

TP 6 : La gestion d'état

1. La gestion d'état coté serveur :

Il existe deux moyens en ASP.NET pour pouvoir passer des informations entre les pages du côté serveur : *Application State* et *Session State*. Le premier, *Application State*, va permettre de stocker des informations qui seront accessibles dans toutes les pages Web de l'application. Il ne regarde pas quel utilisateur demande l'accès à ces informations, elles sont accessible par tout le monde. En revanche le *Session State* vérifiera l'utilisateur. La session étant spécifique à un utilisateur (avec un identifiant unique), seule la personne ayant le bon identifiant pourra accéder aux données de la session. Là encore chaque page pourra accéder aux informations de la session. Le *Session State* perdra ses informations si on l'arrête ou qu'on le vide. Par contre les deux perdront leurs données si l'application est arrêtée ou redémarrée.

2. Application State

État d'application

L'état d'application est un mécanisme de stockage global qui est accessible à partir de toutes les pages de l'application Web, ce qui le rend très utile pour le stockage des informations utilisateur qui doivent être conservées lors des allers-retours vers le serveur et entre les pages.

Variables d'application

ASP.NET contient les variables d'application (une structure de dictionnaire clé-valeur) nécessaires à la conservation de l'état d'application. Les données idéales à insérer dans les variables d'application sont celles qui sont partagées par plusieurs sessions et qui changent peu.

Syntaxe d'une variable d'application : `Application["NomVariable"]`

3. Session State

État de session

L'état de session est similaire à l'état d'application, excepté qu'il est limité à la session de navigateur en cours. Lorsque plusieurs utilisateurs utilisent une application Web, chacun d'entre eux possède un état de session différent. En outre, lorsqu'un utilisateur quitte une application Web, puis y retourne de nouveau, son état de session est différent de celui qu'il possédait auparavant.

Variables de session

ASP.NET contient les variables de session (une structure de dictionnaire clé-valeur) nécessaires à la conservation de l'état de session. Les données idéales à insérer dans les variables d'état de session sont celles qui sont temporaires, sensibles et spécifiques à une session donnée.

Syntaxe d'une variable de session : Session["NomVariable"]

L'objet Session permet de stocker les informations nécessaires à une session utilisateur spécifique. Les variables stockées dans l'objet Session ne sont pas ignorées lorsque l'utilisateur se déplace dans les pages de l'application Web.

Elles persistent tout au long de la session utilisateur.

Pour améliorer la sécurité de l'application, vous devez autoriser les utilisateurs à se déconnecter de votre application. À ce moment-là, l'application doit appeler la méthode Abandon. Cela réduit la possibilité pour un utilisateur malveillant d'obtenir l'identificateur unique de l'URL et de l'utiliser pour récupérer les données utilisateur privées stockées dans la session.

Durée des variables d'application et de session

Le protocole HTTP est un protocole sans état. Un serveur Web ne dispose d'aucun moyen pour déterminer le moment précis où un utilisateur quitte un site Web. À défaut, il détecte qu'un certain laps de temps s'est écoulé depuis la dernière demande d'une page. Il suppose alors que l'utilisateur a quitté le site Web, et supprime tous les éléments de l'état de session associés à celui-ci.

- Les variables de session ont une durée définie d'après le dernier accès
Par défaut, une session se termine lorsqu'un utilisateur n'a pas demandé de page au cours des 20 dernières minutes. Si un même utilisateur demande une page après ce délai, il est considéré en tant que nouvel utilisateur.
- La durée de la session peut être modifiée dans Web.config :

```
<configuration>
  <system.web>
    <sessionState timeout="10" />
  </system.web>
</configuration>
```

- Les variables d'application persistent jusqu'au déclenchement de l'événement `Application_End`

4. Le fichier Global.asax

Le fichier Global.asax est un fichier déclaratif utilisé pour gérer des événements lors de l'exécution de votre application Web. Ce fichier peut gérer des événements d'application et de session (début et fin) pour initialiser les variables d'application et de session.

Les événements les plus courants qui sont gérés par le fichier Global.asax sont les suivant:

Application_Start Cet événement est levé lorsqu'une application Web ASP.NET démarre.

Application_End Cet événement est également un événement à occurrence unique.

Contrairement à l'événement **Application_Start**, il est levé lorsqu'une application Web ASP.NET s'arrête.

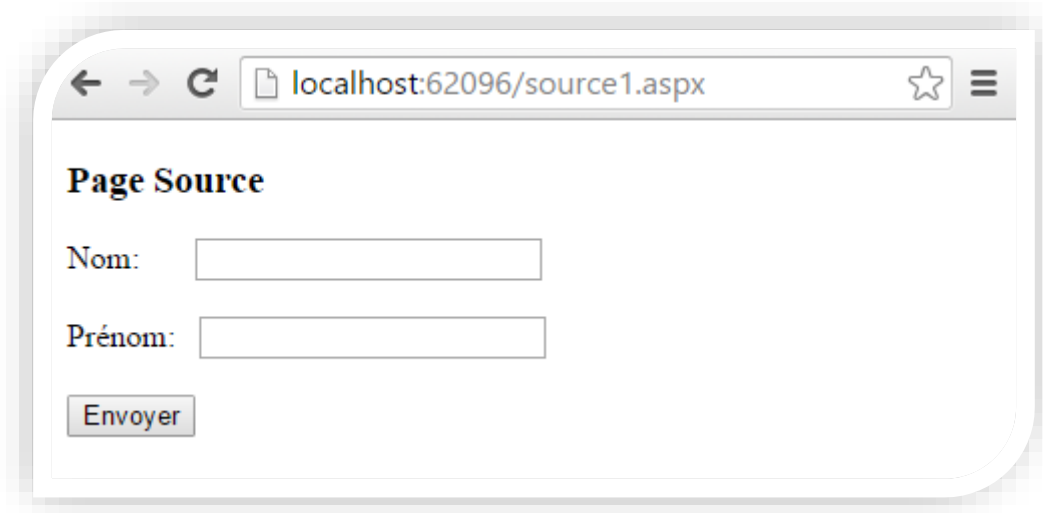
Session_Start Cet événement est levé lorsqu'une session d'un utilisateur est ouverte dans une application Web ASP.NET.

Session_End Contrairement à l'événement **Session_Start**, cet événement est levé lorsqu'une session d'un utilisateur est fermée.

Application_Error Cet événement est déclenché lorsqu'une erreur non gérée se produit dans une application Web ASP.NET.

Atelier 1 : Transférer des informations vers plusieurs pages

1. Créer une page *source1.aspx* contenant le formulaire suivant :



2. Ajoutez le code suivant dans l'événement click du bouton *Envoyer* :

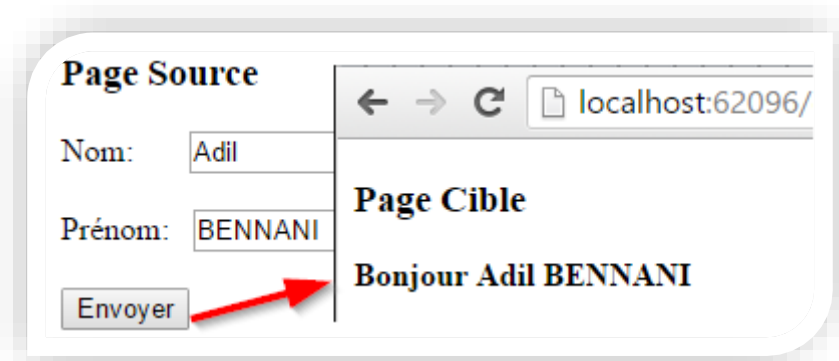
```
protected void Button1_Click(..){
    Session["nom"] = txtNom.Text ;
    Session["prenom"] = txtPrenom.Text ;

    Response.Redirect("cible1.aspx")
}
```

3. Créer une deuxième page *cible1.aspx* content un Label et un contrôle HyperLink
4. Ajouter le code suivant pour récupérer le nom et le prénom dans l'événement Load de la page :

```
protected void Page_Load(..){
    Label1.Text = "Bonjour " + Session["nom"] + " " +Session["prenom"];
}
```

5. Exécuter la page *source1.aspx* et cliquer sur le bouton



6. Créer une troisième page cible2.aspx content un Label
7. Ajouter le code suivant dans l'événement Load de la page :

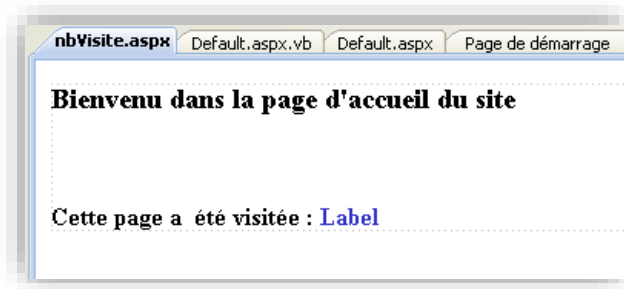
```
protected void Page_Load(..){
Label1.Text = "Bienvenu " + Session["nom"] + " " + Session["prenom"];
}
```

8. Donner la valeur "cible2.aspx" à la propriété NavigateUrl du contrôle HyperLink de la page cible1.aspx
9. Exécuter la page source1.aspx et cliquer sur le bouton
10. Cliquer maintenant sur le lien dans la page cible1.aspx
Qu'est-ce que vous remarquez ?

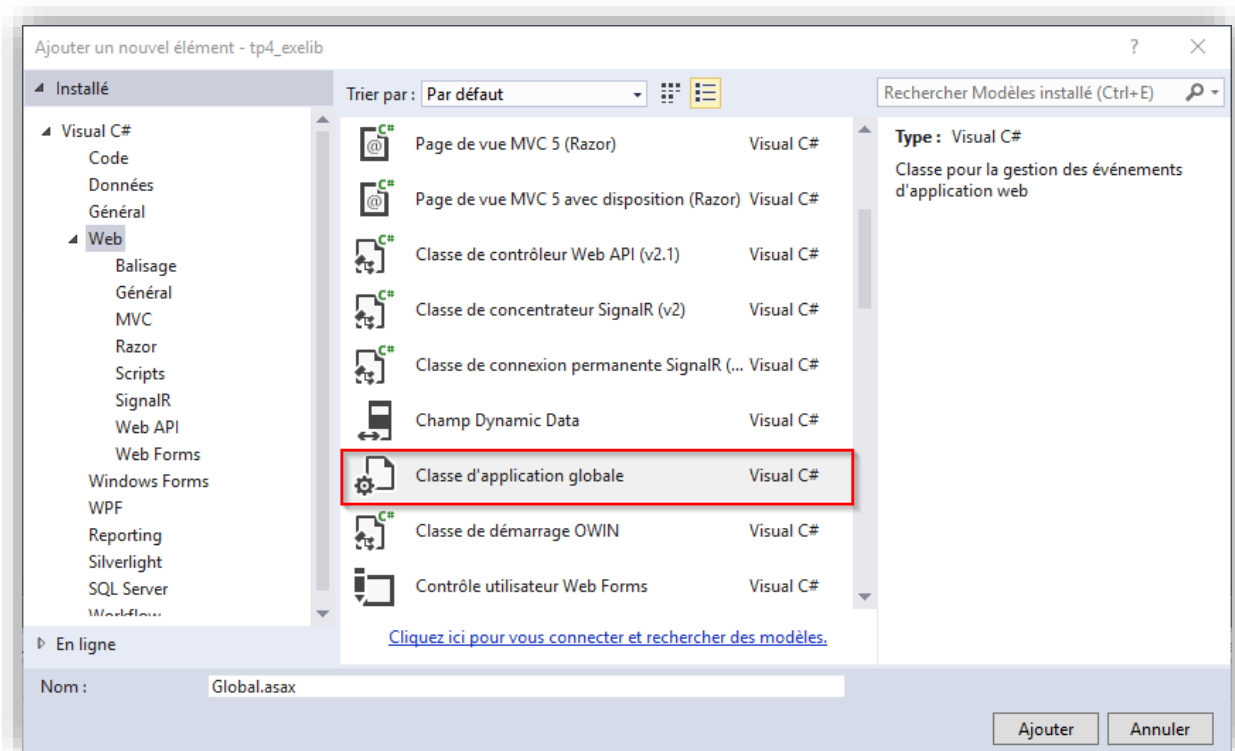
Atelier 2 : Utiliser le fichier Global.asax

Dans cet atelier nous allons créer un compteur de visite d'une page (le nombre de chargement de la page), en utilisant une variable d'application et le fichier Global.asax

1. Ajouter la page suivante nbVisite.aspx dans votre application



2. Dans le menu **Projet**, Cliquer sur **Ajouter un nouvel élément** puis choisir **Classe d'application globale**



3. Analyser le contenu du fichier crée (Global.asax.cs)
4. Ajouter le code suivant dans l'événement Application_Start :

```
protected void Application_Start(..){
    ' Se déclenche lorsque l'application est démarrée
    Application["Visites"] = 0 ;
}
```

5. Dans la procédure d'événement Page_Load du fichier nbVisite.aspx, ajoutez le code qui permet d'incrémenter de 1 la valeur de la variable d'application *Visites* et qui affiche cette variable dans le Label.

```
protected void Page_Load(...){  
    Application.Lock();  
    Application["Visites"] = (int) Application["Visites"] + 1 ;  
    Application.UnLock();  
    Label1.Text = Application["Visites"].ToString() ;  
}
```

Note : *Étant donné que les variables d'application sont accessibles dans un environnement multi-utilisateurs, vous devez empêcher plusieurs utilisateurs ou applications de mettre à jour les données Application simultanément.*

ASP.NET contient un ensemble de méthodes de verrouillage simples, `Application.Lock()` et `Application.UnLock()`, qui peut être utilisé pour autoriser un seul accès à la fois.

6. Compiler le code et naviguer sur la page `nbVisites.aspx`
7. Rafraîchir la page. Qu'est-ce que vous remarquez ?
8. Ouvrir la page `nbVisites.aspx` dans un nouvel onglet (sans compiler la page)
Qu'est-ce que vous remarquez ?

Exercice 1 : Formulaire d'authentification avec sessions

L'objectif de cet exercice est de protéger les pages d'un site en ajoutant une deuxième page secrète à l'exercice Formulaire d'authentification avec redirection.

Créer un formulaire (`login.aspx`) qui demande à l'utilisateur la saisie de son pseudo et son mot de passe et qui appelle la page protégée (`secret1.aspx`).

L'accès au contenu de deux pages `secret1.aspx` et `secret2.aspx` ne sera autorisé que si le pseudo et le mot de passe sont corrects. Donc Il faut tester que la personne qui accède à ces deux pages est bien passée par le formulaire de connexion.

Si l'utilisateur demande les pages secrètes directement, il devra être redirigé vers la page d'authentification. Utilisez la méthode `response.redirect(url_page)` qui permet de rediriger l'utilisateur vers la page spécifiée en paramètre.

Pour ce faire, créer une variable de session après l'authentification et dans les pages secrètes tester si cette variable de session est bien définie.

Afficher dans les pages secrètes un message de bienvenu avec le login de l'utilisateur (Par exemple : *Bienvenu Kamal*, avec Kamal est le login)

Vous devez donc créer 3 pages web :

- **login.aspx** : contient un simple formulaire.
- **secret1.aspx** et **secret2.aspx**: contiennent les informations confidentielles mais ne les affiche que si on lui donne le mot de passe.

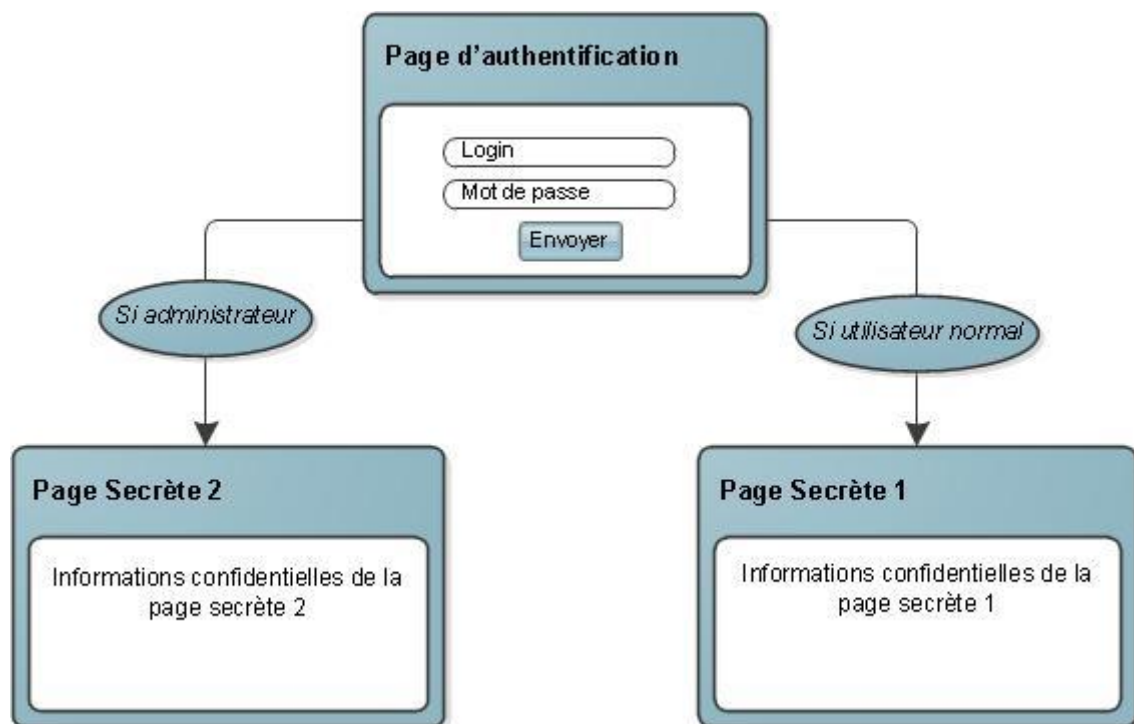
Exercice 2 : Formulaire d'authentification avec compte Administrateur

a) Refaire l'exercice 1 mais cette fois-ci la page secret2.aspx ne sera accessible que par un administrateur.

Un administrateur est un utilisateur qui a le compte : Login=Admin ; Pass=123abc

Donc un utilisateur normal n'est pas autorisé à voir le contenu de la page secret2.aspx

La navigation entre les pages du site sera comme suit :



Remarque : Si l'utilisateur est déjà authentifié et il a accédé à la page login.aspx, on ne doit pas lui afficher le formulaire d'authentification.

b) Ajouter un bouton de déconnexion dans les pages secrètes.

Exercice 3 : Compteur de visiteurs

a) Ajouter dans la page login.aspx un compteur de visiteurs.

Un compteur de visiteurs est un compteur qui compte le nombre de visiteurs qu'un site reçoit. (Les utilisateurs qui ont accédé à la page login.aspx)

b) Modifier le compteur afin d'obtenir un compteur de visiteurs uniques c'est-à-dire le nombre de visiteurs actifs. Une visite est terminée lorsque le visiteur quitte la page