

پس از ایجاد متدها، نوبت به تغییرات App.Config می‌رسد. هرچند خود Visual Studio برای کلاس پیش‌گزیده‌ی خود تنظیماتی را در App.Config افزوده است ولی چنانچه در در خاطر دارید ما آن فایل‌ها را حذف کردیم و فایل‌های جدیدی به جای آن افزودیم. از این رو مراحل زیر را انجام دهید:

1- فایل App.Config را از Solution Explorer باز کنید.

2- به جای عبارت MyNewsWCFLibrary.Service1 در قسمت Service Name این عبارت را بنویسید:

MyNewsWCFLibrary.MyNewsService

3- در قسمت BaseAddress عبارت Design\_Time\_Addresses را حذف کنید.

4- در قسمت BaseAddress شماره پورت را به 8080 تغییر دهید.

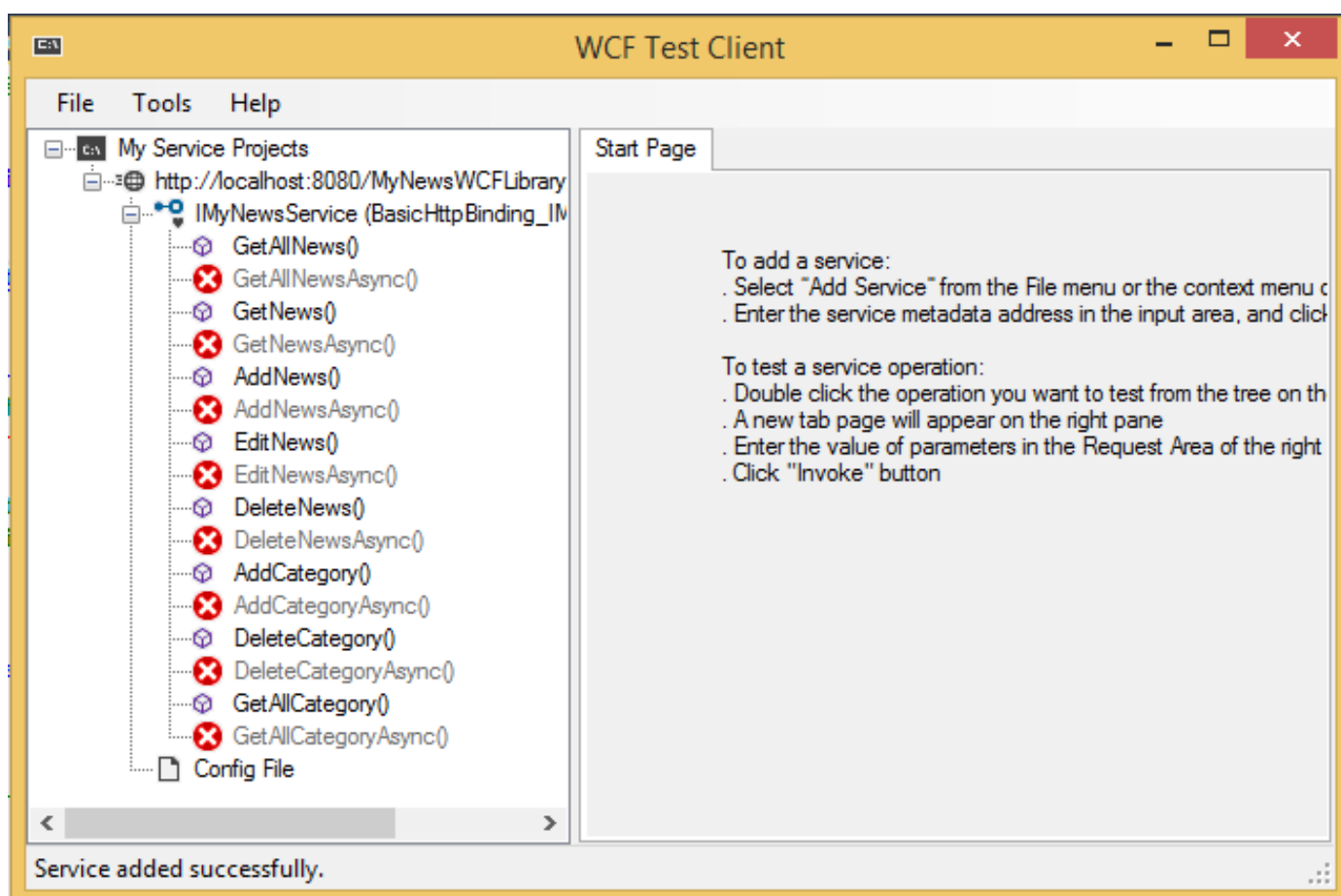
5- در قسمت BaseAddress به جای Service1 بنویسید: MyNewsService

6- در قسمت endpoint به جای عبارت MyNewsWCFLibrary.IService1 بنویسید: MyNewsWCFLibrary.IMyNewsService

در پایان تگ Service در App.Config باید همانند کد زیر باشد:

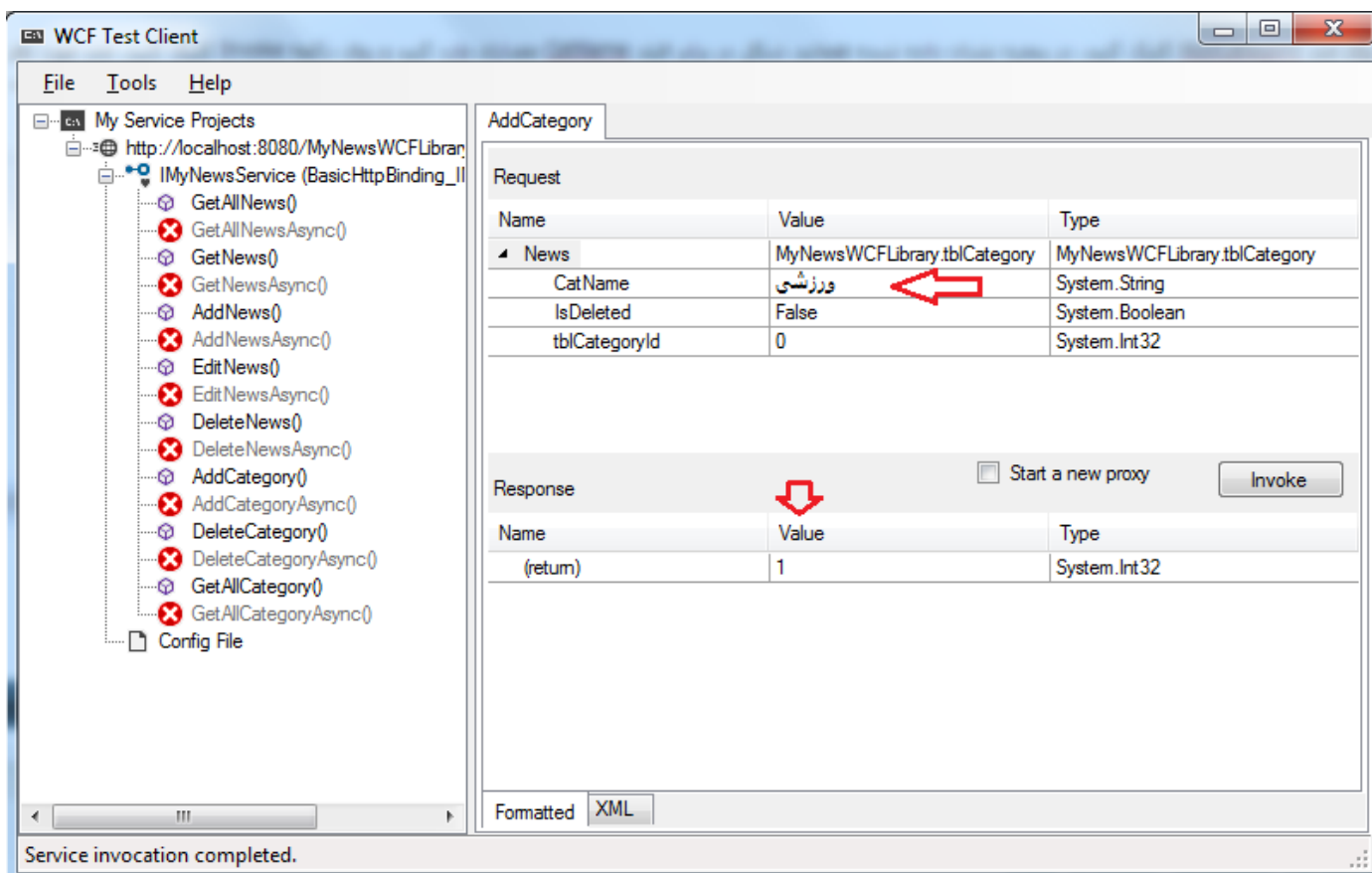
```
<services>
  <service name="MyNewsWCFLibrary.MyNewsService">
    <host>
      <baseAddresses>
        <add baseAddress="http://localhost:8080/MyNewsWCFLibrary/MyNewsService/" />
      </baseAddresses>
    </host>
    <!-- Service Endpoints -->
    <!-- Unless fully qualified, address is relative to base address supplied above -->
    <endpoint address="" binding="basicHttpBinding" contract="MyNewsWCFLibrary.IMyNewsService">
      <!--
        Upon deployment, the following identity element should be removed or replaced to reflect
        the identity under which the deployed service runs. If removed, WCF will infer an
        appropriate identity automatically.
      -->
      <identity>
        <dns value="localhost" />
      </identity>
    </endpoint>
    <!-- Metadata Endpoints -->
    <!-- The Metadata Exchange endpoint is used by the service to describe itself to clients. -->
    <!-- This endpoint does not use a secure binding and should be secured or removed before
    deployment -->
    <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />
  </service>
</services>
```

تغییرات را ذخیره کنید و پروژه را اجرا کنید. باید پنجره‌ای شبیه به پنجره‌ی زیر نشان داده شود:

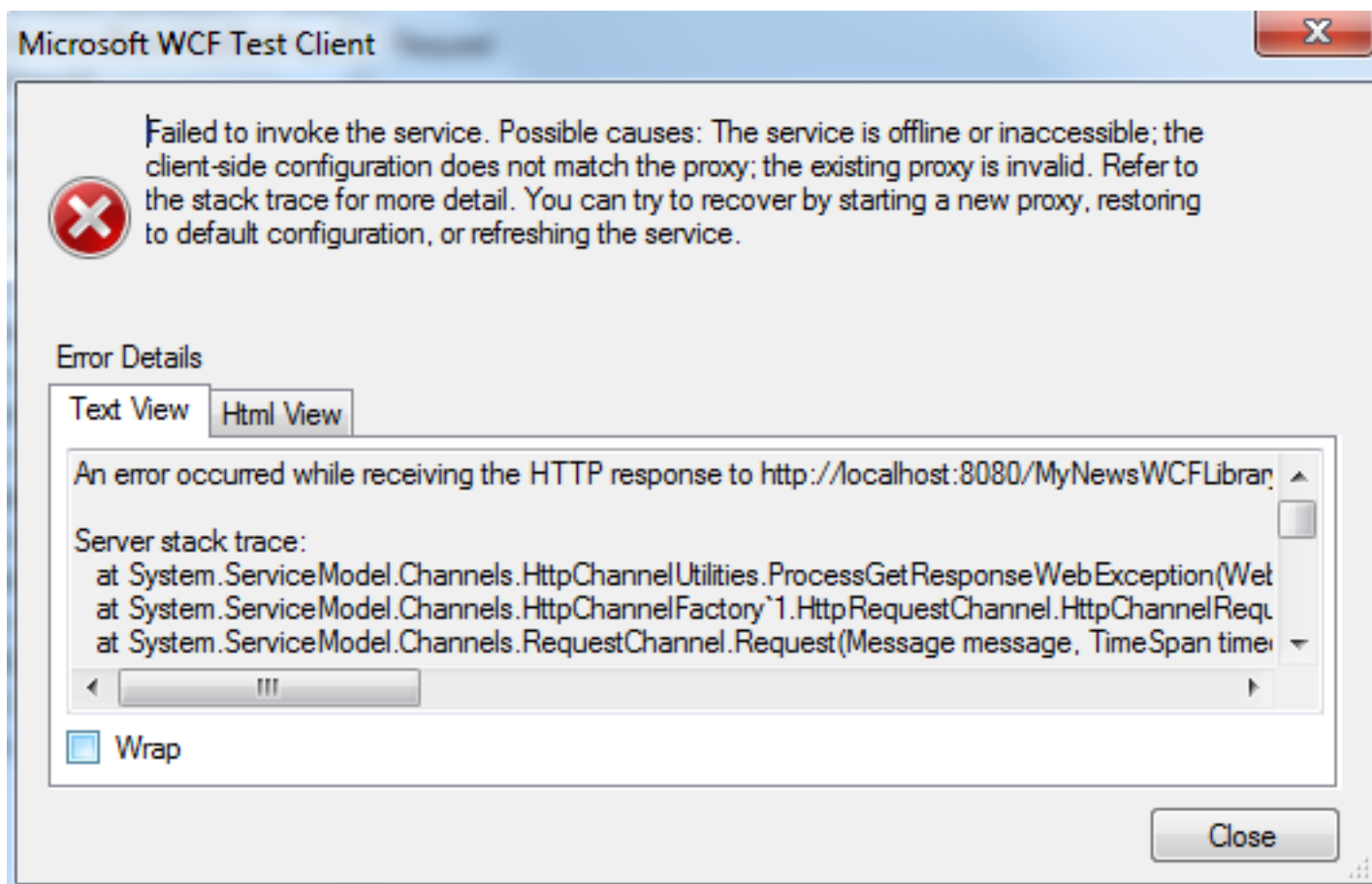


در صورت مشاهده پیام خطا، ویژوال استودیو را ببندید و این بار به صورت Run as administrator باز کنید.

برای نمونه روی متد `AddCategory` کلیک کنید. در پنجره نشان داده شده همانند شکل در برابر فیلد `CatName` مقداری وارد کنید و روی دکمه `Invoke` کلیک کنید. متد مورد نظر اجرا شده و مقداری که وارد کرده ایم در پایگاه داده‌ها ذخیره می‌شود. مقداری که در قسمت پایین دیده می‌شود خروجی متد است که در اینجا شناسه رکورد درج شده است.



بار دیگر برای مشاهده رکورد درج‌شده روی متد `GetAllCategory` کلیک کنید. به علت این‌که این متد ورودی ندارد در قسمت بالا چیزی نشان داده نمی‌شود. روی دکمه `Invoke` کلیک کنید. با پیغام خطای زیر روبه‌رو خواهید شد:



افزودن ویژگی Virtual به tblNews و tblCategory در [بخش دوم](#) خواندید؛ باعث می‌شود که Entity Framework در هنگام اجرا کلاس‌هایی با عنوان "پروکسی‌های پویا" به کلاس‌های Address و Customer بیفزاید و بنابراین قابلیت Lazy Loading برای این کلاس‌ها در زمان اجرای برنامه فراهم می‌گردد.

ولی با افزودن پروکسی‌های پویا به کلاس‌های ما، این کلاس‌ها قابلیت انتقال خود از طریق سرویس‌های WCF را از دست می‌دهند زیرا پروکسی‌های پویا به طور پیش‌گزینه قابلیت سریالایز و دیسریالایز شدن را ندارند!

خوشبختانه می‌توانیم این ویژگی را در کلاس DbContext غیرفعال کنیم. برای این منظور قالب سازنده‌ی آن یا MyNewsModel.Context.tt را از Solution Explorer باز کنید و کد زیر را در آن پیدا کنید:

```
<#Accessibility.ForType(container)> partial class <#code.Escape(container)> : DbContext
{
    public <#code.Escape(container)>()
        : base("name=<#container.Name>")
    {
```

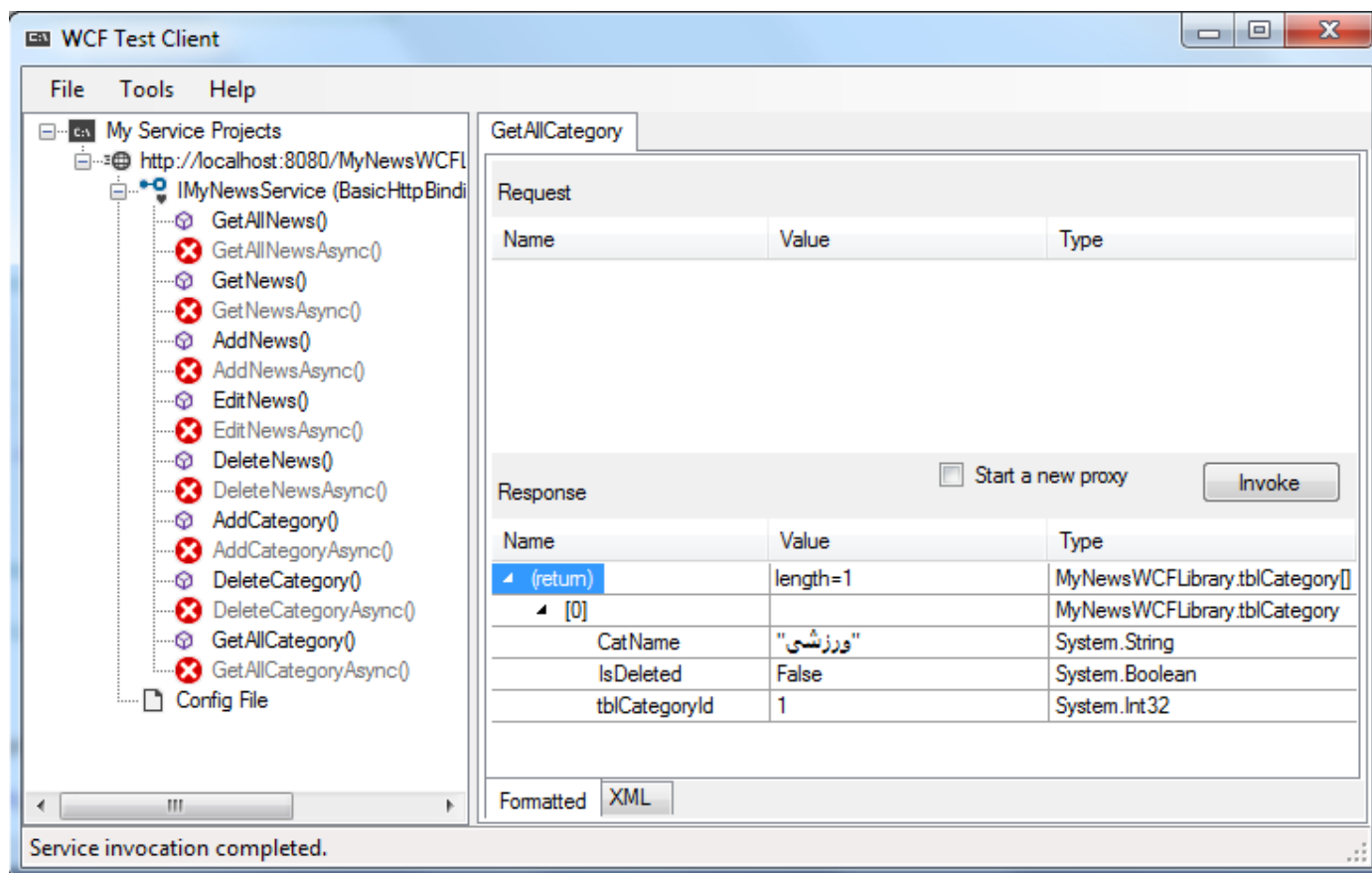
سپس در ادامه‌ی آن کد غیرفعال کردن پروکسی پویا را به این شکل بنویسید:

```
<#Accessibility.ForType(container)> partial class <#code.Escape(container)> : DbContext
{
    public <#code.Escape(container)>()
        : base("name=<#container.Name>")
    {
        Configuration.ProxyCreationEnabled = false;
```

اکنون اگر فایل را ذخیره کنیم سپس فایل MyNewsModel.Context.cs را از Solution Explorer باز کنید؛ خواهید دید که این خط

کد در جای خود قرار گرفته است.

بار دیگر پروژه را اجرا کنید روی متد GetAllCategory کلیک کنید. این بار اگر دکمه Invoke را بفشارید با همانند شکل زیر را خواهید دید:



در بخش ششم پیرامون ارتباط جدول‌های tblNews و tblCategory و نمایش محتویات وابسته جدول خبر به دسته و تنظیمات آن در t4 و کلاس Service

در بخش هفتم پیرامون میزبانی WCFLibrary در یک Web Application

و در بخش هشتم پیرامون ایجاد یک برنامه‌ی ویندوزی جهت استفاده از سرویس‌های WCF خواهم نوشت.