

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

درک Observable در Knockout.js

مدرس : مهندس افشین رفوآ

درک Observable در Knockout.js

مقدمه

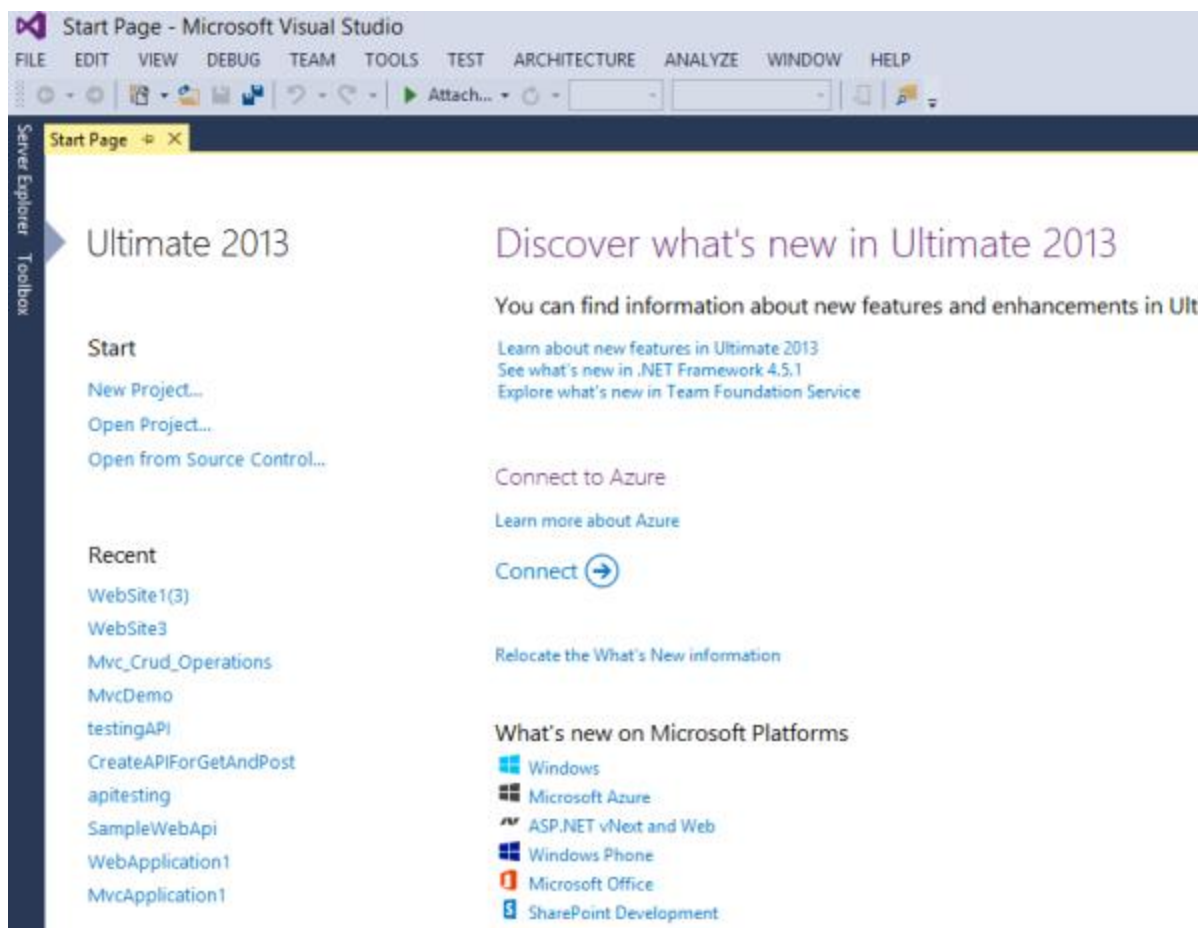
قبل از شروع این مقاله باید ابتدا مقاله قبلی با عنوان " تنظیم Visual Studio 2013 برای توسعه Knockout.js" را مطالعه کرده باشید.

این مقاله چگونگی استفاده از آرایه ها در Knockout.js را توضیح خواهیم داد. هنگامی که می خواهیم با استفاده از observableها، یک object را شناسایی کرده و به تغییرات آن پاسخ دهیم، Knockout.js نقش مهمی را بازی میکند.

Observable در خیلی از جاها استفاده می شود از جمله، زمانی که چندین مقدار را نمایش می دهیم یا ویرایش می کنیم و نیاز است با حذف و اضافه کردن هر آیتم، بخش های تکراری از UI ظاهر شوند. مزیت اصلی KO این است که زمانی view model تغییر می کند، به صورت خودکار بروزرسانی می شود. برای اعمال کردن این ویژگی های KO باید ویژگی های مدلمان را به صورت observable تعریف کنیم. چون این JavaScript است که تغییرات را به اطلاع subscriberها (مشترکین) می رساند و به صورت خودکار وابستگی ها را حذف می کند. KO مهمترین ویژگی برای تازه سازی (Refresh) خودکار UI است. این کار زمانی انجام می شود که تغییرات در ViewModel، به View اعلام شود.

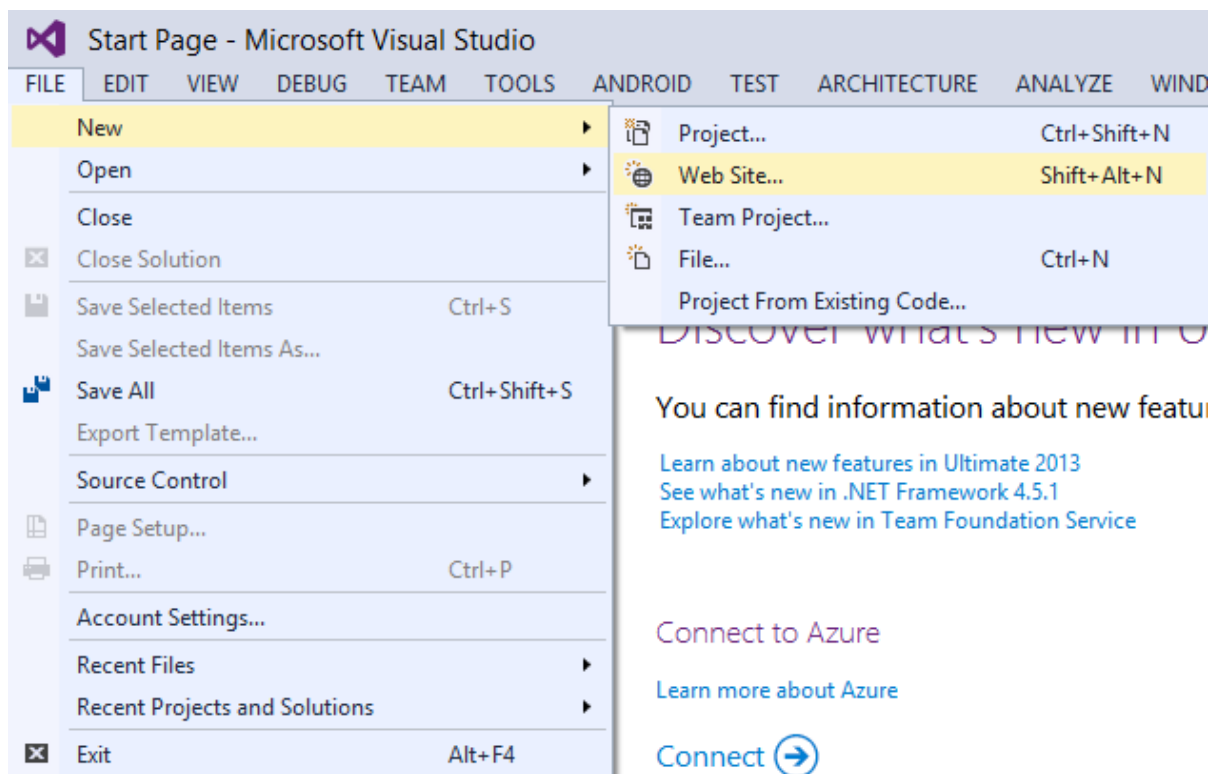
مرحله 1

Visual Studio را باز می کنیم:



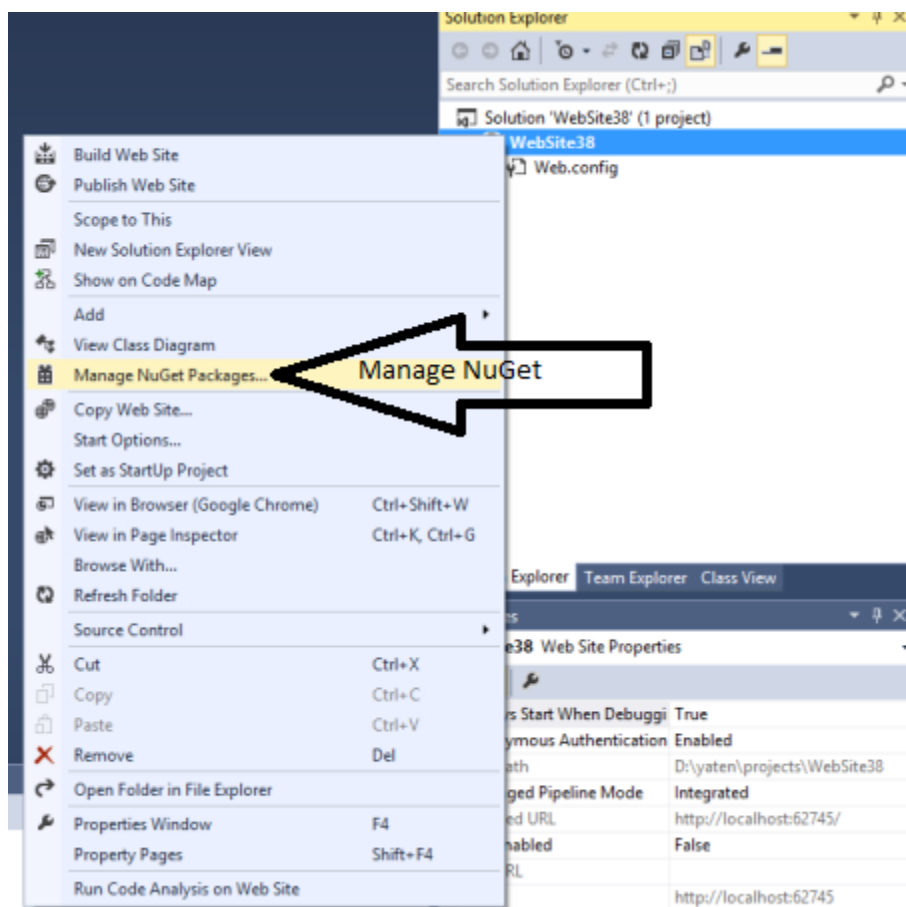
مرحله 2

حالا برای ایجاد یک وب سایت به گزینه **File>New>Website** می رویم.



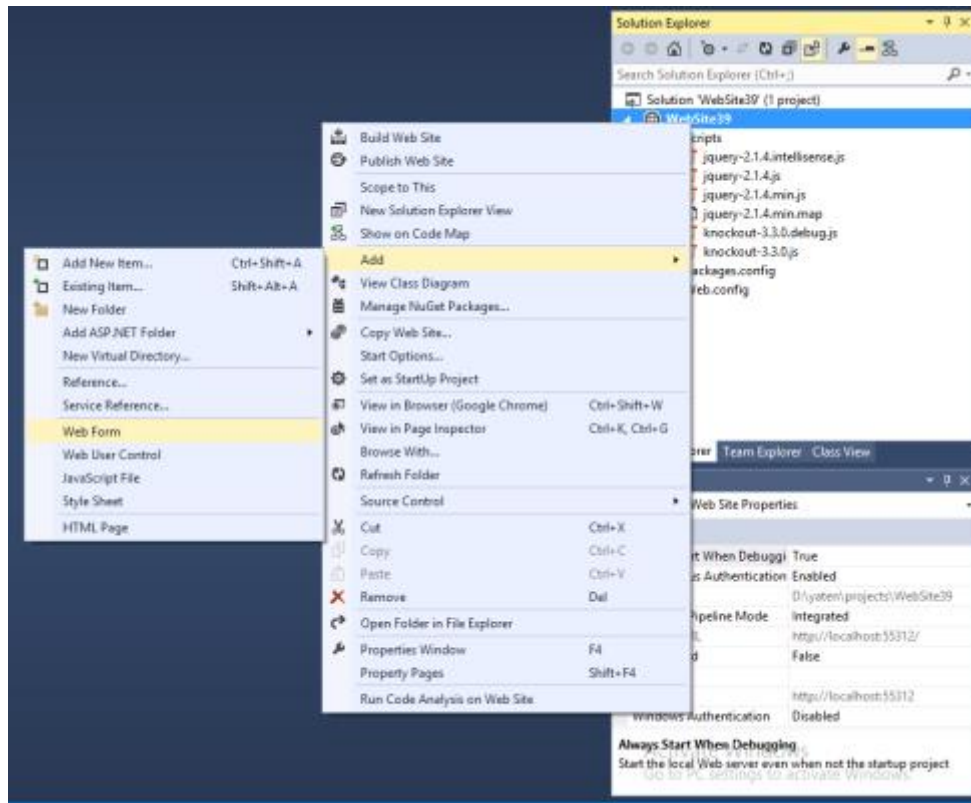
مرحله 3

حالا باید کتابخانه **knockout.js** را به وب سایتمان اضافه کنیم. برای این کار بر روی وب سایت راست کلیک کرده و گزینه **NuGet Packages** را انتخاب کنید.

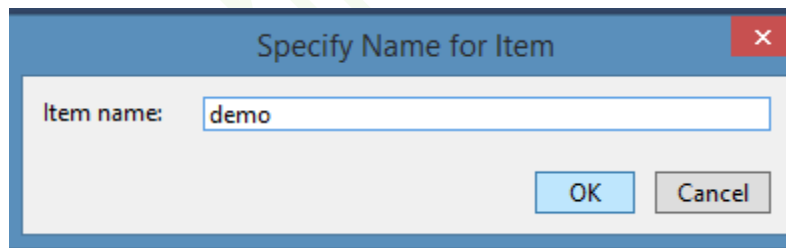


مرحله 4

در این مرحله نیز همانند مقاله قبل، باید کتابخانه **Knockout.js** و **jQuery** را جستجو کرده و به وب سایتمان اضافه کنیم. پس از اضافه کردن کتابخانه **Knockout.js**، باید یک **Web Form** به وب سایت اضافه کنیم. این کار را با راست کلیک بر روی وب سایت و انتخاب **Add>WebForm** انجام می دهیم.



سپس یک نام برای آن انتخاب میکنیم.



مرحله 5

کد آرایه **observable** به صورت زیر خواهد بود:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Understanding Observables in Knockout.js</title>
</head>
<body>
  <form id="form1" runat="server">
```

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

```

<div>
  <p>First Name :<span data-bind="text: firstName" /></p>
  <p>Middle Name :<span data-bind="text: MiddleName" /></p>
  <p>Last Name:<span data-bind="text: LastName"></span></p>
</div>
<hr />
<div>
  <p>First Name<asp:TextBox ID="txname" runat="server" data-bind="value:firstName"></asp:TextBox></p>
  <p>Middle Name<asp:TextBox ID="txtlstnme" runat="server" data-
bind="value:MiddleName"></asp:TextBox></p>
  <p>Last Name<asp:TextBox ID="txtlastname" runat="server" data-
bind="value:LastName"></asp:TextBox></p>
</div>
<hr />
<p>Full Name :<span data-bind="text: fullName"> </span></p>
</form>
<script src="Scripts/knockout-3.3.0.js"></script>
<script type="text/javascript">
var vm =
{
  firstName: ko.observable("yatendra"),
  MiddleName: ko.observable("Kumar"),
  LastName: ko.observable("Sharma")
};
vm.fullName = ko.dependentObservable(function () {
  return vm.firstName() + " " + vm.MiddleName() + " " + vm.LastName();
});
ko.applyBindings(vm);
</script>
</body>
</html>

```

در این کد با استفاده از **observable**ها تغییرات یک شی را شناسایی می کنیم و به آن پاسخ می دهیم. بنابراین **observable**ها زمانی مفید هستند که بخواهیم چند مقدار تغییر کنند و نیاز است بخش های تکراری ای از ال، با حذف و اضافه کردن آیتم ها ظاهر و مخفی شوند.

مرحله 6

پس از نوشتن کد باید وب سایت را با استفاده از کلید **F5** اجرا کنیم. پس از اجرا صفحه زیر نمایش داده می شود:

localhost:55312/Default.aspx

First Name :yatendra

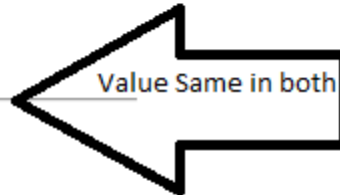
Middle Name :Kumar

Last Name:Sharma

First Name yatendra

Middle Name Kumar

Last Name Sharma



مرحله 7

اگر نام را در **textbox** تغییر دهیم، نام در بخش بالایی پنجره هم به صورت خودکار تغییر می کند.

localhost:55312/Default.aspx

First Name :yatendrasharma

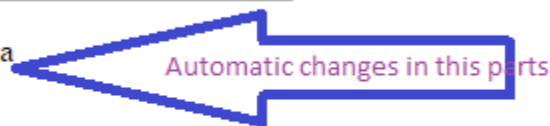
Middle Name :Kumar

Last Name:Sharma

First Name yatendrasharma

Middle Name Kumar

Last Name Sharma



بنابراین، **observable** به خوبی کار می کند. زمانی که یک بخش را تغییر می دهیم، تغییرات مشابهاً در بخش

متصل دیگر هم اتفاق می افتد.

در این مقاله نحوه کارکرد **observable** ها را در **Knockout.js** و نیز چگونگی استفاده از **Observable** در **Knockout.js** در **Visual Studio 2013** را توضیح دادیم.

www.tahlildadeh.com