بنام خدا

Relay Command چیست

مترجم و نویسنده : محمد فیروزی
mfiroozi66@yahoo.com

لطفا پیشنهادات و سوالات خود را در سایت www.papro-co.ir مطرح کنید

Relay Command چیست

وقتی که وارد مبحث WPF می شویم به جذابیت های آن پی خواهیم برد،به نظر من چیزی که از همه بیشتر WPF را جذاب می سازد، الگوی M-V-VM است.با فرض اینکه شما خواننده ی عزیز از WPF می دانید و با الگوی طراحی M-V-VM آشنا هستید، قصد دارم به مبحث مهمی در این الگو یعنی ، RelayCommand بپردازم.

RelayCommand چیست و چگونه از آن باید استفاده کرد؟

اگر با الگوی m-v-vm آشنایی داشته باشید می دانید که هر View،کد پس زمینه آن خالی است ،به جز کلاس سازنده InitializeComponent که در آن وجود داردو(در واقع شما نیازی به نوشتن کدی در view ندارید) که حتی می توانید فایل مربوط به کد پس زمینه آن را نیز از پروژه پاک کنید. و با این وجود خواهید دید که پروژه شما به درستی اجرا خواهد شد.

با وجود عدم تعریف متد Event Handling ای در View ،هنگامی که کاربر بر روی دکمه ای کلیک کند،برنامه واکنش نشان داده و درخواست کاربر ر اجرا خواهد کرد.

واقعا چرا و چگونه؟

زیرا عمل binding هایی که ما بر روی Command های لینک ، دکمه و آیتم های منو بوجود آورده ایم در UI نشان داده می شوند.این binding ها مطمئن می شوند هنگامی که کاربر بر روی کنترل کلیک کرد شی ICommand تحت تاثیر اجرای ViewModel قرار گرفته باشند.شما می تونید فکر کنید که command یک object رابط است که کار شما را برای مصرف یکی از قابلیت های ViewModel از یک View تعریف شده در XAML ساده می کند.

وقتی که یک ViewModel یک نمونه از Property ار نوع ICommand را تحت تاثیر قرار می دهد ، Command Object به طور معمول آن شی ViewModel که وظیفه اش انجام می دهد استفاده می کند.

یک الگوی پیاده سازی ممکن این است که یک کلاس تودرتو Private در کلاس ViewModel بوجود آوریم.بنابراین آن Command می توانند به اعضای Private ای که حاوی ViewModel هستند و NameSpace را آلوده نکرده اند،دسترسی پیدا کنند.

پیاده سازی کلاس تودرتو ICommand interface و یک رفرنس به شی حاوی ViewModel به سازنده آن تزریق می شود. به هرحال ،بوجود آوردن بک کلاس تودرتو که پیاده سازی ICommand برای هر Command تحت تاثیر ViewModel است باعث افزایش حجم کلاس ViewModel می شود.که مدهای زیاد ممکن است مشکل بزرگی را بوجود آورد.

.در نمونه کد زیر کلاس RelayCommand این مشکل را حل می کند.RelayCommand به شما اجازه خواهد داد که Command های منطقی را بوسیله نمایندگان گذشته به سازنده آن تزریق کند

این روش اجازه می دهد که به طور مختصر و مفید ، Command مختصری را در کلاس ViewModel اجرا کند.

RelayCommand ساده شده ی DelegateCommand است .

```
public class RelayCommand : ICommand
{
 #region Fields
 readonly Action<object> _execute;
readonly Predicate<object> _canExecute;
 #endregion // Fields

 #region Constructors
public RelayCommand(Action<object> execute) : this(execute, null) { }
 public RelayCommand(Action<object> execute, Predicate<object> canExecute)
 {
if (execute == null)
 throw new ArgumentNullException("execute");
_execute = execute;
 _canExecute = canExecute;
 }
#endregion // Constructors
 #region ICommand Members
 [DebuggerStepThrough]
public bool CanExecute(object parameter)
{
return _canExecute == null ? true : _canExecute(parameter);
}
 public event EventHandler CanExecuteChanged
 {
 add
 { CommandManager.RequerySuggested += value; }
 Remove
 { CommandManager.RequerySuggested -= value; }

 }
```

```
public void Execute(object parameter)
{
_execute(parameter);
}
#endregion // ICommand Members
}
```

رویداد CanExecuteChanged که بخشی از پیاده سازی ICommand interface می باشد .