ liệu tùy biến. Ví dụ bạn có thể lưu trữ tên người sử dụng và mật khẩu trong một file XML, database, hay một file Text cơ bản.

2.1.1 Cấu hình Form Authentication

Các lựa chọn cấu hình riêng được chỉ định đến Form Authentication

- Cookieless: Cho phép bạn sử dụng sự kiện Form authentication khi trình duyệt không hỗ trợ Cookie, có thể là các giá trị: UseCookies, UseUri, AutoDetect, và UseDeviceProfile. Mặc định giá trị là UseDeviceProfile.
- defaultUrl: Cho phép bạn chỉ định trang mà sau khi người sử dụng được kiểm chứng chuyển tới. mặc định là giá trị Default.aspx.
- domain: cho phép bạn chỉ định domain được kết hợp mới kiểm chứng Cookie, giá trị

mặc định là rỗng.

- enableCrossAppRedirects: Cho phép người sử dụng kiểm chứng qua ứng dụng bằng cách thẻ xác thực trong một chuỗi truy vấn.. Giá trị mặc định là fasle.
- loginUrl: Cho phép bạn chỉ định đường dẫn tới trang Login. Giá trị mặc định là Login.aspx
- name: Cho phép bạn chỉ định tên của cookie kiểm chứng. giá trị mặc định là .ASPXAUTH.
- path: Cho phép bạn chỉ định đường dẫn kết hợp với cookie kiểm chứng mặc định giá trị là /.
- Protection: cho phép bạn chỉ định cookie kiểm chứng được mã hóa như thế nào. Giá trị có thể là All, Encryption, None và Validation, giá trị mặc định là All.
- requiresSSL: Cho phép bạn yêu cầu một SSL(Secure Sockets Layer) kết nối khi truyền cookie kiểm chứng. mặc định giá trị là false.
- slidingExpiration: Cho phép bạn ngăn cản cookie xác thực hết hạn như là người sử dụng tiếp tục tạo một yêu cầu trong một khoảng thời gian, có thể có giá trị là False hoặc True, mặc định là Fasle.
- timeout: Cho phép bạn chỉ định một lượng thời gian hết hạn của cookie xác thực tính bởi phút. Giá trị mặc định là 30.

Ví dụ sau sẽ thay đổi tên của cookie authentication.

Listing 2.1 web1.config

Mã:

```
<?xml version="1.0"?>
<configuration>
  <appSettings/>
  <connectionStrings/>
  <system.web>
    <authentication mode="Forms" >
      <forms name="MyApp" />
    </authentication>
  </system.web>
</configuration>
```

2.1.2 Sử dụng kiểm chứng Cookieless Forms.

Bình thường, kiểm chứng Form sử dụng một cookie để xác định người sử dụng, tuy nhiên Forms authentication hỗ trợ một thuộc tính đặt tên là cookieless authentication. Khi cookieless authentication được cho phép, một người sử dụng có thể được định danh ngoài cookie của trình duyệt.

Bởi việc thêm vào kiểm chứng cookieless, bạn có thể sử dụng Forms Authentication và ASP.NET Membership để kiểm chứng người sử dụng, một người sử dụng có thể được định nghĩa bởi một thẻ duy nhất được thêm vào địa chỉ URL. Nếu người sử dụng sử dụng các URL quan hệ tới đường dẫn từ trang này tới trang khác, sau đó thẻ này được truyền qua giữa các trang tự động và người sử dụng có thể được định danh trên nhiều trang. Khi bạn gọi một trang mà yêu cầu xác thực và xác thực cookieless được cho phép, địa chỉ URL trên trình duyệt nhìn giống như sau:

[http://localhost:2500/Original/\(F\(WfAne ... VlIOKdQkRk](http://localhost:2500/Original/(F(WfAne ... VlIOKdQkRk)

tOqV7cfcrUJ2NKxNhH9dTA7fgzZ-cZwyr4ojyU6EnarC-bbf8g4sl6m4k5kk6Nmcs1))
/SecretFiles/Secret2.aspx

Bạn cấu hình kiểm chứng cookieless bởi việc gán một giá trị của thành phần form trong file web.config, thuộc tính cookieless chấp nhận một vài thuộc tính sau:

- UseCookies: luôn luôn sử dụng cookie xác thực.
- UseUri: Không bao giờ sử dụng cookie xác thực.
- AutoDetect: tự động phát hiện để sử dụng cookie xác thực.
- UseDeviceProfile: sử dụng profile để định rõ khi nào để sử dụng cookie xác thực.

Mặc định là giá trị UseDeviceProfile. Bởi mặc định ASP.NET Framework là một cookie chỉ khi nào một kiểu riêng của thiết bị hỗ trợ cookie. ASP.NET Framework duy trì một cơ sở dữ liệu tùy thuộc khả năng thiết bị trong thiết lập của các file chưa đựng theo đường dẫn sau:

\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CONFIG\Browsers

Bởi mặc định ASP.NET Framework không bao giờ sử dụng xác thực cookieless với một trình duyệt như là IE. Nếu bạn muốn ASP.NET Framework tự động phát hiện trình duyệt có hỗ trợ cookie hay không thì bạn thiết lập thuộc tính cookieless có giá trị là AutoDetect.

Listing 2.2

Mã:

```
<configuration>
  <system.web>
    <authentication mode="Forms" >
      <forms cookieless="AutoDetect"></forms>
    </authentication>
  </system.web>
</configuration>
```

2.1.3 Sử dụng sự hết hạn trượt với Forms Authentication

Bởi mặc định Forms Authentication sử dụng các giải quyết trượt hết hạn, Người sử dụng không yêu cầu trang trong vòng 30 phút, trang sẽ tự động đăng xuất.

Nếu bạn có yêu cầu bảo mật đúng đắn, bạn có thể sử dụng giải pháp thời hạn tuyệt đối thay cho trượt thời hạn. Trong trường hợp này, bạn có thể hiệu lực một người sử dụng đăng nhập lại sau một khoảng thời gian riêng.

Listing 2.3

Mã:

```
<?xml version="1.0"?>
<configuration>
  <appSettings/>
  <connectionStrings/>
  <system.web>
    <authentication mode="Forms" >
      <forms slidingExpiration="false" timeout="1" />
    </authentication>
  </system.web>
</configuration>
```

```
</system.web>  
</configuration>
```

2.1.4 Sử dụng Forms authentication ngang qua ứng dụng

Trong phần trước bạn được học chia sẻ xác thực cookie qua các ứng dụng khác nhau trong cùng server hoặc server khác nhau, trong phần này bạn sẽ học cách chia sẻ xác thực cookie qua nhu domain.

Một cookie trình duyệt luôn quan hệ với domain, Vi dụ website Amazon không thể đọc cookie thiết lập bởi website itechpro hoặc vietnamnet. Tuy nhiên bạn có thể khám phá rằng bạn cần chia sẻ thông tin xác thực qua nhiều website với nhiều domain khác nhau. Bạn có thể làm việc quanh vấn đề này bởi việc truyền thẻ xác thực trong một chuỗi truy vấn hơn là trong một cookie Không có gì ngăn cản bạn truyền tham số qua các domain khác nhau.

Để có thể cho phép trong ngữ cảnh này, bạn phải cấu hình ứng dụng của bạn chấp nhận thẻ xác thực được truyền trong chuỗi truy vấn, như ví dụ sau:

Listing 2.4 web4.config

Mã:

```
<configuration>  
  <system.web>  
    <authentication mode="Forms" >  
      <forms enableCrossAppRedirects="true" />  
    </authentication>  
    <machineKey decryption="AES" validation="SHA1"  
      decryptionKey="306C1FA852AB3B0115150DD8BA30821CDFD125538A0C606DAC  
      A53DBB3C3E0AD2"  
      validationKey="61A8E04A146AFFAB81B6AD19654F99EA7370807F18F5002725DA  
      B98B8EFD19C711337E26948E26D1D174B159973EA0BE8CC9CAA6AAF513BF84E  
      44B2247792265" />  
    </system.web>  
  </configuration>
```

Nếu bạn cấu hình như Listing 2.4 cho phép hai ứng dụng khác nhau định vị trên các domain khác nhau, hai ứng dụng khác nhau có thể chia sẻ thẻ kiểm chứng khác nhau. Khi bạn liên kết hoặc chuyển san trang từ một trang khác bạn phải truyền thẻ kiểm chứng này trong chuỗi tham số truy vấn

Listing 2.5 QueryStringAuthenticate.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeFile="QueryStringAuthenticate.aspx.cs" Inherits="QueryStringAuthenticate" %>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<script runat="server">
    void Page_Load()
    {
        string cookieName = FormsAuthentication.FormsCookieName;
        string cookieValue = FormsAuthentication.GetAuthCookie(User.Identity.Name,
false).Value;
        lnkOtherDomain.NavigateUrl += String.Format("{0}={1}", cookieName,
cookieValue);
    }
</script>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:HyperLink id="lnkOtherDomain" Text="liên kết tới Domain khác"
NavigateUrl="http://www.OtherDomain.com/Secret.aspx" Runat="server" />
        </div>
    </form>
</body>
</html>

```

2.1.5 Sử dụng Lớp FormsAuthentication

Giao tiếp lập trình ứng dụng chính cho tương tác với kiểm chứng Forms là lớp FormsAuthentication. Lớp này hỗ trợ các thuộc tính sau:

- CookieDomain: Trả về domain kết hợp với cookie xác thực
- CookieMode: Trả về kiểu xác thực cookieless. Có thể là các giá trị: AutoDetect, UseCookies, UseDeviceProfile, and UseUri.
- CookiesSupported: Trả về đúng khi trình duyệt hỗ trợ cookie và xác thực Forms được cấu hình để sử dụng cookies.
- DefaultUrl: Trả về URL của trang mà người sử dụng được chuyển tới sau khi được kiểm chứng.
- EnableCrossAppRedirects: Trả về true khi thẻ kiểm chứng có thể gỡ bỏ từ chuỗi truy vấn
- FormsCookieName: trả về tên của cookie xác thực
- FormsCookiePath: Trả về đường dẫn kết hợp với cookie kiểm chứng.
- LoginUrl: trả về URL của trang mà người sử dụng được chuyển tới khi sẽ được kiểm chứng.
- RequireSSL: Trả về đúng khi cookie kiểm chứng phải được truyền thông với SSL.
- SlidingExpiration: Trả về True khi cookie kiểm chứng sử dụng chính sách trượt quá hạn.

Các thuộc tính này trả về các thiết lập cấu hình trong file web.config
Lớp FormsAuthentication hỗ trợ các phương thức sau:

- Authenticate: Cho phép bạn kiểm tra lại UserName và Password dựa vào một danh sách UserName và Password được lưu trữ trong file web.config.
 - Decrypt: cho phép bạn giải mã một cookie xác thực
 - GetAuthCookie: Cho phép bạn lấy thông tin cookie xác thực.
 - GetRedirectUrl: Cho phép bạn lấy thông tin đường dẫn trang ban đầu gây ra chuyển tới trang Login.aspx.
 - HashPasswordForStoringInConfigFile: Cho phép bạn lưu trữ một mật khẩu mà nó có thể được lưu trữ trong file web.config.
 - RedirectFromLoginPage: Cho phép bạn chuyển người sử dụng quay trở lại trang ban đầu được yêu cầu trước khi người sử dụng được chuyển tới trang Login.aspx.
 - RedirectToLoginPage: cho phép chuyển người sử dụng tới trang Login.aspx
 - RenewTicketIfOld: Cho phép bạn cập nhật thời gian hết hạn của cookie kiểm chứng.
 - SetAuthCookie: Cho phép bạn tạo và đưa ra một cookie kiểm chứng.
 - SignOut: Cho phép bạn gỡ bỏ một cookie kiểm chứng và đăng xuất người sử dụng.
- Bạn có thể sử dụng các phương thức và thuộc tính của lớp FormsAuthentication để xây dựng người đăng ký sử dụng và hệ thống kiểm chứng ngoài việc sử dụng ASP.NET Membership. Ví dụ Listing 2.6 chứa đựng một danh sách tên sử dụng và mật khẩu

Listing 2.6 web6.config

Mã:

```
<configuration>
  <system.web>
    <authentication mode="Forms">
      <forms>
        <credentials passwordFormat="Clear">
          <user name="Bill" password="secret" />
          <user name="Jane" password="secret" />
          <user name="Fred" password="secret" />
        </credentials>
      </forms>
    </authentication>
  </system.web>
</configuration>
```

Listing 2.6 chứa đựng thành phần forms mà chứa đựng thành phần credentials. credentials bao gồm một danh sách UserName và Password.

Chú ý rằng thành phần credentials chứa một thuộc tính PasswordFormat mà được thiết lập với giá trị Clear. Nếu bạn thích lưu trữ mật khẩu trong Text hơn bạn có thể lưu trữ mật khẩu trong các giá trị hash. Với con đường đó thì bất cứ ai trên webserver không thể nhìn thấy mật khẩu của người khác. Trường hợp 2 giá trị của PasswordFormat có thể là MD5 và SHA1.

Listing 2.7 FormsLogin.aspx

Mã:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="FormsLogin.aspx.cs"
  Inherits="FormsLogin" %>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
    protected void btnLogin_Click(object sender, EventArgs e)
    {
        if (FormsAuthentication.Authenticate(txtUserName.Text, txtPassword.Text))
            FormsAuthentication.RedirectFromLoginPage(txtUserName.Text,
chkRememberMe.Checked);
        else
            lblError.Text = "Invalid user name/password";
    }
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Đăng nhập hệ thống</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label id="lblError" EnableViewState="false" ForeColor="Red"
Runat="server" />
            <br /><br />
            <asp:Label id="lblUserName" Text="User Name:" AssociatedControlID="txtUserName" Runat="server" />
            <br />
            <asp:TextBox id="txtUserName" Runat="server" />
            <br /><br />
            <asp:Label id="lblPassword" Text="Password:" AssociatedControlID="txtPassword" Runat="server" />
            <br />
            <asp:TextBox id="txtPassword" TextMode="Password" Runat="server" />
            <br /><br />
            <asp:CheckBox id="chkRememberMe" Text="Remember Me" Runat="server" />
            <br /><br />
            <asp:Button id="btnLogin" Text="Login" OnClick="btnLogin_Click"
Runat="server" />
        </div>
    </form>
</body>
</html>

```

Khi bạn nhấn vào nút Button Login, hàm btnLogin_Click() được thực thi và phương thức FormsAuthentication.Authenticate() được sử dụng để kiểm tra tên sử dụng và mật khẩu nhập trong Textbox có trong file web.config không. Nếu người sử dụng xác thực thành công thì phương thức FormsAuthentication.RedirectFromLoginPage() được gọi.

Phương thức RedirectFromLoginPage() làm hai việc: thêm một cookie xác thực vào trình duyệt của người sử dụng và chuyển người sử dụng tới trang đầu tiên bị chuyển sang trang Login.aspx. Nếu người sử dụng yêu cầu trực tiếp trang Login.aspx thì nó sẽ chuyển về trang Default.aspx.

Tham số thứ 2 truyền tới phương thức RedirectFromLoginPage() cho biết có bạn có muốn sử dụng một session hay một persistent cookie hay không. Nếu bạn tạo một persistent cookie thì bạn không cần phải đăng nhập trang web khi bạn trở lại trong một thời gian sau đó.

2.1.6 Sử dụng lớp User

Bạn có thể sử dụng thuộc tính Page.User hoặc HttpContext.User để lấy thông tin về người sử dụng hiện tại. Thuộc tính Page.User đưa ra một đối tượng Principal mà hỗ trợ phương thức sau:

IsInRole: Cho phép kiểm tra người sử dụng có phải là một thành viên của Role riêng hay không.

Ví dụ Khi Windows Authentication được cho phép, bạn có thể sử dụng phương thức IsInRole để kiểm tra người sử dụng có phải là thành viên của nhóm riêng trong MS Windows như là nhóm BUILTIN\Administrators hay không?

```
if (User.IsInRole("BUILTIN\Administrators"))
{
    // thực hiện công việc của quản trị viên hệ điều hành
}
```

Đối tượng Principal chỉ bao gồm một thuộc tính Identity cho phép bạn lấy thông tin về đặc tính của người sử dụng hiện tại. Đối tượng Identity hỗ trợ ba thuộc tính sau:

AuthenticationType: cho phép bạn xác định người sử dụng được kiểm chứng như thế nào có thể là các giá trị: Forms, Basic, và NTLM.

IsAuthenticated: cho phép bạn xác định người sử dụng có được kiểm chứng hay không.
Name: cho phép lấy thông tin tên của người sử dụng.