

Procédure pour se connecter à une bdd MySql avec du C#

Table des matières

Mise en place	2
La bibliothèque nécessaire	4
Class Personne	4
Class DBConnect	5
Organisation du formulaire	9
Les actions réalisés selon le bouton	10

Procédure pour se connecter à une bdd MySql avec du C#

Mise en place

Ici je me sers de l'aragon, Il faut installer MySQL Connector/NET sur le site officiel :
<https://dev.mysql.com/downloads/connector/net/6.1.html>

Créer une base de donnée avec Mysql.

```
create database ConnectCsharpToMysql;
```

```
use ConnectCsharpToMysql;
```

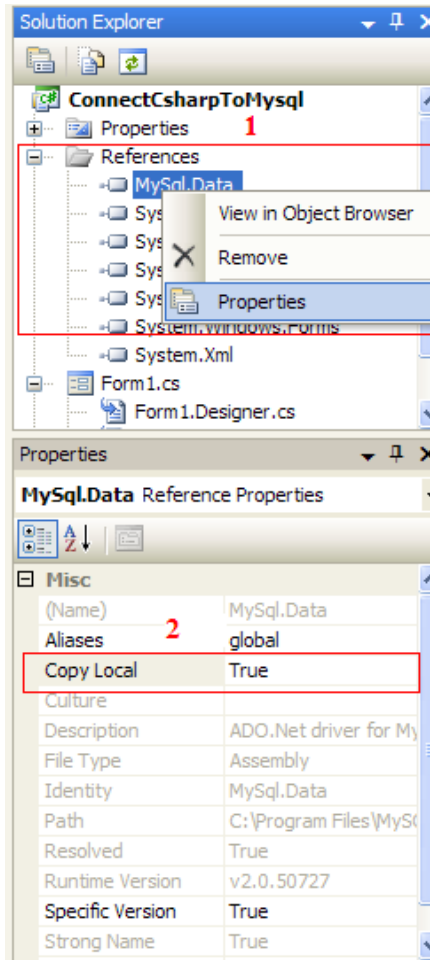
```
create table matable  
(  
  id INT NOT NULL AUTO INCREMENT,  
  name VARCHAR(30),  
  age INT,  
  PRIMARY KEY(id)  
);
```

Créer le projet voulu

ici je fais un windows form

Procédure pour se connecter à une bdd MySql avec du C#

Ajouter la librairie de MySql Library à notre librairie et y ajouter une référence → Assemblies → Extensions → MySql.Data



Vérifier dans les propriétés (1) de mysql.data que Copy Local soit True.(2)

Ajouter un projet librairie
ici DBConnect

```
//Add MySql Library  
using MySql.Data.MySqlClient;
```

Dans le code du windows form je n'oublie pas la connexion à la librairie
using Dbconnect

La bibliothèque nécessaire

Class Personne

// La class personne reçoit les valeurs la base de données

```
public class Personne
{
    private string nom;
    private DateTime date_naissance;

    //accesseurs
    public string Nom { get => nom; set => nom = value; }
    public DateTime Date_naissance { get => date_naissance; set => date_naissance =
value; }

    //constructeurs
    // L'un sans paramètres et l'autre, correspondant à celles de la bdd qui récupère les
valeurs contenu dans les zones de texte du formulaire
    public Personne()
    {
    }

    public Personne(string nom, DateTime date_n)
    {
        this.nom = nom;
        date_naissance = date_n;
    }

    // Méthode de conversion SQL <-> C# (les formats sont différents)
    public String Convert2MySql(DateTime dt)
    {
        return dt.ToString("yyyy-MM-dd HH:mm:ss");
    }

    private DateTime Convert2C(String sqlDate)
    {
        return DateTime.Parse(sqlDate);
    }

    // affiche les valeurs de personne
    public string affiche()
    {
        return "personne = " + nom + " " + date_naissance.ToString();
    }
}
```

Procédure pour se connecter à une bdd MySql avec du C#

Class DBConnect

```
public class DBConnect
{
    /*
     * On va maintenant créer la class DBConnect dans la bibliothèque, qui va accueillir
     nos différentes requêtes SQL.
     */

    //Il faut initialisé les différents composants nécessaires pour se connecter: l' ip, la
    database, le user et le password.
    private MySqlConnection connection;
    private string server;
    private string database;
    private string User;
    private string password;

    //Constructor
    //Le constructeur aura les composants de connexion en paramètres
    public DBConnect(string ip, string bdd, string user, string pwd)
    {
        // récupération des futures zones de textes dans notre formulaire pour la connexion
        puis on utilise Initialize()
        server = ip;
        database = bdd;
        User = user;
        password = pwd;
        Initialize();
    }

    //Initialize values
    // déclare une classe MySqlConnection en utilisant les paramètres de DBConnect()
    private void Initialize()
    {
        string connectionString;

        connectionString = "SERVER=" + server + ";" + "DATABASE=" +
        database + ";" + "user=" + User + ";" + "PASSWORD=" + password + ";";

        connection = new MySqlConnection(connectionString);
    }

    //open connection to database
    // On va se servir de la class MySqlConnection du nom de connection pour maintenant
    essayer de se connecter
    // si cela ne fonctionne pas on renvoie l'erreur
    public bool OpenConnection()
    {
        try
        {
            connection.Open();
            return true;
        }
        catch (MySqlException ex)
        {
            return false;
        }
    }
}
```

Procédure pour se connecter à une bdd MySql avec du C#

```
    }  
}  
  
//Close connection  
//On ferme la connexion et on renvoi si il y a une erreur  
public bool CloseConnection()  
{  
    try  
    {  
        connection.Close();  
        return true;  
    }  
    catch (MySqlException ex)  
    {  
        MessageBox.Show(ex.Message);  
        return false;  
    }  
}  
  
//Insert statement  
// Insert prends en paramètre une classe Personne qui va récupéré a deux paramètres Nom  
et Date_naissance  
// Insert envoi la requête avec les dites valeurs qui sont celles récupérés dans le  
formulaire avec la zone de texte et la date time  
//puis envoie tout simplement le tout dans la bdd avec là encore OpenConnection  
public void Insert(Personne nouveau)  
{  
    string query = "INSERT INTO matable(nom, ddn) VALUES ('" + nouveau.Nom + "', '" +  
nouveau.Convert2MySql(nouveau.Date_naissance) + "')";  
  
    //ouvre connection  
    if (this.OpenConnection() == true)  
    {  
        //créer une requête et attribuer la requête et la connexion depuis le  
constructeur  
        MySqlCommand cmd = new MySqlCommand(query, connection);  
  
        //Execute la requête  
        cmd.ExecuteNonQuery();  
  
        //ferme connection  
        this.CloseConnection();  
    }  
}  
  
//Update statement  
//Même principe que pour au dessus sauf qu'il faut une autre personne pour contenir  
avoir les anciennes valeurs à remplacer  
// Nouvelle contient les nouvelles valeurs et editer pour le where  
public void Update(Personne nouvelle, Personne editer)  
{  
    string query = "UPDATE matable SET nom='" + nouvelle.Nom + "', ddn='" +  
nouvelle.Convert2MySql(nouvelle.Date_naissance) + "' WHERE name='" + editer.Nom + "'";  
  
    //Open connection  
    if (this.OpenConnection() == true)  
    {  
        //créer la requête sql
```

Procédure pour se connecter à une bdd MySql avec du C#

```
MySQLCommand cmd = new MySQLCommand();
//Assigne the query
cmd.CommandText = query;
//Assign the connection using Connection
cmd.Connection = connection;

//Execute query
cmd.ExecuteNonQuery();

//ferme connection
this.CloseConnection();
}

//Delete statement
// comme Insert sauf que la requête DELETE la ligne
public void Delete( string nomsup )
{
    string query = "DELETE FROM matable WHERE name='" + nomsup + "'";

    if (this.OpenConnection() == true)
    {
        MySQLCommand cmd = new MySQLCommand(query, connection);
        cmd.ExecuteNonQuery();
        this.CloseConnection();
    }
}

//Select statement
// La requête récupère tout le contenu de matable
public List<Personne> Select()
{
    string query = "SELECT * FROM matable";

    //Create une liste pour y ranger le résultat de la requête
    List<Personne> list = new List<Personne>();

    //Ouvre connection
    if (this.OpenConnection() == true)
    {
        //Create Command
        MySQLCommand cmd = new MySQLCommand(query, connection);
        //Create a data reader and Execute the command
        MySQLDataReader dataReader = cmd.ExecuteReader();

        //initialisation d'un objet personne
        Personne unepersonne;

        //Lit les données de la list au travers d'une boucle avec la class
        while (dataReader.Read())
        {
            unepersonne = new Personne(dataReader["nom"].ToString(),
            DateTime.Parse(dataReader["ddn"].ToString()));
            list.Add(unepersonne);
        }

        //close Data Reader
    }
}
```

Procédure pour se connecter à une bdd MySql avec du C#

```
dataReader.Close();

//close Connection
this.CloseConnection();

//return list to be displayed
return list;
}
else
{
    return list;
}
}
```


Procédure pour se connecter à une bdd MySql avec du C#

Organisation du formulaire

Respecter le schéma suivant pour le formulaire

The screenshot shows a Windows Form titled "Form1" with a light blue title bar. The form contains the following elements:

- Login Section:** Four text boxes labeled "IP Serveur", "Nom de la base de données", "Identifiant", and "Mot de passe". The "IP Serveur" box contains "localhost", "Nom de la base de données" contains "csharp", "Identifiant" contains "root", and "Mot de passe" contains "root". To the right of these boxes are two buttons: "Se Connecter" and "Voir la liste".
- Data Management Section:** A section titled "Liste des personnes" containing a table with two columns: "Nom" and "Date de naissance". The table has a header row and one data row with an asterisk (*) in the "Nom" column. To the right of the table are two buttons: "Editer la personne" and "Supprimer".
- Form Fields:** Below the table, there are two more form fields: "Nom :" with a text box, and "Date de naissance :" with a date picker showing "jeudi 15 octobre 2020".
- Action Buttons:** At the bottom right, there are two buttons: "Mettre à jour" and "Ajouter".

Procédure pour se connecter à une bdd MySql avec du C#

Les actions réalisés selon le bouton

//Initialisation de deux objets

```
DBConnect test;  
Personne editer;
```

// fonction pour ... remplir la liste du tableau

```
private void remplirListe()  
{
```

```
    dataGridView1.Rows.Clear();
```

```
    List<Personne> lespersonne = test.Select();
```

```
    foreach (Personne quelquun in lespersonne)
```

```
    {
```

```
        dataGridView1.Rows.Add(quelquun.Nom, quelquun.Date_naissance);
```

```
    }
```

```
}
```

//Bouton Se connecter

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
    //récupération du contenu dans les zones texte à gauche du bouton
```

```
    string ip = Textip.Text;
```

```
    string bdd = Textbdd.Text;
```

```
    string id = Textid.Text;
```

```
    string pwd = Textpwd.Text;
```

```
    //instanciation de l'objet DBConnect
```

```
    test = new DBConnect(ip, bdd, id, pwd);
```

```
    //Renvoie un messagebox si c'est bon ou non pour tester la connexion
```

```
    if (test.OpenConnection() == true)
```

```
    {
```

```
        MessageBox.Show("Vous êtes connectez !");
```

```
    }
```

```
    else
```

```
    {
```

```
        MessageBox.Show("Erreur de connexion");
```

```
    }
```

```
    test.CloseConnection();
```

```
}
```

//Bouton Voir la liste

```
private void butListe_Click(object sender, EventArgs e)
```

```
{
```

```
    //Tout est dans le nom
```

```
    remplirListe();
```

```
}
```

//Bouton Editer la personne

```
private void butEditer_Click(object sender, EventArgs e)
```

```
{
```

```
    if (dataGridView1.CurrentRow.Selected)
```

```
    {
```

```
        //editer, class personne qui se "construit" avec les valeurs dans la datagrid  
        selectionner par l'utilisateur
```

Procédure pour se connecter à une bdd MySql avec du C#

```
        editor = new Personne(dataGridView1.CurrentRow.Cells[0].Value.ToString(),
DateTime.Parse(dataGridView1.CurrentRow.Cells[1].Value.ToString()));

        //les valeurs se placent dans les zones en bas à droite
        textBox1.Text = editor.Nom;
        dateTimePicker1.Value = editor.Date_naissance;
    }
}

//Bouton Supprimer
private void butSupprimer_Click(object sender, EventArgs e)
{
    //conversion d'une ligne du tableau en string
    string suppr = dataGridView1.CurrentRow.Cells[0].Value.ToString();

    if (MessageBox.Show("êtes vous sur de vouloir supprimer " + suppr + " ?",
"avertissement ", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        // suppression de la ligne correspondant à suppr
        test.Delete(suppr);

        //re remplit la liste du tableau pour la mettre à jour
        remplirListe();
    }
}

//Bouton Mettre à jour
private void butMaj_Click(object sender, EventArgs e)
{
    //Init d'un objet Peronne()
    Personne nouvelle = new Personne();

    // récupération des informations dans les zones en bas à droite pour envoyer la
requête de mise à jour dans la bdd
    // puis maj de la liste visuel pour l'utilisateur
    if (textBox1.Text != "")
    {
        nouvelle.Nom = textBox1.Text;
        nouvelle.Date_naissance = dateTimePicker1.Value;
        test.Update(nouvelle, editor);
        remplirListe();
    }
    else
    {
        MessageBox.Show("rentrer un nom ! ");
        textBox1.Focus();
    }
}

//Bouton Ajouter
// Même procédure que vu pour le Bouton Mise à jour sauf qu'ici on n'a pas de
récupération des anciennes valeurs
private void butAjouter_Click(object sender, EventArgs e)
{
    Personne nouveau = new Personne();

    if (textBox1.Text != "")
```

Procédure pour se connecter à une bdd MySql avec du C#

```
{
    nouveau.Nom = textBox1.Text;
    nouveau.Date_naissance = dateTimePicker1.Value;
    test.Insert(nouveau);
    remplirListe();
}
else
{
    MessageBox.Show("rentrez un nom ! ");
    textBox1.Focus();
}
}
```