ADO.Net

Nội dung

- Tổng quan về ADO.Net
- Connected model

Giới thiệu ADO.Net

- Các công nghệ cũ
  - ODBC
  - OLEDB
  - ActiveX Data Object
- ADO.Net
  - Dùng cho môi trường .Net
  - Là một phần của .Net Framework
  - Các lớp chứa trong **System.Data.dll**

Các loại CSDL quan tâm

- XML / json
- SQL Server

- Ngoài ra còn có
  - Excel
  - Oracle
  - MySQL / MariaDB / Postgres
  - SQLite

Tại sao lại là các CSDL này?

Tại sao lại phải quan tâm thêm những cái này?

hot trends

- MongoDB
- Firebase

Để bắt đầu, cần tập trung

- Microsoft.Data.SqlClient: **SQL Server**

- Câu hỏi
  - Tại sao lại là SQL Server?

5 bước chính khi thao tác

1. **Mở kết nối** tới CSDL
2. Chuẩn bị câu truy vấn
3. *Thực thi* câu truy vấn
4. *Xử lí kết quả* trả về
5. **Đóng kết nối** tới CSDL

Thêm NuGet package

Microsoft.Data.SqlClient (chỉ dành cho .net 8)

1. Mở kết nối

Connection string cho các CSDL

- Connectionstrings.com

Tạo kết nối

```
string connectionString = """
    Server = localhost;
    Database = MyShop;
    User Id = sa;
    Password = SqlServer@123;
    TrustServerCertificate = True;
 """;

var connection = new SqlConnection(connectionString);
try {
    connection.Open();
} catch (Exception ex) {
    Console.WriteLine($"Cannot connect. Reason: {ex.Message}");
}
```

pinggy

```
var connectionString = """
    Server = rniwo-23-97-62-146.a.free.pinggy.link,33777;
    Database = master;
    User Id = sa;
    Password = SqlServer@123;
    TrustServerCertificate = True;
""";
```

Tạo chuỗi kết nối

- Dùng windows authentication, localhost, bản express, kết nối tới CSDL có tên: QLNhanVien

```
var connectionString =
    "Server=.\SQLEXPRESS;
    Database=DemoDB;
    Trusted_Connection=yes;
    TrustServerCertificate=True
    “
```

Tạo chuỗi kết nối – SQL Server

- Localhost, kết nối có username và password, bản đầy đủ

connectionString =

"*Server= .\SQLEXPRESS;*

Database=QLNhanVien;

User ID = tester1; Password = 123;

TrustServerCertificate=True

";

Mở kết nối

```
var connection = new SqlConnection(connectionString);

connection.Open();
```

Một số cách đặt tên biến khác

- con
- cnn

Đưa kết nối vào thread khác

Mở kết nối chiếm dụng main thread khiến UI không phản hồi

Giải pháp: đưa vào thread khác

```
connection = await Task.Run(() => {
     var _connection = new SqlConnection(connectionString);
    _connection.Open();

    // Test khi chạy quá nhanh
    System.Threading.Thread.Sleep(3000);
    return _connection;
});
```

Lưu chuỗi kết nối ở đâu?

- Biến môi trường ở Settings Debug

Sử dụng builder

```
var builder = new SqlConnectionStringBuilder();
builder.DataSource = @".\sqlexpress";
builder.InitialCatalog = "MyShop1";
builder.UserID = "sale1";
builder.Password = "1234";

var db = new MS02Entities(builder.ConnectionString);
if (db.CanConnect()) {
    MessageBox.Show("OK");
} else
{
    MessageBox.Show("Wrong information");
}
```

2. Chuẩn bị câu truy vấn

Truy vấn bình thường

```
var sql = "select * from NhanVien";
```

Một số cách đặt tên biến khác

- query
- queryString
- sqlQuery

Truy vấn

```
var sql = "select id, category_name from Category";
var command = new SqlCommand(sql, connection);
var reader = command.ExecuteReader();

while (reader.Read()) { // Duyệt qua mỗi dòng của bảng kết quả
    int id = (int) reader["id"];
    string name = (string)reader["category_name"];

    Console.WriteLine($"{id} - {name}");
}
```

Truy vấn phức tạp – Nối chuỗi

- Ví dụ đăng nhập

var sql = "*select * from NhanVien where Username='*" + username + "'"

Rất nguy hiểm, nên tránh việc cộng chuỗi với toán tử +

⇒ Tiềm ẩn lỗi **sql injection** giúp kẻ tấn công đăng nhập mà không cần biết username cũng như không cần biết mật khẩu!!! (username = "' or '1'='1 ")

select * from NhanVien where Username='' or '1'='1'

Sử dụng parameter

```
sql = "select * from NhanVien where Username=@User";

var command = new SqlCommand(sql, connection);

command.Parameters.Add("@User", OleDbType.NVarChar);
command.Parameters["@User"].Value = username;
```

Viết tắt:

```
cmd.Parametes.Add("@User", OleDbType.NVarChar).Value = "";
```

Ví dụ GetById

```
int idToFind = 3;
var sql = "select id, category_name from Category where id=@id";
var command = new SqlCommand(sql, connection);
command.Parameters
    .Add("@id", System.Data.SqlDbType.Int)
    .Value = idToFind;
```

Ví dụ DeleteById

```
int idToFind = 3;
var sql = "delete from Category where id=@id";
var command = new SqlCommand(sql, connection);
command.Parameters
    .Add("@id", System.Data.SqlDbType.Int)
    .Value = idToFind;

var count = command.ExecuteNonQuery();

if (count == 1) {
    Console.WriteLine("Delete successfully!");
} else {
    Console.WriteLine($"Cannot delete category with id={idToFind}");
}
```

3. Thực thi câu truy vấn

Các kiểu lệnh cần thực thi

- ExecuteReader: là câu truy vấn lấy dữ liệu gồm nhiều dòng nhiều cột.
- ExecuteNonQuery: không có kết quả trả về, thường là **Thêm**, **Xóa**, **Sửa**
- ExecuteScalar: lấy một giá trị đơn. Ví dụ số lớn nhất, trung bình, nhỏ nhất, ID vừa chèn...

Command & DataReader

```
var command = new SqlCommand(sql, connection);
var reader = command.ExecuteReader();
```

- Kết quả trả về là bảng, vẫn đang giữ kết nối
- Biến reader đang trỏ đến dòng thứ nhất của bảng
- Phải đọc lần lượt từng dòng trong kết quả

4. Xử lí kết quả trả về

Đọc dữ liệu từ DataReader

```
while ( reader.Read() ) { ‘ Đọc từng dòng
    var id = (int) reader[0];
    var fullName = (string) reader[1];
}
```

- Cách tốt hơn (Dễ bảo trì)

```
var id = (int) reader[“ID”];
var fullName = (string) reader[“Name”];
```

Cập nhật dữ liệu - Command

```
var sql = "update SinhVien set Name=@name where ID=@id"
var command = new SqlCommand(sql, connection)

int rows = command.ExecuteNonQuery();
```

5. Đóng kết nối

Đóng kết nối

- Quá dễ: connection.Close();

Những việc làm thông dụng

- GetAll
  - Với danh sách nhỏ thì lấy toàn bộ
  - Với danh sách lớn cần
    - Phân trang (vd: 10 dòng mỗi trang, hiện trang 4)
    - Sắp xếp (vd: giảm dần theo ngày tháng, tăng dần theo giá)
    - Lọc (vd: tìm trong khoảng giá, CPU, Ram)
    - Tìm kiếm (vd: tên / mô tả có chứa "3080")
- GetByID (Lấy thông tin 1 đối tượng dựa vào ID)
- Insert – Thêm
- DeleteByID – Xóa
- UpdateByID – Sửa

Mini quiz 3

1. GetAll: Những bảng có đặc điểm nào thì sẽ lấy hết vào bộ nhớ?
2. Những bảng có đặc điểm nào không thích hợp lấy hết vào bộ nhớ?
3. Nếu không lấy hết thì phải lấy như thế nào?

Một số vấn đề khác

Lấy ID của đối tượng vừa chèn

```
string sql = "insert into Category(Name) values(@Name);
                select ident_current('Category');";
var command = new SqlCommand(sql, _connection);
command.Parameters.Add("@Name", SqlDbType.NText)
                .Value = newCategory.Name;
int id = Convert.ToInt32(command.ExecuteScalar());
```

Chèn và lấy id

```
var sql = """
    insert into Category(category_name) values(@name);
    select ident_current('Category');
"""; // Eager loading
var command = new SqlCommand(sql, connection);
command.Parameters
    .Add("@name", System.Data.SqlDbType.NVarChar)
    .Value = "Microsoft";

var createdId = (int)((decimal)command.ExecuteScalar());
Console.WriteLine($"Insert successfully. Newly created id = {createdId}");
```

Làm việc với stored procedure

```
var sql = "StoredName";
var command = new SqlCommand(sql, connection);
command.CommandType = CommandType.StoredProcedure;

var RetVal = command.Parameters.Add ("RetVal", SqlDbType.Int);
RetVal.Direction = ParameterDirection.ReturnValue;

command.Parameters.Add("@id", SqlDbType.VarChar, 11)
      .Direction = ParameterDirection.Input;

var NumTitles  =    command.Parameters.Add("@titlesout", SqlDbType.VarChar, 11);
NumTitles.Direction = ParameterDirection.Output;
```

Phân trang – Các yếu tố

- Tổng số lượng trang có thể có?
  - Tổng số dòng kết quả trả về? - TotalItems
  - Số lượng kết quả mỗi trang? – RowsPerPage
  - TotalPages = TotalItems / RowsPerPage + 0 / 1
  - Mất một lần truy vấn
- Trang hiện tại là trang nào? Trang kế?
- Trước SQL Server 2012 sử dụng hàm Row_Number()

Phân trang từ SQL Server 2012

- Bỏ qua 10 kết quả đầu tiên, lấy toàn bộ còn lại

```
Select * from NhanVien order by ID offset 10 rows
```

(Bắt buộc phải có order by)

- Bỏ qua 10 kết quả đầu tiên, lấy 7 kết quả kế

```
Select * from NhanVien order by ID offset 10 rows
    fetch next 7 rows only
```

Trả ra tổng item

select ID, Name, count(*) over() as TotalItems

1 – Laptop – 3

2– Mouse – 3

3 – Keyboard – 3

Vấn đề trạng thái đơn hàng

Các trạng thái có thể có:

- Mới tạo, Đã hủy, Đã hoàn thành, Đang giao
- New, Cancelled, Completed, Shipping

Tạo ra enum ánh xạ 1-1

	EnumKey	Value	Description
1	New	1	Đơn hàng mới được tạo
2	Completed	2	Khách hàng đã thanh toán
3	Cancelled	3	Đơn hàng đã hủy
4	Shipping	4	Đã thanh toán và đã giao

```
enum PurchaseStatus
{
    All = -1,
    New = 1,
    Cancelled = 2,
    Completed = 3,
    Shipping = 4
}
```

Ảnh lưu ở đâu

Phạm vi môn học

- Dành một thư mục chứa ảnh tập trung
- Khi nộp bài nhớ nộp kèm

Thực tế

- Cần setup file server - nginx

Postgres

Tạo ra container và chạy

```
$ docker create --name db -e POSTGRES_PASSWORD=1234
-p 5432:5432 postgres

$ docker start db
```

Lấy id vừa tạo ra

```
SELECT currval(pg_get_serial_sequence('Category','id'));
```

Paging

select * from category offset 1 limit 3;

Get the total rows of the query

SELECT *, count(*) OVER() AS Total

FROM category OFFSET 1 LIMIT 3;