External Format Processing

Parsing JSON, JSON.NET

SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Contents

Have a Question?

SoftUni Foundation

sli.do

#csharp-db

The JSON Data Format

Definition and Syntax

JSON Data Format

SoftUni Foundation

- **JSON** (**J**ava**S**cript **O**bject **N**otation) is a lightweight data format
  - Human and machine-readable plain text
  - Based on **JavaScript** objects
  - Independent of development platforms and languages
  - JSON data consists of:
    - Values (strings, numbers, etc.)
    - Key-value pairs: **{ key : value }**
    - Arrays: **[value1, value2, …]**

JSON Data Format (2)

SoftUni Foundation

- The JSON data format follows the rules of object creation in JS
  - **Strings**, **numbers** and **Booleans** are valid JSON:

```
"this is a string and is valid JSON"
```

```
3.14
```

```
true
```

  - **Arrays** are valid JSON:

```
[5, "text", true]
```

  - **Objects** are valid JSON (key-value pairs):

```
{
  "firstName": "Vladimir", "lastName": "Georgiev",
  "jobTitle": "Technical Trainer", "age": 25
}
```

Processing JSON

Parsing JSON in C# and .NET

Built-in JSON Support

- .NET has a built-in **DataContractJsonSerializer** class
- Contained in **System.Runtime.Serialization** assembly
- Supports **deserializing** (parsing) strings and **serializing** objects
- Including **DataContractJsonSerializer** into a project:

```
using System.Runtime.Serialization.Json;
```

Serializing JSON

SoftUni Foundation

- DataContractJsonSerializer can serialize an object:

```
static string SerializeJson<T>(T obj)
{
  var serializer = new DataContractJsonSerializer(obj.GetType());
  using (var stream = new MemoryStream())
  {
    serializer.WriteObject(stream, obj);
    var result = Encoding.UTF8.GetString(stream.ToArray());
    return result;
  }
}
```

```
var product = new Product();
```

product {Json.Product}

Cost	25
Description	null
Id	0
Name	"Oil Pump"

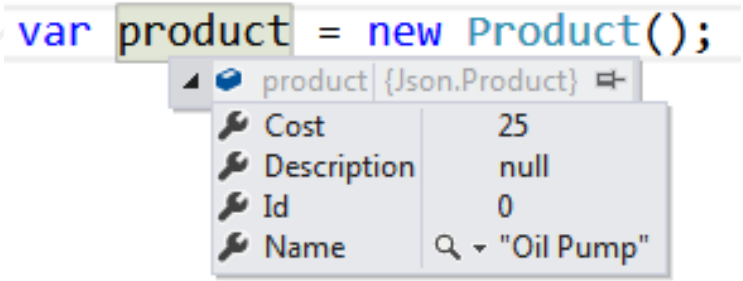
```
{
  "Id":0,
  "Name":"Oil Pump",
  "Description":null,
  "Cost":25
}
```

Deserializing JSON

- DataContractJsonSerializer can deserialize a JSON string:

```
static T DeserializeJson<T>(string jsonString)
{
  var serializer = new DataContractJsonSerializer(typeof(T));
  var jsonStringBytes = Encoding.UTF8.GetBytes(jsonString);
  using (var stream = new MemoryStream(jsonStringBytes))
  {
    var result = (T)serializer.ReadObject(stream);
    return result;
  }
}
```

```
{
  "Id":0,
  "Name":"Oil Pump",
  "Description":null,
  "Cost":25
}
```



JSON.NET

Better JSON Parsing for .NET Developers

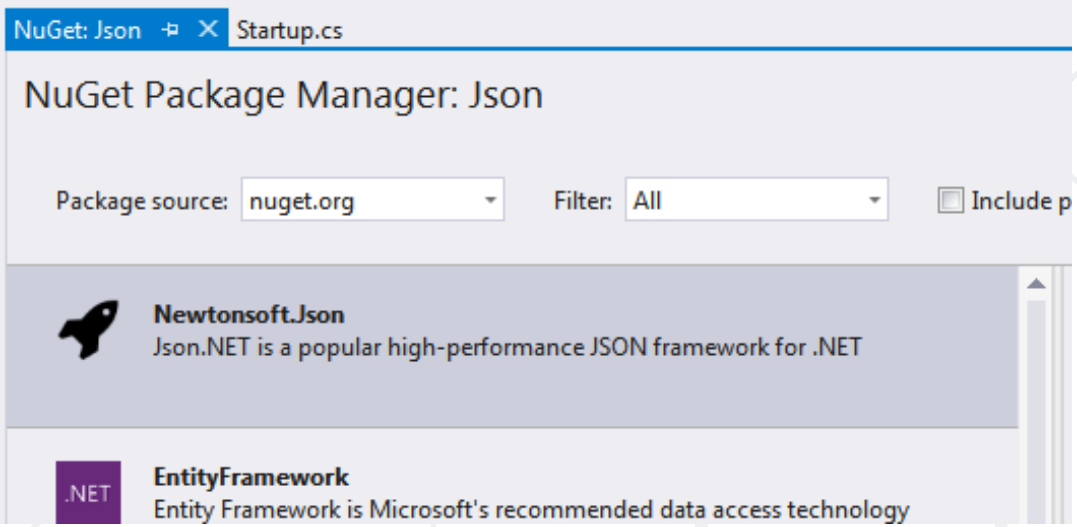
What is JSON.NET?

SoftUni Foundation

- **JSON.NET** is a JSON **framework** for .NET
  - **More functionality** than built-in functionality
  - Supports **LINQ-to-JSON**
  - Out-of-the-box support for parsing between **JSON** and **XML**
  - Open-source project: **http://www.newtonsoft.com**
  - Json.NET vs .NET Serializers: **https://www.newtonsoft.com/json/help/html/JsonNetVsDotNetSerializers.htm**

Installing JSON.NET

- To install JSON.NET use the **NuGet Package Manager**:



- Or with a command in the Package Manager Console:

```
Install-Package Newtonsoft.Json
```

General Usage

- JSON.NET exposes a static service **JsonConvert**
- Used for parsing and configuration
  - To **Serialize** an object:

```
var jsonProduct = JsonConvert.SerializeObject(product);
```

  - To **Deserialize** an object:

```
var objProduct =
  JsonConvert.DeserializeObject<Product>(jsonProduct);
```

JSON.NET Features

SoftUni Foundation

- JSON.NET can be configured to:
  - **Indent** the output JSON string
  - To convert JSON to **anonymous types**
  - To control the **casing** and **properties** to parse
  - To skip errors
- JSON.NET also supports:
  - LINQ-to-JSON
  - Direct parsing between XML and JSON

Configuring JSON.NET

- By default, the result is a **single line of text**
- To indent the output string use **Formatting.Indented**

```
JsonConvert.SerializeObject(products, Formatting.Indented);
```

```
{
  "pump": {
    "Id": 0,
    "Name": "Oil Pump",
    "Description": null,
    "Cost": 25.0
  },
  "filter": {
    "Id": 0,
    "Name": "Oil Filter",
    "Description": null,
    "Cost": 15.0
  }
}
```

Configuring JSON.NET

SoftUni Foundation

- Deserializing to **anonymous** types:

Incoming JSON

```
var json = @"{ 'firstName': 'Vladimir',
               'lastName': 'Georgiev',
               'jobTitle': 'Technical Trainer' }";
var template = new
{
    FirstName = string.Empty,
    LastName = string.Empty,
    Occupation = string.Empty
};
var person = JsonConvert.DeserializeAnonymousType(json, template);
```

Template objects

JSON.NET Parsing of Objects

SoftUni Foundation

- By default JSON.NET takes each property / field from the class and parses it
  - This can be controlled using attributes:

```
public class User
{
  [JsonProperty("user")]
  public string Username { get; set; }

  [JsonIgnore]
  public string Password { get; set; }
}
```

Parse Username to user

Skip the property

JSON.NET Parsing of Objects

SoftUni Foundation

- By default JSON.NET takes each property / field from the class and parses it
  - This can be controlled using **ContractResolver**:

```
DefaultContractResolver contractResolver = new DefaultContractResolver()
{
    NamingStrategy = new SnakeCaseNamingStrategy()
};

var serialized = JsonConvert.SerializeObject(person, new JsonSerializerSettings()
{
    ContractResolver = contractResolver,
    Formatting = Formatting.Indented
});
```

LINQ-to-JSON

- LINQ-to-JSON works with JObjects
  - Create from JSON string:

```
JObject obj = JObject.Parse(jsonProduct);
```

  - Reading from file:

```
var people = JObject.Parse(File.ReadAllText(@"c:\people.json"))
```

  - Using **JObject**:

```
foreach (JToken person in people)
{
  Console.WriteLine(person["FirstName"]); // Ivan
  Console.WriteLine(person["LastName"]); // Petrov
}
```

LINQ-to-JSON (2)

SoftUni Foundation

- JObjects can be queried with LINQ

```
var json = JObject.Parse(@"{'products': [
  {'name': 'Fruits', 'products': ['apple', 'banana']},
  {'name': 'Vegetables', 'products': ['cucumber']}]}");

var products = json["products"].Select(t =>
  string.Format("{0} ({1})",
    t["name"],
    string.Join(", ", c["products"])
))

// Fruits (apple, banana)
// Vegetables (cucumber)
```

XML-to-JSON

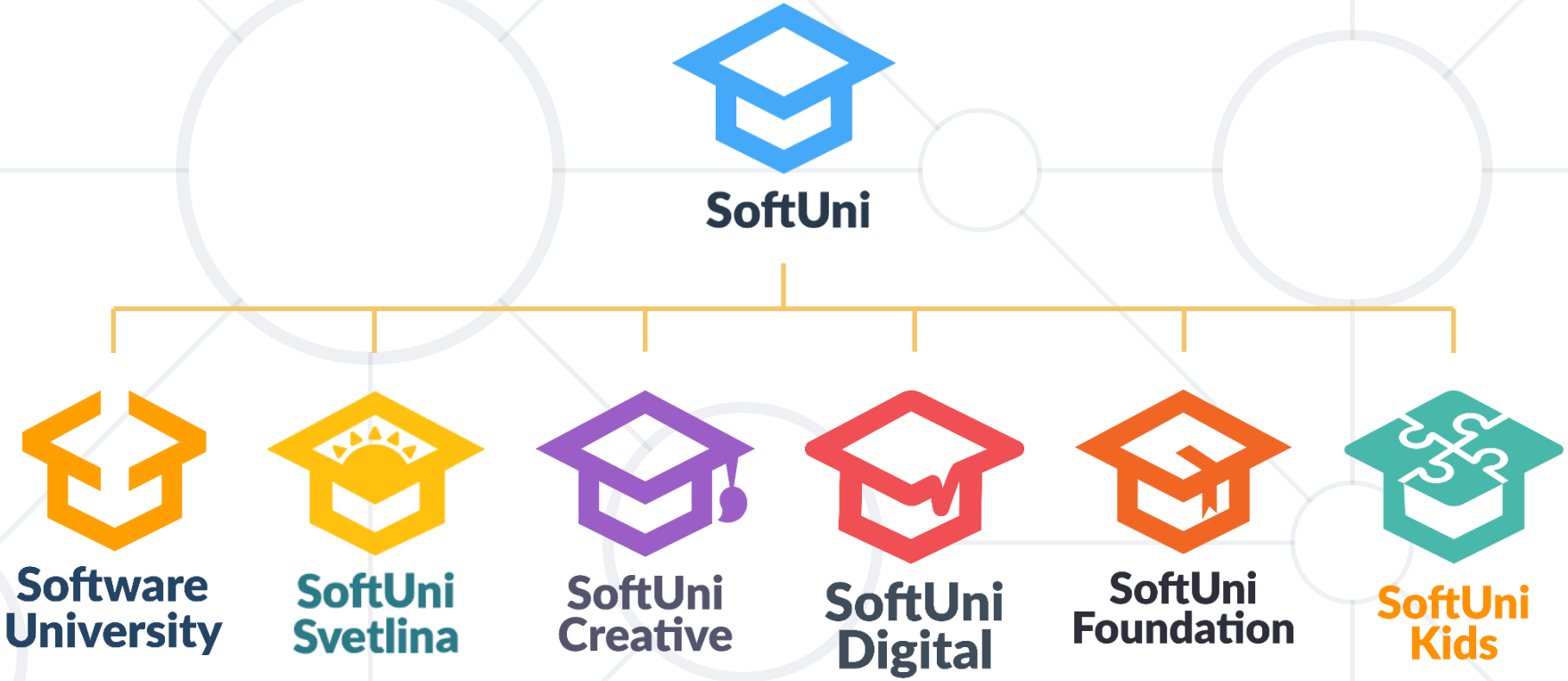
```
string xml = @"<?xml version='1.0' standalone='no'?>
 <root>
    <person id='1'>
        <name>Alan</name>
        <url>www.google.com</url>
    </person>
    <person id='2'>
        <name>Louis</name>
        <url>www.yahoo.com</url>
    </person>
</root>";

XmlDocument doc = new XmlDocument();
doc.LoadXml(xml);
string jsonText = JsonConvert.SerializeXmlNode(doc);
```

```
{
    "?xml": {
    "@version": "1.0",
    "@standalone": "no"
    },
    "root": {
        "person": [
        {
            "@id": "1",
            "name": "Alan",
            "url": "www.google.com"
        },
        {
            "@id": "2",
            "name": "Louis",
            "url": "www.yahoo.com"
        }
        ]
    }
}
```

Summary

- JSON is cross platform data format
- **DataContractJsonSerializer** is the default JSON Parser in C#
- **JSON.NET** is a fast framework for working with JSON data

Questions?


SoftUni
Software University
SoftUni Svetlina
SoftUni Creative
SoftUni Digital
SoftUni Foundation
SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech
we know sports

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

INDEAVR
Serving the high achievers

INFRAGISTICS

STEMO®
Computer Systems & Software

SUPERHOSTING.BG

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

One SOFTW

Lukanet.com

WORLD OF MYTHS

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license