

در این قسمت قصد داریم از امکانات جدید اعتبار سنجی تعریف شده در فضای نام استاندارد System.ComponentModel.DataAnnotations استفاده نمائیم. از سیلورلایت سه به بعد امکان استفاده از این فضای نام به سادگی در برنامه‌های سیلورلایت میسر است (همچنین در برنامه‌های ASP.Net MVC)؛ اما برای کار با آن در WPF نیاز به تعدادی متد کمکی می‌باشد...

فهرست مطالب:

فصل 5- تعیین اعتبار ورودی کاربر و الگوی MVVM

مقدمه

معرفی برنامه فصل

مدل برنامه‌ی فصل

ViewModel برنامه فصل

View برنامه فصل

[دریافت قسمت پنجم](#)

[دریافت مثال قسمت پنجم](#)

تعدادی از منابع و مآخذ مورد استفاده در این سری:

1.

[Model-View-ViewModel \(MVVM\) Explained](#)

2.

[Model View ViewModel](#)

3.

[DataModel-View-ViewModel pattern](#)

4.

[Minute Overview of MVVM in Silverlight 5](#)

5.

[A Field Guide to WPF Presentation Patterns](#)

6.

[An attempt at simple MVVM with WPF](#)

7.

[WPF: If Heineken did MVVM Frameworks Part 1 of n](#)

8.

[Modal dialogs with MVVM and Silverlight 4](#)

9.

[How do I do... With the Model-View-ViewModel pattern](#)

10.

[Intro to WPF MVVM](#)

11.

[Introduction to MVVM pattern in WPF](#)

12.

[Learning WPF M-V-VM](#)

13.

[Binding Combo Boxes in WPF with MVVM](#)

14.

[Model-View-ViewModel Pattern](#)

15.

[Unit Testable WCF Web Services in MVVM and Silverlight 4](#)

16.

[MVVM Part 1: Overview](#)

17.

[?Which came first, the View or the Model](#)

18.

[Stackoverflow's questions tagged with MVVM](#)

19.

[WPF: MVVM \(Model View View-Model\) Simplified](#)

20.

[WPF and MVVM tutorial 01, Introduction](#)

21.

[?...WPF patterns : MVC, MVP or MVVM or](#)

22.

[Silverlight, MVVM and Validation Part III](#)

23.

['DotNetKicks.com - Stories recently tagged with 'MVVM](#)

24.

[DotNetShoutout - Stories tagged with MVVM](#)

25.

[MVVM Light Toolkit](#)

26.

[MVVM screen casts](#)

27.

[What's new in MVVM Light V3](#)

28.

[Using RelayCommand in Silverlight 3 and WPF](#)

29.

[WPF Apps With The Model-View-ViewModel Design Pattern](#)

30.

[WPF MVVM and Showing Dialogs](#)

نظرات خوانندگان

نویسنده: sAeid

تاریخ: ۱۴:۳۳:۴۳ ۱۳۸۹/۰۲/۰۴

با سلام

ممنون از مطالب جالب تون در این که بطور کلی در سایت هست و مخصوص در باره MVVM می خواستم تشکر کنم که فایل ها رو بصورت PDF قرار دادید و خواهش کنم اگر امکان دارد مطالب مربوط به NHibernate رو هم لطف کنید و بصورت PDF قرار بدهید چون من مشکل پرینت و مطالعه این مطالب رو داشتم می دونم کار وقت گیری هست اما زکات علم نشر آن است یک دنیا تشکر سعید محمدهاشم

نویسنده: وحید نصیری

تاریخ: ۱۴:۴۱:۲۷ ۱۳۸۹/۰۲/۰۴

سلام

ممنون. برای پرینت می‌تونید از فایل chm قرار داده شده (خلاصه وبلاگ، منوی سمت راست بالای سایت) هم استفاده کنید.

نویسنده: C# Builder

تاریخ: ۲۰:۱۱:۳۳ ۱۳۸۹/۰۴/۱۴

بسیار بسیار مفید بود با تشکر فراوان

نویسنده: داود رضانی

تاریخ: ۱۲:۰۳:۱۷ ۱۳۸۹/۰۹/۱۸

لطفا این آموزش رو ادامه بدید

نویسنده: رضا

تاریخ: ۱۱:۴۵ ۱۳۹۱/۰۷/۰۸

آقای نصیری من با توجه به مطالب این فصل یک سوال داشتم: وقتی ما از Entity Framework Code First استفاده میکنیم، و تمام کلاس‌های مربوطه را داخل یک پروژه دیگر به اسم DAL قرار میدهیم، روش درست این است که به ازای هر موجودیت یک کلاس دیگر ساخته و INotifyPropertyChanged و ... را در آن پیاده سازی کنیم؟ به عنوان مثال وقتی کلاس Code First ما Customer می‌باشد ما یک کلاس دیگر مثلاً به اسم CustomerModel بسازیم (برای پیاده سازی INotifyPropertyChanged و IDataErrorInfo و ...) و بعد هنگام انجام عملیات (ثبت و ویرایش و حذف) این دو کلاس رو به هم Map کنیم؟ ممنون.

نویسنده: وحید نصیری

تاریخ: ۱۱:۵۶ ۱۳۹۱/۰۷/۰۸

دو نوع کلاس اینجا وجود دارند:

domain models : کلاس‌های معادل جداول و موجودیت‌های بانک اطلاعاتی

viewmodels : مقصود از این viewmodel ها، کلاس مدل معادل عناصر بصری UI است و منظور viewmodel تعریف شده در MVVM نیست که دقیقاً معادل Controller در MVC است.

بنابراین اگر domain model شما با مدل معادل view یکی است، همه رو یکجا هم می‌تونید تعریف کنید ولی عموماً این‌ها یکی

نیستند. بنابراین نیاز است بین این دو فرق گذاشت و در صورت نیاز نگاشت لازم را انجام داد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۹:۴۲

[لینک کمکی](#) دریافت این سری