

Exemple d'interface graphique avec MongoDB

Étape 1 : Dans Studio 3T, on crée la base de données **Kba**, on crée ensuite la collection **joueurs** et on y insère **quelques documents**

```
use Kba; // Création de la base de données Kba
```

```
db.createCollection("joueurs") // création de la collection joueurs
```

```
// insertion du document 1
```

```
db.joueurs.insertOne(
  {
    "_id": 1,
    "alias": "primogene",
    "solde": 1000,
    "mechant" :
      { "nom": "la bête",
        "force": "trés fort",
        "arme": "feu"
      }
  }
)
```

```
// insertion du document 2
```

```
db.joueurs.insertOne(
  {
    "_id": 2,
    "alias": "patoche",
    "solde": 800,
    "mechant" :
      { "nom": "la bête",
        "force": "trés fort",
        "arme": "feu"
      }
  }
)
```

```
// insertion du document 3
```

```
db.joueurs.insertOne(
  {
    "_id": 3,
    "alias": "simba",
    "solde": 1200,
    "mechant" :
      { "nom": "orc",
        "force": "trés fort",
        "arme": "épée"
      }
  }
)
```

```
// insertion du document 4
```

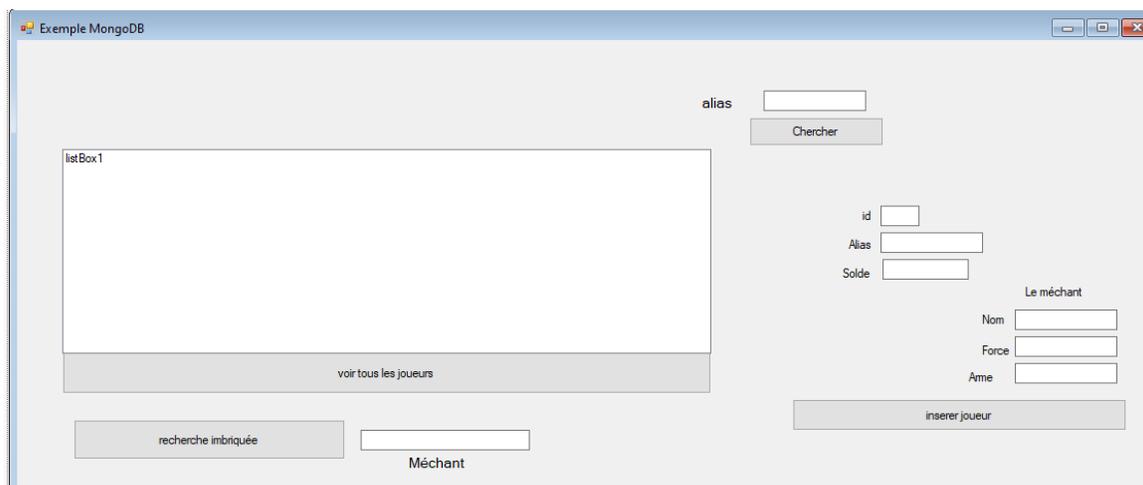
```
db.joueurs.insertOne(
  {
    "_id": 4,
    "alias": "piwi",
    "solde": 1000,
    "mechant" :
      { "nom": "orc",
        "force": "trés fort",
        "arme": "feu"
      }
  }
)
```

```
// insertion du document 5
```

```
db.joueurs.insertOne(
  {
    "_id": 5,
    "alias": "Tigris",
    "solde": 800,
    "mechant" :
      { "nom": "ogopogo",
        "force": "trés très fort",
        "arme": "feu"
      }
  }
)
```

Remarquez que votre document contient un document imbriqué

Étape 2 on crée l'interface graphique comme suit :

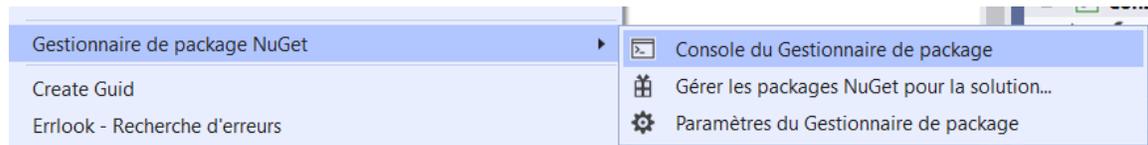


Sur l'interface nous avons

1. Un bouton pour insérer un joueur
2. Un bouton pour voir tous les joueurs
3. Un bouton pour rechercher selon un joueur
4. Un bouton pour faire la recherche selon le mechant (document imbriqué). On veut savoir quels sont les joueurs qui ont combattu un mechant donnée
5. Tous les affichages se font dans un listBox

Avant de commencer à coder , inclure le package nécessaire pour avoir le driver MongoDB pour C#. Allez sur le site : <https://www.nuget.org/packages/mongodb.driver>

Dans la console du gestionnaire des Packages, tapez la commande suivante:



Install-Package MongoDB.Driver -Version 2.14.0

Puis ajouter les espaces de noms pour avoir les classes de driver MongoDB

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MongoDB.Driver;
using MongoDB.Bson;
```

Dans notre cas MongoDB est installée sur une machine locale (localhost) et écoute par le port 27017. Ce qui veut dire que la chaîne de connexion est de la forme:

```
"mongodb://localhost:27017":
```

```
private const string connectionString = "mongodb://localhost:27017";
```

Et pour ouvrir une chaîne de connexion, il suffit d'instancier un client MongoDB en utilisant la chaîne de connexion (comme SqlConnection ou OracleConnection)

```
private const string connectionString = "mongodb://localhost:27017";
MongoClient mong = new MongoClient(connectionString);
```

Code du bouton afficher tous les joueurs :

Dans MongoDB, l'instruction à exécuter est : **db.joueurs.find();**

Aucun critère n'est fourni. Ce qui se traduit par le Filter.Empty

La variable filter est de type document. Dans C# c'est un BsonDocument

```
private void btnAfficher_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    IMongoDatabase db = mong.GetDatabase("Kba");
    IMongoCollection<BsonDocument> c1 = db.GetCollection<BsonDocument>("joueurs");

    var filter = Builders<BsonDocument>.Filter.Empty;
    var res = c1.Find(filter).ToList();
    foreach (var re in res)
    {
        listBox1.Items.Add(re);
    }
}
```

Code du bouton pour afficher un joueur selon son alias : (chercher un joueurs

Dans MongoDB, le critère de recherche est un document { "alias": "alias" }

db.joueurs.find({"alias": "Tigris"});

il suffit d'indiquer le filtre dans le code C#.

```

private void butnchercher_Click_1(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    string varalias = textAlias.Text;
    IMongoDatabase db = mong.GetDatabase("Kba");

    IMongoCollection<BsonDocument> c1 = db.GetCollection<BsonDocument>("joueurs");

    var filter = Builders<BsonDocument>.Filter.Eq("alias", varalias);
    var res1 = c1.Find(filter).ToList();

    foreach (var re in res1)
    {
        listBox1.Items.Add(re);
    }
}

```

Code du bouton afficher les joueurs ayant combattu une créature donnée : (recherche imbriquée)

Ici nous voulons savoir par exemple, qui sont les joueurs qui ont combattu un orc. L'orc est dans le document « mechant » qui est à l'intérieur du document joueur.

Dans MongoDB la recherche se fera ainsi :

```
db.joueurs.find({"mechant.nom": "orc"})
```

Pour le code C #, il suffit de l'indiquer que le critère de recherche se trouve dans un document

```

private void rechercher2_Click(object sender, EventArgs e)
{
    string mechant = textMechant.Text;
    IMongoDatabase db = mong.GetDatabase("Kba");
    IMongoCollection<BsonDocument> c1 = db.GetCollection<BsonDocument>("joueurs");

    var filter = Builders<BsonDocument>.Filter.Eq("mechant.nom", mechant);
    var res1 = c1.Find(filter).ToList();

    foreach (var re in res1)
    {
        listBox1.Items.Add(re);
    }
}

```

Code du bouton pour insérer un document :

Ici, notre document est imbriqué, alors il sera important de le préciser pour C#. Il n'y a pas d'auto-increment aussi. C'est possible d'avoir un auto-increment avec un trigger, mais ce n'est pas le but ici. Dans MongoDB, l'insertion se fait comme suit.

```
db.joueurs.insertOne(
  {
    "_id": 3,
    "alias": "simba",
    "solde": 1200,
    "mechant" :
    {
      "nom": "orc",
      "force": "trés fort",
      "arme": "épée"
    }
  }
)
```

Dans C#, il faudra d'abord créer le document Bson, puis effectuer l'insertion.

```
private void btnInser1_Click(object sender, EventArgs e)
{
    IMongoDatabase db = mong.GetDatabase("Kba");
    IMongoCollection<BsonDocument> c1 = db.GetCollection<BsonDocument>("joueurs");
    int id = int.Parse(textId.Text.ToString());
    string alias = textAliass.Text;
    int solde = int.Parse(textSolde.Text.ToString());
    string nom = textNom.Text;
    string force = textForce.Text;
    string arme = textArme.Text;

    var documnt = new BsonDocument
    {
        { "_id", id},
        { "alias", alias},
        { "solde", solde},
        { "mechant", new BsonDocument {
            { "nom", nom},
            { "force", force},
            { "arme", arme},
        }
        }
    };

    c1.InsertOne(documnt);
}
```

Si le document à insérer contient une liste comme suit :

```
db.joueurs.insertOne(
  {
    "_id": 100,
    "alias":["Lucky","Patate"],
    "solde": 1000,
    "mechant" :
      {"nom":"la bête",
       "force": "trés fort",
       "arme": "feu"
      }
  }
);
```

Alors le code C# du document serait :

```
{
  { "_id", id},
  // { "alias", alias},
  {"alias", new BsonArray(new[] {"Lucky", "Patate"})},
  { "solde", solde},

  { "mechant" , new BsonDocument {
    {"nom", nom},
    { "force", force},
    { "arme", arme},
  }
  }
};

cl.InsertOne(documnt);
}
```