

JSON

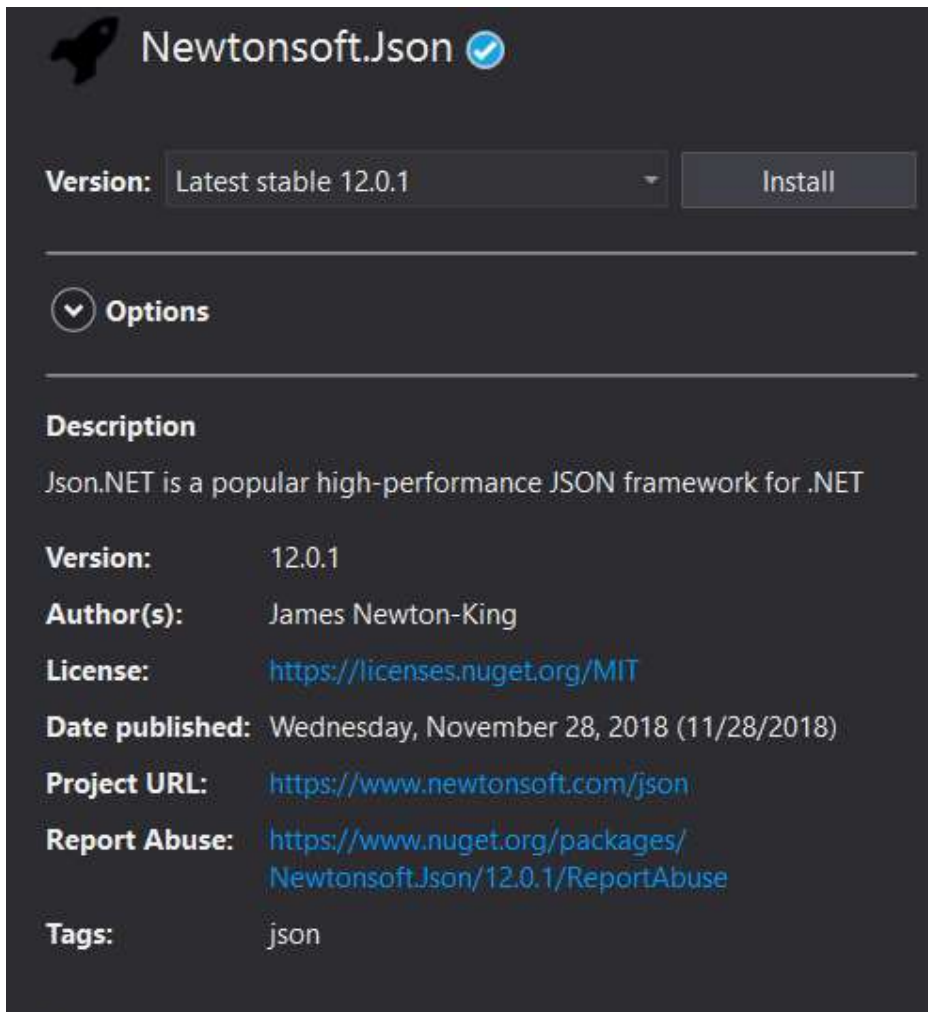
What is JSON?

JSON VS XML


```
3 <menu id="file" value="File">
4   <popup>
5     <menuitem value="New" onclick="CreateNewDoc()" />
6     <menuitem value="Open" onclick="OpenDoc()" />
7     <menuitem value="Close" onclick="CloseDoc()" />
8   </popup>
9 </menu>
```


```
3 = {"menu": {
4   "id": "file",
5   "value": "File",
6   "popup": {
7     "menuitem": [
8       {"value": "New", "onclick": "CreateNewDoc()"},
9       {"value": "Open", "onclick": "OpenDoc()"},
10      {"value": "Close", "onclick": "CloseDoc()"}
11    ]
12  }
13 }}
```


NuGet JSON Library



The screenshot shows the NuGet package page for Newtonsoft.Json. At the top, there is a rocket icon and the package name 'Newtonsoft.Json' with a verified badge. Below this, a 'Version:' dropdown menu is set to 'Latest stable 12.0.1', and an 'Install' button is to its right. A horizontal line separates this from the 'Options' section, which is indicated by a downward arrow and the word 'Options'. Another horizontal line follows. The 'Description' section contains the text: 'Json.NET is a popular high-performance JSON framework for .NET'. Below the description, several metadata fields are listed: 'Version:' (12.0.1), 'Author(s):' (James Newton-King), 'License:' (a link to https://licenses.nuget.org/MIT), 'Date published:' (Wednesday, November 28, 2018 (11/28/2018)), 'Project URL:' (a link to https://www.newtonsoft.com/json), 'Report Abuse:' (a link to https://www.nuget.org/packages/Newtonsoft.Json/12.0.1/ReportAbuse), and 'Tags:' (json).

Newtonsoft.Json 

Version: Latest stable 12.0.1  **Install**

 **Options**

Description

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

Version: 12.0.1

Author(s): James Newton-King

License: <https://licenses.nuget.org/MIT>

Date published: Wednesday, November 28, 2018 (11/28/2018)

Project URL: <https://www.newtonsoft.com/json>

Report Abuse: <https://www.nuget.org/packages/Newtonsoft.Json/12.0.1/ReportAbuse>

Tags: json

JsonConvert

```
// Create an Object
Product product = new Product();
// Set some properties
product.Name = "Apple";
product.ExpiryDate = new DateTime(2008, 12, 28);
product.Price = 3.99M;
product.Sizes = new string[] { "Small", "Medium", "Large" };
// Serializing
string json = JsonConvert.SerializeObject(product);
// Deserializing
Product dp = JsonConvert.DeserializeObject<Product>(json);
```

JSON and LINQ

```
JsonObject myObj = JsonObject.Parse(@"{
    'CPU': 'Intel',
    'Drives': [
        'DVD read/writer',
        '500 gigabyte hard drive'
    ]
}");

string cpu = (string)myObj["CPU"];
// Intel

string firstDrive = (string)myObj["Drives"][0];
// DVD read/writer

IList<string> allDrives = myObj["Drives"].Select(t => (string)t).ToList();
foreach (var v in allDrives)
    Console.WriteLine(v);
// DVD read/writer
// 500 gigabyte hard drive
```