بسم اللّه الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

استفاده کردن از Web API در Web Form های Asp.Net

مدرس : مهندس افشین رفوآ

دوره آموزشی Web API

اگرچه Asp.net Web API برای Asp.Net MVC بیشتر مورد استفاده قرار می گیرد، می توان آن را به آسانی به Web Form های Asp.Net سنتی اضافه کرد. در ادامه این مقاله قصد دارد این مطلب را توضیح دهد.

چکیده

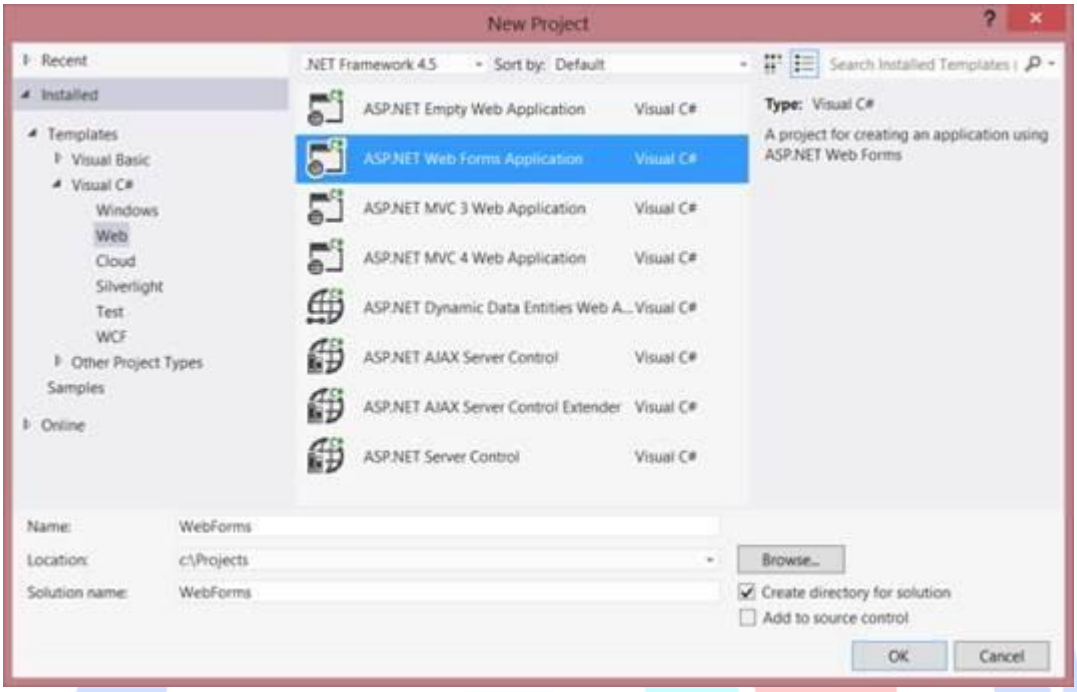
برای استفاده از Web API در Web form ها دو روش اصلی وجود دارد:

- اضافه کردن کنترولر Web API که از کلاس ApiController مشتق شده است.
- اضافه کردن جدول مسیریابی به متد Application_start

ساخت یک پروژه Web Form

Visual studio را باز کنید و new project را در صفحه start انتخاب کنید و یا در منو file، new و سپس project را انتخاب کنید.

در قسمت Templates گزینه installed templates راکلیک کنید تا visual c# را باز کنید. در زیر visual c#، web را انتخاب کنید.در صفحه، Asp.net web forms application را انتخاب کنید و نام را وارد کرده و ok را انتخاب کنید.

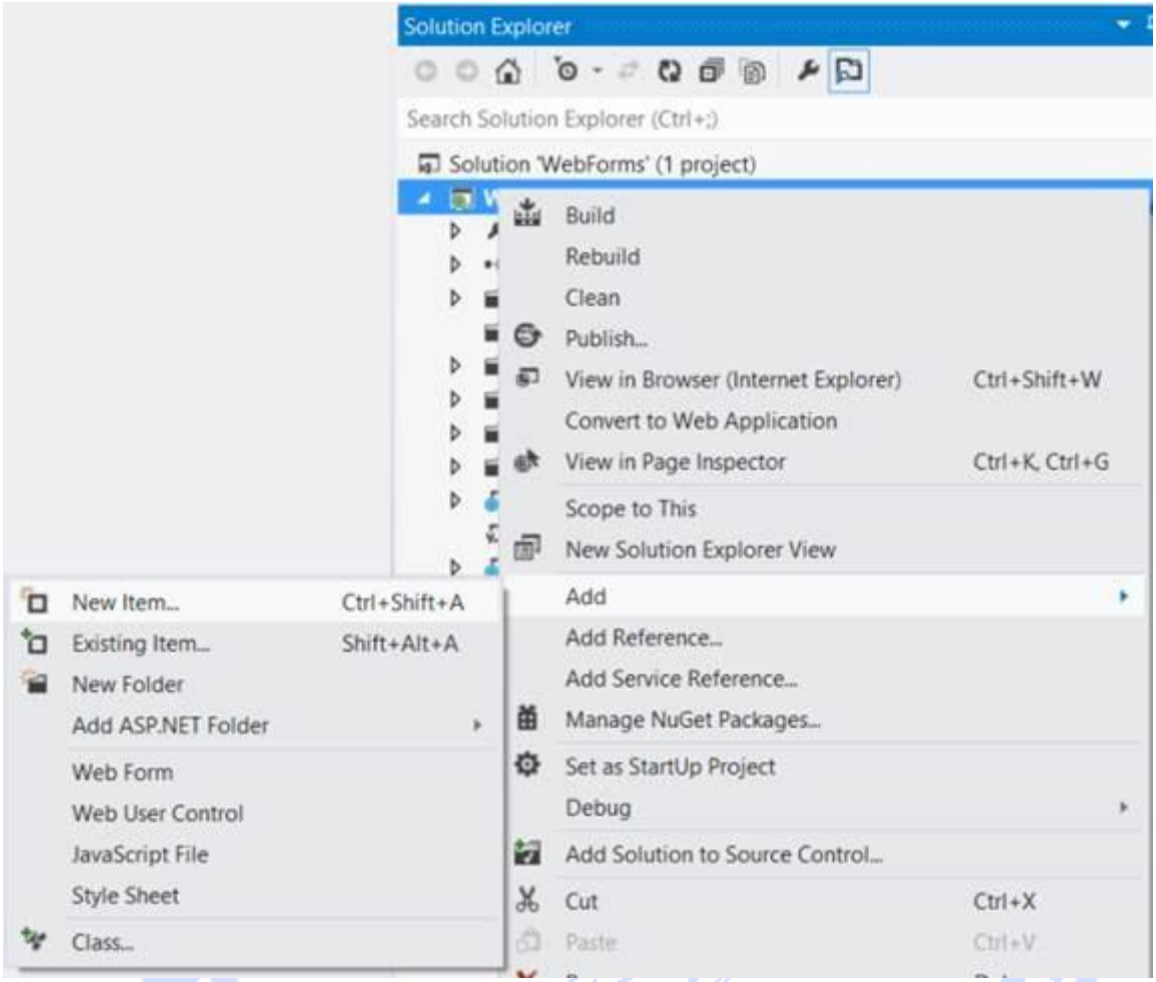


ساخت Model و Controller

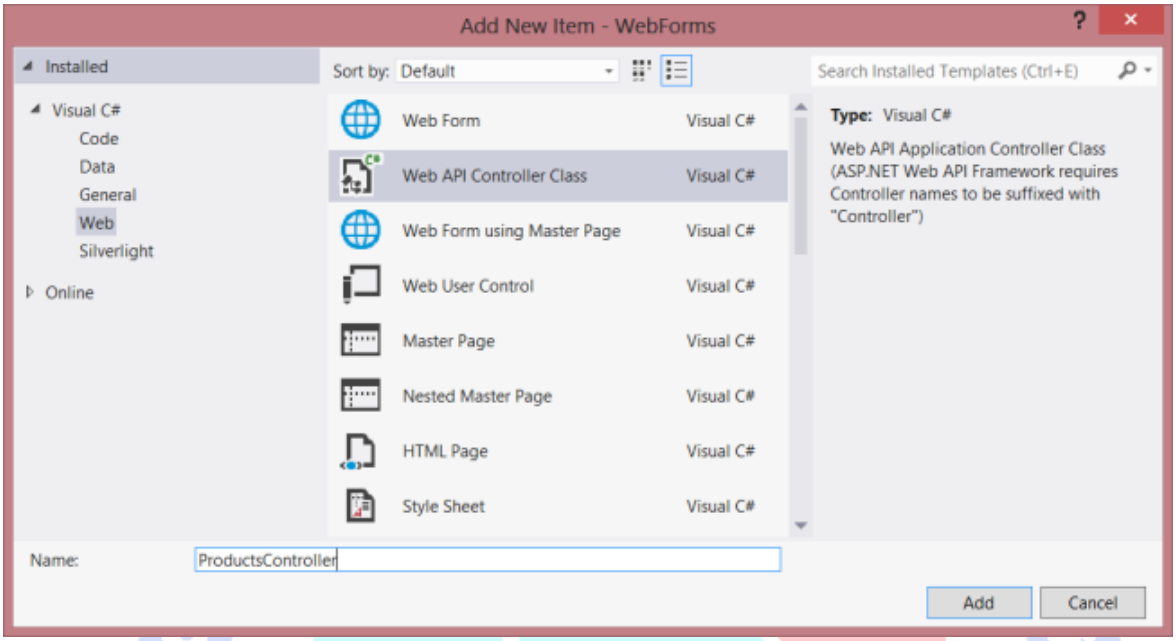
برای اضافه کردن کلاس Model ابتدا در solution explorer روی پروژه کلیک راست کنید و Add class را انتخاب کنید. نام کلاس را product انتخاب کرده و مقادیر زیر را در آن پیاده سازی کنید:

```
public class Product
{
   public int Id { get;set;}
   public string Name { get;set;}
   public decimal Price { get;set;}
   public string Category { get;set;}
}
```

سپس، یک کنترولر web API را به پروژه اضافه کنید. یک کنترولر، یک object است که درخواست های HTTP را برای web API مدیریت می کند. در solution explorer روی پروژه کلیک راست کنید و add new item را انتخاب کنید.



در زیر گزینه installed templates ، visual c# را باز کنید و Web را انتخاب کنید. سپس از لیست template ها، Web API Controller Class را انتخاب کنید و نام آن را ProductsController قرار دهید و Add کنید.



متد های داخل آن کلاس را حذف کنید و متد های زیر را به آن اضافه کنید:

```
namespace WebForms
{
   using System;
   using System.Collections.Generic;
   using System.Linq;
   using System.Net;
   using System.Net.Http;
   using System.Web.Http;
   public class ProductsController : ApiController
   {
       Product[] products = new Product[]
       {
           new Product { Id = 1, Name = "Tomato Soup", Category = "Groceries", Price = 1 },
           new Product { Id = 2, Name = "Yo-yo", Category = "Toys", Price = 3.75M },
           new Product { Id = 3, Name = "Hammer", Category = "Hardware", Price = 16.99M }
       };
       public IEnumerableGetAllProducts()
       {
           return products;
       }
       public Product GetProductById(int id)
       {
           var product = products.FirstOrDefault((p) =>p.Id == id);
           if (product == null)
           {
               throw new HttpResponseException(HttpStatusCode.NotFound);
           }
          return product;
       }
       public IEnumerableGetProductsByCategory(string category)
       {
           return products.Where(
               (p) =>string.Equals(p.Category, category,
                   StringComparison.OrdinalIgnoreCase));
       }
```

```
    }
}
```

اضافه کردن اطلاعات مسیریابی

حالا ما یک مسیر URI اضافه می کنیم که URI ها از فرم /api/products/ به Controller، مسیریابی شده است.

در solution explorer روی Global.asax دو بار کلیک کنید تا Global.asax.cs باز شود و using زیر را به آن اضافه کنید.

```
using System.Web.Http;
```

سپس کد های زیر را به متد Application_start آن اضافه کنید.

```
RouteTable.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{controller}/{id}",
    defaults: new { id = System.Web.Http.RouteParameter.Optional }
    );
```

اضافه کردن Ajax سمت کاربر

تمام چیزی که ما برای Web API احتیاج داریم این است که کاربران به آن دسترسی داشته باشند.حال می خواهیم یک صفحه Html که از jQuery برای فراخوانی API استفاده می کند بسازیم. Default.aspx را باز کنید.

کد های زیر را در content section اصلی جا به جا کنید:

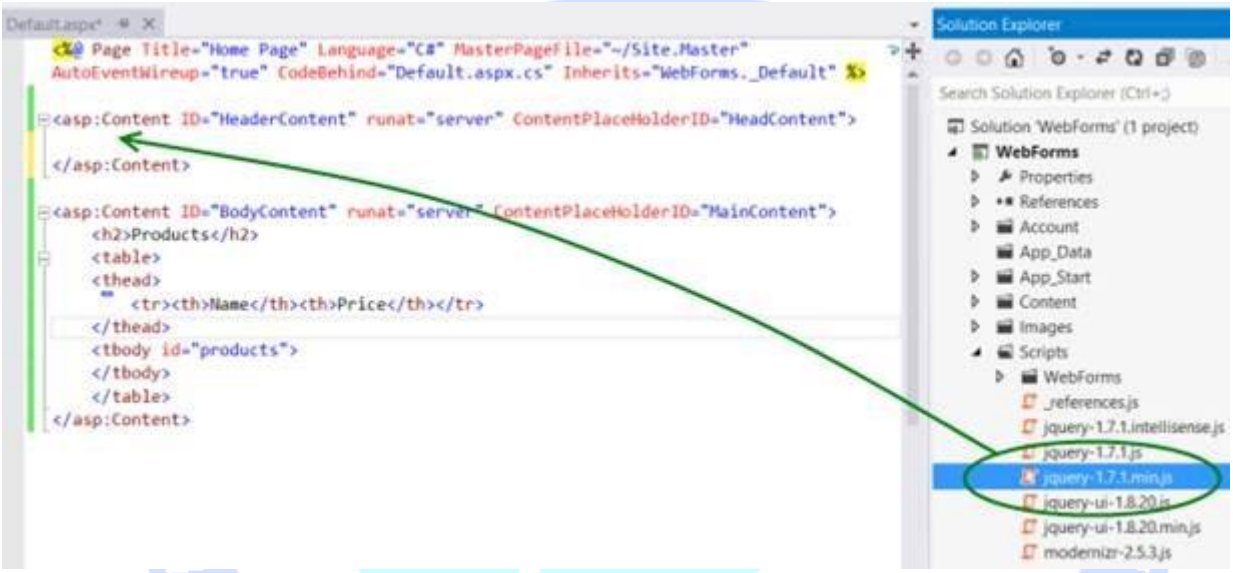
```
%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
    AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebForms._Default" %>
asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
/asp:Content>

asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    h2>Products
    table>
    thead>
        tr>NamePrice
    /thead>
    tbody id="products">
    /tbody>
    /table>
/asp:Content>
```

سپس، رفرنس jquery را به HeaderContent اضافه کنید.

```
asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
   script src="Scripts/jquery-1.7.1.min.js" type="text/javascript">
/asp:Content>
```

نکته: شما همچنین می توانید با drag and drop کردن، رفرنس script ها را از solution explorer به صفحه اضافه کنید.



قسمت jQuery زیر را به صفحه اضافه کنید.

```
function getProducts() {
    $.getJSON("api/products",
        function (data) {
            $('#products').empty();// Clear the table body.
            // Loop through the list of products.
            $.each(data, function (key, val) {
                // Add a table row for the product.
                var row = '<td>' + val.Name + '</td><td>' + val.Price + '</td>';
                $('<tr/>', { text: row }) // Append the name.
                    .appendTo($('#products'));
            });
        });
}
$(document).ready(getProducts);
```

وقتی این صفحه بارگذاری می شود، یه درخواست به صورت Ajax به سمت api/products ارسال می شود. این درخواست، لیستی از محصولات را در قالب JSON بر می گرداند.این اسکریپت اطلاعات محصول را به جدول Html اضافه می کند.