# EF第二章：Basic CRUD Functionality

## HTMLHelper类

传送门：https://msdn.microsoft.com/zh-cn/library/system.web.mvc.htmlhelper(v=vs.108).aspx

## PartialView, EditorFor 和 DisplayFor

**相同之处：**

PartialView, EditorFor 和 DisplayFor 都可以用作来实现页面的公共部分，其他页面可以根据需求来引用。

**不同之处：**

PartialView 是从Page的角度来处理，因此主 Page 和 PartialView 一般用相同的Model，这样做可以实现PartialView中所对应字段的编辑功能；如果PartialView使用的是主Page中Model的子Model，那么只能实现Partial View所对应Model的显示功能。

具体的引用方式为： @Html.Partial("~/Views/Shared/\_ProductPartial.cshtml")

 @Html.Partial("~/Views/Shared/\_ProductPartial.cshtml", Model.Products)

   @Html.RenderPartial("~/Views/Shared/\_ProductPartial.cshtml")

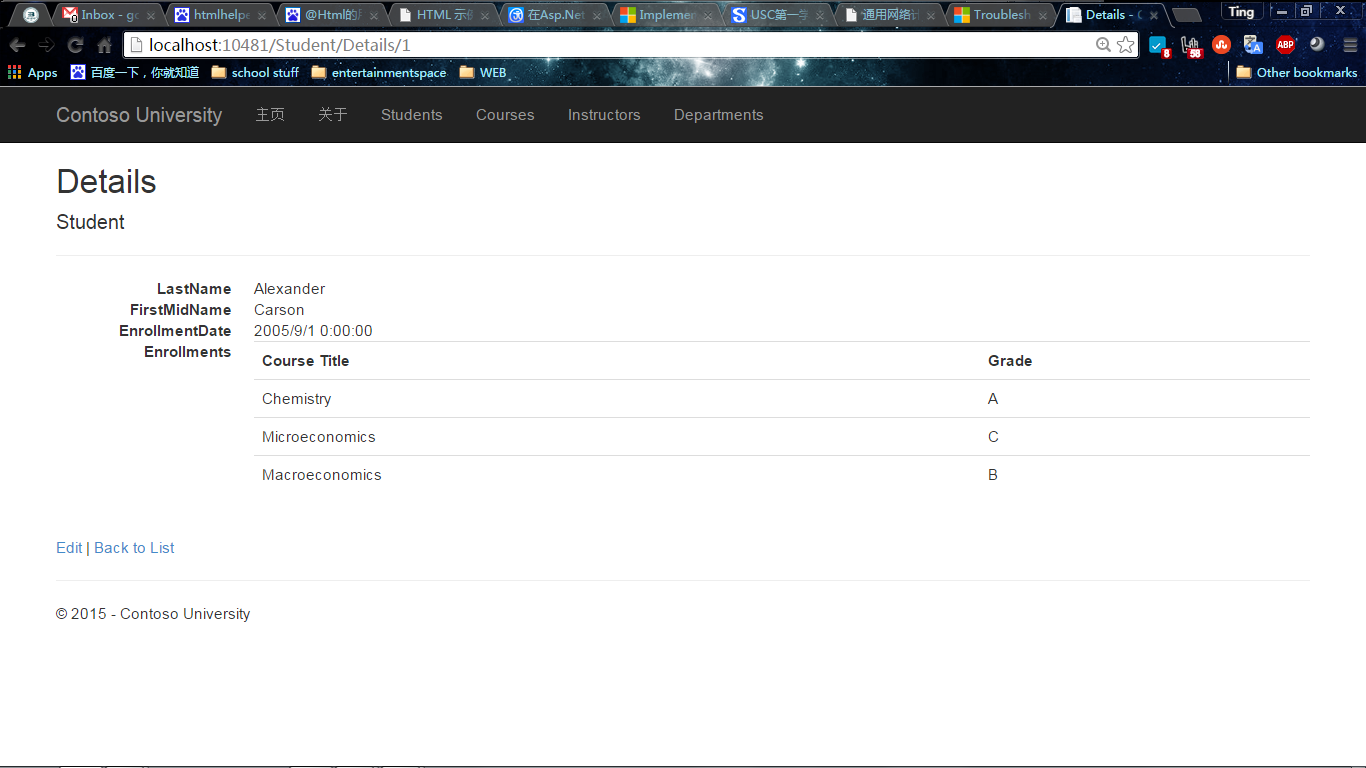
EditorFor和DisplayFor是从Model的角度来处理，此时DisplayFor实现的是子Model的显示功能，EditorFor实现的是子Model的编辑功能。

具体的引用方式为： @Html.EditorFor(m => m.Products)

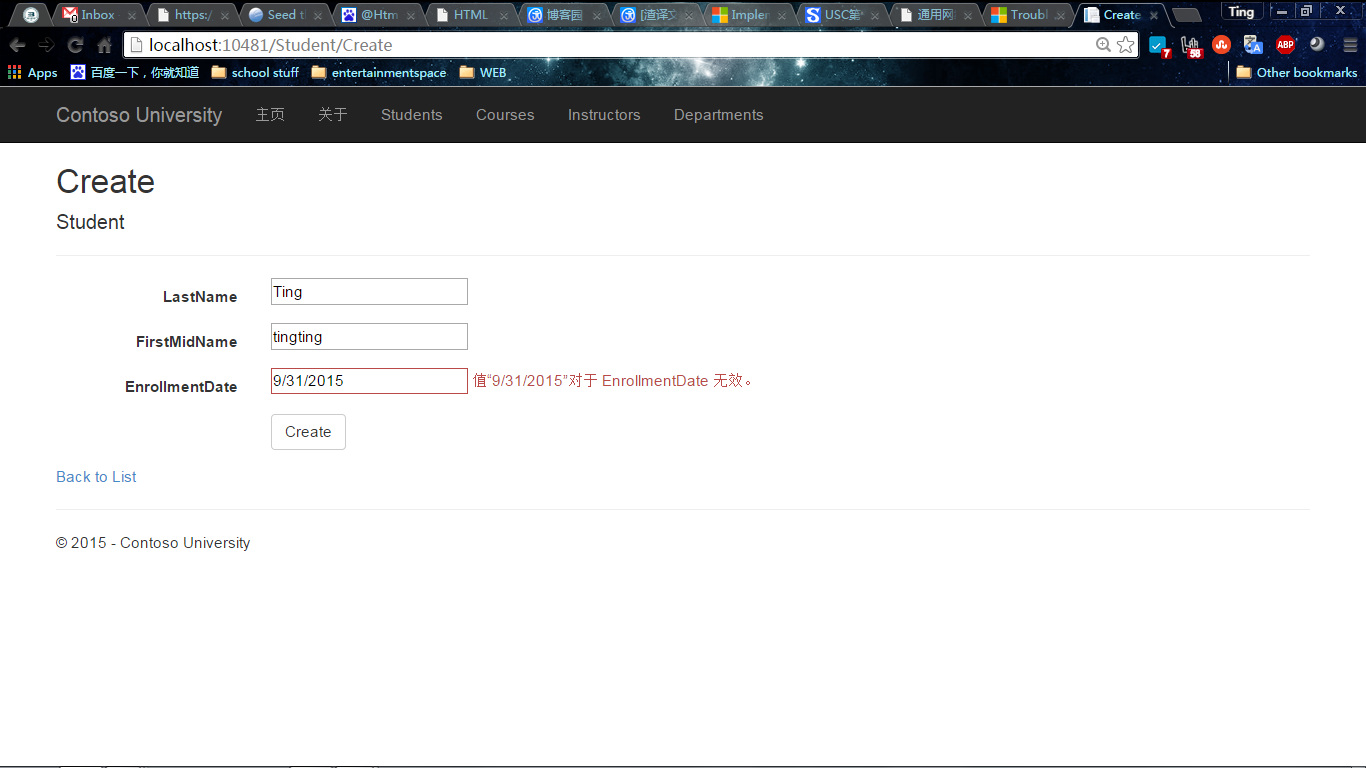
@Html.DisplayFor(m => m.Products)

单独运行detail不行，因为 reached from a link that specifies the student to display

直接跳到主页运行detail就好啦~



对于create更改：加了try和catch



问题：

加上了一段edit的代码，用editpost来加了个异常处理，页面404了

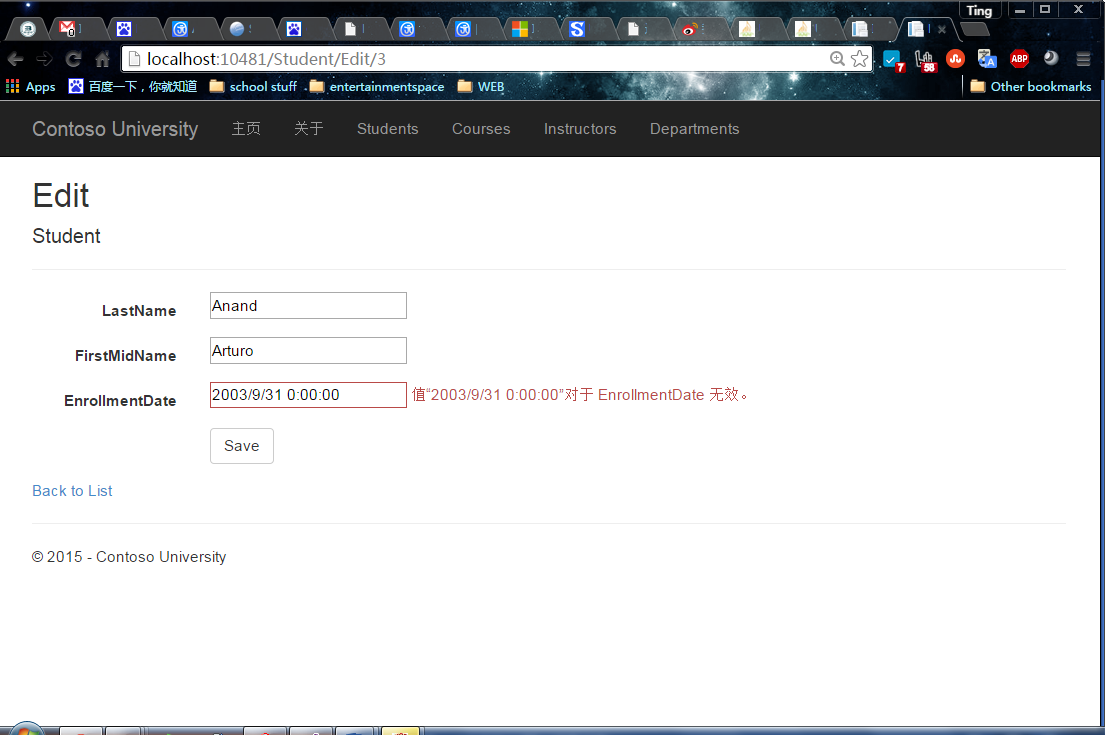
解决：

带这个的[HttpPost]

[ValidateAntiForgeryToken]

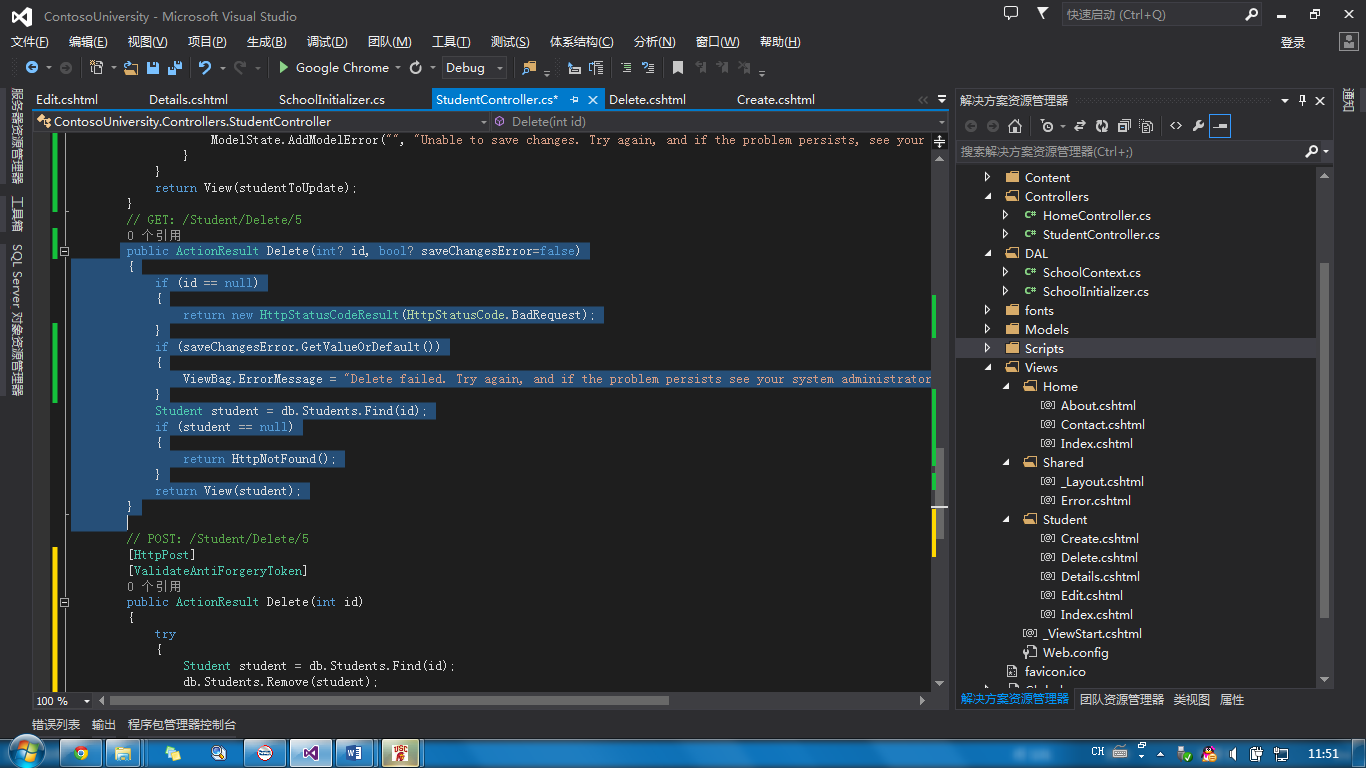
是带有html特性的，第二个Edit操作方法的上面有HttpPost属性。此属性指定了Edit方法的重载，此方法仅被POST 请求所调用。您可以将HttpGet属性应用于第一个编辑方法，但这是不必要的，因为它是默认的属性。

原来是改带html特性的那个edit就好了



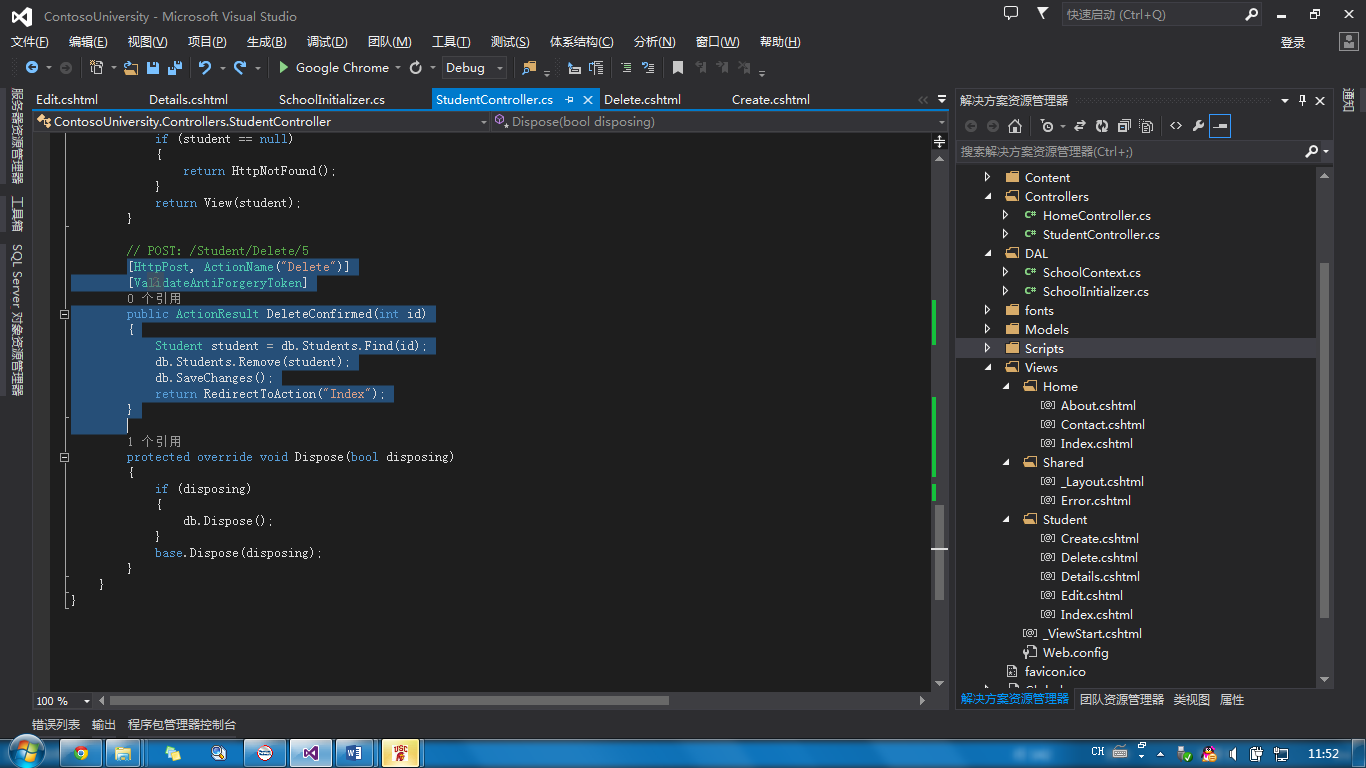
## Updating the Delete Page

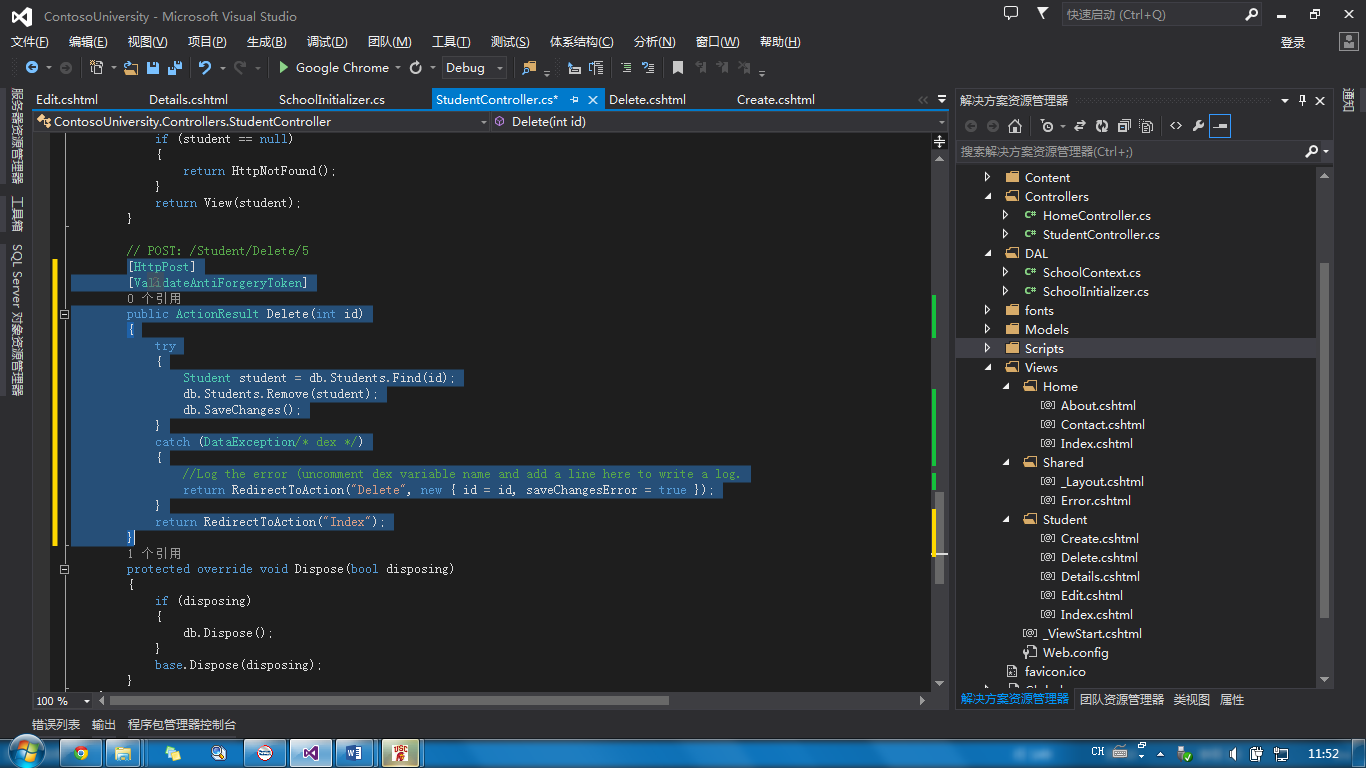
对于delete换血了两个地方



1. Replace the HttpGet Delete action

whether the method was called after a failure to save changes

换成：



1. Replace the HttpPost Delete action method (named DeleteConfirmed

performs the actual delete operation and catches any database update errors.

1. In *Views\Student\Delete.cshtml*,

加上提醒