عنوان: **آموزش WAF (بررسی Commandها)**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۳/۰۱/۱۶ ۲۲:۵۰
آدرس: www.dotnettips.info
گروه‌ها: MVVM, WPF, WAF

در این پست قصد داریم مثال قسمت قبل را توسعه داده و پیاده سازی Commandها را در آن در طی یک مثال بررسی کنیم. از این جهت دکمه‌ای، جهت حذف آیتم انتخاب شده در دیتا گرید، به فرم BookShell اضافه می‌نماییم. به صورت زیر:

```
<Button Content="RemoveItem" Command="{Binding RemoveItemCommand}" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="75"/>
```

Command تعریف شده در Button مورد نظر به خاصیتی به نام RemoveItemCommand در BookViewModel که نوع آن ICommand است اشاره می‌کند. پس باید تغییرات زیر را در ViewModel اعمال کنیم:

```
public ICommand RemoveItemCommand { get; set; }
```

از طرفی نیاز به خاصیتی داریم که به آیتم جاری در دیتاگرید اشاره کند.

```
public Book CurrentItem
        {
            get
            {
                return currentItem;
            }
            set
            {
                if(currentItem != value)
                {
                    currentItem = value;
                    RaisePropertyChanged("CurrentItem");
                }
            }
        }
        private Book currentItem;
```

همان طور که در پست قبلی توضیح داده شد پیاده سازی‌ها تعاریف ViewModel در Controller انجام می‌گیرد برای همین منظور باید تعریف DelegateCommand که یک پیاده سازی خاص از ICommand است در کنترلر انجام شود. :

```
[Export]
    public class BookController
    {
        [ImportingConstructor]
        public BookController(BookViewModel viewModel)
        {
            ViewModelCore = viewModel;
        }

        public BookViewModel ViewModelCore
        {
            get;
            private set;
        }

        public DelegateCommand RemoveItemCommand
        {
            get;
            private set;
        }

        private void ExecuteRemoveItemCommand()
        {
            ViewModelCore.Books.Remove(ViewModelCore.CurrentItem);
        }

        private void Initialize()
        {
            RemoveItemCommand = new DelegateCommand(ExecuteRemoveItemCommand);
            ViewModelCore.RemoveItemCommand = RemoveItemCommand;
```

```
        }

        public void Run()
        {

            var result = new List<Book>();
            result.Add(new Book { Code = 1, Title = "Book1" });
            result.Add(new Book { Code = 2, Title = "Book2" });
            result.Add(new Book { Code = 3, Title = "Book3" });

            Initialize();

            ViewModelCore.Books = new ObservableCollection<Models.Book>(result);

            (ViewModelCore.View as IBookView).Show();
        }
    }
```

تغییرات:

»خاصیتی به نام RemoveItemCommand که از نوع DelegateCommand است تعریف شده است؛

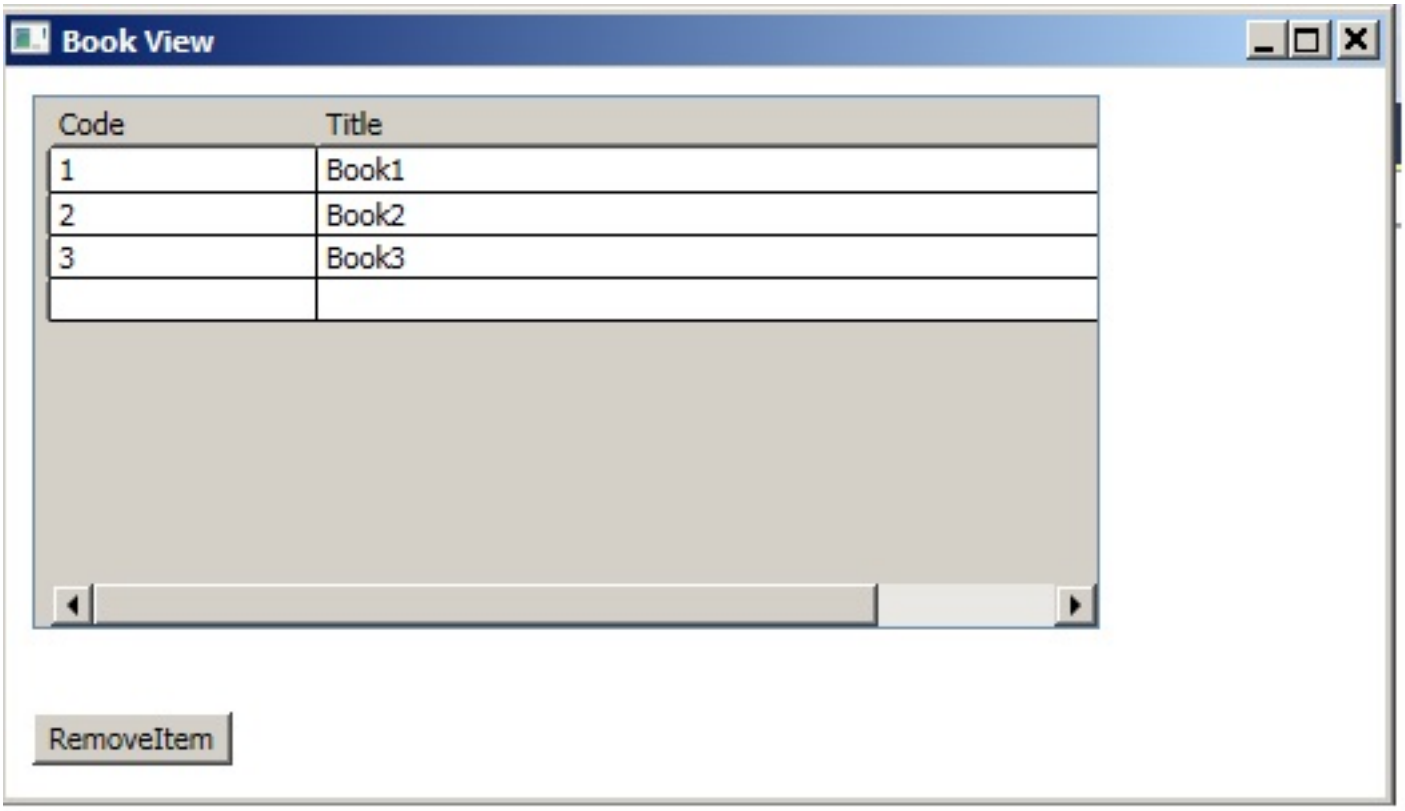
»متدی به نام Initialize اضافه شد که متدهای Execute و CanExecute برای Commandها را در این قسمت رجیستر می‌کنیم.

»در نهایت Command تعریف شده در کنترلر به Command مربوطه در ViewModel انتساب داده شد.

حال کافیست خاصیت SelectedItem دیتاگرید BookShell به خاصیت CurrentItem موجود در ViewModel مقید شود:

```
<DataGrid ItemsSource="{Binding Books}" SelectedItem="{Binding CurrentItem
,Mode=TwoWay,UpdateSourceTrigger=PropertyChanged}" HorizontalAlignment="Left" Margin="10,10,0,0"
VerticalAlignment="Top" Width="400" Height="200">
            <DataGrid.Columns>
                <DataGridTextColumn Header="Code" Binding="{Binding Code}"
Width="100"></DataGridTextColumn>
                <DataGridTextColumn Header="Title" Binding="{Binding Title}"
Width="300"></DataGridTextColumn>
            </DataGrid.Columns>
        </DataGrid>
```

اگر پروژه را اجرا نمایید، بعد از انتخاب سطر مورد نظر و کلیک بر روی دکمه RemoveItem مورد زیر قابل مشاهده است:




```
1 reference
private void ExecuteRemoveItemCommand()
{
    ViewModelCore.Books.Remove(ViewModelCore.CurrentItem);
}
```