

## Bài 9.3: Đọc ghi file JSON

---

- ✓ Tổng quan về JSON
- ✓ Đọc dữ liệu từ file JSON
- ✓ Ghi dữ liệu ra file JSON
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

# Tổng quan về JSON

- ✓ JSON viết tắt của JavaScript Object Notation. Là một định dạng chuẩn cho lưu trữ và vận chuyển dữ liệu giữa các thiết bị đầu cuối dạng text.
- ✓ JSON dễ đọc hiểu và có cú pháp tự bản thân mô tả ý nghĩa sử dụng.
- ✓ Quy tắc cú pháp của JSON:
  - ✓ Dữ liệu lưu trữ dạng cặp name:value.
  - ✓ Các dữ liệu được phân tách nhau bằng dấu phẩy.
  - ✓ Đối tượng được bao bởi cặp ngoặc nhọn {}.
  - ✓ Mảng các đối tượng được bao bởi cặp ngoặc vuông [].
- ✓ Giá trị của phần value có thể là string, số, đối tượng, mảng, giá trị null, giá trị kiểu bool.
- ✓ Để thao tác thuận tiện với JSON trong C# ta cài gói mở rộng Newtonsoft.Json.

# Hình ảnh về tài liệu JSON

```
{
  "students": [
    {
      "id": "SV101",
      "age": 20,
      "major": "CNTT",
      "gpa": 3.25,
      "fullname": {
        "first": "Hoàng",
        "last": "Nguyễn",
        "mid": "Đình"
      },
      "address": {
        "wards": "Trung Hòa",
        "district": "Cầu Giấy",
        "city": "Hà Nội"
      }
    },
    ...
  ],
  ...
}
```

name

value

giá trị số

phân tách

string

Đối tượng

mảng

# Đọc file JSON

- ✓ Khi nói về đọc file JSON, ta ngầm hiểu là đọc dữ liệu ra từ file text hoặc tải dữ liệu về từ internet.
- ✓ Sau đó chuyển đổi dữ liệu từ dạng string sang đối tượng cần sử dụng trong C#.
- ✓ Quá trình chuyển đổi từ dữ liệu dạng JSON sang đối tượng trong C# gọi là deserialization(sự giải tuần tự hóa). Quá trình ngược lại gọi là serialization(sự tuần tự hóa).
- ✓ Để thực hiện được việc chuyển đổi, ta sẽ tạo các lớp mô tả dữ liệu của các đối tượng trong file JSON. Sau đó tiến hành:
  - ✓ Nếu dữ liệu đọc được mô tả đối tượng JSON, ta dùng lớp JObject.Parse(str) để tạo JSON object.
  - ✓ Nếu dữ liệu đọc được mô tả một mảng các đối tượng, dùng lớp JSONArray.Parse(str) để tạo đối tượng mảng JSON object.
  - ✓ Tiếp đó chuyển đổi đối tượng hoặc token nhận được sang lưu trữ trong đối tượng trong C# bằng phương thức ToObject<Type>().

# Ví dụ: lớp Student

```
13 references
class Student // lớp mô tả thông tin sinh viên
{
    [JsonProperty("id")]
    2 references
    public string Id { get; set; }
    [JsonProperty("age")]
    2 references
    public int Age { get; set; }
    [JsonProperty("major")]
    2 references
    public string Major { get; set; }
    [JsonProperty("gpa")]
    2 references
    public float Gpa { get; set; }
    [JsonProperty("fullname")]
    2 references
    public FullName FullName { get; set; }
    [JsonProperty("address")]
    2 references
    public Address Address { get; set; }

    0 references
    public Student() { }
    5 references
    public Student(string id, int age, string major,
        float gpa, FullName fullName, Address address) ...
    0 references
    public override string ToString() ...
}
```

# Ví dụ: lớp FullName và Address

```
internal class FullName
```

```
{  
    [JsonProperty("first")]  
    public string FirstName { get; set; }  
    [JsonProperty("last")]  
    public string LastName { get; set; }  
    [JsonProperty("mid")]  
    public string MidName { get; set; }  
  
    public FullName() { }  
    public FullName(string first, string last, string mid) ...  
    public override string ToString() ...  
}
```

```
9 references  
internal class Address
```

```
{  
    [JsonProperty("wards")]  
    public string Wards { get; set; }  
    [JsonProperty("district")]  
    public string District { get; set; }  
    [JsonProperty("city")]  
    public string City { get; set; }  
  
    public Address() { }  
    public Address(string ward, string district, string city) ...  
    public override string ToString() ...  
}
```



# Ví dụ đọc và chuyển đổi JSON sang C# object

```
var fileName = "data.json";
FileInfo file = new FileInfo(fileName);
using (var reader = file.OpenText())
{
    var data = reader.ReadToEnd();
    var jobject = JObject.Parse(data);
    var peopleJToken = jobject["students"];
    List<Student> students = new List<Student>();
    foreach (var item in peopleJToken)
    {
        //people.Add(JsonConvert.DeserializeObject<Person>(item.ToString()));
        students.Add(item.ToObject<Student>());
    }
    ShowData(students);
}
```

C:\WINDOWS\system32\cmd.exe

```
Student[Id=SV101, Age=20, Major=CNTT, Gpa=3.25, FullName=Nguyễn Đình Hoàng, Address=Trung Hòa - Cầu Giấy - Hà Nội]
Student[Id=SV102, Age=21, Major=CNTT, Gpa=3.45, FullName=Văn Tiến Lâm, Address=Nghĩa Đô - Cầu Giấy - Hà Nội]
Student[Id=SV103, Age=24, Major=CNTT, Gpa=3.75, FullName=Lưu Thế Mạnh, Address=Dịch Vọng - Cầu Giấy - Hà Nội]
Student[Id=SV104, Age=21, Major=CNTT, Gpa=3.15, FullName=Lưu Đức Bình, Address=Mai Dịch - Cầu Giấy - Hà Nội]
Student[Id=SV105, Age=20, Major=CNTT, Gpa=3.15, FullName=Đinh Thị Mai, Address=Quan Hoa - Cầu Giấy - Hà Nội]
Press any key to continue . . .
```

# Quá trình đọc và chuyển đổi

- ✓ B1: xác định file JSON và định dạng lưu trong file đó.
- ✓ B2: đọc toàn bộ dữ liệu trong file với `ReadToEnd()`.
- ✓ B3: chuyển đổi tương tự đọc được dạng string sang đối tượng JSON với `JsonObject.Parse()`.
- ✓ B4: lấy mảng student bằng cách truy cập vào `jObject["students"]` với `jObject` là tên đối tượng `JsonObject`, `students` là tên của trường dữ liệu trong đối tượng javascript object.
- ✓ B5: chuyển đổi từng `JToken` sang dạng đối tượng trong c# với phương thức `ToObject<T>()`. Với `T` là tên kiểu dữ liệu trong C#.



# Ghi file JSON

- ✓ Để ghi file JSON, trước hết ta phải chuyển các thuộc tính, trường dữ liệu của đối tượng về dạng text chuẩn JSON.
- ✓ Ta có thể chuyển đổi một đối tượng, 1 mảng, 1 collection về dạng tài liệu JSON với phương thức `JsonConvert.SerializeObject(object, format);`
- ✓ Sau đó ghi dữ liệu dạng text đã chuyển đổi sang JSON vào file có đuôi .json ta được file JSON.

# Ví dụ



1 reference

```
static Student[] CreateStudents()
{
    return new Student[] {
        new Student("SV101", 20, "CNTT", 3.25f,
            new FullName("Hoàng", "Nguyễn", "Đình"),
            new Address("Trung Hòa", "Cầu Giấy", "Hà Nội")),
        new Student("SV102", 21, "CNTT", 3.45f,
            new FullName("Lâm", "Văn", "Tiến"),
            new Address("Nghĩa Đô", "Cầu Giấy", "Hà Nội")),
        new Student("SV103", 24, "CNTT", 3.75f,
            new FullName("Mạnh", "Lưu", "Thế"),
            new Address("Dịch Vọng", "Cầu Giấy", "Hà Nội")),
        new Student("SV104", 21, "CNTT", 3.15f,
            new FullName("Bình", "Lưu", "Đức"),
            new Address("Mai Dịch", "Cầu Giấy", "Hà Nội")),
        new Student("SV105", 20, "CNTT", 3.15f,
            new FullName("Mai", "Đình", "Thị"),
            new Address("Quan Hoa", "Cầu Giấy", "Hà Nội"))
    };
}
```

# Ví dụ

- ✓ Gói dữ liệu trong một lớp vô danh với tên thuộc tính là students sau đó chuyển dữ liệu sang dạng JSON.
- ✓ Cuối cùng ghi vào file:

```
// gói data bên trong một lớp vô danh có tên là tên phần tử muốn tạo
// để tạo name
var root = new
{
    students = CreateStudents()
};
var strJson = JsonConvert.SerializeObject(root, Formatting.Indented);
Console.WriteLine(strJson);
File.WriteAllText("data.json", strJson);
```



# Nội dung tiếp theo

## Thao tác với CSDL