عنوان: **Serialization #2**
نویسنده: سیدمجتبی حسینی
تاریخ: ۱۳۹۱/۰۷/۱۵ ۲۳:۱۷
آدرس: www.dotnettips.info
برچسب‌ها: C#, Serialization

مطابق آنچه در قسمت قبل گفته شد برای آن‌که بتوان از مدل News برای سریالی‌کردن استفاده کرد، باید آن را به شکل ذیل پیاده‌سازی کرد:

```
[DataContract]
public class News
{
     [DataMember] public int Id;
     [DataMember] public string Body;
     [DataMember] public DateTime NewsDate;
}
```

با Override کردن [ DataContract ] به صورت [DataContract(Name=”MyCustomNews")] می‌توان نام ریشه XML فایل را به MyCustomNews تغییر داد. همچنین با Override کردن [DataMember] بصورت [DataMember(Name=”MyCustomFieldName")] می‌شود به هر فیلدی عنوان دلخواهی داد و همچنین با تعیین عبارت NameSpace به صورت [DataContract(Name =") "MyCustomNews", Namespace = "http://www.my.com"] می‌شود فضای نام را تغییر داد که با این تغییرات، خروجی زیر حاصل می‌شود:

```
<?xml version="1.0" encoding="utf-8"?>
<MyCustomNews xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.my.com">
  <Body>NewsBody</Body>
  <MyCustomFieldName>111</MyCustomFieldName>
  <NewsDate>2012-10-04T00:00:00</NewsDate>
</MyCustomNews>
```

ویژگی [DataMember] هم ازفیلد ها و هم از property ها، پشتیبانی می‌کند، خواه عمومی باشند یا خصوصی و نوع فیلد یا Property می‌تواند به یکی از اشکال زیر باشد:

انواع اولیه .

انواع Uri ، Guid ، TimeSpan ، DateTime و انواع Enum

انواع پوچ پذیر هر کدام از موارد بالا

نوع []byte

انواع تعریف شده توسط کاربر که توسط صفت [DataContract] محصور شده‌اند.

هر نوع IEnumerable

هر نوعی که با صفت [Serializable] محصور شود و یا اینترفیس ISerializable را پیاده سازی کند.

هر نوعی که اینترفیس IXmlSerializble را پیاده سازی نماید.

**تعیین فرمت باینری برای سریالی‌کردن:**

برای سریالی کننده‌های DataContractSerializer و NetDataContractSerializer می‌توان به روش زیر فرمت خروجی را به شکل فرمت باینری درآورد که خروجی آن تاحد زیادی کوچک‌تر و کم حجم‌تر می‌شود:

```
var s = new MemoryStream();
using (XmlDictionaryWriter w=XmlDictionaryWriter.CreateBinaryWriter(s))
{
        ds.WriteObject(w,news);
}
```

و برای Deserialize کردن آن به شیوه زیر عمل می‌کنیم:

```
var s2 = new MemoryStream(s.ToArray());
News deserializednews;
using (XmlDictionaryReader r=XmlDictionaryReader.CreateBinaryReader(s2,XmlDictionaryReaderQuotas.Max))
{
     deserializednews = (News)ds.ReadObject(r);
}
```

که در آن از ویژگی Max کلاس XmlDictionaryReaderQuotas برای به دست آوردن حداکثر سهمیه فضای دیسک مربوط به XmlDictionaryReaders استفاده می‌شود.

## نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۱۶ ۱:۸

با تشکر. یک کتابخانه XML Serialization هم توسط آقای سینا ایروانیان تهیه شده که یک سری از محدودیت‌های کتابخانه‌های توکار دات نت را برطرف کرده.