

Extension methods شما را قادر می‌سازند تا به type‌های موجود بدون اینکه کلاس جدیدی ایجاد کنید که از آن‌ها به ارث رفته باشند، متدهای جدیدی اضافه نمائید و بیشترین استفاده آن‌ها در `System.Collections.IEnumerable` است. به طور مثال این امکان وجود ندارد که بتوان بر روی `IEnumerable`‌ها از دستور [Foreach](#) استفاده کرد. برای نمونه من برای اینکه `foreach` داشته باشم، آن‌را به لیست تبدیل می‌کردم و سپس از امکان `foreach` بهره‌مند می‌شدم که شاید کار درستی نباشد.

اما با Extension Method افزودن متد `foreach` به `IEnumerable` به نحو زیر میسر است:

```
public static void ForEach<T>(this IEnumerable<T> collection, Action<T> action)
{
    foreach (var item in collection)
        action(item);
}
```

نظرات خوانندگان

نویسنده:

بهروز راد

تاریخ:

۱۱:۱۲ ۱۳۹۱/۰۴/۰۹

من فکر نمی‌کنم این روش نوشتن مطلب صحیح باشه. شما خیلی کلی می‌نویسید. بهتره با جزئیات بیشتری به مطلب بپردازید. مثلاً لزوم ذکر نوع داده ای که قصد دارید اون رو توسعه بدید بعد از کلمه‌ی کلیدی this در پارامتر اول یا الزام وجود متد در یک کلاس static و تعریف خود متد. اینها موضوعاتی هستند که برای فردی که با Extension Method ها آشنا نیست باید گفته بشه.

نویسنده:

Amin

تاریخ:

۱۲:۱۰ ۱۳۹۱/۰۴/۰۹

من هم با نظر آقای راد موافقم. کلی گویی بدرد کسی که آشنایی زیادی نداره، نمیخوره. شرمنده اما حسم از این پست این بود که یه جووری موضوع از سر خودتوت باز کردین.

نویسنده:

وحید نصیری

تاریخ:

۱۲:۱۳ ۱۳۹۱/۰۴/۰۹

لطفا این بحث رو ادامه ندید. در یک سری از دسترسی‌های نویسنده‌ها به زودی تجدید نظر خواهد شد. چه کسانی که ابراز تمایل کرده بودند و فقط وقت من رو جهت ثبت آن‌ها نصف روزی تلف کردند؛ چه کسانی که مطالب سطحی ارسال می‌کنند. با تشکر

ضمن اینکه یک مطلب را هم مد نظر داشته باشید. اینجا هدف بیشتر ذکر یک سری نکته است به همین جهت اسم سایت **tips** دارد (نکات ریز).

اسم سایت **encyclopedia** نیست که برای هر مطلبی قرار باشد کتاب نوشته شود. اگر نوشته شد، چقدر خوب؛ اگر نه ... یک نکته ریز جدید یاد گرفتید. این هم خوب. همچنین هدف از این سایت خواننده عام با سطح مطالعه و اطلاعات صفر نیست و نبوده. از روز اولش اینطور نبوده و نخواهد بود.

یک مطالب رو هم فراموش نکنید. به قول مولوی: «گر تو بهتر میزنی بستان بزن» در طی یک ماهی که این سایت در حالت تعلیق بود ... من هر چقدر سایت‌های فارسی زبان فنی برنامه نویسی رو گشتم که دوتا مطلب به درد بخور پیدا کنم ... چیزی نیافتم. هیچی! واقعا دریغ از یک مطلب فنی به درد بخور که منتشر شده باشد. حالا همین عده منتظرند سریع بریزند سر یک نفر شروع کنند به داد و قال. باز تکرار می‌کنم: «گر تو بهتر میزنی بستان بزن»

نویسنده:

نیلوفر نوروزی

تاریخ:

۱۴:۵۴ ۱۳۹۱/۰۴/۰۹

من اصلا قصد این را نداشتم که کلی گویی کنم
اگر مطلب درست نوشته نشده معذرت می‌خواهم
هدفم فقط این بود که مطلبی که به خودم خیلی کمک کرده در اختیار دوستان قرار دهم.

نویسنده:

peyman

تاریخ:

۱۸:۱۴ ۱۳۹۱/۰۴/۰۹

والا تا جایی که من یادم هست پایه این وبلاگ بر این بوده که افراد تا حدی آشنایی با مفاهیم ابتدایی رو دارند. اما در هر صورت ساده نویسی در بیان مطالب پیچیده میتونه این وبلاگ رو بطور خیلی گسترده‌تر مفید واقع کنه حتی برای افرادی که مبتدی هستند. در پایان هم باید بگم که من دست و پای اون فردی رو میبوسم که به من حتی یک کلمه یاد بده ... چه آقای راد در وب سایت برنامه نویسی. چه آقای نصیری و دوستان گرامی‌شون در این وبلاگ که مدت زیادی هست وبلاگ مفیدشون رو مطالعه میکنم ...

اگر با نوع داده Enum آشنایی ندارید [قسمت یکم این مطلب](#) را بخوانید.

```
public enum Grade
{
    Failing = 5,
    BelowAverage = 10,
    Average = BelowAverage + 5, // = 15
    VeryGood = 18,
    Excellent = 20
}
```

بازنویسی متد ToString(): امکان بازنویسی متد ToString() در نوع Enum وجود ندارد. بنابراین برای چاپ عبارت Very Good به جای VeryGood تکنیک زیر جالب به نظر می‌رسد. هر چند استفاده از آرایه و ترکیب اندیس آن با Enum و یا استفاده از HashTable راه‌هایی است که در ابتدا به ذهن ما خطور می‌کند اما لطفاً به ادامه مطلب توجه فرمایید! با در نظر گرفتن مثال قبل، یک Custom Attribute به نوع داده شمارشی اضافه می‌کنیم. برای این منظور بصورت زیر عمل می‌کنیم.

1. ایجاد کلاس Description که از کلاس Attribute مشتق شده است و تعریف خصوصیت Text:

```
class Description : Attribute
{
    public string Text;
    public Description(string text)
    {
        Text = text;
    }
}
```

2. به سراغ نوع Enum تعریف شده رفته و جهت استفاده از صفت جدید که در مرحله قبل پیاده سازی کردیم، تغییرات را به شکل زیر اعمال می‌کنیم:

```
public enum Grade
{
    [Description("Mardood")]
    Failing = 5,

    [Description("Ajab Shansi")]
    BelowAverage = 10,

    [Description("Bad Nabood")]
    Average = BelowAverage + 5,

    [Description("Khoob Bood")]
    VeryGood = 18,

    [Description("Gol Kashti")]
    Excellent = 20
}
```

تنها کاری که باقی مانده یاری گرفتن از متدهای الحاقی (Extension Methods) جهت خواندن مقدار Description است:

```
public static class ExtensionMethodCls
{
    public static string GetDescription(this Enum enu)
    {
        Type type = enu.GetType();
```

```

        MemberInfo[] memInfo = type.GetMember(enu.ToString());
        if (memInfo != null && memInfo.Length > 0)
        {
            object[] attrs = memInfo[0].GetCustomAttributes(typeof(Description), false);
            if (attrs != null && attrs.Length > 0)
                return ((Description)attrs[0]).Text;
        }
        return enu.ToString();
    }
}

```

حال نوع Enum ما کمی توسعه یافته است و توسط متد GetDescription می توان متن دلخواه و متناسب با مقدار را نمایش داد:

```
Console.WriteLine(grade.GetDescription()); // Print Bad Nabood
```

کد کامل مثال بررسی شده نیز بصورت زیر خواهد بود:

```

using System;
using System.Reflection;

namespace CSharpEnum
{
    class Description : Attribute
    {
        public string Text;
        public Description(string text)
        {
            Text = text;
        }
    }

    public enum Grade
    {
        [Description("Mardood")]
        Failing = 5,

        [Description("Ajab Shansi")]
        BelowAverage = 10,

        [Description("Bad Nabood")]
        Average = BelowAverage + 5,

        [Description("Khoob Bood")]
        VeryGood = 18,

        [Description("Gol Kashti")]
        Excellent = 20
    }

    public static class ExtensionMethodCls
    {
        public static string GetDescription(this Enum enu)
        {
            Type type = enu.GetType();
            MemberInfo[] memInfo = type.GetMember(enu.ToString());
            if (memInfo != null && memInfo.Length > 0)
            {
                object[] attrs = memInfo[0].GetCustomAttributes(typeof(Description), false);
                if (attrs != null && attrs.Length > 0)
                    return ((Description)attrs[0]).Text;
            }
            return enu.ToString();
        }
    }
}

```

```
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        const Grade grade = Grade.Average;  
        Console.WriteLine("Underlying type: {0}", Enum.GetUnderlyingType(grade.GetType()));  
        Console.WriteLine("Type Code      : {0}", grade.GetTypeCode());  
        Console.WriteLine("Value          : {0}", (int)grade);  
        Console.WriteLine("-----");  
        Console.WriteLine(grade.ToString()); // name of the constant  
        Console.WriteLine(grade.ToString("G")); // name of the constant  
        Console.WriteLine(grade.ToString("F")); // name of the constant  
        Console.WriteLine(grade.ToString("x")); // value is hex  
        Console.WriteLine(grade.ToString("D")); // value in decimal  
        Console.WriteLine("-----");  
        Console.WriteLine(grade.GetDescription()); // Print Bad Nabood  
        Console.ReadKey();  
    }  
}
```

با استفاده از این تکنیک (مخصوصاً ما فارسی زبان ها) به راحتی می‌توانیم از مقادیر Enum استفاده بهتری ببریم. برای مثال اگر بخواهیم یک مقدار Enum را بصورت فارسی در یک Drop Down List نمایش دهیم این تکنیک بسیار مفید خواهد بود.

نظرات خوانندگان

نویسنده: حسین
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۱۹

سلام
کدهایی که به این صورت نوشته میشن چی بهشون گفته میشه؟ چی هستن؟
خیلی جاها این کدها رو دیدم به صورت های مختلف
کمی تو گوگل سرچ کردم نتیجه مطلوبی نگرفتم یعنی نمیدونم دنبال چی بگردم
لطفا راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۲۳

به Attribute ای که در اینجا توسعه داده شده (یا از آن استفاده شده)، اصطلاحا data annotation هم گفته می شود. یک سری از فریم ورک ها به صورت توکار قادر به استفاده از آن ها هستند مانند ASP.NET MVC برای نمایش توضیحات مرتبط یا نمایش برچسب ها به صورت خودکار.
مطالب فوق رو می تونید پایه طراحی این نوع کتابخانه ها در نظر بگیرید.

نویسنده: حسین
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۳۰

کدهایی که تو linq هم استفاده میشن از همین دسته اند؟
مثلا

```
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID", AutoSync=AutoSync.Always, DbType="Int NOT NULL IDENTITY", IsDbGenerated=true)]
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۴۶

بله. به این ها [Attribute](#)، meta-data، و یا data annotation گفته می شود.

نویسنده: ایمان محمدی
تاریخ: ۱۳۹۱/۰۵/۰۶ ۱۴:۰۹

نیازی به تعریف کلاس Description نیست.

کافیه از فضای نام System.ComponentModel استفاده کنید و در مقدار بازگشتی متد GetDescription بجای

```
return ((Description)attrs[0]).Text;
```

بنوسید

```
return ((DescriptionAttribute)attrs[0]).Description;
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۶ ۱۴:۲۰

هدف این مطلب بررسی زیر ساخت این نوع طراحی بوده نه صرفا مصرف کننده محض بودن. تشابه اسمی اهمیتی ندارد. روش انجام کار مهم بوده در اینجا.

نویسنده: ایمان محمدی
تاریخ: ۱۶:۵۱۳۹۱/۰۵/۰۶

وقتی کلاس Description در فضای نام System.ComponentModel وجود دارد دلیلی نداره کلاس مشابه ای تعریف کنیم. بخاطر اینکه مصرف کننده محض نباشیم یک متد الحاقی به نام GetEnumerator() اضافه کردم که لیست اعضای یک Enum رو برای استفاده در کمبو باکس و ... بر می گردونه :
ابتدا کلاس زیر به کلاس ExtensionMethodCls اضافه می کنیم :

```
public class EnumObject
{
    public Enum ValueMember { get; set; }
    public int intValueMember
    {
        get { return int.Parse(ValueMember.ToString("D")); }
    }
    public string stringValueMember
    {
        get { return ValueMember.ToString(""); }
    }
    public string DisplayMember
    {
        get { return ValueMember.GetDescription(); }
    }
}
```

و متد الحاقی زیر رو برای گرفتن لیست تعریف می کنیم:

```
public static List<EnumObject> GetEnumerator(this Enum enu)
{
    List<EnumObject> li = new List<EnumObject>();
    foreach (var item in enu.GetType().GetEnumValues())
    {
        li.Add(new EnumObject { ValueMember = (Enum)item });
    }
    return li;
}
```

نحوه استفاده :

```
comboBox1.DataSource = Grade.VeryGood.GetEnumerator();
comboBox1.DisplayMember = "DisplayMember";
comboBox1.ValueMember = "ValueMember";
```

همون طوری که در کد بالا می بینید برای گرفتن لیست مجبور شدیم یکی از اعضای enum رو انتخاب کنیم (Grade.VeryGood) شاید انتخاب یکی از اعضا و بعد درخواست لیست اعضا رو کردن کار قشنگی نباشه به همین دلیل متد زیر رو تعریف کردیم :

```
public static List<EnumObject> EnumToList<T>()
{
    Type enumType = typeof(T);
    if (enumType.BaseType != typeof(Enum))
        throw new ArgumentException("T must be of type System.Enum");

    List<EnumObject> li = new List<EnumObject>();
    foreach (var item in enumType.GetEnumValues())
    {
        li.Add(new EnumObject { ValueMember = (Enum)item });
    }
}
```

```
    return li;
}
```

نحوه استفاده :

```
comboBox1.DataSource = ExtensionMethodCls.EnumToList<Grade>();
comboBox1.DisplayMember = "DisplayMember";
comboBox1.ValueMember = "ValueMember";
```

کد کامل :

```
public static class ExtensionMethodCls
{
    public class EnumObject
    {
        public Enum ValueMember { get; set; }
        public int intValueMember
        {
            get { return int.Parse(ValueMember.ToString("D")); }
        }
        public string stringValueMember
        {
            get { return ValueMember.ToString(""); }
        }
        public string DisplayMember
        {
            get { return ValueMember.GetDescription(); }
        }
    }

    public static List<EnumObject> EnumToList<T>()
    {
        Type enumType = typeof(T);
        if (enumType.BaseType != typeof(Enum))
            throw new ArgumentException("T must be of type System.Enum");

        List<EnumObject> li = new List<EnumObject>();
        foreach (var item in enumType.GetEnumValues())
        {
            li.Add(new EnumObject { ValueMember = (Enum)item });
        }
        return li;
    }

    public static List<EnumObject> GetEnumList(this Enum enu)
    {
        List<EnumObject> li = new List<EnumObject>();
        foreach (var item in enu.GetType().GetEnumValues())
        {
            li.Add(new EnumObject { ValueMember = (Enum)item });
        }
        return li;
    }

    public static string GetDescription(this Enum enu)
    {
        Type type = enu.GetType();
        MemberInfo[] memInfo = type.GetMember(enu.ToString());
        if (memInfo != null && memInfo.Length > 0)
        {
            object[] attrs = memInfo[0].GetCustomAttributes(typeof(DescriptionAttribute), false);
            if (attrs != null && attrs.Length > 0)
                return ((DescriptionAttribute)attrs[0]).Description;
        }
        return enu.ToString();
    }
}
```



```
}
```

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۵/۰۷ ۹:۴۵

با سلام خدمت شما دوست عزیز. همانطور که آقای نصیری اشاره کردند هدف این مطلب و البته این سری از مطالب آشنایی قدم به قدم با مفاهیم این نوع داده بوده است. انقیاد کنترل‌ها با اشیاء از نوع Enum موضوع بعدی این سری از مطالب بود. البته باز هم از شما تشکر می‌کنم بخاطر این نظر. اگر این قابلیت در سایت ایجاد شود که بتوان به نظرات لینک داد بهتر است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۷ ۱۰:۰۰

علامت آبی رنگ # کنار هر نظر، لینک مستقیم به همان نظر است.

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۵/۰۷ ۲۲:۰۸

مرسی. خیلی خیلی خوب...

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۱۷ ۲۰:۵۴

باسلام؛ لطفاً به توضیحی در باره‌ی MemberInfo بدید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۸ ۹:۱۱

مراجعه کنید به [MSDN](#) و مثال‌های آن در همان صفحه.

چقدر خوب می‌شد اگر،

نوع داده String دارای متدی جهت حذف تگ‌های HTML داشت:

```
string htmlStr = "<h1>.Net Tips</h1>";
htmlStr.ClearHtmlTags();
```

کلاس Image دارای متدی جهت تغییر اندازه (Resize) داشت:

```
image1.Resize(50, 80);
```

کنترل DropDownList متدی جهت انقیاد داده‌ها داشت:

```
dropDownList1.Bind((List<Category>)categories, "Name", "Id");
```

متدهای الحاقی به همین منظور متولد شده اند. در واقع هر زمان بدنه کلاسی (نوع داده، کنترل و تمام اشیاء دات نت) در اختیار ما نباشد امکان اضافه کردن متدهای الحاقی به آنها وجود دارد. برای این منظور کفایت چند نکته را رعایت کنید:

کلاس دربرگیرنده متد یا متدهای الحاقی باید Public و Static باشد.

متد الحاقی باید Public و Static باشد.

اولین پارامتر متد الحاقی باید با کلمه کلیدی this همراه باشد و این پارامتر اشاره به کلاسی دارد که متد جاری به آن الحاق (یا ضمیمه) خواهد شد.

یک مثال:

در این مثال متد الحاقی برای بهبود نوع داده String را خواهیم دید. وظیفه‌ی این متد شمارش تعداد کلمات موجود در رشته است.

```
public static class StringExtensions
{
    /// <summary>
    /// Count all words in a given string
    /// </summary>
    /// <param name="input">string to begin with</param>
    /// <returns>int</returns>
    public static int WordCount(this string input)
    {
        var count = 0;
        try
        {
            // Exclude whitespaces, Tabs and line breaks
            var re = new Regex(@"^[^\s]+");
            var matches = re.Matches(input);
            count = matches.Count;
        }
        catch (Exception)
        {
            return -1;
        }
        return count;
    }
}
```

نحوه استفاده:

```
var s = "i Love Dot Net Tips.";
var wordCount = s.WordCount();
```

در ضمن وب سایتی جهت به اشتراک گذاری این متدها به عنوان یکی از بهترین مراجع در دسترس است:

<http://extensionmethod.net>

با توجه به این مطلب توسعه پروژه ای در همین سایت با عنوان "[متدهای الحاقی](#)" آغاز شده است. در این پروژه ضمن پوشش متدهای الحاقی پرکاربرد سعی به توسعه متدهای الحاقی داریم که بیشتر در برنامه های فارسی کاربرد دارند.

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۱/۰۵ ۰:۵۵

ممنون. در صورت امکان لطفا آدرس مخزن کد را به صورت <http://projectName.codeplex.com> ارائه کنید.

نویسنده: علیرضا اسمرام
تاریخ: ۱۳۹۲/۰۱/۰۵ ۱:۵

سلام. برداشت من از آدرس مخزن کد، آدرسی بود که دوستان بتوانند از طریق VisualStudio.NET روی توسعه این پروژه بنده را همراهی کنند.
آدرس مورد نظر شما <http://dnextensions.codeplex.com> است. اگر اشتباه برداشت کردم لطفاً بفرمایید تا آدرس را در پروژه اصلاح کنم.
همچنین سوالات مربوط به پروژه را در [بخش مربوط به آن](#) مطرح کنید.
با سپاس از لطف شما

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۵ ۱:۱۰

بله. لطفا آدرس عمومی پروژه را لحاظ بفرمائید. با تشکر.

نویسنده: علیرضا اسمرام
تاریخ: ۱۳۹۲/۰۱/۰۵ ۱:۱۹

با تشکر، اصلاح شد.

نویسنده: M.Q
تاریخ: ۱۳۹۲/۰۱/۰۵ ۱۵:۵۷

من در زمان کانکت شدن به مخزن کد با خطای زیر مواجه می‌شوم. همچنین فقط کاربران عضو سایت می‌توانند همکاری کنند یا بازدیدکنندگان هم اجازه همکاری دارند؟

Microsoft Visual Studio
TF31002: Unable to connect to this Team Foundation Server: <http://dnextensions.codeplex.com/>.
Team Foundation Server Url: <http://dnextensions.codeplex.com/>.

Possible reasons for failure include:
- The name, port number, or protocol for the Team Foundation Server is incorrect.
- The Team Foundation Server is offline.
- The password has expired or is incorrect.

Technical information (for administrator):
The remote server returned an error: (404) Not Found.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۵ ۱۶:۴۲

- این مورد مختص تمام سورس کنترلرها است. فقط اعضای تیم و اعضای تعریف شده دسترسی commit دارند.
البته می‌تونید یک کپی فقط خواندنی با checkout مسیر <https://dnextensions.svn.codeplex.com/svn> تهیه کنید. روش کار [در اینجا](#) توضیح داده شده.
و یا در همان سایت کدپلکس در قسمت سورس‌ها، آخرین سورس‌ها را دانلود کنید.
+ می‌تونید برای همکاری حداقل دو کار را انجام دهید:
الف) وصله‌های خودتون رو [در اینجا](#) ارسال کنید.

ب) و یا مواردی را که مدنظر دارید در سایت جاری د [ر قسمت بازخوردهای پروژه](#) مطرح کنید.

نویسنده: علیرضا اسم‌رام
تاریخ: ۲۳:۱۹ ۱۳۹۲/۰۱/۰۵

سلام، توضیحات آقای نصیری کامل بود. همچنین اضافه کنم که دسترسی به سورس برنامه از طریق [این لینک](#) + نیز میسر است.

نویسنده: M.Q
تاریخ: ۱۲:۲۰ ۱۳۹۲/۰۱/۰۶

ممنون

من سعی می‌کنم از روش هاش هایی که آقای نصیری گفتند همکاری کنم.

با سپاس

نویسنده: م.خ
تاریخ: ۱۳:۵۳ ۱۳۹۲/۰۱/۰۶

سلام

مشابه Class helper در دلفی است...

برای بررسی صحت کد ملی باید کمی با ساختار این کد آشنا شویم. کد ملی ۰۰۷-۷۶۱۱۶۱-۶ را در نظر بگیرید. همانطور که مشاهده می‌کنید این کد به ۳ قسمت تقسیم شده است. قسمت اول که یک عدد ۳ رقمی است نشان دهنده محل تولد فرد است. ۶ رقم وسط شماره شناسایی فرد و رقم آخر، رقم کنترل است. در حقیقت ساختار ۹ رقم اول طبق الگوریتمی قابل بررسی است که خروجی آن الگوریتم همان رقم آخر است. اگر خروجی الگوریتم با رقم کنترل برابر باشد می‌توان گفت کد ملی وارد شده معتبر است و در غیر اینصورت کد ملی معتبر نخواهد بود.

برای این منظور دو متد الحاقی زیر به [پروژه DNT.Extensions](#) اضافه شده است که بدنه آنها به شرح زیر است:

```
/// <summary>
/// Validate IR National Code
/// </summary>
/// <param name="nationalcode">National Code</param>
/// <returns></returns>
public static bool IsValidNationalCode(this string nationalcode)
{
    int last;
    return nationalcode.IsValidNationalCode(out last);
}

/// <summary>
/// Validate IR National Code
/// </summary>
/// <param name="nationalcode">National Code</param>
/// <param name="lastNumber">Last Number Of National Code</param>
/// <returns></returns>
public static bool IsValidNationalCode(this string nationalcode, out int lastNumber)
{
    lastNumber = -1;
    if (!nationalcode.IsItNumber()) return false;
    var array = nationalcode.ToCharArray();
    if (array.Length != 10) return false;
    var j = 10;
    var sum = 0;
    for (var i = 0; i < array.Length - 1; i++)
    {
        sum += Int32.Parse(array[i].ToString(CultureInfo.InvariantCulture)) * j;
        j--;
    }
    var div = sum / 11;
    var r = div * 11;
    var diff = Math.Abs(sum - r);

    if (diff <= 2)
    {
        lastNumber = diff;
        return diff == Int32.Parse(array[9].ToString(CultureInfo.InvariantCulture));
    }
    var temp = Math.Abs(diff - 11);
    lastNumber = temp;
    return temp == Int32.Parse(array[9].ToString(CultureInfo.InvariantCulture));
}
```

طریقه استفاده از این متدها نیز به شرح ذیل است:

```
bool b = "0077611616".IsValidNationalCode();
```

پی نوشت:

لطفاً بخش نظرات خوانندگان را نیز دنبال کنید.

نظرات خوانندگان

نویسنده: علیرضا
تاریخ: ۱۷:۳۵ ۱۳۹۲/۰۱/۰۶

مشخصا برای اینکه متد قابل اطمینان باشد نباید از متد Parse استفاده کرد. به جای آن از TryParse استفاده کنید تا از وقوع exception جلوگیری شود

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۰ ۱۳۹۲/۰۱/۰۶

اون وقت اگر در این بین به مشکل برخورد چطور؟ استثناء اصلا چیز بدی نیست؛ کرش بسیار پدیده مطلوبی است! چون نشان وجود مشکل در سیستم است. بجای اینکار بهتر است در همان بدو امر بررسی شود که رشته دریافتی عدد است یا خیر. چون طول رشته زیاد است می‌شود از [مثلا Regex](#) استفاده کرد:

```
public static bool IsItNumber(this string inputvalue)
{
    var isnumber = new Regex("[^0-9]");
    return !isnumber.IsMatch(inputvalue);
}
```

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۷:۵۰ ۱۳۹۲/۰۱/۰۶

با تشکر از توجه شما،
یک متد الحاقی با عنوان IsItNumber به پروژه اضافه شد. همچنین متد IsValidNationalCode اصلاح شد.

```
public static bool IsValidNationalCode(this string nationalcode, out int lastNumber)
{
    lastNumber = -1;
    if (!nationalcode.IsItNumber()) return false;
    var array = nationalcode.ToCharArray();
    if (array.Length != 10) return false;
    var j = 10;
    var sum = 0;
    for (var i = 0; i < array.Length - 1; i++)
    {
        sum += Int32.Parse(array[i].ToString(CultureInfo.InvariantCulture)) * j;
        j--;
    }
    var div = sum / 11;
    var r = div * 11;
    var diff = Math.Abs(sum - r);

    if (diff <= 2)
    {
        lastNumber = diff;
        return diff == Int32.Parse(array[9].ToString(CultureInfo.InvariantCulture));
    }
    var temp = Math.Abs(diff - 11);
    lastNumber = temp;
    return temp == Int32.Parse(array[9].ToString(CultureInfo.InvariantCulture));
}
```

نویسنده: صابر فتح الهی
تاریخ: ۱۹:۱۲ ۱۳۹۲/۰۱/۰۶

سلام برای بررسی کد ملی من از یک الگوریتم دیگه استفاده کردم توی این [پست](#)

نویسنده: _mehdi
تاریخ: ۵:۲۷ ۱۳۹۲/۰۱/۰۷

سلام

من متوجه دلیل استفاده از متدهای الحاقی برای بررسی صحت کد ملی (جایی دیگه هم بررسی صحت ایمیل را دیده بودم) نمیشم . چون به نظر من البته اگر درست باشه با فلسفه وجودی متدهای الحاقی جور در نیامد . صرف اینکه کد ملی را از نوع string میگیریم بررسی صحت اون قابلیت نیست که بخواهیم به دلیل بسته بودن کلاس string به اون اضافه کنیم . چرا که ارزیابی صحت چیزی که منطق حاکم بر اون هیچ ربطی به کلاس string نداره را نمیتوان بهش افزود . بهتر نیست این نوع بررسیها به یه کلاس یوتیلیتی منتقل بشه .

نویسنده: محسن
تاریخ: ۹:۴۷ ۱۳۹۲/۰۱/۰۷

این نوع کلاسها هم تماما کمکی هستند. به همان روش سابق نام کلاس نام متد هم قابل دسترسی اند. به علاوه، به این ترتیب، علاوه بر روش قدیمی فراخوانی، حالت فراخوانی fluent هم میسر می شود که از دیدگاه زبان انگلیسی خواندنش ساده تر است و روان تر.

نویسنده: مجید
تاریخ: ۱۲:۵۵ ۱۳۹۲/۰۱/۰۷

سلام

جالبه! فقط یه نکته میشه به جای

```
var div = sum / 11;  
var r = div * 11;  
var diff = Math.Abs(sum - r);
```

نوشت :

```
var diff = sum % 11;
```

نویسنده: علیرضا اسمرام
تاریخ: ۱۳:۲۹ ۱۳۹۲/۰۱/۰۷

سلام. بله دو دستور فوق معادل یکدیگرند. من قبل از انتشار نسخه نهایی با مقادیر r و div کارهای دیگری انجام داده بودم که با توجه به حذف آن موارد می توان باقیمانده تقسیم sum بر 11 را تنها نگه داشت.

نویسنده: RezaNBT
تاریخ: ۱:۵۰ ۱۳۹۲/۰۱/۱۳

با سلام الگوریتم طراحی شده با reminder کمتر یا مساوی 2 مقدار غلط بر میگردد. بعنوان مثال مقدار محاسبه شده برای کد 0010350829 که معتبر می باشد برابر با 2 می باشد که باید از دوره تناوب کنترل کننده ی نقلی یعنی عدد 11 کم شود. یعنی :

```
if (diff <= 2)  
{  
    lastNumber = diff;  
    return diff ==(11- Int32.Parse(array[9].ToString(CultureInfo.InvariantCulture)));  
}
```


به نظر بنده نیازی به چک کردن و شاخه ای کردن کد نمی باشد و این بلوک شرطی می بایستی حذف گردد و در هر حالتی 11 از رقم نقلی کم شود و با مقدار حساب شده مقایسه شود.

از طرفی چون این فرمول نقلی به ازای هر عدد ده رقمی دارای تکرار فقط یک رقم خروجی صحیح دارد می بایستی این مورد هم کنترل گردد. یعنی در ابتدای تابع الحاقی باید داشته باشیم:

```
var allDigitEqual = new[] { "0000000000", "1111111111", "2222222222", "3333333333", "4444444444",  
"5555555555", "6666666666", "7777777777", "8888888888", "9999999999" };  
if (allDigitEqual.Contains(nationalcode)) return false;
```

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۳ ۳:۵۲

البته به جای این کد می توان [از این هم](#) استفاده کرد که به نظر خواناتر هست.

نویسنده: مجید
تاریخ: ۱۳۹۲/۰۱/۱۳ ۱۲:۲۲

شما اطمینان دارید که کد ملی 0010350829 معتبر است؟

نویسنده: علیرضا اسمرام
تاریخ: ۱۳۹۲/۰۱/۱۳ ۱۲:۳۳

با تشکر از شما متد فوق بصورت زیر اصلاح شد و کلاس مربوطه بروزرسانی شده است:

```
/// <summary>  
/// Validate IR National Code  
/// </summary>  
/// <param name="nationalcode">National Code</param>  
/// <param name="lastNumber">Last Number Of National Code</param>  
/// <returns></returns>  
public static bool IsValidNationalCode(this string nationalcode, out int lastNumber)  
{  
    lastNumber = -1;  
    if (!nationalcode.IsItNumber()) return false;  
    var invalid = new[]  
    {  
        "0000000000", "1111111111", "2222222222", "3333333333",  
        "4444444444", "5555555555",  
        "6666666666", "7777777777", "8888888888", "9999999999"  
    };  
    if (invalid.Contains(nationalcode)) return false;  
    var array = nationalcode.ToCharArray();  
    if (array.Length != 10) return false;  
    var j = 10;  
    var sum = 0;  
    for (var i = 0; i < array.Length - 1; i++)  
    {  
        sum += Int32.Parse(array[i].ToString(CultureInfo.InvariantCulture)) * j;  
        j--;  
    }  
    var diff = sum % 11;  
    if (diff < 2)  
    {  
        lastNumber = diff;  
        return diff == Int32.Parse(array[9].ToString(CultureInfo.InvariantCulture));  
    }  
    var temp = Math.Abs(diff - 11);  
    lastNumber = temp;  
    return temp == Int32.Parse(array[9].ToString(CultureInfo.InvariantCulture));  
}
```

}

البته من فرض کردم که کد ملی 0010350829 معتبر و کد 0010350822 نامعتبر است. آیا شما مطمئن هستید که کد 0010350829 معتبر است؟

نویسنده: صابر فتح الهی
تاریخ: ۸:۴۸ ۱۳۹۲/۰۱/۱۴

سلام
مهندس [این تکه کد](#) توسط آقای [ابراهیم بیاگوی](#) بهبود داده شده و به نظر شخصی من بهتره.

نویسنده: RezaNBT
تاریخ: ۲۰:۴۶ ۱۳۹۲/۰۱/۱۴

بله از صحت کد ملی 0010350829 اطمینان کامل دارم. شما هم می‌توانید محض اطمینان با الگوریتم‌های دیگر تست کنید.

نویسنده: sh-a
تاریخ: ۲۱:۲۷ ۱۳۹۲/۰۱/۱۴

با سلام
من از الگوریتم زیر استفاده میکنم گفتم اینجا بنویسم شاید به کار بیاد

```
Private Function ValidateNcode(ByVal NC As Int64) As Boolean
    If IsNumeric(NC) = False Or NC.ToString.Length < 10 Then
        Return False
    End If
    Dim vnc As String = NC.ToString
    Dim pos As Integer = 10
    Dim sum As Integer = 0

    For i = 0 To 8
        sum += Integer.Parse(vnc.ToCharArray()(i).ToString()) * pos
        pos = pos - 1
    Next i

    Dim remind As Integer = sum Mod 11
    Dim controlNum As Integer = Integer.Parse(vnc.ToCharArray()(9))

    If remind <= 2 Then
        If controlNum = remind Then
            Return True
        Else
            Return False
        End If
    Else
        If (11 - remind) = controlNum Then
            Return True
        Else
            Return False
        End If
    End If
End Function
```

```
private bool ValidateNcode(Int64 NC)
{
    if (IsNumeric(NC) == false | NC.ToString.Length < 10) {
        return false;
    }
    string vnc = NC.ToString;
    int pos = 10;
    int sum = 0;

    for (i = 0; i <= 8; i++) {
        sum += int.Parse(vnc.ToCharArray()(i).ToString()) * pos;
        pos = pos - 1;
    }
}
```

```
}  
int remind = sum % 11;  
int controlNum = int.Parse(vnc.ToCharArray()(9));  
  
if (remind <= 2) {  
    if (controlNum == remind) {  
        return true;  
    } else {  
        return false;  
    }  
} else {  
    if ((11 - remind) == controlNum) {  
        return true;  
    } else {  
        return false;  
    }  
}  
}
```

نویسنده: f.beigirad
تاریخ: ۱۴۰۱/۰۶/۱۷

اتباع خارجی هم کد شناسایی دارن.
آیا میشه متدیم برای تشخیص صحت این کد پیشنهاد بدین.

نویسنده: افشین
تاریخ: ۱۳۹۲/۰۸/۲۷ ۷:۵۹

یک حالت جدید پیدا کردم که مطمئنا خود ثبت احوال هم نمی‌دونه! هر عدد ده رقمی که به فرم قریبه وارد بشه، صحیح فرض میشه!
مثلا a00000000a یا 1234554321 یا 1346886431 همگی صحیح فرض میشن حال آنکه اشتباه هستن!

نویسنده: سعید
تاریخ: ۱۳۹۲/۰۸/۲۷ ۱۷:۵۱

من براتون یه کد ملی میارم که کل این الگوریتم رو نقض کنه. کد ملی 1111111111 متعلق به آقای در شهر رضا هست! من چون به تعداد بالایی کد ملی دسترسی دارم، بررسی کردم و الان مطمئن هستم ثبت احوال در سالهای ابتدایی تخصیص کد ملی، خطا داشته.

نویسنده: وحید
تاریخ: ۱۳۹۳/۰۴/۲۲ ۱:۳۹

سلام به دوستان عزیز، این الگوریتم برای حدود 10 درصد از کد ملی‌ها درست عمل نمی‌کند به عنوان نمونه کدهای زیر در الگوریتم فوق فاقد اعتبار هستند (از صحت کد ملی اطمینان دارم)

1381647899
0922351873
4122587241
4312281949
4220196526
6219937997
0077910345
4569973963
2133288158
2003987765
2758531405
4251235692

2156345454

1222365215

5412568954

از این دست می‌توانم تعداد بسیاری را مثال بزنم

با Html Helper ما میتوانیم لینک‌های متن دار را ایجاد نماییم. شاید گاهی پیش آید که بجای لینک‌های متنی، از تصویر بجای لینک بخواهید استفاده نمایید. می‌توان هر زمانی، لینکی را که حاوی یک تصویر باشد، ایجاد کنید. اما با Asp.net MVC یکی از راه‌های مناسب برای انجام این کار استفاده از Extension methods است که از تکرار کد نویسی نیز جلوگیری می‌نماید. همان طور که در کد زیر مشاهده می‌نمایید، این کد تشکیل شده است از نوشته لینک و نام اکشن متد و نام کنترلر آن، همراه با آی دی و تعریف css. کاری که قرار است صورت گیرد همانند ActionLink می‌باشد، با این تفاوت که تصویری هم به لینک ما اضافه خواهد شد:

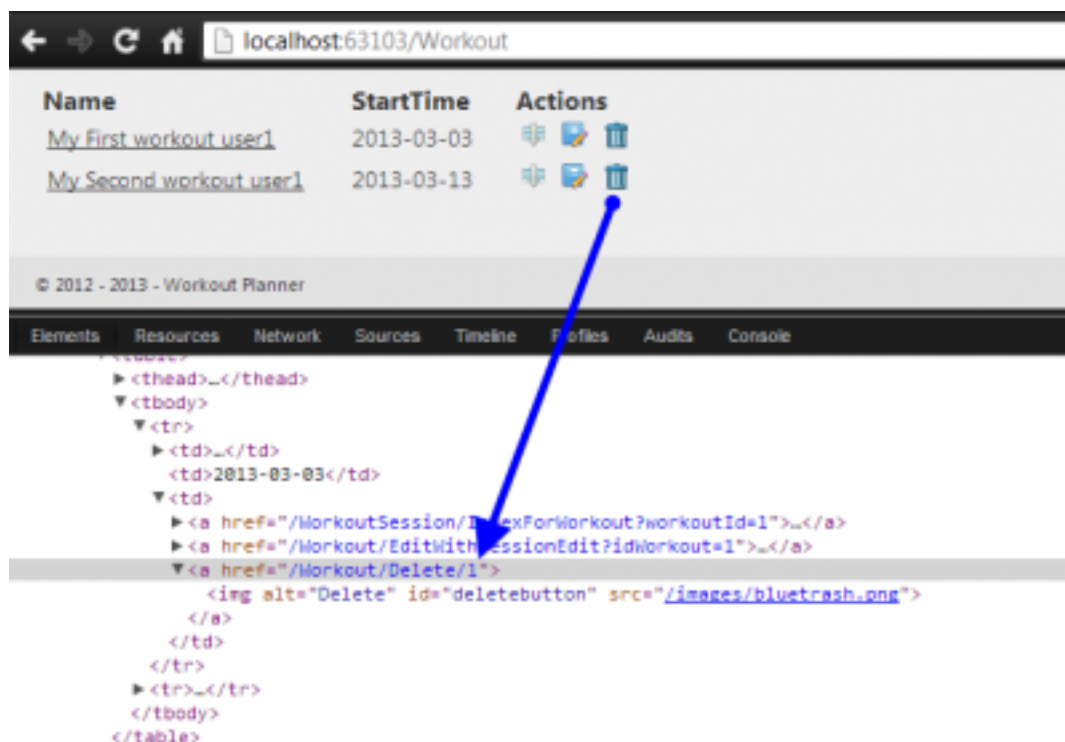
```
@Html.ActionLink("Text to display", "ActionNameHere", "ControllerNameHere", new { id = 123 },
new{@class="myClass"})
```

برای نوشتن این متد جهت تولید لینک‌هایی به همراه تصویر، از کدهای زیر استفاده می‌کنیم:

```
public MvcHtmlString ActionImage(this HtmlHelper htmlHelper,
    string controller,
    string action,
    object routeValues,
    string imagePath,
    string alternateText = "",
    object htmlAttributes = null)
{
    var anchorBuilder = new TagBuilder("a");
    var url = new UrlHelper(htmlHelper.ViewContext.RequestContext);
    anchorBuilder.MergeAttribute("href", url.Action(action, controller, routeValues));
    var imgBuilder = new TagBuilder("img");
    imgBuilder.MergeAttribute("src", url.Content(imagePath));
    imgBuilder.MergeAttribute("alt", alternateText);
    var attributes = (IDictionary<string, object>)
    htmlHelper.AnonymousObjectToHtmlAttributes(htmlAttributes);
    imgBuilder.MergeAttributes(attributes);
    string imgHtml = imgBuilder.ToString(TagRenderMode.SelfClosing);
    anchorBuilder.InnerHtml = imgHtml;
    return MvcHtmlString.Create(anchorBuilder.ToString());
}
```

اولین پارامتر اجازه گسترش HtmlHelper را می‌دهد و توسط آن می‌توان متدهای جدید خود را با کلمه کلیدی @Html شروع کنید. 4 پارامتر بعدی، پارامترهای ضروری هستند. یکی controller، پس از آن routeValues و سپس آدرس تصویر است. در نهایت، ما می‌توانیم در صورت نیاز متن جایگزین و ویژگی‌های اضافی متنی را نیز مشخص نماییم. در رابطه با توضیحات مربوط به بدنه متد: ابتدا لینک‌ها را ایجاد خواهد کرد با استفاده از RequestContext و UrlHelper. سپس بخش ساخت تصویر و اضافه کردن آن به لینک است. برای استفاده از متد زیر استفاده نمود:

```
@Html.ActionImage("Workout", "Delete", new { id = @workout.Id }, "/images/bluetrash.png", "Delete", new
{ id="deletebutton"})
```



در شکل بالا خروجی HTML تولید شده را مشاهده می‌نمایید.

نظرات خوانندگان

نویسنده: Hamid NCH
تاریخ: ۱۰:۱ ۱۳۹۲/۰۵/۲۵

سلام و ممنون از مطلب ارسالی خیلی مفیدتون. از این خط خطا گرفته میشه:

```
var url = new UrlHelper(HtmlHelper.ViewContext.RequestContext);
```

و خطای گرفته شده:

Error 1 An object reference is required for the non-static field, method, or property
"System.Web.Mvc.HtmlHelper.ViewContext.get"

نویسنده: Hamid NCH
تاریخ: ۱۰:۶ ۱۳۹۲/۰۵/۲۵

با عرض معذرت. مشکل از کد خودم بود و در این کد:

```
var url = new UrlHelper(htmlHelper.ViewContext.RequestContext);
```

htmlHelper را به صورت HtmlHelper نوشته بودم و اصلا متوجه این بخش نبودم. ممنون

نویسنده: داود
تاریخ: ۱۴:۱۸ ۱۳۹۲/۱۰/۰۷

سلام

مرسی از مطالب مفیدتون. میشه راهنمایی کنید که چطوری توی یه پروژه MVC از این کد باید استفاده کرد؟

نویسنده: محسن خان
تاریخ: ۱۹:۳۶ ۱۳۹۲/۱۰/۰۷

یک کلاس استاتیک جدید درست کنید. بعد متد الحاقی ActionImage رو بهش اضافه کنید. حالا در کدهای Razor مطابق توضیحی که دادند Html.ActionImage قابل استفاده می‌شود. اگر لازم بود فضا نام کلاس را هم ابتدای فایل View اضافه کنید.

نویسنده: داود
تاریخ: ۹:۲۹ ۱۳۹۲/۱۰/۰۸

```
Compiler Error Message: CS0103: The name 'workout' does not exist in the current context
@using ActionImage.Extentions;
@Html.ActionImage("Workout", "Delete", new { id = @workout.Id }, "/images/bluetrash.png", "Delete", new
{ id="deletebutton"})
```

نویسنده: محسن خان
تاریخ: ۹:۵۱ ۱۳۹۲/۱۰/۰۸

بجای اون id هرچی دوست داشتی بنویس. احتمالا یک حلقه بوده که workout یکی از اشیاء اون هست. مهم نیست. جزو کار نیست.

نویسنده: رشوند
تاریخ: ۱۸:۴۷ ۱۳۹۳/۰۴/۲۹

سپاس، بسیار مفید بود.
البته در کد آقا مهرداد هم به صورت H نوشته شده که لطفا اصلاح بفرمایید...

نویسنده: رشوند
تاریخ: ۱۶:۳۰ ۱۳۹۳/۰۵/۱۱

یه سوال برام پیش اومد
فرض کنیم کاربری لاگین کرده رولش هم مثلا خبرنگاره،
در پروفایلش لیست خبر هایی که ثبت کرده را می بینه با توجه به مثالی که شما هم زدید ، اگر مثلا عدد "1" لینک ویرایش (یا حذف) که نشانگر id است رو به صورت دستی(تغییر کد html و ...) تغییر بده، می تونه به خبر دیگه ای دسترسی داشته باشه ؟
خبری که ثبت کنندش خودش نیست بتبع هم اجازه ویرایش و حذف ان را نباید داشته باشه...
راه حل چیست؟ آیا باید قبل از نشان دادن صفحه ویرایش یا قبل از عمل حذف ثبت کننده خبر هم چک بشه (که فردی که این خبر رو ثبت نکرده دسترسی به ان نداشته باشه)؟
لطفا راهنمایی بفرمایید.
سپاس.

نویسنده: وحید نصیری
تاریخ: ۱۸:۳۵ ۱۳۹۳/۰۵/۱۱

- وجود [anti forgery token](#) جهت مقابله با جعل صفحه و درخواست است.
- امکان [رمزنگاری خودکار فیلدهای مخفی](#) وجود دارد.
- بررسی های سمت سرور هم جای خودش را دارند و لازم هستند.

شاید برای شما هم پیش آمده باشد که بخواهید در هر بار واکنشی لیستی از اطلاعات، مثلاً از دیتابیس، آیتمهای آن را بصورت تصادفی مرتب کنید.

من در پروژه اخیرم برای نمایش یک سری سوال مجبور بودم که در هر بار نمایش سوالات، لیست را به صورت رندوم مرتب کنم و به کاربر نمایش بدم. برای حصول این مهم، یک extension method به شکل زیر نوشتم:

```
public static class RandomExtentions
{
    public static void Shuffle<T>(this IList<T> list)
    {
        Random rng = new Random();
        Thread.Sleep(100);
        int n = list.Count;
        while (n > 1)
        {
            n--;
            int k = rng.Next(n + 1);
            T value = list[k];
            list[k] = list[n];
            list[n] = value;
        }
    }
}
```

در این تابع که اسمش را Shuffle گذاشتم، با دریافت یک لیست از نوع T، آیتمهای درون لیست را به صورت تصادفی مرتب می‌کند.

مثال :

```
var x = new List<int>();
x.Add(1);
x.Add(2);
x.Add(3);
x.Add(4);
x.Add(5);
x.Shuffle();
```

در این مثال لیست x که از نوع int میباشد پس از فراخوانی Shuffle به یک لیست نامرتب تبدیل میشود که نحوه چیدمان در هر بار فراخوانی، تصادفی خواهد بود.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۴:۱۶ ۱۳۹۳/۰۲/۰۷

اگر از EF استفاده می‌کنید، برای اینکار یک ستون Guid پویا را اضافه می‌کند. سپس بر اساس این ستون، مرتب سازی را انجام می‌دهد. [اطلاعات بیشتر](#)

نویسنده: بهزاد دات نت
تاریخ: ۱۴:۳۰ ۱۳۹۳/۰۲/۰۷

با سپاس از شما. در صورت استفاده از EF روشی که شما فرمودین بهتر و کارآمدتر هستش.