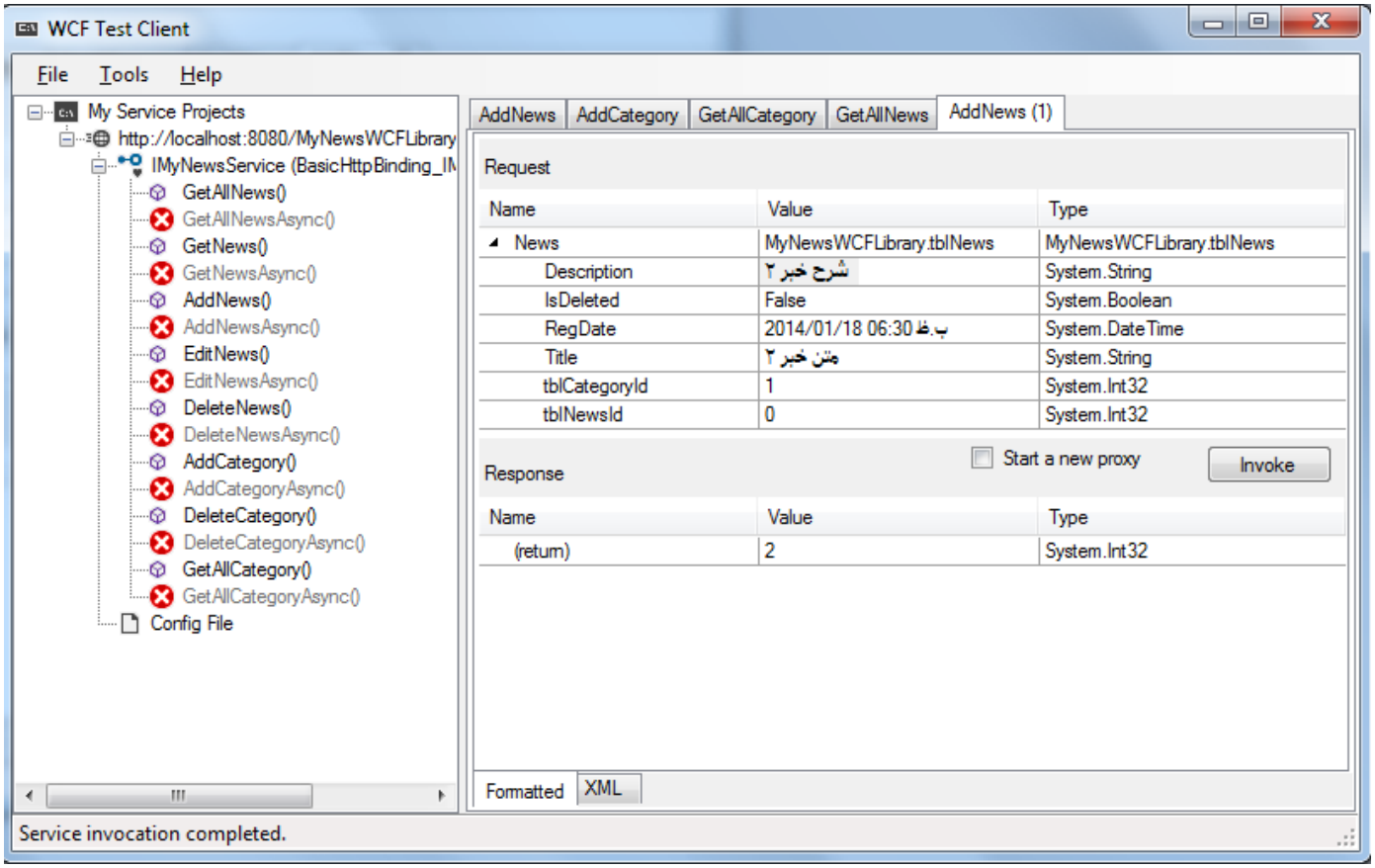
عنوان: **ایجاد سرویس چندلایه‌ی WCF با Entity Framework در قالب پروژه - 6**
نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۱۰/۲۹ ۵:۱۰
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, WCF, Entity Framework 6

پروژه را اجرا کنید و در WCF Test Client به وسیله‌ی متد AddNews دو خبر جدید درج کنید.



روی متدهای GetAllCategory و GetAllNews به صورت جداگانه کلیک کنید. متوجه خواهید شد که هرچند در کلاس tblNews شی‌ای از نوع tblCategory و در کلاس tblCategory شی‌ای از نوع مجموعه‌ی tblNews به صورت Virtual تعریف شده است ولی در بر خلاف انتظارمان اثری از آن در این‌جا دیده نمی‌شود. نتیجه‌ی مشاهده‌شده به خاطر است که در هر دو تعریف صفت DataMember را به ویژگی‌های ناوبری اختصاص نداده ایم و این می‌تواند راهبرد ما در طراحی WCF باشد. ولی اگر می‌خواهید ویژگی ناوبری میان موجودیت‌ها در متدهای ما هم دیده شود ادامه‌ی این درس را بخوانید وگرنه ممکن است تصمیم داشته باشید در صورت نیاز به پیوند میان موجودیت‌ها، متد جدیدی بنویسید و از دستورهای Linq استفاده کنید و یا برای این‌کار از Stored Procedured بهره ببرید.

در اینجا من این سناریو را دنبال می‌کنم که در صورتی که متد GetAllNews اجرا شود؛ بدون این‌که نیاز باشد برای دانستن نام دسته‌ی خبر از متد دیگری مانند GetAllCategory استفاده کنیم؛ رکورد وابسته موجودیت دسته در هر خبر نشان داده شود.

از Solution Explorer فایل MyNewsModel.tt را باز کنید و دنبال کد زیر بگردید:

```
public string NavigationProperty(NavigationProperty navigationProperty)
    {
        var endType = _typeMapper.GetTypeName(navigationProperty.ToEndMember.GetEntityType());
        return string.Format(
```

```
            CultureInfo.InvariantCulture,
            "{0} {1} {2} {{ {3}get; {4}set; }}",
            AccessibilityAndVirtual(Accessibility.ForProperty(navigationProperty)),
            navigationProperty.ToEndMember.RelationshipMultiplicity == RelationshipMultiplicity.Many ?
("ICollection<" + endType + ">") : endType,
            _code.Escape(navigationProperty),
            _code.SpaceAfter(Accessibility.ForGetter(navigationProperty)),
            _code.SpaceAfter(Accessibility.ForSetter(navigationProperty)));
    }
```

سپس آن‌را به صورت زیر ویرایش کنید:

```
public string NavigationProperty(NavigationProperty navigationProperty)
    {
        var endType = _typeMapper.GetTypeName(navigationProperty.ToEndMember.GetEntityType());
        return string.Format(
            CultureInfo.InvariantCulture,
            "{0}{1} {2} {3} {{ {4}get; {5}set; }}",
navigationProperty.ToEndMember.RelationshipMultiplicity != RelationshipMultiplicity.Many ?
"[DataMember]" + Environment.NewLine : "",
            AccessibilityAndVirtual(Accessibility.ForProperty(navigationProperty)),
            navigationProperty.ToEndMember.RelationshipMultiplicity == RelationshipMultiplicity.Many ?
("ICollection<" + endType + ">") : endType,
            _code.Escape(navigationProperty),
            _code.SpaceAfter(Accessibility.ForGetter(navigationProperty)),
            _code.SpaceAfter(Accessibility.ForSetter(navigationProperty)));
    }
```

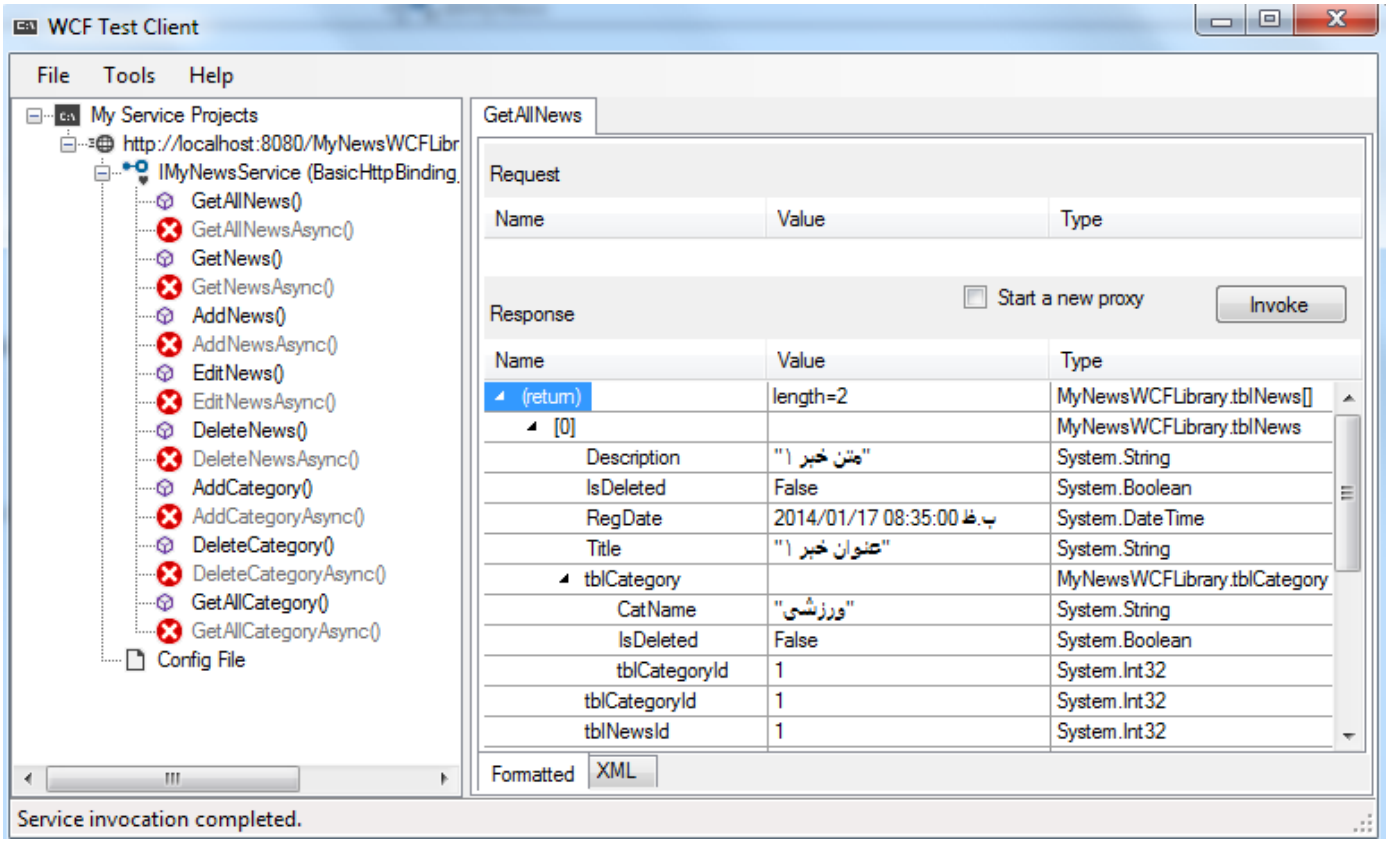
پس از ذخیره‌ی فایل، خواهید دید که صفت DataMember در کلاس tblNews پیش از ویژگی tblCategory افزوده شده است. بار دیگر پروژه را اجرا کنید. روی متد GetAllNews کلیک کنید و روی دکمه Invoke بفشارید. خواهید دید که هرچند tblCategory در ویژگی‌های آن قرار گرفته است ولی مقدار آن Null است. برای حل این مشکل باید از Solution Explorer فایل MyNewsService.cs را باز کنید و به به جای کد مربوط به متدهای GetAllNews و GetNews کدهای زیر را قرار دهید:

```
public List<tblNews> GetAllNews()
        {
            return dbMyNews.tblNews.Include(p=>p.tblCategory).Where(c=>c.IsDeleted == false).ToList();
```

```
}

        public tblNews GetNews(int tblNewsId)
        {
            return dbMyNews.tblNews.Include(p => p.tblCategory).FirstOrDefault(p => p.tblNewsId ==
tblNewsId);
        }
```

این بار اگر پروژه را اجرا کنید با نتیجه‌ای مانند شکل زیر روبه‌رو خواهید شد:



در بخش هفتم پیرامون میزبانی WCF Library خواهم نوشت.