

Travaux pratiques

Série 7

Programmation Web à l'aide de la syntaxe Razor C#

Février

Manipulation d'une base de données SQL Server

Ce Travail pratique décrit comment :

- Utiliser la classe **Database** pour se connecter à une base de données à partir d'un script **Razor** et exécuter des requêtes **SQL** dans les Pages **Web ASP.NET**.
- Accéder aux données de la base de données et les afficher dans les pages **Web ASP.NET**.
- Insérer, supprimer et mettre à jour des enregistrements dans les tables d'une base de données.
- Utiliser l'assistant **WebGrid** pour afficher les données d'une table.

I. Insertion d'un nouveau contact dans la base de données *Carnet.mdf*

Il s'agit dans cette partie de créer une nouvelle page **Web** qui affiche un formulaire permettant aux utilisateurs d'ajouter un nouveau contact à la table **Contactes** du carnet d'adresses représenté par la base de données **Carnet.mdf** créé dans la série 5.

Une fois le nouveau contact sera inséré dans la table **Contactes**, la page affichera le contenu de cette table mise à jour, en redirigeant le script de traitement du formulaire vers la page **ListeContactes.cshtml** créé dans la série 5.

Le script d'action effectuera une validation des données du formulaire pour s'assurer que les données des champs obligatoires dans la table **Contactes** ont bien été saisies par l'utilisateur et sont conformes à la base de données. Par exemple, le nom et le prénom d'un contact qui ne doivent pas être vides.

Insertion de données dans une table

```
INSERT [INTO] table [(champ1, champ2,...)] VALUES ('valeur1', 'valeur2',...), ('valeur1', ...);
```

Permet d'insérer une ou plusieurs lignes dans une table. Si la liste des valeurs est dans le même ordre que la liste des colonnes de la table et s'il y a une valeur pour chaque colonne dans la table, alors la liste de nom de colonnes peut être omise.

1. Sous **WebMatrix**, ouvrez le site **Web Carnet** créé dans la série 5.

2. Créez une nouvelle page **Web CSHTML** nommée *InsertContacte.cshtml* qui présente le formulaire suivant d'insertion d'un contacte dans la base de données. Le script d'action associé au formulaire sera la page elle-même.

```
<h1>Formulaire d'ajout d'un nouveau contacte dans le carnet d'adresse</h1>
<p>@messNom</p>
<p>@messPrenom</p>
<p>@messSQL</p>
<p>(*) : Champs obligatoires</p>
<form method="post" action="" style="width: 30%">
  <fieldset>
    <legend>Ajout d'un nouveau contacte</legend>
    <p>Nom (*):<br />
    <input type="text" size="30" maxlength="30" name="nom" value="@Request["nom"]" />
    <p>Prénom (*): <br />
    <input type="text" size="30" maxlength="30" name="prenom" value="@Request["prenom"]" />
    <p>Adresse : <br />
    <textarea rows="4" cols="50" name="adresse">@Request["adresse"]</textarea>
    <p>Email : <br />
    <input type="text" size="30" maxlength="30" name="email" value="@Request["email"]" />
    <p>Téléphone : <br />
    <input type="text" size="10" maxlength="10" name="telephone" value="@Request["telephone"]" />
    <p><input type="submit" value="Insérer le contacte" /></p>
  </fieldset>
</form>
```

Formulaire d'ajout d'un nouveau contacte dans le carnet d'adresse

(*) : Champs obligatoires

Le code **Razor C#** suivant doit être placé en haut de la page de formulaire *InsertContacte.cshtml*.

```
@using WebMatrix.Data
@{
  var db = Database.Open("Carnet");
```

```

var messNom = "";
var messPrenom = "";
var messSQL = "";
if (IsPost)
{
    // Récupération des données du formulaire.

    var nom = Request["nom"];    var prenom = Request["prenom"];
    var adresse = Request["adresse"]; var email = Request["email"];
    var telephone = Request["telephone"];
    if(!nom.IsEmpty() && !prenom.IsEmpty())
    {
        /* Definition de la requête SQL d'insertion du contacte dans la table Contactes. Les valeurs à
        assigner aux champs de la table sont transmises en paramètres de la méthode Execute. */
        var insertQuery = "INSERT INTO Contactes (Nom, Prénom, Adresse, Email, Téléphone)" +
            "VALUES (@0, @1, @2, @3, @4)";

        try
        {
            db.Execute(insertQuery, nom, prenom, adresse, email, telephone);
            // Redirection vers la page ListeContactes.cshtml pour afficher le carnet d'adresses.
            Response.Redirect("ListeContactes.cshtml");
        }
        catch(Exception e)
        {
            messSQL = "Erreur d'insertion du contacte : " + e.Message;
        }
    }
    else // Le nom ou le prénom est vide.
    {
        if (nom.IsEmpty())
        {
            messNom = "Le nom du contacte doit obligatoirement être fourni.";
        }
        if (prenom.IsEmpty())
        {
            messPrenom = "Le prénom du contacte doit obligatoirement être fourni.";
        }
    }
}
}

```

Remarque : Par mesure de sécurité, il faut toujours passer des valeurs à une instruction **SQL** en utilisant les paramètres de la méthode **Execute**, exactement comme dans cet exemple. Cela permet de se protéger contre toutes tentatives d'envoi de commandes malveillantes dans la base de données appelées **attaques d'injection SQL**.

II. Mettre à jour des données dans une base de données

Dans cette partie, il s'agit de créer deux pages **Web** : La première nommée **EditerContactes.cshtml**, affiche les contacts dans un tableau et permet à un utilisateur d'en sélectionner un pour faire des modifications; la seconde nommée **ModifierContacte.cshtml**, permet à l'utilisateur de réaliser les modifications et de les enregistrer dans la base de données.

La mise à jour (ou modification)

UPDATE table SET champ1=expr1 [, champ2=expr2 ...] [WHERE condition]

UPDATE modifie les valeurs d'une ou de plusieurs champs d'une ou de plusieurs lignes d'une table. La condition dans la clause **WHERE** permet de fournir les lignes à modifier.

Suppression de lignes dans une table

DELETE FROM table [WHERE condition]

DELETE permet de supprimer des lignes dans une table. Si aucune condition ni données, toutes les lignes sont supprimées. Sinon, les lignes pour lesquelles la condition est vérifiée sont supprimées.

1. Créez une nouvelle page **Web CSHTML** nommée *EditerContactes.cshtml* qui présente le tableau suivant permettant de sélectionner un contact afin de modifier les informations du contact.

Editer les informations sur les contacts

	Nom	Prénom	Adresse	Email	Téléphone
Editer	Bakkali	Ahmed	Tétouan	bakkali@gmail.com	0690504825
Editer	Taleb	Fatiha	Tanger	talib.fatiha@hotmail.com	0667646570
Editer	Oulad Ali	Zineb	Martil	ouladalo@yahoo.fr	0539925165
Editer	Dahmani	Anouar	Larache	dahmani-anouar@gmail.com	0668646570
Editer	Ben Jbara	Badr	Tétouan	benjbara@hotmail.fr	0660643245
Editer	Khamlichi	Salim	Tétouan	khamlichi@yahoo.fr	0539424235

```
@using WebMatrix.Data
```

```
@{
```

```
var db = Database.Open("Carnet");
```

```
var selectQueryString = "SELECT * FROM Contactes";
```

```
}
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<title>Editer des contacts</title>
```

```
</head>
```

```
<body>
```

```
<h1 style="text-align: center">Editer les informations sur les contacts</h1>
```

```
<table border="2" align="center" width="60%">
```

```
<tr style="text-align: left">
```

```
<th>&nbsp;</th><th>Nom</th><th>Prénom</th><th>Adresse</th>
```

```
<th>Email</th><th>Téléphone</th>
```

```
</tr>
```

```
@foreach (var l in db.Query(selectQueryString))
```

```
{
```

```
<tr>
```

```
<td><a href="@("ModifierContacte.cshtml?Id=" + l.Id)">Editer</a></td>
```

```
<td>@l.Nom</td><td>@l.Prénom</td><td>@l.Adresse</td>
```

```
<td>@l.Email</td><td>@l.Téléphone</td>
```

```

        </tr>
    }
</table>
</body>
</html>

```

- Il faut maintenant créer la page **ModifierContacte.cshtml** qui permet à l'utilisateur de mettre à jour les informations sur un contact. Celle-ci inclut du code de validation pour contrôler les données que l'utilisateur saisit. Par exemple, toutes les colonnes obligatoires doivent contenir une valeur.

```

@using WebMatrix.Data
@{
    var db = Database.Open("Carnet");
    var id = Request["Id"];
    if (id.IsEmpty())
    {
        Response.Redirect("EditerContactes.cshtml");
    }
    string messNom = "", messPrenom = "";
    var selectQueryString = "SELECT * FROM Contactes WHERE Id=@0";
    var l = db.QuerySingle(selectQueryString, id);
    string nom = l.Nom, prenom = l.Prenom;
    string adresse = l.Adresse, email = l.Email, telephone = l.Téléphone;
    if (IsPost) // La page a été appelée comme script d'action pour traiter le formulaire.
    {
        nom = Request["nom"]; prenom = Request["prenom"];
        if (!nom.IsEmpty() && !prenom.IsEmpty())
        {
            adresse = Request["adresse"]; email = Request["email"];
            telephone = Request["telephone"];
            var updateQueryString = @"UPDATE Contactes SET Nom=@0, Prenom=@1, Adresse=@2,
                                     Email=@3, Téléphone=@4 WHERE Id=@5";
            db.Execute(updateQueryString, nom, prenom, adresse, email, telephone, id);
            Response.Redirect("EditerContactes.cshtml");
        }
        else
        {
            if (nom.IsEmpty())
            {
                messNom = "Le nom du contact doit obligatoirement être fourni.";
            }
            if (prenom.IsEmpty())
            {
                messPrenom = "Le prénom du contact doit obligatoirement être fourni.";
            }
        }
    }
}
<!DOCTYPE html>

```

```

<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Modifier un contact</title>
</head>
<body>
  <h1>Modification des informations d'un contact</h1>
  <p>@messNom</p>
  <p>@messPrenom</p>
  <p>(*) : Champs obligatoires</p>
  <form method="post" action="" style="width: 30%">

    <fieldset>
      <legend>Ajout d'un nouveau contact</legend>
      <p>Nom (*):<br />
      <input type="text" size="30" maxlength="30" name="nom" value="@nom" />
      </p>
      <p>Pr&eacute;nom (*): <br />
      <input type="text" size="30" maxlength="30" name="prenom" value="@prenom" />
      </p>
      <p>Adresse : <br />
      <textarea rows="4" cols="50" name="adresse">@adresse</textarea>
      </p>
      <p>Email : <br />
      <input type="text" size="30" maxlength="30" name="email" value="@email" />
      </p>
      <p>T&eacute;l&eacute;phone : <br />
      <input type="text" size="10" maxlength="10" name="telephone" value="@telephone" />
      </p>
      <p><input type="submit" value="Modifier le contact" /></p>
    </fieldset>
  </form>
</body>
</html>

```

Il faut noter que pour supprimer les données de la base de données Carnet.sdt, on peut procéder de la même manière que pour la modification de données : On peut créer deux pages **Web** : La première affiche les contacts dans un tableau et permet à un utilisateur d'en sélectionner un pour le supprimer et la seconde, permet à l'utilisateur de confirmer sa demande de suppression et de réaliser s'il accepte la suppression du contact de la base de données.