

## Module : Développement web coté serveur :

### Chapitre 9 : Sécurité(Partie1)

---

#### Authentification par formulaires

L'authentification par formulaires désigne un système dans lequel les demandes non authentifiées sont redirigées vers un formulaire HTML (Hypertext Markup Language), grâce à la redirection HTTP (Hypertext Transfer Protocol) côté client. L'utilisateur envoie ses informations d'identité par le biais d'un formulaire. Si l'application valide les informations d'identification du formulaire, le système envoie un cookie d'authentification à l'utilisateur. Toute demande ultérieure effectuée par celui-ci contiendra le cookie d'authentification dans ses en-têtes, ce qui permettra à l'utilisateur d'être authentifié.

#### 1.1. Configuration de l'authentification par formulaire

On ajoute dans le fichier web.config la section suivant :

```
<authenticationmode="Forms">
<formsloginUrl="loginForm.aspx"defaultUrl="Index.aspx">

</forms>
</authentication>
```

en respectant la forme suivante :

```
<configuration>
<system.web>
<authentication mode="authmode">
</authentication>
</system.web>
</configuration>
```

#### 1.2. Sécurisation des pages

Utilisation de la classe : **FormsAuthentication**

Méthodes :

**RedirectToLoginPage**:redirige un navigateur vers le **LoginUrl** configuré pour permettre aux utilisateurs de se connecter à une application

**RedirectFromLoginPage** : redirige un utilisateur authentifié vers l'URL d'origine qu'il a demandée ou vers le DefaultUrl

Exemple :

```
If ((UserEmail.Text = "jchen@contoso.com") && (UserPass.Text = "37Yj*99Ps"))
FormsAuthentication.RedirectFromLoginPage (UserEmail.Text, false)
```

**SignOut** : Supprime du navigateur le ticket d'authentification par formulaire.

**Authenticate** : Valide un nom d'utilisateur et un mot de passe par rapport aux informations d'identification stockées dans le fichier de configuration d'une application.

```
FormsAuthentication.Authenticate(TextBox1.Text, TextBox2.Text)
```

Remarque : vous pouvez ajouter des utilisateurs dans le fichier de configuration comme suivant :

```
<authentication mode="Forms">
<forms loginUrl="Authentification.aspx" defaultUrl="WebForm1.aspx" >
<credentials passwordFormat="Clear">

<user name="user1" password="user1"></user>
    <user name="user2" password="user2"></user>
</credentials>
</forms>
</authentication>
```

### 1.3. *Sécurisation d'une application complète :*

Pour marquer l'ensemble de l'application comme étant sécurisée, créez une section `<authorization>` dans la section `<system.web>`, comme illustré dans l'exemple de code suivant :

```
<configuration>
<system.web>
<authenticationmode="Forms">
<formsloginUrl="loginForm.aspx">
</forms>
</authentication>
<authorization>
<denyusers="?" />
</authorization>
</system.web>
</configuration>
```

### 1.4. *Sécurisation de pages spécifiques*

Pour marquer uniquement des pages spécifiques comme étant sécurisées, créez dans la section `<configuration>` une section `<location>` avec des sous-sections `<system.web>` et `<authorization>` pour chaque page sécurisée dans votre application Web :

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

```

<authenticationmode="Forms">
<formsloginUrl="loginForm.aspx">
</forms>
</authentication>
<authorization>
<denyusers="?" />
</authorization>
</system.web>

<locationpath="UploadS.aspx">
<system.web>
<authorization>
<allowusers="toto" />
    <denyusers="?" />
</authorization>
</system.web>
</location>
</configuration>

```

## 1.5. Contrôle de Connexion : LoginName, LoginStatut, LoginView

### 1.5.1. LoginStatus

Le LoginStatus est affiche tout simplement un lien "login" si l'utilisateur est anonyme ou un lien "LogOut" si l'utilisateur est connecté.

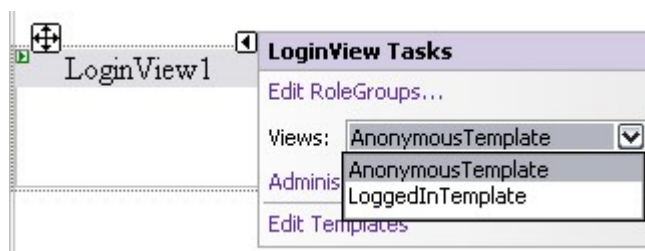
### 1.5.2. LoginName

LoginName, vu dans l'exemple du LoginView, permet de récupérer le Login de l'utilisateur connecté, pratique pour personnaliser un message avec le login pour chaque utilisateur.

### 1.5.3. LoginView :

Le loginView permet de gérer des templates différents:

- le template pour les utilisateurs anonymes
- le template pour les utilisateurs connectés



Dans l'exemple ci-dessous, un utilisateur anonyme verrait le un zone pour se connecter, alors qu'un utilisateur connecté verra "Bonjour [Login]".

```
asp:LoginViewID="LoginView1"runat="server">
<AnonymousTemplate>
    Nom Utilisateur :
    <asp:TextBoxID="TextBox1"runat="server"></asp:TextBox><br/>

    Mot de Passe :
    <asp:TextBoxID="TextBox2"runat="server"TextMode="Password"></asp:TextBox><br/>
    <asp:ButtonID="Button1"runat="server"Text="Se Connecter"/><br/>
    <asp:LoginStatusID="LoginStatus1"
        LogoutPageUrl="~/default.aspx"runat="server"/>
</AnonymousTemplate>
<LoggedInTemplate>
    Bonjour <asp:LoginNameID="LoginName1"
        runat="server"/>!
    <asp:LoginStatusID="LoginStatus1"runat="server"/>
</LoggedInTemplate>
</asp:LoginView>
```

## TP 9 : Partie1

On souhaite réaliser une gestion plus poussée de la sécurité pour le projet GestionFormation,

- 1) Ajouter des enregistrements à la table Utilisateur, utiliser les nom utilisateurs : User1Alaoui, User2Ismail, admin1, admin2
- 2) Les utilisateurs non inscrits ne peuvent pas accéder à notre site
- 3) Les utilisateurs suivants : User1Alaoui, User2Ismail, peuvent seule accéder aux pages AjouterFormation.aspx, FormationLangage.aspx, AjouterLangage.aspx :
- 4) Les utilisateurs dont le nom est admin1, admin2, peuvent seule accéder aux pages AjouterPersonnel.aspx
- 5) Refaire les boutons de déconnexion, utiliser les contrôles serveurs web LoginName, LoginStatut et LoginView