

عنوان: پیاده سازی رخدادهای یک شیء در الگوی MVP

نویسنده: امیر هاشم زاده

تاریخ: ۱۴۰۳/۰۳/۱۴

آدرس: www.dotnettips.info

برچسب‌ها: ASP.Net, Design patterns, MVP, Telerik, Grid, RadGrid

یکی از ضروریات دنیای برنامه نویسی امروز، داشتن یک الگوی مناسب می‌باشد. یکی از الگوهای مناسب برای وب فرم‌ها، استفاده از الگوی MVP است.

اگر در خلال پیاده سازی، گاهی اوقات نیاز به handle کردن رخدادها را داشته باشید بدین منظور به روش زیر عمل می‌کنیم: (توجه: شیء مورد نظر ما در این پست RadGrid از کنترل‌های Telerik در نظر گرفته شده است).

```
// ASPX page
<telerik:RadGrid ID="RadGrid1" runat="server"></telerik:RadGrid>
// Asp.Net Code Behind
protected void Page_Load(object sender, EventArgs e)
{
    GridPresenter presenter = new GridPresenter(this);
}
// view interface
public interface IGridView
{
    Telerik.Web.UI.RadGrid myGrid { get; }
}
// presenter
protected readonly IGridView _view;
public GridPresenter(IGridView view)
{
    _view = view;
    _view.myGrid.UpdateCommand += new Telerik.Web.UI.GridCommandEventHandler(onUpdateCommand);
    _view.myGrid.InsertCommand += new Telerik.Web.UI.GridCommandEventHandler(onInsertCommand);
    _view.myGrid.EditCommand += new Telerik.Web.UI.GridCommandEventHandler(onEditCommand);
}
private void onUpdateCommand(object sender, Telerik.Web.UI.GridCommandEventArgs e)
{
    // Code for updating
}
private void onInsertCommand(object sender, Telerik.Web.UI.GridCommandEventArgs e)
{
    // Code for inserting
}
private void onEditCommand(object sender, Telerik.Web.UI.GridCommandEventArgs e)
{
    // Code for editcommand
}
```

نظرات خوانندگان

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۳/۲۹ ۰:۴۹

خوبه
اما من به قسمت رو متوجه نشدم. ببینید ما می‌گیم (MVP) (Model View Presenter).
خب حالا سوال اینجاست که در بخش Asp.net Code Behind از چه کدی استفاده می‌کنیم؟
یعنی کدهای View و Presenter چه موقع فراخوانی میشوند؟
از کجا اجرا میشود و به کجا ختم میشود؟
مدل ما کجاست؟
و ...

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۳/۲۹ ۱۱:۱

ابتدا وب فرم مورد نظر اینترفیس IGridView را پیاده سازی می‌کند و سپس در Code behind در Page_Load یک وهله از presenter با پارامتر this ایجاد می‌کنید، در نتیجه کلیه رخدادها در presenter مدیریت می‌شوند.
نکته در این است که چون هر بارگذاری صفحه نیاز به متصل کردن رخدادها به شیء هستیم مجبوریم در Page_Load یک وهله از presenter ایجاد کنیم.
در این پست هدف پیاده سازی مدل نبوده است، ولی شما می‌توانید مدل خود را با توجه به الگوی MVP طراحی کنید و از آن در presenter استفاده کنید.

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۳/۲۹ ۱۴:۲۶

مرسی
مشکل قطعه کد زیر بود که باعث شده بود که نحوه استفاده رو متوجه نشم!

```
GridPresenter presenter = new GridPresenter(this);
```

در مورد Model هم شما درست می‌گید. هدف این پست استفاده از Model نبود.

نویسنده: جلال
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۲:۴۱

سلام،
با این قضیه ارجاع مستقیم به یک کنترل خاص در IView می‌تونم کنار بیام، ولی با ارجاع به اون در CPresenter نه!
دلیل خاصی داشته؟ آخه معمولاً برای رخدادگردانی، EventHandler‌ها رو در IView معرفی می‌کنن و در CPresenter استفاده می‌کنند.

البته در [این مقاله](#)، کلاً منکر استفاده از EventHandler‌ها در IView میشه و ترجیح میده فقط تابع Presenter رو از Code Behind فراخوانی کنه!

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۸/۱۶ ۱۸:۴۸

یکی از دلایل استفاده از این روش، تمیز نگه داشتن ویو (با توسعه برنامه پارامترهای لازم برای 4 عمل اصلی اضافه می‌شد

که نگهداری پارامترهای رخداد گردانها پیچیده و دشوار می شد) بود، زمانی که داشتم این پیاده سازی رو در پروژه انجام می دادم به این نتیجه رسیده بودم بهتر است کلیه عملیات های افزودن، حذف، ویرایش و حذف شیء تلریک گرید و یو رو در presenter انجام بدم چون قرار بود یک سری عملیات منطقی بر روی 4 عمل اصلی انجام بدم.