

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

Computed Observable ها در Knockout.js

مدرس : مهندس افشین رفوآ

مقدمه

Computed Observable ها در Knockout.js، توابعی هستند که به یک یا چند Observable دیگر وابستگی دارند و وقتی observable های پایه شان تغییر کنند، به صورت خودکار بروزرسانی می شوند. یک Computed Observable از یک یا چند Observable دیگر مشتق شده است، بنابراین Computed Observable ها فقط خواندنی هستند. یک computed observable قابل نوشتن (writable computed observable) دقیقاً مثل یک observable معمولی، با قوانین دلخواه جهت خواندن و نوشتن، استفاده می شود.

Pure Computed Observables

اگر یکسری محاسبات ساده در Computed Observable انجام شود و یک مقداری را برگرداند و همزمان یک عملیات بروزرسانی انجام دهد، می تواند به عنوان Pure Computed observable در نظر گرفته شود. از Pure Computed observables ها در Knockout برای مدیریت مصرف حافظه به صورت موثر و قابل ارزیابی مجدد، استفاده می شود.

برخی از ویژگی های Computed Observable ها عبارتند از:

ko.isComputed

ko.isObservable

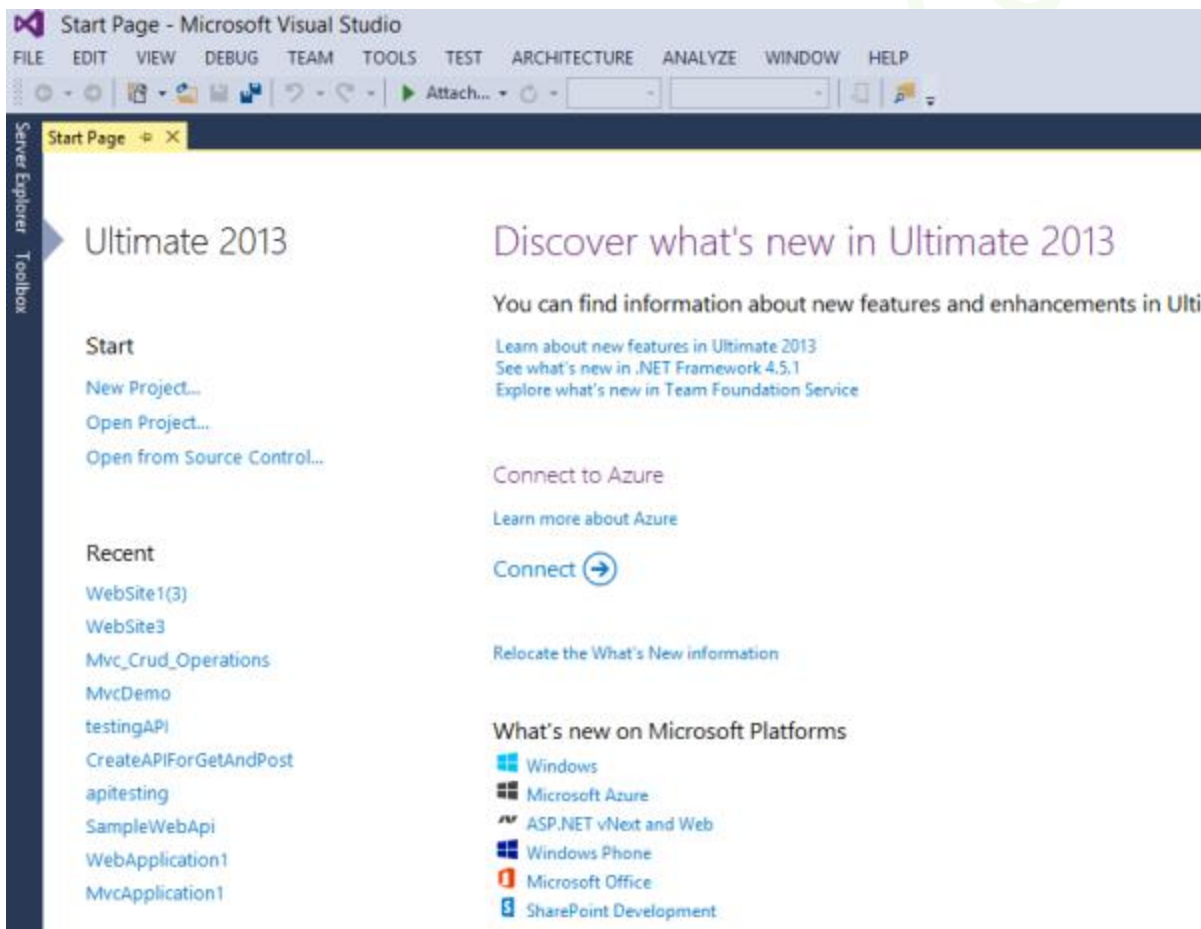
ko.isWritableObservable

Observable قابل نوشتن (Writable Observable)

همانطور که می دانیم **Computed Observable** ها فقط خواندنی هستند. اما می توان آنها را قابل نوشتن هم کرد. برای این منظور، باید یک **callback function** فراهم کنیم که عملیات نوشتن در مقادیر را انجام دهد.

مرحله 1

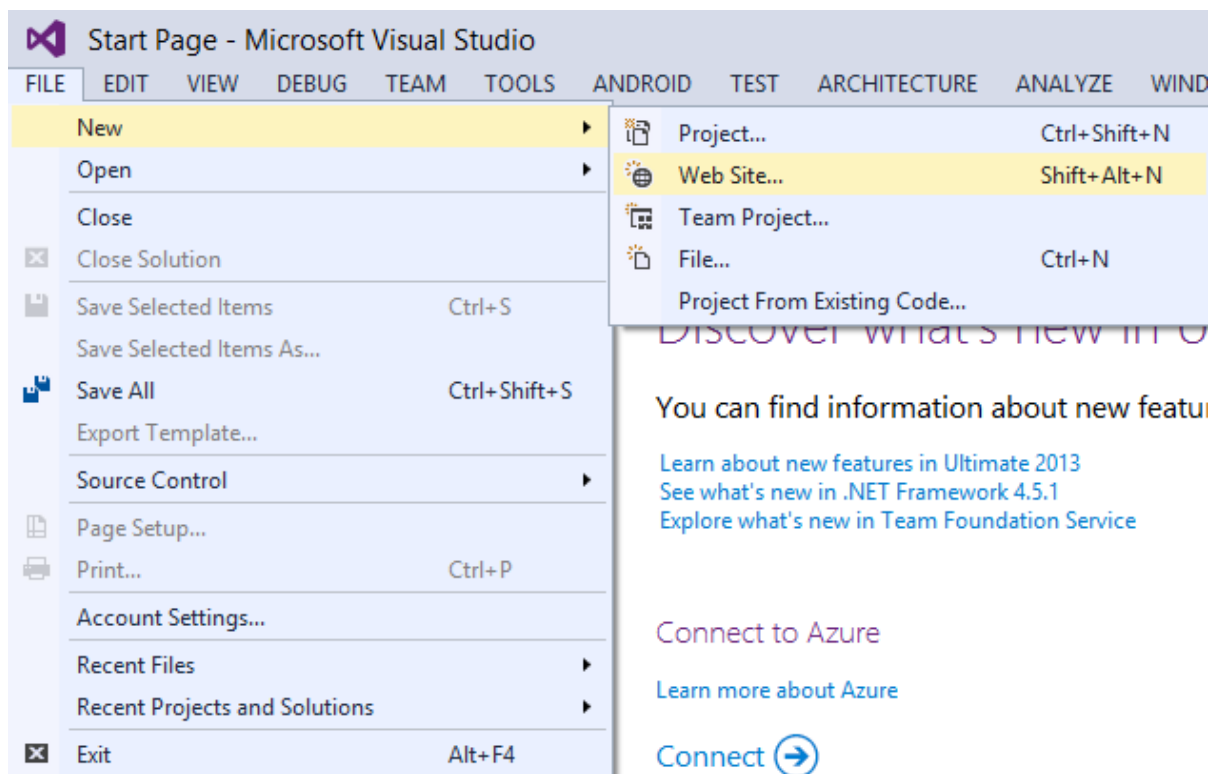
Visual Studio را باز کنید.



مرحله 2

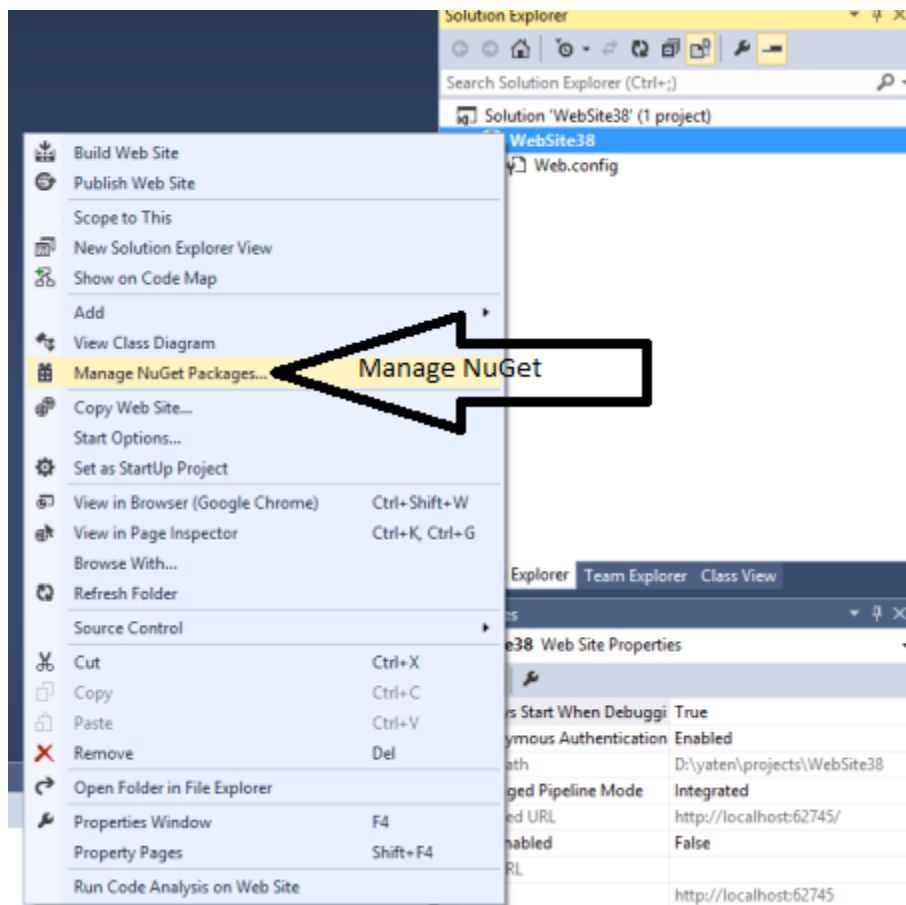
باید یک **Web Site** ایجاد کنیم. بر روی گزینه **File>New>Web Site** کلیک کنید.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7
88146323 - 88446780 - 88146330



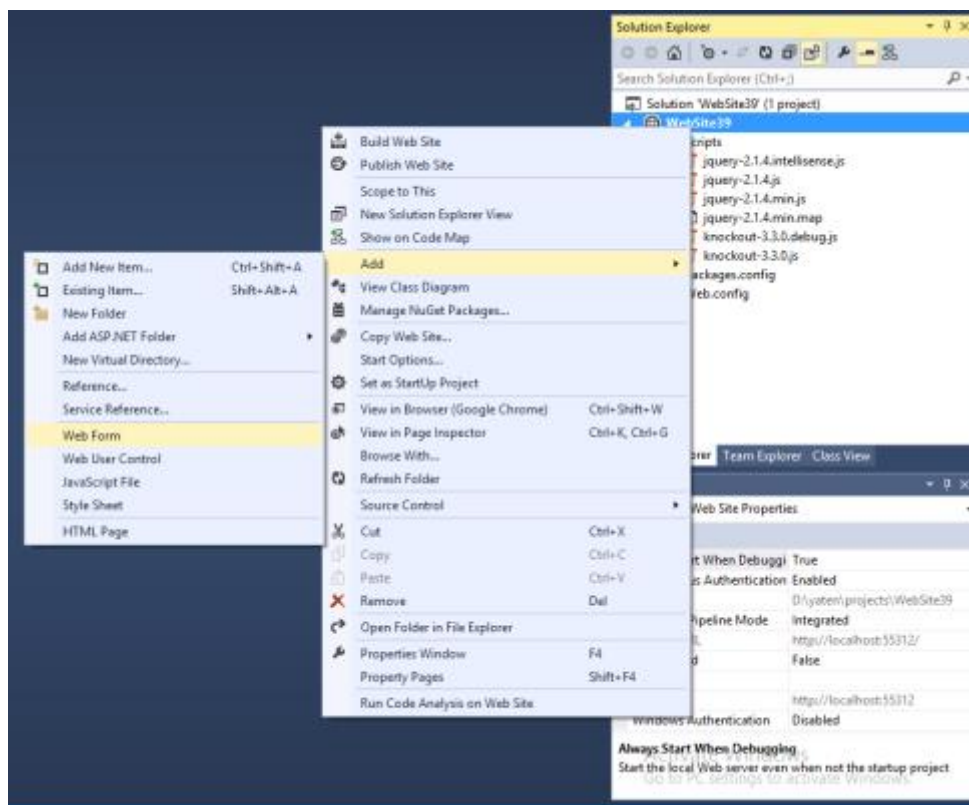
مرحله 3

در این مرحله باید کتابخانه **Knockout.js** را به **website** اضافه کنیم. به همین منظور بر روی **website** راست کلیک کرده و **NuGet Package** را انتخاب می کنیم.



مرحله 4

پس از اضافه کردن کتابخانه **Knockout.js** بر روی **web site** راست کلیک کرده و یک **web form** به **website** اضافه می کنیم.



مرحله 5

حالا باید کد **Computed Observable** ها را بنویسیم.

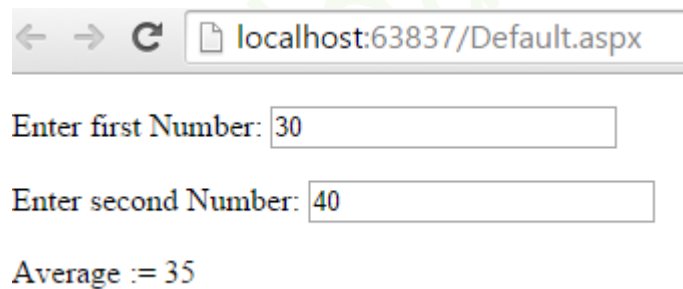
در این کد از دو **observable** و یک **Computed Observable** استفاده می کنیم تا نشان بدهیم چطور **Computed Observable** ها وابسته به یک یا چند **Observable** کار می کنند و مقادیر 30 و 40 را به پارامترها ارسال می کنیم.

```
function MyViewModel()
{
  this.fir = ko.observable(30);
  this.sec = ko.observable(20);
  this.Multi = ko.computed(function () {
    if (typeof (this.fir()) !== "number" || typeof (this.sec()) !== "number") {
      this.fir(Number(this.fir()));
      this.sec(Number(this.sec()));
    }
    total = (this.fir() * this.sec());
    return total;
  }, this);
}
```

در این تصویر سعی داریم وابستگی **Computed Observable** ها به **Observable** های معمولی را نشان دهیم.

مرحله 6

پس از نوشتن کد، **website** را با کلید **F5** اجرا می کنیم. پنجره زیر نمایش داده می شود:



مرحله 7

اگر تغییراتی را در **Observable** اول ایجاد کنیم، **Computed Observable** یا همان مقدار میانگین، به صورت خودکار تغییر می کند. به این ترتیب نشان داده می شود که **Computed Observable** وابسته به **observable** ها است.

localhost:63837/Default.aspx

Enter first Number: 67

Enter second Number: 40

Average := 53.5

Changes in the Observable(First Number, Automatic changes in the Average Value