عنوان: **یکسان سازی "ی" و "ک" دریافتی در حین استفاده از WCF RIA Services**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۵/۱۶ ۱۴:۰۱:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: Silverlight

یکی از مواردی که در تمام برنامه‌های فارسی "باید" رعایت شود (مهم نیست به چه زبانی یا چه سکویی باشد یا چه بانک اطلاعاتی مورد استفاده است)، بحث اصلاح "ی" و "ک" دریافتی از کاربر و یکسان سازی آن‌ها می‌باشد. به عبارتی برنامه‌ی فارسی که اصلاح خودکار این دو مورد را لحاظ نکرده باشد دیر یا زود به مشکلات حادی برخورد خواهد کرد و "ناقص" است : اطلاعات بیشتر ؛
برای مثال شاید دوست نداشته باشید که دو کامران در سایت شما ثبت نام کرده باشند؛ یکی با ک فارسی و یکی با ک عربی! به علاوه همین کامران امروز می‌تواند لاگین کند و فردا با یک کامپیوتر دیگر و صفحه کلیدی دیگر پشت درب خواهد ماند. در حالیکه از دید این کامران، کلمه کامران همان کامران است!
بنابراین در دو قسمت "باید" این یکسان سازی صورت گیرد:
الف) پیش از ثبت اطلاعات در بانک اطلاعاتی (تا با دو کامران ثبت شده در بانک اطلاعاتی مواجه نشوید)
ب) پیش از جستجو (تا کامران روزی دیگر با صفحه کلیدی دیگر بتواند به برنامه وارد شود)

راه حل یکسان سازی هم شاید به نظر این باشد: رخداد فشرده شدن کلید را کنترل کنید و سپس جایگزینی را انجام دهید (مثلا ی عربی را با ی فارسی جایگزین کنید). این روش چند ایراد دارد:
الف) Silverlight به دلایل امنیتی اصلا چنین اجازه‌ای را به شما نمی‌دهد! (تا نتوان کلیدی را جعل کرد)
ب) همیشه با یک TextBox ساده سر و کار نداریم. کنترل‌های دیگری هم هستند که امکان ورود اطلاعات در آن‌ها وجود دارد و آن وقت باید برای تمام آن‌ها کد نوشت. ظاهر کدهای برنامه در این حالت در حجم بالا، اصلا جالب نخواهد بود و ضمنا ممکن است یک یا چند مورد فراموش شوند.

راه بهتر این است که دقیقا حین ثبت اطلاعات یا جستجوی اطلاعات در لایه‌ای که تمام ثبت‌ها یا اعمال کار با بانک اطلاعاتی برنامه به آنجا منتقل می‌شود، کار یکسان سازی صورت گیرد. به این صورت کار یکپارچه سازی یکبار باید انجام شود اما تاثیرش را بر روی کل برنامه خواهد گذاشت، بدون اینکه هرجایی که امکان ورود اطلاعات هست روال‌های رخداد گردان هم حضور داشته باشند.

در مورد مقدمات WCF RIA Services که درSilverlight و ASP.NET کاربرد دارد می‌توانید به این مطلب مراجعه کنید: +

جهت تکمیل این بحث متدی تهیه شده که کار یکسان سازی ی و ک دریافتی از کاربر را حین ثبت توسط امکانات WCF RIA Services انجام می‌دهد (دقیقا پیش از فراخوانی متد SubmitChanges باید بکارگرفته شود):

```
namespace SilverlightTests.RiaYeKe
{
  public static class PersianHelper
  {
      public static string ApplyUnifiedYeKe(this string data)
      {
          if (string.IsNullOrEmpty(data)) return data;
          return data.Replace("ی" ,"ی").Replace("ک" ,"ک");
      }
  }
}
```

```
using System.Linq;
using System.Windows.Controls;
using System.Reflection;
using System.ServiceModel.DomainServices.Client;

namespace SilverlightTests.RiaYeKe
{
```

```
  public class RIAHelper
  {
      /// <summary>
      /// یک دست سازی ی و ک در عبارات ثبت شده در بانک اطلاعاتی پیش از ورود به آن
      /// این متد باید پیش از فراخوانی متد
      /// SubmitChanges
      /// استفاده شود
      /// </summary>
      /// <param name="dds"></param>
      public static void ApplyCorrectYeKe(DomainDataSource dds)
      {
          if (dds == null)
              return;

          if (dds.DataView.TotalItemCount <= 0)
              return;

          //پیدا کردن موجودیت‌های تغییر کرده
          var changedEntities = dds.DomainContext.EntityContainer.GetChanges().Where(
              c => c.EntityState == EntityState.Modified ||
                   c.EntityState == EntityState.New);

          foreach (var entity in changedEntities)
          {
              //یافتن خواص این موجودیت‌ها
              var propertyInfos = entity.GetType().GetProperties(
                      BindingFlags.Public | BindingFlags.Instance
                      );

              foreach (var propertyInfo in propertyInfos)
              {
                  //اگر این خاصیت رشته‌ای است ی و ک آن را استاندارد کن
                  if (propertyInfo.PropertyType != typeof (string)) continue;
                  var propName = propertyInfo.Name;
                  var val = new PropertyReflector().GetValue(entity, propName);
                  if (val == null) continue;
                  new PropertyReflector().SetValue(
                      entity,
                      propName,
                      val.ToString().ApplyUnifiedYeKe());
              }
          }
      }
  }
}
```

توضیحات:

از آنجائیکه حین فراخوانی متد SubmitChanges فقط موجودیت‌های تغییر کرده جهت ثبت ارسال می‌شوند، ابتدا این موارد یافت شده و سپس خواص عمومی تک تک این اشیاء توسط عملیات Reflection بررسی می‌گردند. اگر خاصیت مورد بررسی از نوع رشته‌ای بود، یکبار این یک دست سازی اطلاعات ی و ک دریافتی صورت خواهد گرفت (و از آنجائیکه این تعداد همیشه محدود است عملیات Reflection سربار خاصی نخواهد داشت).

اگر در کدهای خود از DomainDataSource استفاده نمی‌کنید باز هم تفاوتی نمی‌کند. متد ApplyCorrectYeKe را از قسمت DomainContext.EntityContainer به بعد دنبال کنید.

اکنون تنها مورد باقیمانده بحث جستجو است که با اعمال متد ApplyUnifiedYeKe به مقدار ورودی متد جستجوی خود، مشکل حل خواهد شد.

کلاس PropertyReflector بکارگرفته شده هم از اینجا به عاریت گرفته شد.

دریافت کدهای این بحث

نظرات خوانندگان

نویسنده: شنگول
تاریخ: ۱۳۸۹/۰۵/۱۶ ۲۲:۴۱:۵۸

سلام مهندس نصیری
خسته نباشید و دستمریزاد
اما اینکار که شما گفتین هم یه مشکل جدید دیگه را به وجود میاره و اون اینه که « ک » در مرتب سازی بعد از « ی » قرار میگیره! « ک » در جای خودش قرار میگیره یعنی بعد از حرف «ق».
این میتونه خیلی از کاربران را در خیلی از برنامه‌ها دچار مشکل کنه. یعنی خیلی وقتها (بخصوص کارمندان اداری) کار سرچ را توسط برنامه انجام نمیدن و در یک دیتاگرید اطلاعات لود شده را می‌بینن و با چشم دنبالش میگردن (من اینو در خیلی از برنامه‌ها و خیلی از ادارات و سازمان‌ها دیده‌م، حتی در خیلی از برنامه‌های حسابداری و فروش)
حالا فرض کنید کاربر داره دنبال یک نام و یا یک محصول میگرده که با حرف « ک » شروع شده ولی هرچی لیست را حول و حوش « ق » و « م » بالا و پایین می‌کنه چیزی نمیبینه و باید بدونه که لازمه به انتهای لیست بره تا اونایی را با حرف « ک » شروع شده‌ن بتونه پیدا کنه!
مشکل وقتی بدتر میشه که بدونیم این شکل از مرتب سازی فقط مورد حرف اول نیست و هرجایی از کلمه که حرف « ک » وجود داشته باشه، اگر کلمات دیری شبیه اون هم وجود داشته باشن، اون کلمه که « ک » داره به انتهای لیست میره (در محدودهٔ همون کلمات) و این خودش میتونه ریپورت‌های خروجی برنامه را اگر بر حسب اون فیلد مرتب شده باشن دچار اشکال کنه و اگه پرینت هم لازم باشه مشکل کم‌کم بزرگتر و پیچیده‌تر هم میشه.

خلاصه اینکه همونطور که خودتون توی مطلب قبلی هم نوشتین باید راهکاری اندیشیده بشه که سیستم‌ها (بخصوص پایگاه‌های داده) این حروف را یکی در نظر بگیرن، ولی تا حصول این مقصود که چندان هم نزدیک بنظر نمیرسه، شخصاً فکر میکنم بجای تبدیل تمام « ک » ها (کاف عربی) به « ک » (کاف فارسی) بهتره که همهٔ اونها را به « ک » عربی تبدیل کنیم و برای نمایش اوناه از فونتهای فارسی که « ک » را بصورت « ک » نشون میدن استفاده کنیم ولی در مورد «ی» یای فارسی اولویت داره.

نظرات و پیشنهادات دیگر شما و باقی دوستان را با اشتیاق دنبال میکنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۵/۱۶ ۲۳:۱۱:۱۷

سلام،
نمی‌دونم اون نرم افزاری که این مشکل را داشته از چه بانک اطلاعاتی استفاده می‌کرده، چون من الان بر اساس گفته‌های شما با SQL Server تست کردم و مشکلی در مرتب سازی نبود:
نتیجه‌ی بررسی مرتب سازی
بنابراین بهتر است همه چیز تبدیل و همه چیز یک دست شود تا مشکلات ذکر شده را نداشته باشیم.

نویسنده: Meysam
تاریخ: ۱۳۸۹/۰۵/۱۸ ۱۲:۴۴:۴۵

ایده ی جالبیه.

نویسنده: mohammad
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۱:۵۸:۳۵

من یک سال پیش یک اکستنشن دقیقا مثل این روش نوشتم و استفاده می کنم.
البته من تو bussinesobject این کارو می کنم.
با آپدیت دیتا بیس قبلی ها هم آپدیت می شن. دوباره کاری هم نمی شه