

تشریح مسئله : KnownTypeAttribute چیست و چگونه از آن استفاده کنیم؟

پیش نیاز : آشنایی اولیه با مفاهیم WCF برای فهم بهتر مطالب

در ابتدا یک Wcf Service Application ایجاد کنید و مدل زیر را بسازید:

```
[DataContract]
public abstract class Person
{
    [DataMember]
    public int Code { get; set; }

    [DataMember]
    public string Name { get; set; }
}
```

{ یک کلاس پایه برای Person ایجاد کردیم به صورت abstract که وهله سازی از آن میسر نباشد و 2 کلاس دیگر می‌سازیم که از کلاس بالا ارث ببرند:

کلاس #1

```
[DataContract]
public class Student : Person
{
    [DataMember]
    public int StudentId { get; set; }
}
```

کلاس #2

```
[DataContract]
public class Teacher : Person
{
    public int TeacherId { get; set; }
}
```

فرض کنید قصد داریم سرویسی ایجاد کنیم که لیست تمام اشخاص موجود در سیستم را در اختیار ما قرار دهد. (هم Student و هم Teacher). ابتدا Contract مربوطه را به صورت زیر تعریف می‌کنیم:

```
[ServiceContract]
public interface IStudentService
{
    [OperationContract]
    IEnumerable<Person> GetAll();
}
```

همان طور که می‌بینید خروجی متد GetAll از نوع Person است (نوع پایه کلاس Student , Teacher). سرویس مربوطه بدین شکل خواهد شد.

```
public class StudentService : IStudentService
{
    public IEnumerable<Person> GetAll()
    {
        List<Person> listOfPerson = new List<Person>();

        listOfPerson.Add( new Student() { Code = 1, StudentId = 123, Name = "Masoud Pakdel" } );
        listOfPerson.Add( new Student() { Code = 1, StudentId = 123, Name = "Mostafa Asgari" } );
        listOfPerson.Add( new Student() { Code = 1, StudentId = 123, Name = "Saeed Alizadeh" } );

        listOfPerson.Add( new Teacher() { Code = 1, TeacherId = 321, Name = "Mahdi Rad" } );
    }
}
```

```

        listOfPerson.Add( new Teacher() { Code = 1, TeacherId = 321, Name = "Mohammad Heydari" } );
        listOfPerson.Add( new Teacher() { Code = 1, TeacherId = 321, Name = "Saeed Khatami" } );

        return listOfPerson;
    }

```

در این سرویس در متد GetAll لیستی از تمام اشخاص رو ایجاد می‌کنیم. 3 تا Student و 3 تا Teacher رو به این لیست اضافه میکنیم. برای نمایش اطلاعات در خروجی یک پروژه Console Application ایجاد کنید و سرویس بالا رو از روش AddServiceReference به پروژه اضافه کنید سپس در کلاس Program کدهای زیر رو کپی کنید.

```

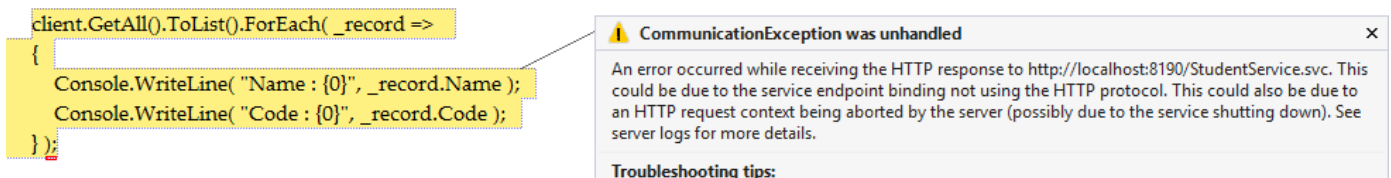
class Program
{
    static void Main( string[] args )
    {
        StudentService.StudentServiceClient client = new StudentService.StudentServiceClient();

        client.GetAll().ToList().ForEach( _record =>
        {
            Console.Write( "Name : {0}", _record.Name );
            Console.WriteLine( "Code : {0}", _record.Code );
        } );

        Console.ReadLine();
    }
}

```

پروژه رو کامپایل کنید. تا اینجا هیچ گونه مشکلی مشاهده نشد و انتظار داریم که خروجی مورد نظر رو مشاهده کنیم. بعد از اجرای پروژه با خطای زیر متوقف می‌شویم:



مشکل از اینجا ناشی می‌شود که هنگام عمل سریالایز، WCF Runtime با توجه به وهله سازی از کلاس Person می‌دونه که باید کلاس Student یا Teacher رو سریالایز کنه ولی در هنگام عمل دی سریالایز، WCF Runtime این موضوع رو درک نمی‌کنه به همین دلیل یک Communication Exception پرتاب می‌کنه. برای حل این مشکل و برای اینکه WCF Deserialize Engine رو متوجه نوع وهله سازی کلاس‌های مشتق شده از کلاس پایه کنیم باید از KnownTypeAttribute استفاده کنیم. فقط کافیست که این Attribute رو بالای کلاس Person به ازای تمام کلاس‌های مشتق شده از اون قرار بدید. بدین صورت:

```

[DataContract]
[KnownType( typeof( Student ) )]
[KnownType( typeof( Teacher ) )]
public abstract class Person
{
    [DataMember]
    public int Code { get; set; }

    [DataMember]
    public string Name { get; set; }
}

```

حالا پروژه سمت سرور رو دوباره کامپایل کنید و سرویس سمت کلاینت رو Update کنید. بعد پروژه رو دوباره اجرا کرده تا خروجی زیر رو مشاهده کنید.

```
Name : Masoud Pakdel Code : 1
Name : Mostafa Asgari Code : 1
Name : Saeed Alizadeh Code : 1
Name : Mahdi Rad Code : 1
Name : Mohammad Heydari Code : 1
Name : Saeed Khatami Code : 1
```

با وجود KnownType دیگه WCF Deserialize Engine میدونه که باید از کدام DataContact برای عمل دی سریالاز نمونه ساخته شده از کلاس Person استفاده کنه. دانستن این مطلب هنگام پیاده سازی [مفاهیم ارث بری در ORM ها](#) زمانی که از WCF استفاده می کنیم ضروری است.

نظرات خوانندگان

نویسنده: مصطفی عسگری
تاریخ: ۱۳۹۲/۰۳/۱۱ ۹:۴۹

ممنون ... استفاده بردیم.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۱۵ ۱۸:۹

ترجمه این مطلب در code project به تاریخ Jun 4 که میشه دیروز البته

[What is KnownType Attribute and How to Use It in WCF Technology](#)

نویسنده: مسعود م. پاکدل
تاریخ: ۱۳۹۲/۰۳/۱۵ ۲۲:۵۷

ممنون از دقت و اطلاع شما