

LẬP TRÌNH NÂNG CAO

Lưu trữ dữ liệu bằng file JSON



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

Quản lý

Quản lý

Quản lý

Quản lý sinh viên



Tỉnh

Tỉnh

Tỉnh

Tỉnh

Khoa

Lớp

Sinh viên

Mã tỉnh

Mã khoa

Khoa

Mã lớp

C

6

6

Tên tỉnh

Tên khoa

Tên lớp

Mã SV

Họ tên

Giới tính ☐ Nam ☐ Nữ

Ngày sinh 1/ 1/2000

Nơi sinh

Hà Nội

Khoa

Lớp

60TH1

Thêm

Sửa

Xóa

Mã SV

Họ tên

Ngày sinh

Giới tính

Quê quán

Khoa

Lớp

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



Ví dụ

	Mã khoa	Tên khoa
▶	CNTT	Công nghệ thông tin
	CT	Công trình
	KT	Tài chính kế toán
	QTKD	Quản trị kinh doanh

[{"Mã khoa":"CNTT","Tên khoa":"Công nghệ thông tin"},
{"Mã khoa":"CT","Tên khoa":"Công trình"},
{"Mã khoa":"KT","Tên khoa":"Tài chính kế toán"},
{"Mã khoa":"QTKD","Tên khoa":"Quản trị kinh doanh"}]



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 []



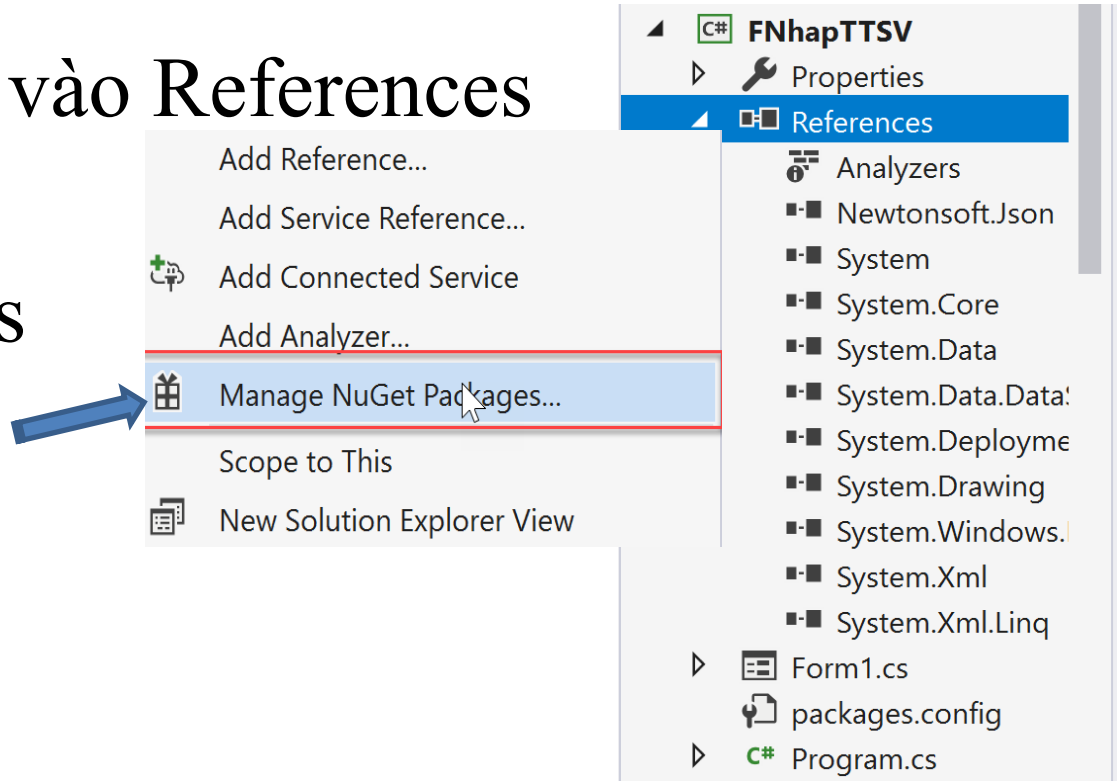
Kiểu dữ liệu trong JSON

- string: phải đặt trong cặp dấu ngoặc kép “”
- number: là một số nguyên hoặc là một số thực
- object: một đối tượng thuộc kiểu JSON
- array: một mảng được bao trong cặp dấu ngoặc vuông []
- Boolean
- NULL



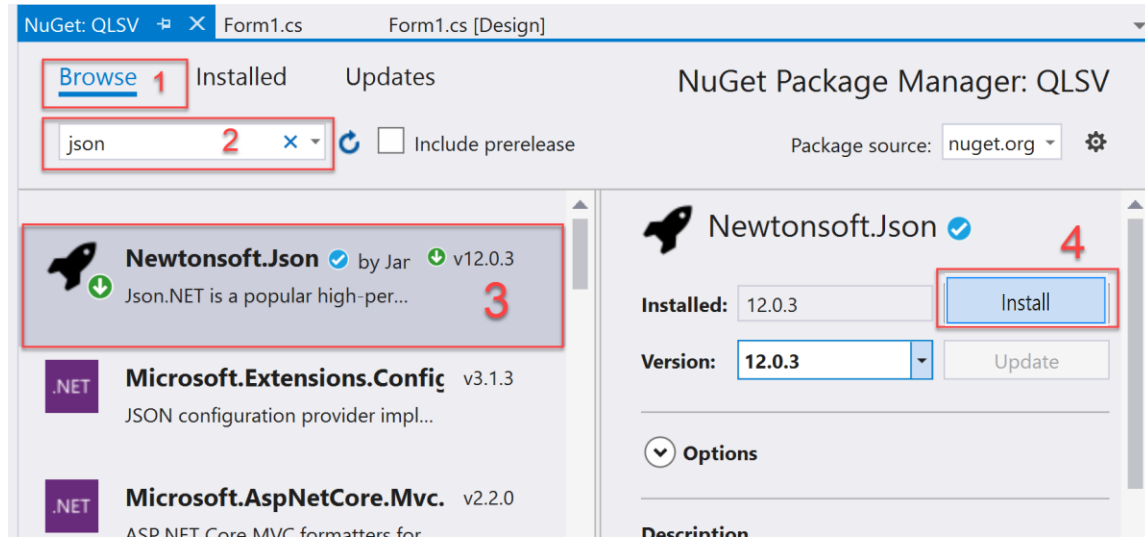
Cài đặt JSON vào C#

- Bấm chuột phải vào References
- Chọn Manage NuGet Packages

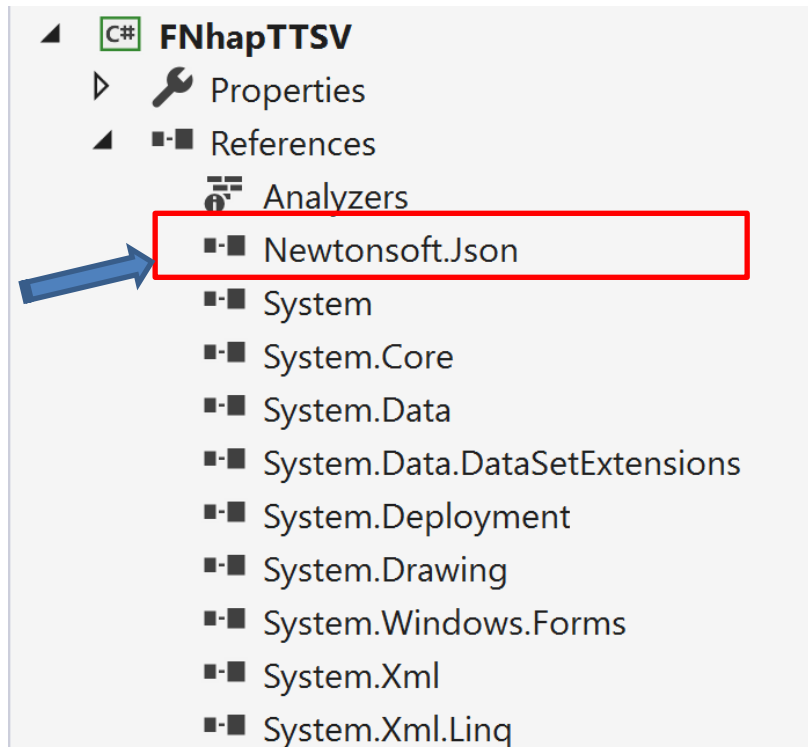


Cài đặt JSON vào C#

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



Cài đặt JSON vào C#



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



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);
```



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();  
reader.Close();//đóng file sau khi đọc
```



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

- 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 1

- 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 1

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

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



Cách làm 1

- Chuyển đổi bảng thành chuỗi JSON

```
string jsonstr;  
//chuyển đổi bảng thành chuỗi JSON  
jsonstr = JsonConvert.SerializeObject(dtTinh);
```



Cách làm 1

- Chuyển đổi chuỗi JSON thành bảng

//Chuyển đổi chuỗi json thành bảng

```
dtTinh = JsonConvert.DeserializeObject<DataTable>(jsonstr);
```

//Gán dữ liệu của bảng vào datagridview

```
dgvTinh.DataSource = dtTinh;
```



Cách làm 2

- 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)
    {
        id = i; name = n;
    }
}
```



Cách làm 2

- 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 2

- 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 2

- 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 2

- 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));
```



Bài tập

Quản lý sinh viên

Tỉnh Khoa Lớp **Sinh viên**

Mã SV Nơi sinh

Họ tên Khoa

Giới tính ☐ Nam ☐ Nữ Lớp

Ngày sinh

Mã SV	Họ tên	Ngày sinh	Giới tính	Quê quán	Khoa	Lớp
-------	--------	-----------	-----------	----------	------	-----

- Tạo form cho phép thêm, sửa, xóa các loại dữ liệu Tỉnh, Khoa, Lớp và thông tin Sinh viên.
- Trong đó dữ liệu của Tỉnh, Khoa, Lớp trong giao diện thông tin sinh viên được cập nhật ngay khi cập nhật các thông tin đó ở các giao diện tương ứng.