

Getting Started with JSON in C# Using Json.NET

THE CASE FOR JSON AND JSON.NET



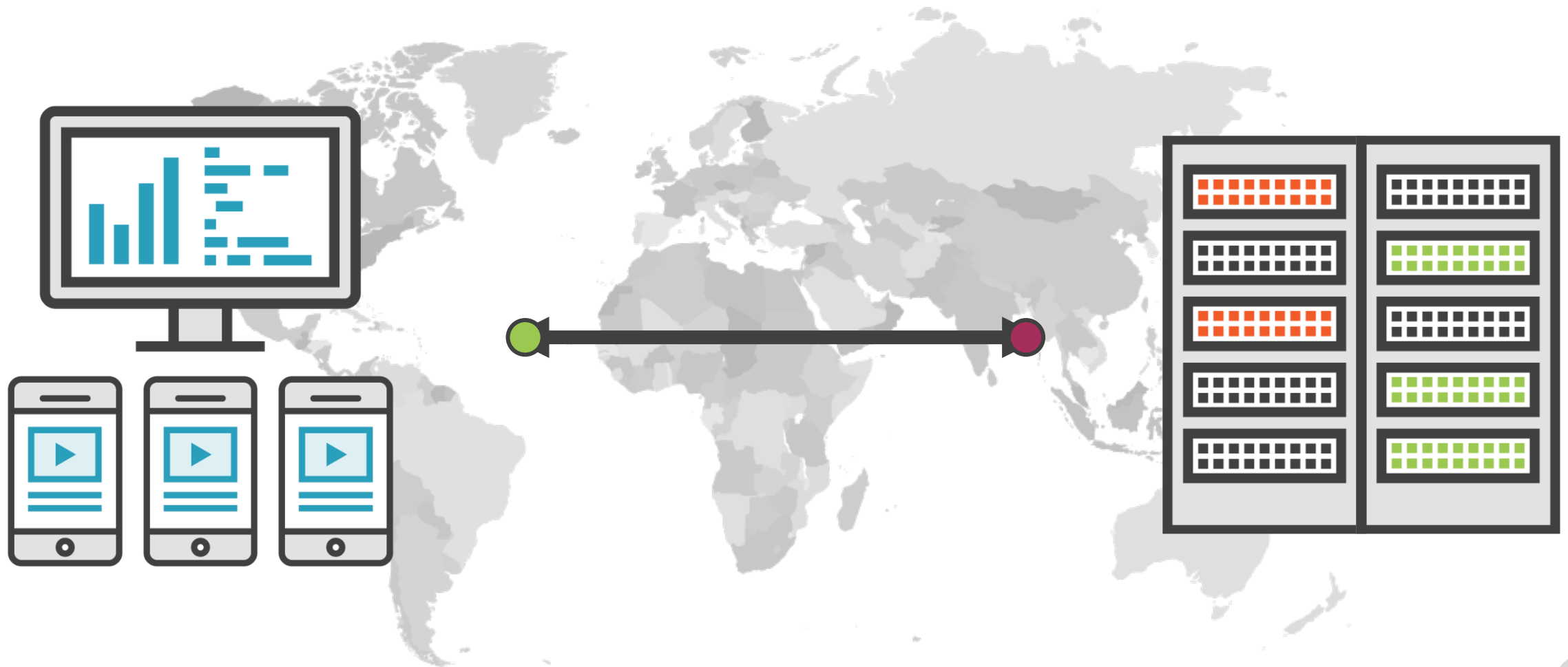
Xavier Morera

HELPING DEVELOPERS UNDERSTAND SEARCH & BIG DATA

@xmorera www.xaviermorera.com



The Case for JSON and Json.NET



Used By

MongoDB

Solr

ElasticSearch

Microsoft SQL Server



Search for packages...



Newtonsoft.Json 12.0.3

Json.NET is a popular high-performance JSON framework for .NET

Requires NuGet 2.12 or higher.

[Package Manager](#)[.NET CLI](#)[PackageReference](#)[Paket CLI](#)

```
PM> Install-Package Newtonsoft.Json -Version 12.0.3
```



> Dependencies

> GitHub Usage

∨ Version History

Version	Downloads	Last updated
12.0.3	13,357,784	5 months ago
12.0.2	37,348,690	4/22/2019
12.0.1	33,962,424	11/27/2018
11.0.2	39,712,618	3/24/2018

Info

last updated 5 months ago

[Project Site](#)

[Source repository](#)

[Dependent repositories](#) (1.8K+)

License: [MIT](#)

[Contact owners](#)

[Report](#)

[Download package](#) (2.48 MB)

[Download symbols](#) (887.73 KB)

Statistics

↓ 441,646,182 total downloads

13,357,784 downloads of current version

130,471 downloads per day (avg)

[View full stats](#)

Where to Go?

- Case for JSON and Json.NET
- LINQ to JSON
- Serialization Fundamentals
- JSON & XML
- Settings & Attributes
- Binary JSON (BSON)
- Custom Serialization
- Json.NET Schema
- Performance Tips
- System.Text.Json vs. Json.NET & .NET Core and 5



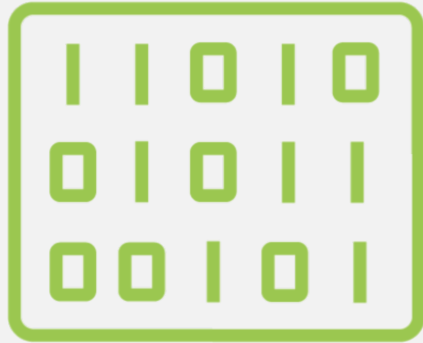


What is JSON?

What is JSON?



JavaScript
Object Notation



Data
Interchange
Format



Human
Readable Text /
Open Standard



Data Objects as
Key Value Pairs



```
{  
  "name": "Xavier Morera",  
  "courses": [  
    "Solr",  
    "Spark"  
  ],  
  "since": "2014-01-14T00:00:00",  
  "happy": true,  
  "issues": null,  
  "car": {  
    "model": "Land Rover Series III",  
    "year": 1976  
  }  
}
```

Hello JSON!



History of JSON



Douglas Crockford

Computer programmer

Douglas Crockford is an American computer programmer and entrepreneur who is best known for his ongoing involvement in the development of the JavaScript language, for having popularized the data format ... [Wikipedia](#)

Born: [Minnesota](#)

Known for: [JSON](#)

Education: [San Francisco State University](#)

JavaScript Object Notation

By Douglas Crockford

In 2001

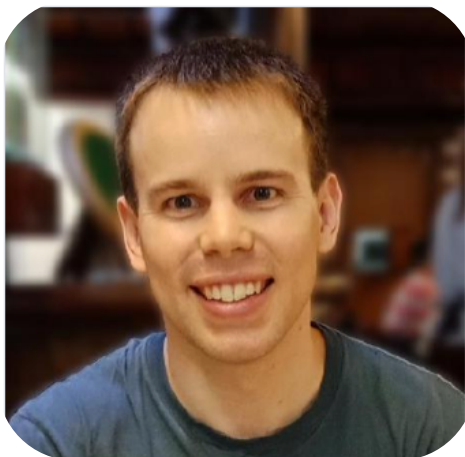
Official site

- <http://www.json.org>

Described by two standards

- RFC 8259 and ECMA-404





It's complicated

James Newton-King

JamesNK

Follow

Software Developer. Author of Json.NET.
Not Batman.

Microsoft

Wellington, New Zealand

<http://james.newtonking.com>

Block or report user

Organizations



History of Json.NET

JamesNK / [Newtonsoft.Json](#)

Used by

591k

Watch

526

★ Star

7.7k

Fork

2.6k

Code

Issues 402

Pull requests 45

Actions

Projects 0

Security

Insights

1,821 commits

20 branches

0 packages

62 releases

95 contributors

MIT

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

Silver Fix wrong define is used in StringUtils.ToLower() (#2304)

✓ Latest commit 666d976 19 days ago

Build	Update version to 13.0.1-beta and remove portable builds (#2228)	5 months ago
Doc	Update NullValueHandling.Ignore.xml (#2226)	4 months ago
Src	Fix wrong define is used in StringUtils.ToLower() (#2304)	19 days ago
.appveyor.yml	Add AppVeyor.yml file (#1417)	3 years ago
.gitattributes	-Fixed WinRT build	6 years ago
.gitignore	-Fixed serializing generated types with duplicate property names	2 years ago
CONTRIBUTING.md	HTTPs all the things (#1710)	2 years ago
ISSUE_TEMPLATE.md	HTTPs all the things (#1710)	2 years ago
LICENSE.md	Create LICENSE.md	7 years ago



Why is JSON Important?

Supported By

Ruby

Objective C

Java

C#

Python

And more



Supported Natively in JavaScript

JSON

```
var authors = [  
  { firstName: 'Xavier',  
    lastName: 'Morera',  
    courseCount: 4 },  
  { firstName: 'Lars',  
    lastName: 'Klint',  
    courseCount: 22 },  
  { firstName: 'Jason',  
    lastName: 'Alba',  
    courseCount: 16 }  
];
```

JavaScript

```
<script type="text/javascript">  
var the Author = document.getElementById("oneAuthor");  
var theAuthors = document.getElementById("allAuthors");  
var countAuthors = document.getElementById("totalAuthors");  
theAuthor.innerHTML = authors [0].firstName + " " +  
authors[0].lastName;  
theAuthors.innerHTML = authors [0].firstName + ", " + authors  
[1].lastName;  
countAuthors.innerHTML = "Total authors: " + authors.length;  
</script>
```



Developers Want Out of the Syntax Business

```
<author first-name="Xavier" last-name="Morera"/>  
<author>  
  <first-name>Xavier</first-name>  
  <last-name>Morera</last-name>  
</author>  
<object type="author">  
  <property name="first-name">Xavier</property>  
  <property name="last-name">Morera</property>  
</object>
```



XML

```
<authors>
  <author>
    <firstName>Xavier</firstName>
    <lastName>Morera</lastName>
    <courseCount>4</courseCount>
  </author>
  <author>
    <firstName>Lars</firstName>
    <lastName>Klint</lastName>
    <courseCount>22</courseCount>
  </author>
  <author>
    <firstName>Jason</firstName>
    <lastName>Alba</lastName>
    <courseCount>16</courseCount>
  </author>
</authors>
```



Fat Free Alternative to XML

```
{
  "authors": {
    "author": [
      {
        "firstName": "Xavier",
        "lastName": "Morera",
        "courseCount": "4"
      },
      {
        "firstName": "Lars",
        "lastName": "Klint",
        "courseCount": "22"
      },
      {
        "firstName": "Jason",
        "lastName": "Alba",
        "courseCount": "16"
      }
    ]
  }
}
```



Fat Free Alternative to XML

```
<nodes>  
  <node type="author" firstname="Xavier" lastname="Morera" coursecount="4"></node>  
  <node type="author" firstname="Lars" lastname="Klint" coursecount="22"></node>  
  <node type="author" firstname="Jason" lastname="Alba" coursecount="16"></node>  
</nodes>
```

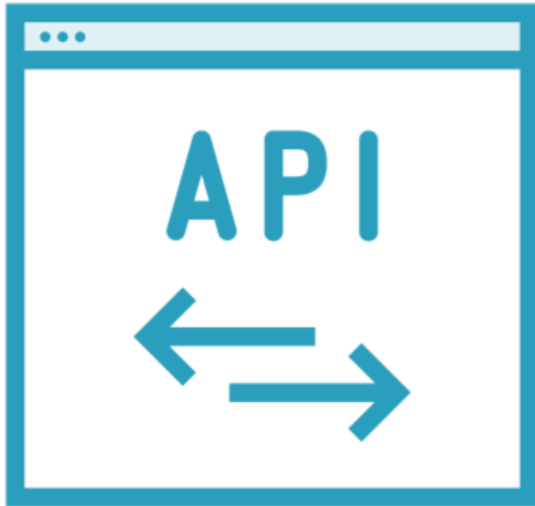


Fat Free Alternative to XML

```
<authors>
  <author>
    <firstName>Xavier</firstName>
    <lastName>Morera</lastName>
    <courseCount>4</courseCount>
  </author>
  <author>
    <firstName>Lars</firstName>
    <lastName>Klint</lastName>
    <courseCount>22</courseCount>
  </author>
  <author>
    <firstName>Jason</firstName>
    <lastName>Alba</lastName>
    <courseCount>16</courseCount>
  </author>
</authors>
```



JSON & APIs



Many applications expose an API

- REST

Different formats supported besides JSON

- Like XML

JSON's readability and being lightweight

- Suitable for data exchange

Large number of APIs

Demo



JSON in JavaScript and APIs





JSON Specification



Collection of name value pairs

Four primitive types:

- Strings
- Numbers
- Booleans
- Null

Two structured types:

- Objects
- Arrays

Duplicate Keys

ECMA-404

No mention of unique keys

```
{  
  "author": "Xavier",  
  "author": "Jason"  
}
```

RFC 8259

Names within an object

SHOULD be unique

All implementations

Agree on name-value mappings

When no duplicates present

Json.NET will use the last value



JSON Data Types

Number

```
"courses": 5,  
"weight": 205.5,  
"likeJSON": 7.02e+23,
```

String

```
"name": "Xavier",
```

Boolean

```
"developer": true,
```

Array

```
"counts":  
  ["one", 2,  
   "three", 4]
```

Object

```
"author": {  
  "firstName": "Xavier",  
  "lastName": "Morera"  
}
```

null

```
"healthyDiet": null,
```

Whitespaces

```
"spaces" : "ignore them outside of key value pairs"
```




```
{  
  "fullName": "Xavier Morera",  
  "developer": true,  
  "courses": 5,  
  "weight": 205.5,  
  "healthyDiet": null,  
  "likeJSONOverXML": 7.02e+23,  
  "author": {  
    "firstName": "Xavier",  
    "lastName": "Morera"  
  },  
  "counts": [  
    "one",  
    2,  
    "three",  
    4  
  ],  
  "spaces": "ignore them outside of key value pairs"  
}
```



```
{  
  "fullName": "Xavier Morera",  
  "developer": true,  
  "courses": 5,  
  "weight": 205.5,  
  "healthyDiet": null,  
  "likeJSONOverXML": 7.02e+23,  
  "author": {  
    "firstName": "Xavier",  
    "lastName": "Morera"  
  }  
  "counts": [  
    "one",  
    2,  
    "three",  
    4  
  ],  
  "spaces": "ignore them outside of key value pairs"  
}
```

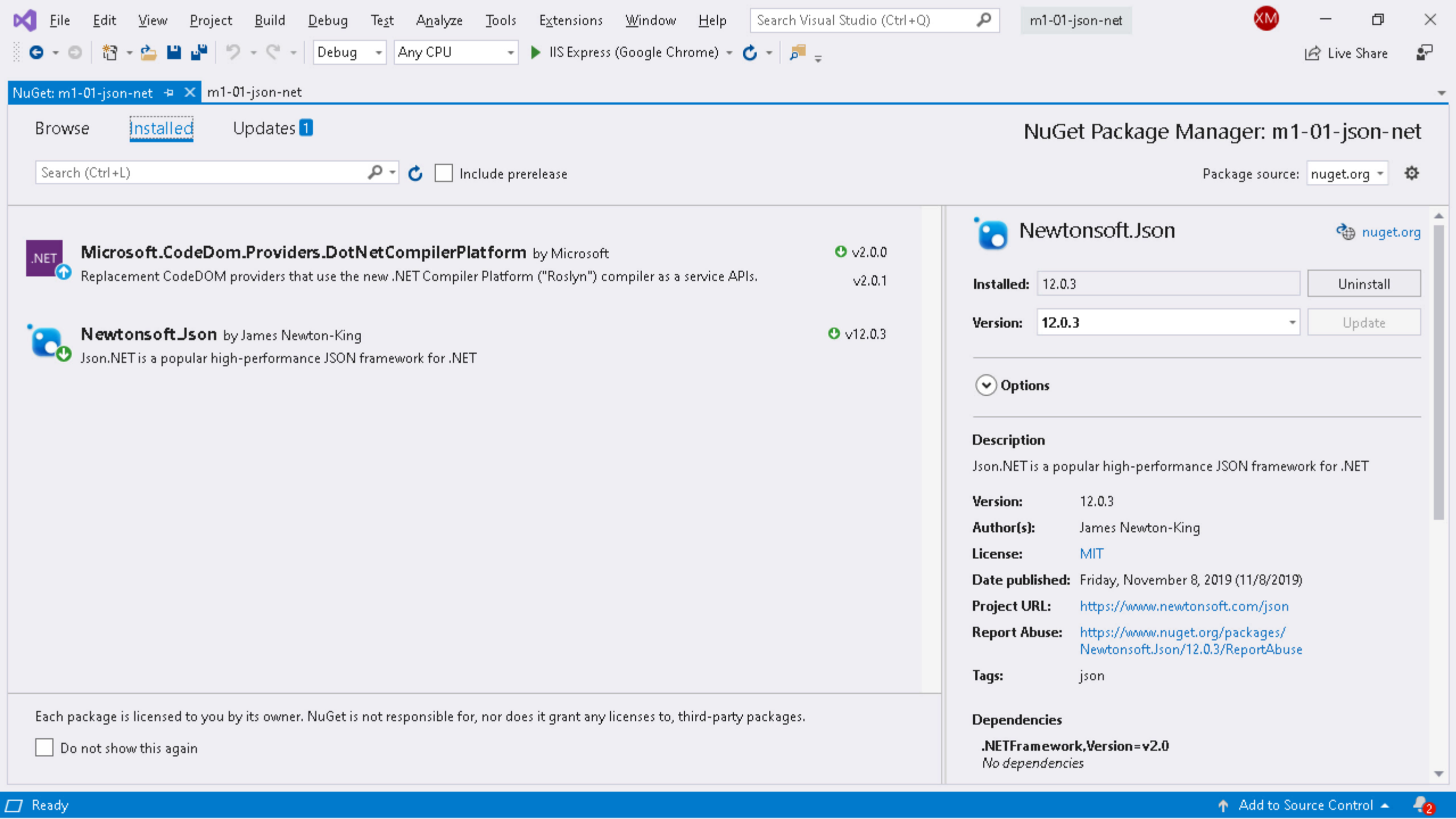


Demo



Adding Json.NET to a .NET Project



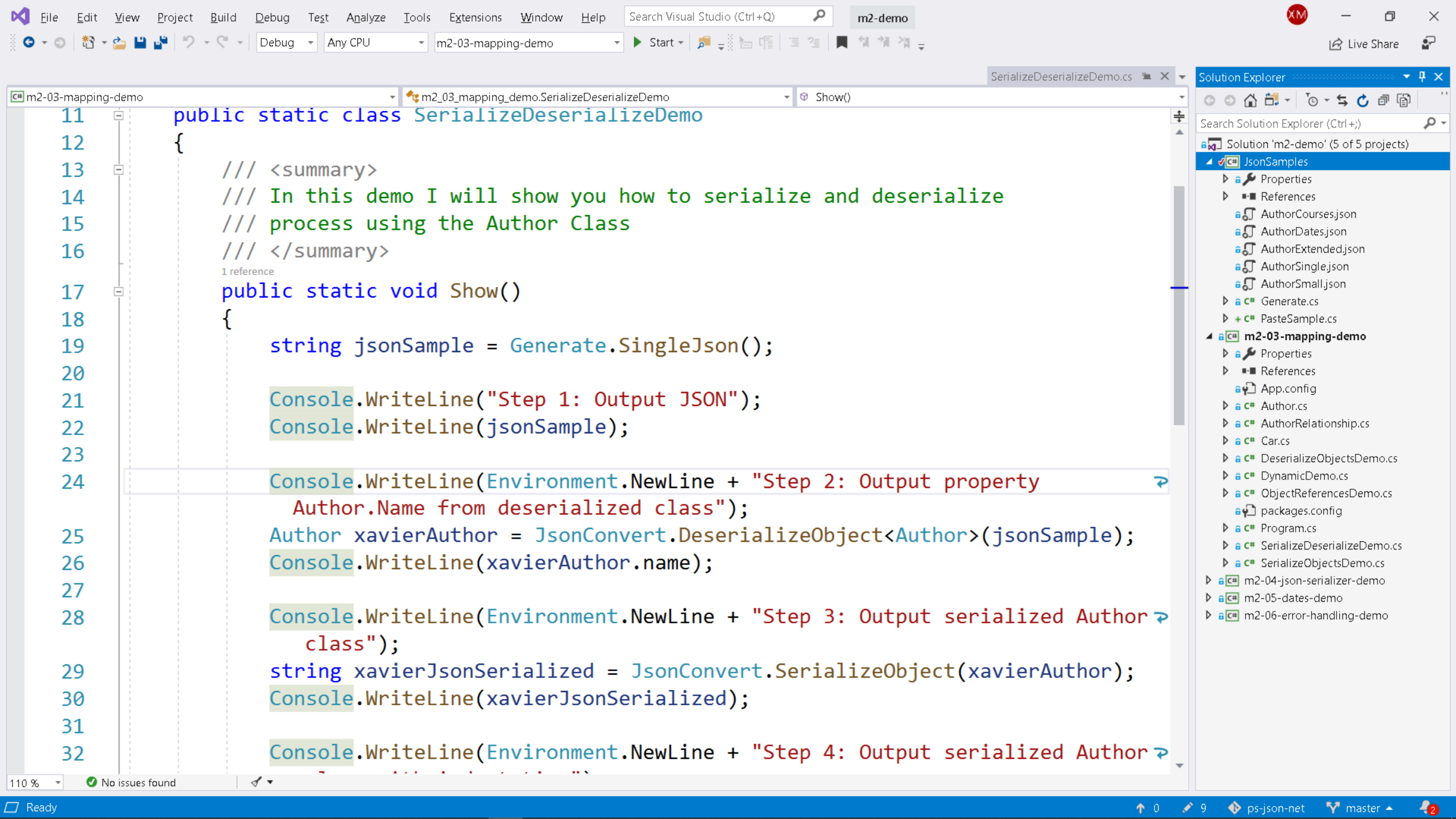


Demo



Demos Overview and One Good Tip





Takeaway



JavaScript Object Notation (JSON)

Human Readable Data Interchange Format

Key/Value Pairs

- RFC 8259
- ECMA 404

Developers Out Of Syntax Business!

Fat Free Alternative to XML

Json.NET

