

عنوان: Serialization #1

نویسنده: سیدمجتبی حسینی

تاریخ: ۲:۴۲ ۱۳۹۱/۰۷/۱۳

آدرس: www.dotnettips.info

برچسب‌ها: C#, Serialization

در این نوشتار به Serialization و Deserialization یعنی مکانیزمی که توسط آن اشیاء می‌توانند به صورت متنی مسطح و یا به شکل باینری درآیند، پرداخته می‌شود.

مفهوم Serialization

سریالی کردن، عملیاتی است که یک شیء و یا مجموعه‌ای از اشیاء که به یکدیگر ارجاع می‌دهند را به شکل گروهی از بایت‌ها یا فرمت XML درآورده که می‌توان آن‌ها را ذخیره کرد و یا انتقال داد. Deserialization معکوس عملیات بالاست که گروهی از داده‌ها را دریافت کرده و بصورت یک شیء و یا مجموعه‌ای از اشیاء که به یکدیگر ارجاع می‌دهند، تبدیل می‌کند. Serialization و Deserialization نوعاً برای موارد زیر بکار می‌روند: انتقال اشیاء از طریق یک شبکه یا مرز یک نرم افزار .

ذخیره اشیاء در یک فایل یا بانک اطلاعاتی.

موتورهای سریالی‌کننده

چهار شیوه برای سریالی کردن در دات نت فریم ورک وجود دارد: سریالی‌کننده قرارداد داده (Data Contract Serializer)

سریالی‌کننده باینری

سریالی‌کننده XML مبتنی بر صفت

سریالی‌کننده اینترفیس IXmlSerializer

سه مورد اول، موتورهای سریالی‌کننده‌ای هستند که بیشترین یا تقریباً همه کارهای سریالی کردن را انجام می‌دهند. مورد آخر برای انجام مواردی است که خودتان قصد سریالی‌سازی دارید که این موتور با استفاده از XmlWriter و XmlReader این کار را انجام می‌دهد. IXmlSerializer می‌تواند به همراه سریالی‌کننده قرارداد داده و یا سریالی‌کننده XML در موارد پیچیده سریالی کردن استفاده شود.

سریالی‌کننده Data Contract

برای انجام این کار دو نوع سریالی‌کننده وجود دارد :

DataContractSerializer، که اصطلاحاً [loosely Coupled](#) شده است به نوع سریالی‌کننده Data Contract.

NetDataContractSerializer که اصطلاحاً tightly Coupled شده است به نوع سریالی‌کننده Data Contract.

مدل زیر را در نظر بگیرید:

```
public class News
{
    public int Id;
```

```

    public string Body;
    public DateTime NewsDate;
}

```

برای سریالی کردن نوع News به شیوه Data Contract, باید:

فضای نام System.Runtime.Serialization را به کد برنامه اضافه کنیم.

صفت [DataContract] را به نوعی که تعریف کرده ایم, اضافه کنیم.

صفت [DataMember] را به اعضای که می خواهیم در سریالی شدن شرکت کنند, اضافه کنیم.

و به این ترتیب کلاس News به شکل زیر درمی آید:

```

using System.Runtime.Serialization;
using System.Xml;
using System;
using System.IO;
namespace ConsoleApplication1
{
    [DataContract]
    public class News
    {
        [DataMember] public int Id;
        [DataMember] public string Body;
        [DataMember] public DateTime NewsDate;
    }
}

```

سپس به شکل زیر از مدل خود نمونه ای ساخته و با ایجاد یک فایل, نتیجه سریالی شده مدل را در آن ذخیره می کنیم .

```

var news = new News
{
    Id = 1,
    Body = "NewsBody",
    NewsDate = new DateTime(2012, 10, 4)
};
var ds = new DataContractSerializer(typeof (News));
using (Stream s=File.Create("News.Xml"))
{
    ds.WriteObject(s, news); //سریالی کردن
}

```

که محتویات فایل News.Xml به صورت زیر است:

```

<News xmlns="http://schemas.datacontract.org/2004/07/ConsoleApplication7"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><Body>NewsBody</Body><Id>1</Id><NewsDate>2012-10-04T00:00:00</NewsDate></News>

```

و برای Deserialize کردن این فایل داریم:

```

News deserializednews;
using (Stream s = File.OpenRead("News.Xml"))
{
    deserializednews = (News)ds.ReadObject(s); //Deserializing
}
Console.WriteLine(deserializednews.Body);

```

همان طور که ملاحظه می کنید فایل ایجاد شده از خوانایی خوبی برخوردار نیست که برای دستیابی به فایلی با خوانایی بالاتر از

XmlWriter استفاده میکنیم:

```
XmlWriterSettings settings = new XmlWriterSettings {Indent = true};
using (XmlWriter writer = XmlWriter.Create("News.Xml", settings))
{
    ds.WriteObject(writer, news);
}
System.Diagnostics.Process.Start("News.Xml");
```

به این ترتیب موفق به ایجاد فایلی با خوانایی بالاتر می‌شویم:

```
<?xml version="1.0" encoding="utf-8"?>
<News xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.datacontract.org/2004/07/ConsoleApplication7">
  <Body>NewsBody</Body>
  <Id>1</Id>
  <NewsDate>2012-10-04T00:00:00</NewsDate>
</News>
```

نظرات خوانندگان

نویسنده: محمد

تاریخ: ۱۱:۴۲ ۱۳۹۱/۰۷/۱۳

دست شما درد نکند خیلی اینا رو میدیدم ولی نمیفهمیدم چین.
ممنون که روشنمون کردی