# CHƯƠNG 2
# BIỂU DIỄN DỮ LIỆU BÁN CẤU TRÚC

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC ĐÀ LẠT

## Mục tiêu

- Giới thiệu một số kiểu dữ liệu bán cấu trúc thông dụng:
  - XML
  - JSON
  - BSON

## Giới thiệu

- Khi xây dựng một chương trình, thường thì chương trình đó phải kết nối với cơ sở dữ liệu để đọc và truy vấn dữ liệu để xử lý.
- Có ba loại dữ liệu được sử dụng là:
  - Dữ liệu phi cấu trúc (*Unstructured Data*)
  - Dữ liệu có cấu trúc (*Structured Data*)
  - Dữ liệu bán cấu trúc (*Semi-Structured Data*)

## Giới thiệu

| Dữ liệu phi cấu trúc | Dữ liệu có cấu trúc | Dữ liệu bán cấu trúc |
|---|---|---|
| • Hồ sơ<br>• Văn bản<br>• Nội dung cuộc gọi điện thoại | • Mẫu tin, trường<br>• Dữ liệu được lưu trong CSDL | • Không nằm trong CSDL nhưng lại có cấu trúc và thuộc tính |

# Giới thiệu

- Đối với các chương trình có kết nối với cơ sở dữ liệu:
  - Thường kết nối với cơ sở dữ liệu dạng có cấu trúc để dễ quản lý
  - Một số chương trình nhỏ, gọn, đơn giản có thể sử dụng dữ liệu dạng bán cấu trúc để lưu trữ và xử lý dữ liệu

# Dữ liệu bán cấu trúc

- Là dạng dữ liệu mà thông tin không được tổ chức và lưu trữ trong hệ quản trị cơ sở dữ liệu nhưng lại có các thuộc tính để quy định cách tổ chức và phân tích dữ liệu một cách dễ dàng
  - Không lộn xộn và khó kiểm soát như dữ liệu phi cấu trúc
  - Không cứng nhắc và dễ định lượng như dữ liệu có cấu trúc
- Ví dụ: sử dụng thẻ HTML để lưu nội dung văn bản Web

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

# Một số loại dữ liệu bán cấu trúc

| Kiểu dữ liệu | Ví dụ | Mô tả |
|---|---|---|
| **Kiểu dữ liệu dạng tập tin** | Là dạng dữ liệu được tạo ra bởi các tập tin, mỗi tập tin chứa nhiều mẫu tin là dòng dữ liệu, mỗi dòng có thể có các cột. | CSV, Excel, Foxpro, File Text… |
| **Kiểu dữ liệu dạng thẻ** | Kiểu dữ liệu cũng dạng tập tin nhưng trong đó chứa các thẻ để chứa dữ liệu, mỗi thẻ có thể có các thuộc tính để phân biệt. | XML, XAML, AIML, HTML… |
| **Kiểu dữ liệu dạng trường cấu trúc** | Là các kiểu dữ liệu có các trường để chứa thông tin, các trường này do người dùng định nghĩa. | JSON, BSON… |
| **Kiểu dữ liệu dạng cấu trúc quy ước** | Là dạng dữ liệu có các trường được tạo ra theo một quy định cho trước. Khi sử dụng kiểu này, người dùng phải lưu thông tin theo các trường có sẵn. | Email, RDF… |

# Kiểu dữ liệu XML

- XML (eXtensible Markup Language)
- Ngôn ngữ đánh dấu mở rộng
- Một chuẩn ngôn ngữ do W3C (World Wide Web Consortium) đề xuất
- Một ngôn ngữ đánh dấu bởi các thẻ và có thể mô tả được nhiều kiểu dữ liệu khác nhau
- Cho phép người dùng tự định nghĩa thẻ để lưu dữ liệu
- Các thẻ được định nghĩa tuân thủ theo cấu trúc cây với các nút lồng nhau

# Kiểu dữ liệu XML

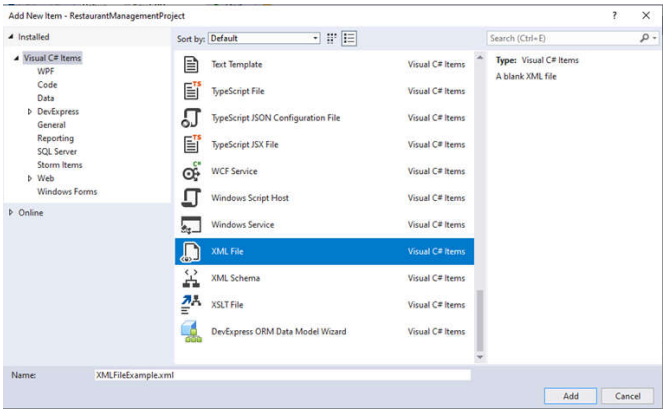
- Các công cụ biên tập: iTaxViewer, Notepad, Visual Studio
- Tập tin XML có phần mở rộng là *.xml
- Cú pháp: <tennut>nội dung</tennut>
  - Trong đó:
    - <tennut> là thẻ mở của nút;
    - </tennut> là thẻ đóng của nút
    - phần ở giữa là nội dung dữ liệu cần chứa.
  - Tên nút do người dùng tự định nghĩa, theo quy tắc không có khoảng trắng, phải bắt đầu bằng chữ cái.

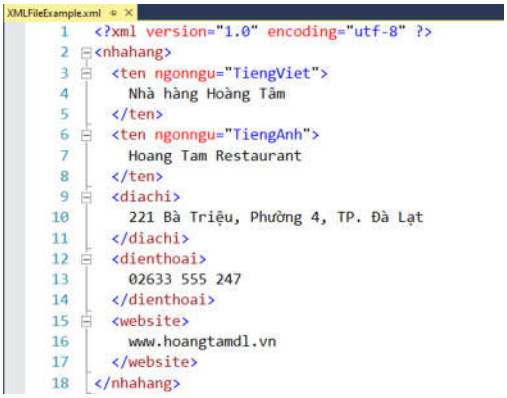
# Kiểu dữ liệu XML

```
<?xml version="1.0" encoding="utf-8" ?>
<nhahang>
    <tentiengViet>
        Nhà hàng Hoàng Tâm
    </tentiengViet>
    <tentiengAnh>
        Hoang Tam Restaurant
    </tentiengAnh>
    <diachi>
        221 Bà Triệu, Phường 4, TP. Đà Lạt
    </diachi>
    <dienthoai>
        02633 555 247
    </dienthoai>
    <website>
        www.hoangtamdl.vn
    </website>
</nhahang>
```

# Tạo tập tin XML bằng Visual Studio

- Click phải lên dự án, chọn Add, chọn New Item




# Kiểu dữ liệu JSON

- JSON (Javascript Object Notation)
- Một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình có thể đọc được
- Có cấu trúc đơn giản, dễ sử dụng → được ứng dụng khá phổ biến, đặc biệt cho các ứng dụng trên Web
- Được dùng để tạo ra các tập tin API

# Kiểu dữ liệu JSON

- Các công cụ biên tập: bất kỳ trình soạn thảo nào
- Tập tin JSON có phần mở rộng là *.json hoặc *.js
- Cú pháp: tuân theo dạng cặp ***key-value***, có hỗ trợ cấu trúc dữ liệu như đối tượng hoặc mảng, bao gồm các thành phần sau:
  - Dữ liệu được biểu diễn dưới dạng cặp: *key-value*;
  - Các dấu ngoặc nhọn lưu trữ tên các đối tượng, tiếp theo là dấu hai chấm, sau đó là giá trị; Các cặp tên/giá trị được phân tách bằng dấu phẩy; Đối tượng hoặc giá trị được để trong dấu ngoặc kép;
  - Dấu ngoặc vuông biểu diễn kiểu dữ liệu mảng, các giá trị cũng được phân cách bằng dấu phẩy

# Kiểu dữ liệu JSON

```
{
    "nhahang": {
        "tentiengViet": "Nhà hàng Hoàng Tâm",
        "tentiengAnh": "Hoang Tam restaurant",
        "diachi": "221 Bà Triệu, Phường 4, TP. Đà Lạt",
        "dienthoai": "02633 555 247",
        "website": "www.hoangtamdl.vn"
    }
}
```

```
{
    "nhahang": [
     {
         "tentiengViet": "Nhà hàng Hoàng Tâm",
         "tentiengAnh": "Hoang Tam restaurant",
         "diachi": "221 Bà Triệu, Phường 4, TP. Đà Lạt",
         "dienthoai": "02633 555 247",
         "website": "www.hoangtamdl.vn"
     },
     {
         "tentiengViet": "Nhà hàng Hoàng Tâm 2",
         "tentiengAnh": "Hoang Tam 2 restaurant",
         "diachi": "250 Phan Đình Phùng, Phường 2, TP. Đà Lạt",
         "dienthoai": "02633 666 257",
         "website": "www.hoangtamdl.vn"
     }
     ]
}
```

# Các kiểu dữ liệu trong JSON

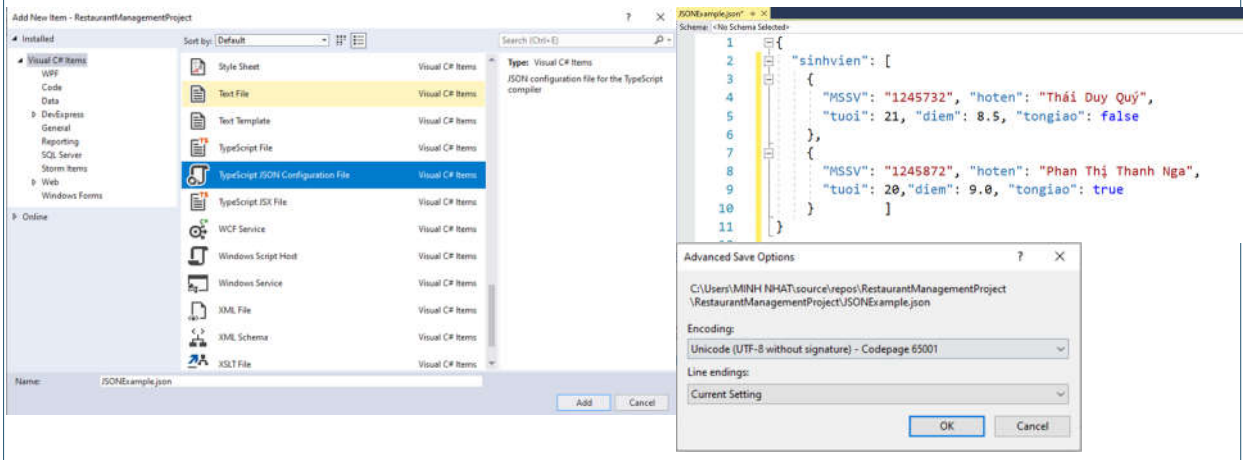
| Kiểu dữ liệu | Mô tả | Ví dụ |
|---|---|---|
| **Number** | Cho phép lưu trữ số nguyên, số thực, hoặc số mũ. | "sinhvien":{<br>"tuoi": 20,<br>"diem":7.5<br>} |
| **String** | Cho phép lưu trữ chuỗi ký tự, chuỗi ký tự phải nằm trong dấu nháy kép. | "sinhvien":{<br>"ho": "Nguyễn",<br>"tenlot": "Văn",<br>"ten": "An"<br>} |
| **Boolean** | Kiểu dữ liệu true hoặc false | "sinhvien":{<br>"tongiao":false<br>} |

# Các kiểu dữ liệu trong JSON

| Kiểu dữ liệu | Mô tả | Ví dụ |
|---|---|---|
| **Array** | Kiểu dữ liệu mảng, sử dụng dấu ngoặc vuông để biểu diễn nhiều dữ liệu | "sinhvien":[<br>{<br>"hoten": "Nguyễn An",<br>"tuoi": 19,<br>"tongiao": false<br>},<br>{<br>"hoten": "Phạm Quốc",<br>"tuoi": 20,<br>"tongiao": true<br>} |
| **Object** | Là một tập hợp các cặp key-value không theo thứ tự, các cặp này được đặt trong dấu ngoặc, key luôn là chuỗi và theo sau là dấu hai chấm | {"id": 1245213, "hoten": "Nguyễn An", "tuoi": 20, "diemtongket": 7.78} |

# Tạo tập tin JSON bằng Visual Studio

- Click phải lên dự án, chọn Add, chọn New Item




# Kiểu dữ liệu JSON

- Demo cách đọc tập tin JSON

# Kiểu dữ liệu BSON

- BSON (Binary JSON)
- Là một kiểu mã hoá và lưu trữ dữ liệu dạng bán cấu trúc giống như JSON, có hỗ trợ thêm kiểu nhị phân (*BinData*) và kiểu ngày tháng (*DateData*)
- Ưu điểm so với JSON: giúp lưu trữ nhẹ hơn, dễ duyệt hơn, và hiệu quả hơn trong quá trình mã hoá và giải mã.
- Thường được dùng trong hệ CSDL MongoDB
- Đều được hỗ trợ bởi các ngôn ngữ cơ bản như C, C#, Java, Python, …

# Mối quan hệ giữa JSON và BSON

| JSON | BSON | Giải thích |
|---|---|---|
| {"hello":"world"} | \x16\x00\x00\x00 | // Kích thước tập tin |
| | \x02 | // 0x02 là kiểu String |
| | hello\x00 | // trường khoá |
| | \x06\x00\x00\x00world\x00 | // trường giá trị |
| | \x00 | // 0x00 là kiểu EOO ('kết thúc tập tin') |
| { | \x31\x00\x00\x00 | // Kích thước tập tin |
| "BSON": | \x04BSON\x00 | // Trường khoá |
| [ | \x26\x00\x00\x00 | |
| "awesome", | \x02\x30\x00\x08\x00\x00\x00awesome\x00 | //Các giá trị trong mảng |
| 5.05, | \x01\x31\x00\x33\x33\x33\x33\x33\x33\x14\x40 | |
| 1986 | \x10\x32\x00\xc2\x07\x00\x00 | |
| ] | \x00 | |
| } | \x00 | |
| | | //Kết thúc |

# Kiểu dữ liệu sử dụng trong BSON

| Kiểu dữ liệu | Mô tả |
|---|---|
| **unicode** | Kiểu dữ liệu chuỗi, sử dụng UTF-8 |
| **integer** | Kiểu số nguyên 32bit hoặc 64bit. |
| **double** | Kiểu dữ liệu thực, sử dụng dấu phẩy động 64bit của IEEE 754. |
| **decimal** | Kiểu số thập phân, theo chuẩn 128-bit IEEE 754-2008. |
| **byte array** | Kiểu dữ liệu mảng, sử dụng các byte để lưu trữ. |
| **boolean** | Kiểu bit, trả về true hoặc false |
| **null** | Kiểu null |
| **object** | Kiểu đối tượng, một đối tượng là một cặp khoá-giá trị |
| **MD5 binary data** | Kiểu mã hoá nhị phân MD5 |