

(JavaScript Object Notation) [JSON](#) یک راه مناسب برای نگهداری اطلاعات است و از لحاظ ساختاری شباهت زیادی به XML ، رقیب قدیمی خود دارد.

وب سرویس و آجاکس برای انتقال اطلاعات از این روش استفاده می‌کنند و بعضی از پایگاه‌های داده مانند [RavenDB](#) بر مبنای این تکنولوژی پایه گذاری شده اند.

هیچ چیزی نمی‌تواند مثل یک مثال؛ خوانایی ، سادگی و کم حجم بودن این روش را نشان دهد :

اگر یک شی با ساختار زیر در سی شارپ داشته باشید :

```
class Customer
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
}
```

ساختار JSON متناظر با آن (در صورت این که مقدار دهی شده باشد) به صورت زیر است:

```
{
  "Id":1,
  "FirstName":"John",
  "LastName":"Doe"
}
```

و در یک مثال پیچیده‌تر :

```
class Customer
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public Car Car { get; set; }
    public IEnumerable<Location> Locations { get; set; }
}

class Location
{
    public int Id { get; set; }
    public string Address { get; set; }
    public int Zip { get; set; }
}

class Car
{
    public int Id { get; set; }
    public string Model { get; set; }
}
```

```
{
  "Id":1,
  "FirstName":"John",
  "LastName":"Doe",
  "Car": {
    "Id":1,
    "Model":"Nissan GT-R"
  },
  "Locations":[
```

```
{
  {
    "Id":1,
    "Address":"30 Mortensen Avenue, Salinas",
    "Zip":93905
  },
  {
    "Id":2,
    "Address":"65 West Alisal Street, #210, Salinas",
    "Zip":95812
  }
}
```

ساختار JSON را مجموعه ای از (نام - مقدار) تشکیل می‌دهد. ساختار مشابه آن در زبان سی شارپ [KeyValuePair](#) است.

مشاهده [این تصاویر](#) ، بهترین درک را از ساختار JSON به شما می‌دهد.

[Json.net](#) یکی از بهترین کتابخانه هایی است که برای کار با این تکنولوژی در .net ارائه شده است. بهترین روش اضافه نمودن آن به پروژه [NuGet](#) است. برای این کار دستور زیر را در Package Manager Console وارد کنید.

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

با استفاده از کد زیر می‌توانید یک Object را به فرمت JSON تبدیل کنید.

```
var customer = new Customer
{
    Id = 1,
    FirstName = "John",
    LastName = "Doe",
    Car = new Car
    {
        Id = 1,
        Model = "Nissan GT-R"
    },
    Locations = new[]
    {
        new Location
        {
            Id = 1,
            Address = "30 Mortensen Avenue,
Salinas",
            Zip = 93905
        },
        new Location
        {
            Id = 2,
            Address = "65 West Alisal Street, #210,
Salinas",
            Zip = 95812
        }
    }
};

var data = Newtonsoft.Json.JsonConvert.SerializeObject(customer);
```

خروجی تابع `SerializeObject` رشته ای است که محتوی آن را در چهارمین بلاک کد که در بالاتر آمده است، می‌توانید مشاهده کنید.

برای `Deserialize` کردن (Cast اطلاعات با فرمت JSON به کلاس موردنظر) از روش زیر بهره می‌گیریم :

```
var customer = Newtonsoft.Json.JsonConvert.DeserializeObject<Customer>(data);
```

آشنایی با این تکنولوژی، پیش درآمدی برای چشیدن طعم NoSQL و معرفی کارآمدترین روش‌های آن است که در آینده خواهیم آموخت...
خوشحال می‌شوم اگر نظرات شما را در باره این موضوع بدانم.

نظرات خوانندگان

نویسنده: احمدعلی شفیعی
تاریخ: ۱۵:۳۳ ۱۳۹۱/۰۴/۱۰

خیلی ممنون. واقعا کاربردی بود.

نویسنده: مهدی پایروند
تاریخ: ۱۶:۰۰ ۱۳۹۱/۰۴/۱۰

بهترین کتابخانه برای سریالایز کردن دیتا به سمت کلاینت است. و هم سرعتی تقریبا 8 برابر سرعت System.Web.Script.Serialization.JavaScriptSerializer دارد

نویسنده: سروش ترک زاده
تاریخ: ۱۷:۴۸ ۱۳۹۱/۰۴/۱۰

در تصدیق حرف شما، دیدن [این مقایسه](#) خالی از لطف نیست. (پایین صفحه : Performance Comparison)

نویسنده: ssm
تاریخ: ۲۰:۱۷ ۱۳۹۱/۰۴/۱۰

با سلام
بسیار عالی بود مشتاقانه منتظر مطالب بعدی شما هستم
موفق باشید

نویسنده: مهدی ترابی
تاریخ: ۲۳:۰۴ ۱۳۹۱/۰۴/۱۰

بسیار خوب.
لطفن JavaScript Object Notation تصحیح شود به JavaScript Object Notation

نویسنده: سروش ترک زاده
تاریخ: ۱۱:۴۶ ۱۳۹۱/۰۴/۱۱

ممنون از دقت نظر شما. تصحیح شد.

نویسنده: رضاب.
تاریخ: ۳:۷ ۱۳۹۱/۰۴/۱۲

با درود،

علت کمرنگ شدن نقش XML (اگر کمرنگ شده؟) بخاطر همین سادگی و سبکی JSON هست؟

نویسنده: سروش ترک زاده
تاریخ: ۱۰:۳۳ ۱۳۹۱/۰۴/۱۲

سلام
JSON رقیبی قوی برای XML محسوب می‌شه، اما به نظر هنوز زود هست که بگیم XML در حال حذف شدن هست. چون هنوز سیستم‌های قدرتمندی مثل Microsoft Workflow Generator یا همین RSS که زیاد در طول روز با اون سروکار داریم، بر مبنای این تکنولوژی کار می‌کنند.

نویسنده: رضا

تاریخ: ۱۳۹۳/۰۳/۲۶ ۹:۳۵

با سلام و تبریک به خاطر مقاله مفیدتون
یه سوال برام پیش اومده ، سرعت عمل و کارایی JSON بهتره یا XML ?
برای انجام یه برنامه لازم هست که بدونم از کدوم بهتره استفاده کنم.
ممنون و متشکر

نویسنده: سروش ترک زاده

تاریخ: ۱۳۹۳/۰۳/۲۶ ۱۱:۵۲

سلام
سرعت عمل، به ابزاری بستگی دارد که به وسیله آن اطلاعات serialize و deserialize می‌شود.
بهترین ابزاری که برای کار با XML معرفی شده است،(البته تا جایی که من خبر دارم) LinqToXML است. کار کردن با آن ساده است
اما دردسرهای خاص خودش رو داره.
از طرفی فرمت JSON نسبت به XML، حجم کمتری دارد (حداقل به این دلیل که نیاز به باز و بسته کردن tag نیست)
در مجموع من JSON رو پیشنهاد می‌کنم.