

سه نوع استثنای مهم ممکن است حین ذخیره سازی تغییرات در EF code first رخ دهند که بررسی جزئیات آن‌ها می‌تواند راهنمای خوبی برای کاربر و همچنین برنامه نویسی در عیب یابی سیستم باشد. این استثناءها باید به صورت مستقل و جداگانه بررسی شوند و نه اینکه از حالت عمومی catch Exception استفاده شود.

این سه نوع استثناء شامل موارد DbUpdateConcurrencyException، DbEntityValidationException و DbUpdateException هستند که به صورت خلاصه به شکل زیر باید تعریف شوند:

```
try
{
    context.SaveChanges();
}
catch (DbEntityValidationException validationException)
{
    //...
}
catch (DbUpdateConcurrencyException concurrencyException)
{
    //...
}
catch (DbUpdateException updateException)
{
    //...
}
```

توضیحات تکمیلی

در حالت **DbEntityValidationException** به جزئیات خطاهای حاصل از اعتبار سنجی اطلاعات خواهیم رسید. برای مثال اگر قرار است طول فیلدی 30 حرف باشد و کاربر 40 حرف را وارد کرده است، نام خاصیت و همچنین پیغام خطای در نظر گرفته شده را دقیقاً در اینجا می‌توان دریافت کرد و به نحو مقتضی به کاربر نمایش داد:

```
catch (DbEntityValidationException validationException)
{
    foreach (var error in validationException.EntityValidationErrors)
    {
        var entry = error.Entry;
        foreach (var err in error.ValidationErrors)
        {
            Debug.WriteLine(err.PropertyName + " " + err.ErrorMessage);
        }
    }
}
```

نوع استثنای **DbUpdateConcurrencyException** به مسایل همزمانی و به روز رسانی یک رکورد توسط دو یا چند کاربر در شبکه مرتبط می‌شود که در **قسمت سوم** سری EF code first با معرفی ویژگی‌های ConcurrencyCheck و Timestamp در مورد آن بحث شد. در اینجا به کلیه موجودیت‌های تداخل دار توسط خاصیت concurrencyException.Entries خواهیم رسید و همچنین به کمک متد GetDatabaseValues می‌توان موارد جدید ثبت شده مرتبط با این موجودیت تداخل دار را از بانک اطلاعاتی نیز دریافت کرد:

```
catch (DbUpdateConcurrencyException concurrencyException)
{
    // بررسی مورد اول
    var dbEntityEntry = concurrencyException.Entries.First();
    var dbPropertyValues = dbEntityEntry.GetDatabaseValues();
}
```

```
}
```

و یا کلاً ممکن است حین به روز رسانی بانک اطلاعاتی مشکلی رخ داده باشد که در اینجا عموماً پیغام حاصل را باید در InnerException تولیدی یافت و همچنین در اینجا لیست موجودیت‌های مشکل دار نیز قابل دریافت و بررسی هستند:

```
catch (DbUpdateException updateException)
{
    if (updateException.InnerException != null)
        Debug.WriteLine(updateException.InnerException.Message);

    foreach (var entry in updateException.Entries)
    {
        Debug.WriteLine(entry.Entity);
    }
}
```

بنابراین بررسی catch exception کلی در EF Code first مناسب نبوده و نیاز است بیشتر به جزئیات ذکر شده، وارد و دقیق شد.

یک نکته:

بهتر است یک کلاس پایه عمومی مشتق شده از DbContext را ایجاد و متد SaveChanges آن را تعریف کرد. سپس سه حالت فوق را به آن اعمال نمود. اکنون می‌توان از این کلاس پایه بارها استفاده کرد بدون اینکه نیازی به تکرار کدهای آن در هر جایی که قرار است از متد SaveChanges استفاده شود، باشد. شبیه به این کار را در [قسمت 14](#) سری EF code first مشاهده نموده‌اید.

نظرات خوانندگان

نویسنده: علی قمشلویی
تاریخ: ۲۲:۸ ۱۳۹۱/۰۳/۳۱

با سلام و تشکر مثل همیشه عالی

نویسنده: Mona
تاریخ: ۲۳:۱۶ ۱۳۹۱/۰۳/۳۱

یک خواهش، بدلیل اینکه سطوح علمی مراجعین متفاوت است، پیشنهاد میکنم در صورت امکان مثل codeproject یک فایل Sample هم برای پست‌ها آپلود شود.
ممنون

نویسنده: محمدرضا
تاریخ: ۲۲:۳۷ ۱۳۹۱/۰۴/۰۴

آیا می‌توان به جای نمایش یک پیام انگلیسی هنگام وقوع خطای اعتبارسنجی مثلا از

```
[Required(ErrorMessage = "لطفا نام نویسنده را مشخص نمایید")]
```

استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۳۹ ۱۳۹۱/۰۴/۰۴

بله. اگر از ASP.NET MVC استفاده کنید به صورت خودکار تبدیل به پیغام‌های اعتبار سنجی سمت کاربر هم می‌شود. ولی در کل اثر بررسی سمت سرور هم دارد و همین پیغام‌ها توسط DbEntityValidationException قابل دریافت هستند .

نویسنده: محمدرضا
تاریخ: ۲۲:۵۹ ۱۳۹۱/۰۴/۰۴

من یک پیام خطا برای یکی از Property ها تعیین کردم.وقتی می‌خوام به طریق زیر پیام رو دریافت کنم بازم همون خطای انگلیسی رو مشاهده می‌کنم!؟

```
catch (DbEntityValidationException validationException)
{
    var errors = validationException.EntityValidationErrors.First();
    foreach (var propertyError in errors.ValidationErrors)
    {
        Console.WriteLine(propertyError.PropertyName + "----" +
propertyError.ErrorMessage);
    }
}
```

نویسنده: وحید نصیری
تاریخ: ۲۳:۳ ۱۳۹۱/۰۴/۰۴

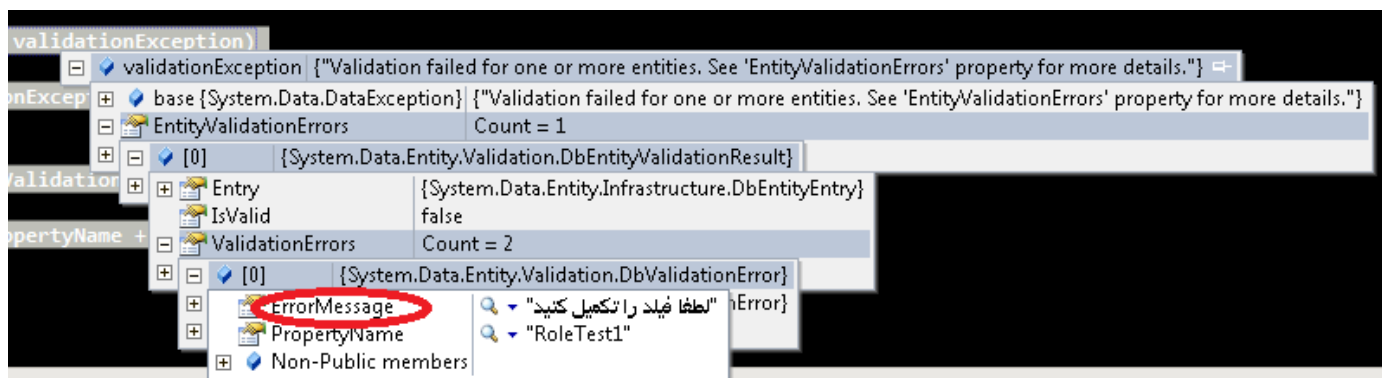
شما از متد First اینجا استفاده کردید. به عبارتی فقط اولین مورد گزارش داده شده رو دارید بررسی می‌کنید و از بقیه موارد صرف‌نظر کردید.

نویسنده: محمدرضا
تاریخ: ۲۳:۵ ۱۳۹۱/۰۴/۰۴

درسته چونکه میخواستم فقط همین خطا رو که عمدی ایجاد کردم ببینم ولی نشد.

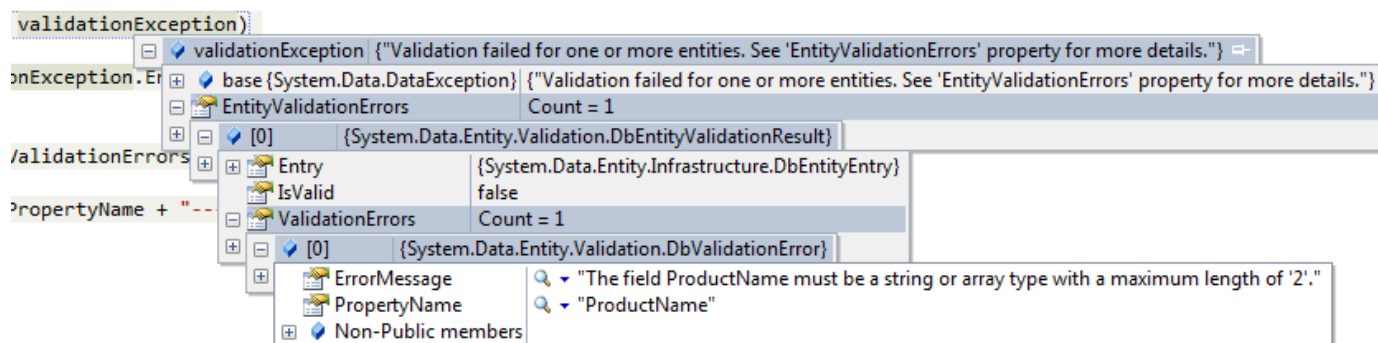
نویسنده: وحید نصیری
تاریخ: ۲۳:۱۴ ۱۳۹۱/۰۴/۰۴

کد کاملی که من در مطلب فوق نوشتم چنین چیزی رو نشون می‌ده:



نویسنده: محمدرضا
تاریخ: ۲۳:۳۵ ۱۳۹۱/۰۴/۰۴

```
[MaxLength(2), Required(ErrorMessage = \"طول فیلد بیش از حد مجاز است\")]  
public string ProductName { get; set; }
```



من از MySQL استفاده کردم. میتونه مشکل از این باشه؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۴۰ ۱۳۹۱/۰۴/۰۴

خیر. علت این است که به ازای تک تک ویژگی‌هایی که تعریف می‌کنید باید ErrorMessage مقدار دهی شود. مثلا در اینجا MaxLength تعریف شده دارای پیغام خطا نیست و آنرا از پیغام خطای Required دریافت نمی‌کند.

نویسنده: محمدرضا
تاریخ: ۲۳:۴۴ ۱۳۹۱/۰۴/۰۴

بله همینطور. باید به این شکل می‌نوشتیم.

```
[MaxLength(2, ErrorMessage = "طول فیلد بیش از حد مجاز است")]
```

خیلی ممنون آقای نصیری.

نویسنده: محمدرضا
تاریخ: ۱۹:۳۸ ۱۳۹۱/۰۴/۰۶

بهتر است یک کلاس پایه عمومی مشتق شده از DbContext را ایجاد و متد SaveChanges آن را تحریف کرد. سپس سه حالت فوق را به آن اعمال نمود. اکنون می‌توان از این کلاس پایه بارها استفاده کرد بدون اینکه نیازی به تکرار کدهای آن در هر جایی که قرار است از متد SaveChanges استفاده شود، باشد. چطور میشه از این استثنای در override متد savechange استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۹ ۱۳۹۱/۰۴/۰۶

به این شکل کلی

```
try
{
    return base.SaveChanges();
}
catch (DbEntityValidationException validationException)
{
    //...
}
//...
```

نویسنده: محمدرضا
تاریخ: ۲۱:۴۴ ۱۳۹۱/۰۴/۰۶

منظورم اینه که متد savechange فقط مقدار int برمیگردونه. اگر استثنایی رخ داد فقط می‌تونیم یه عدد خاص برگردونیم که نشانه‌ی وقوع استثنا هست. در این صورت چطور می‌تونیم پیام خطاهای اعتبارسنجی رو کنترل کنیم؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۵۵ ۱۳۹۱/۰۴/۰۶

اینکه شما چه کاری با این اطلاعات انجام می‌دید به طراحی سیستم خودتون مرتبط است. یکی یک پیام popup نمایش می‌ده یکی یک messagebox یکی هم فقط این‌ها رو لاگ می‌کنه برای بررسی بعدی و یک پیام کلی به کاربر نمایش می‌ده که مشکلی رخ داده. من این‌ها رو لاگ می‌کنم و پیام کلی نمایش می‌دم.

نویسنده: حسینی
تاریخ: ۲۲:۲۳ ۱۳۹۱/۰۸/۰۹

بسم الله الرحمن الرحيم

با سلام

چه جوری میشه بعد از درآوردن اعتبارسنجی‌های مربوط به تک تک پراپرتی‌ها پیام اون‌ها رو در جلوی تکس باکس مربوطه نوشت. راهی که به ذهن بنده میرسه به شکل زیر است:

```
ICollection<DbValidationError> ValidationProperty =  
    context.Entry(product).Property(p => p.Name).GetValidationErrors();  
  
    if (ValidationProperty.Count() > 0)  
    {  
        DbValidationError Error=ValidationProperty.First();  
        errorProvider1.SetError(txtName, Error.ErrorMessage);  
    }
```

خواستم ببینم راه بهتری وجود داره؟

با سپاس فراوان

شب خوش

نویسنده: سارا کیانی
تاریخ: ۱۵:۱۳ ۱۳۹۳/۰۱/۲۷

سلام

چنانچه قصد داشته باشیم در Context خودمان استثنای Connect به سرویس دهنده رو هم در نظر گرفته تا در صورت قطع ارتباط، پیغام خاصی به کاربر نمایش دهیم با کدام Exception در Code First تشخیص داده خواهد شد؟

نویسنده: وحید نصیری
تاریخ: ۱۵:۳۰ ۱۳۹۳/۰۱/۲۷

مانند دوران ADO.NET است:

```
catch (System.Data.SqlClient.SqlException ex)  
{  
    foreach (SqlError error in ex.Errors)  
    {  
        switch (error.Number)  
        {  
            case 1205:  
                System.Diagnostics.Debug.WriteLine("SQL Error: Deadlock condition.");  
                return true;  
  
            case -2:  
                System.Diagnostics.Debug.WriteLine("SQL Error: Timeout expired.");  
                return true;  
  
            case -1:  
                System.Diagnostics.Debug.WriteLine("SQL Error: Timeout expired.");  
                return true;  
        }  
    }  
}
```

ضمناً یک سری مباحث به نام اتصال بهبودپذیر و مقاوم به EF 6 در این زمینه اضافه شده:

[Connection Resiliency Spec](#)

+

[نحوه راه اندازی مجدد یک دیتابیس اس کیوال سرور پس از پر شدن هارد دیسک](#)