

LẬP TRÌNH NÂNG CAO

Lưu trữ dữ liệu



Giảng viên: Nguyễn Thị Phương Dung
Email: dungntp@thu.edu.vn

Giải pháp lưu trữ khi không dùng CSDL

- Sử dụng chuẩn định dạng dữ liệu JSON
- JSON viết tắt của: **J**ava**S**cript **O**bject **N**otation
- JSON sử dụng định dạng key-value cho mỗi thông tin dữ liệu.
 - Key: \approx tên trường hoặc tên thuộc tính
 - Value: \approx dữ liệu của trường / giá trị của thuộc tính
- Lưu trữ dưới dạng file text thường có phần mở rộng là .json hoặc .js



Định dạng chuỗi JSON

- Mỗi đối tượng JSON được bao bọc bởi cặp dấu ngoặc nhọn {}
- Dấu : dùng để phân cách giữa key và value
- Dấu , để phân cách giữa các cặp key-value, hoặc giữa các đối tượng JSON trong mảng
- Các key và value được đặt trong cặp dấu nháy kép “” trừ dữ liệu dạng số và dạng Boolean.
- Nếu trong value có chứa dấu “ thì dùng dấu (\) trước dấu “ đó.
- Nếu value là một mảng các đối tượng JSON khác thì được bao bởi cặp ngoặc []



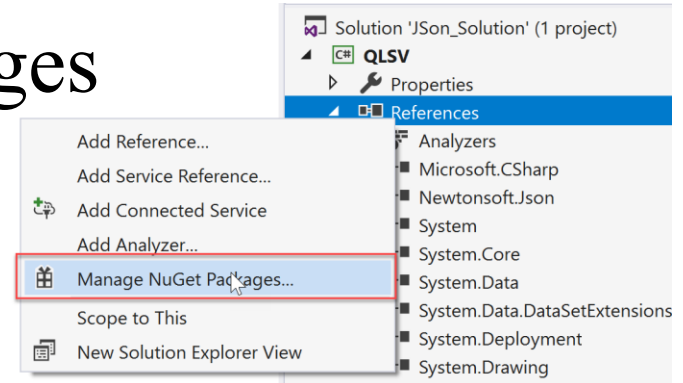
Cách sử dụng JSON trong C#

- Khai báo: `using Newtonsoft.Json;`
- Sử dụng các lệnh chuyển đổi:
- `JsonConvert.SerializeObject(object)`: chuyển đổi object sang định dạng chuỗi JSON
- `JsonConvert.DeserializeObject<object>(json)`: chuyển đổi chuỗi JSON về dạng đối tượng



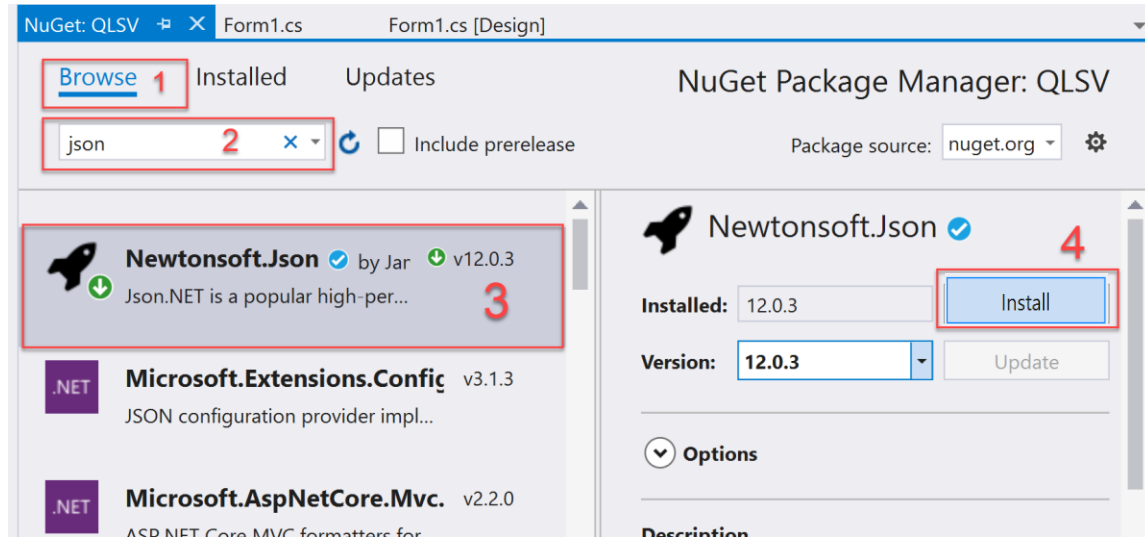
Cài đặt JSON vào C#

- Nếu có lỗi khi khai báo using Newtonsoft.Json; cần cài đặt thêm thư viện json bằng cách:
- Bấm chuột phải vào References
- Chọn Manage NuGet Packages



Cài đặt JSON vào C#

- Thực hiện tìm kiếm và cài đặt theo các bước sau:



Cách đọc/ghi file JSON

- Tạo đối tượng thuộc lớp `System.IO.StreamReader` để đọc file
- Ví dụ:

```
System.IO.StreamReader reader;  
reader = new System.IO.StreamReader(@"data.json");  
string jsonstr = reader.ReadToEnd();
```



Cách đọc/ghi file JSON

- Sử dụng lệnh

`System.IO.File.WriteAllText(filepath, string)`
để ghi string vào file

- Ví dụ:

```
System.IO.File.WriteAllText(@"data.json", jsonstr);
```



Ví dụ

- Viết chương trình cho phép:
 - Nhập danh sách các tỉnh thành
 - Lưu dữ liệu để lần sau có thể mở ra chỉnh sửa hoặc thêm tiếp
- => Cần xác định nội dung cần lưu trữ và xác định định dạng dữ liệu cần lưu trữ



Cách làm 1

- Tạo lớp tính để xác định thuộc tính và giá trị dữ liệu sẽ lưu trữ.

```
class Tinh
{
    int id;
    string name;
    public Tinh(int i, string n) ...
    public int ID
    {
        get { return id; }
        set { id = value; }
    }
    public string Name
    {
        get { return name; }
        set { name = value; }
    }
}
```



Cách làm 1

- Tạo 1 danh sách các đối tượng kiểu Tỉnh để có thể lưu trữ dưới dạng JSON

```
List<Tỉnh> dstinh= new List<Tỉnh>();
```



Cách làm 1

- Mỗi lần thêm mới một đối tượng Tinh thì cập nhật vào danh sách.

```
Tinh t = new Tinh(id, textBox1.Text);  
dstinh.Add(t);
```



Cách làm 1

- Sau đó chuyển danh sách thành dạng chuỗi JSON rồi lưu vào file.

```
string jsonstr = JsonConvert.SerializeObject(dstinh);  
System.IO.File.WriteAllText("tinh.json", jsonstr);
```



Cách làm 1

- Khi nào cần dùng thì đọc file ra để lấy dữ liệu:

```
//tạo đối tượng StringReader để truy cập vào file
System.IO.StreamReader reader = new System.IO.StreamReader(@"tinh.json");
//sử dụng lệnh ReadToEnd() để lấy toàn bộ dữ liệu trong file
string jsonstr = reader.ReadToEnd();
//tạo đối tượng phù hợp với chuỗi JSON đã lưu
//và chuyển dữ liệu từ chuỗi json đã lưu vào đối tượng
List<Tinh> dstinhs = JsonConvert.DeserializeObject<List<Tinh>>(jsonstr);
//sử dụng danh sách chuyển đổi từ chuỗi JSON
dstinhs.ForEach(tinh => dataGridView.Rows.Add(tinh.ID, tinh.Name));
```



Cách làm 2

- Sử dụng các đối tượng có sẵn như:
 - DataTable: cấu trúc dạng bảng, danh sách các dòng
 - DataSet: cấu trúc dạng danh sách các bảng



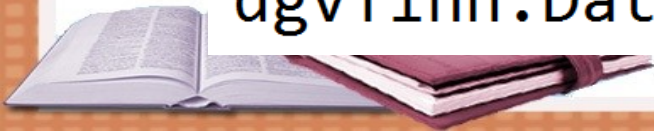
Cách làm 2

- Tạo bảng Tỉnh

```
DataTable dtTinh = new DataTable("Tinh");
```

- Thiết lập các cột cho bảng và gán DataSource của DataGridView bằng bảng vừa tạo

```
dtTinh.Columns.Add("ID");  
dtTinh.Columns.Add("Tên tỉnh");  
dgvTinh.DataSource = dtTinh;
```



Cách làm 2

- Thêm mới dữ liệu vào bảng thì datagridview cũng cập nhật theo:

```
dtTinh.Rows.Add(idtinh, textBox1.Text);
```



Cách làm 2

- Chuyển đổi bảng thành chuỗi JSON

```
string jsonstr;  
//chuyển đổi bảng thành chuỗi JSON  
jsonstr = JsonConvert.SerializeObject(dtTinh);  
//đưa dữ liệu vừa chuyển đổi lên textBox2  
textBox2.Text = jsonstr;
```



Cách làm 2

- Chuyển đổi chuỗi JSON thành bảng

```
//chuyển đổi ngược từ chuỗi JSON trong textBox2 vào bảng  
DataTable dt = JsonConvert.DeserializeObject<DataTable>(textBox2.Text);  
//gán dữ liệu của bảng vào một datagridview1 để xem  
dataGridView1.DataSource = dt;
```

