خصوصیات controller

1. داخل پوشه controllers (‌بعدا میتوان این الزام را تغییر داد)

در اصل کلاس ما باید داخل namespace

.Controllers باشد

1. کلاس کنترلر باید فرزند کلاس Controller باشد
2. نام کلاس باید با کلمه Controller تمام شود.

Convention over configuration

در کلاس کنترلر متد هایی با خروجی از نوع ActionResult وجود دارند ( یا فرزندان ActionResult) به این متد ها Action Method گفته میشود.

این متد در حقیقت پاسخ دهنده های اصلی به درخواست های Httpهستند.

Routing Engine

ابزاری در ASP.NET MVC که وظیفه رساندن درخواست ها به Action ها را بر عهده دارد.

<http://www.abc.com/Users/Edit/4>

<http://www.abc.com/Home/Index>

<http://www.abc.com>

<http://www.abc.com/Home/About>

View

View در MVC وظیفه نمایش رابط کاربری را بر عهده دارد.

View میتواند حاوی کد کلاینت ساید به علاوه کدی با سینکس Razor باشد.

کدی که با سینتکس Razor در View نوشته میشود سمت سرور اجرا شده و خروجی کلاینت ساید آن به کلاینت ارسال میشود.

View ها با دو سینتکس سرور ساید قابل پیاده سازی هستند.

1. Razor
   1. Cshtml
   2. vbhtml
2. Web Form
   1. Aspx
   2. Ascx

روش های انتقال اطلاعات از اکشن به View

1. ViewData:

ViewData یک مجموعه کلید و مقداری است که طی هر درخواست و پاسخ بین اکشن و View به اشتراک گذاشته میشود.

کاربرد اصلی آن انتقال اطلاعات جزیی صفحه است.

1. ViewBag

ViewBag یک dynamic object است که طی هر درخواست و پاسخ بین اکشن و View به اشتراک گذاشته میشود.

در دو مورد بالا مدت زمان عمر آنها تا پایان اولین پاسخ به کلاینت است.

1. TempData:

TempData مشابه ViewData یک مجموعه با کلید های رشته ای و مقادیر object است.

با این تفاوت که مدت زمان عمر آن تا پایان response حاصل redirect زنده میماند.

هر سه حالت بالا برای انتقال اطلاعات جزیی به View استفاده میشوند.

1. Model:

View ها در ASP.NET MVC دو گونه اند

۴.۱ Untyped View: این View ها حالت پیش فرض View ها هستند در این نوع Viewها امکان دریافت Model به عنوان تامین کننده اطلاعات را نداریم

۴.۲ Typed View : این نوع View ها یک نوع داده را به عنوان نوع داده Model دریافت میکنند.

سپس میتوانیم از اکشنی که آن نما را اجرا میکند شی Modelای همنوع نوع تعریف شده برای View ارسال کنیم.

برای تعریف نوع داده مدل در View

@model نام کلاس مدل

برای دسترسی به شی مدل دریافتی در View باید طریق خصوصیت Model در View استفاده کنیم.

Razor

اگر بخواهسم از داخل کد بلاک داخل کد html متنی را بنویسیم: آن را داخل تگ <text> قرار میدهیم

روش های دریافت اطلاعات در Action

* + 1. شی Request در کلاس کنترلر

برای دریافت اطلاعات ارسال شده با متد POST

Request.Form[“نام کلید”]

برای دریافت اطلاعات ارسال شده به روش GET

Request.QueryString[“نام کلید”]

* + 1. استفاده از پارامتر FormCollection

از این پارامتر برای دریافت اطلاعات داخل متد Post استفاده میشود

1. Model Binder

بر اساس تشابه نام کلید های اطلاعات ارسالی با پارامتر های ورودی اکشن متد کار میکند

Entity Framework

1. Database First
2. Model First
3. Code First

برای ایجاد یک مدل Code First

باید اتفاقات زیر بیافتد

1. نصب پکیج ناگت Entity Framework
2. ایجاد یک کلاس ( Context) به عنوان کلاس اصلی مدل پایگاه و به ارث بردن آن از کلاسی به نام DbContext

این کلاس نماینده خود پایگاه داده است

Connection String

رشته ای که حاوی اطلاعات لازم برای برقراری ارتباط با پایگاه داده است

کلید=مقدار; کلید=مقدار; کلید=مقدار;

server/data source

برای مشخص کردن سرور اس کیو ال

Initial catalog/database

نام پایگاه داده

(در روش CodeFirst الزامی وجود ندارد که این پایگاه از قبل موجود باشد)

Windows Authentication

integrated security=SSPI/true;

SQL Server Authentication

userid/uid=نام کاربری

password/pwd=رمز عبور

server=.;database=ToursDb;uid=sa;pwd=developer;

محل های نوشتن مجاز Connection String

1. به عنوان پارامتر سازنده پدر کلاس Context
2. قرار دادن CS در فایل .config و اشاره به اسم آن در پارامتر سازنده کلاس پدر context

ایجاد روابط بین کلاس ها در مدل کد فرست

1. یک به یک :‌این رابطه در مدل کد فرست عموما در قالب وراثت پیاده سازی میشود – پیاده سازی آن در خارج از حالت وراثت هم ممکن است.
2. یک به چند:

در سمت ۱ رابطه: مجموعه ای از جنس کلاس طرف مقابل ایجاد میکنیم.

در سمت چند رابطه: یه خصوصیت از جنس طرف مقابل تعریف میکنیم

اختیاری:

چند به چند: در هر طرف رابطه خصوصیتی مجموعه ای از جنس طرف مقابل تعریف

میکنیم

پس از تعریف هر کلاس entityباید آن را به صورت زیر به کلاس context که نماینده پایگاه داده است معرفی کنیم.

مراحل تولید پایگاه از روی مدل کد فرست

1. اجرای دستور

Enable-migrations

در Nuget Package Manager Console

(migration در حقیقت تصویر تغییرات رخ داده بر روی مدل نسبت به migration قبلی است)

1. ایجاد تصویر از تغییرات مدل

Add-migration نام migration

1. اجرای دستور

Update-database

مراحل انجام عملیات درج رکورد (Create)

1. ایجاد شی Context
2. ایجاد شی از جنس کلاس Entity مورد نظر و مقدار دهی به خصوصیات آن
3. Ctx.نام مجموعه.Add(نام شی Entity(
4. Ctx.SaveChanges();

نوشتن پرس و جو (Query) در سی شارپ

1. LINQ

Language Integrated Query

from متغیر in نام مجموعه

where شرط

select نام متغییر یا بخشهایی از آن;

2. LINQ Extension Methods

Model Binder اطلاعات را از منابع زیر خوانده و سپس با پارامتر های ورودی Action تطبیق میدهد.

1. QueryString (GET)
2. Request Body (POST)
3. Route Value: اطلاعاتی که کلید آنها در Route تعریف شده

هر گاه در MVC متدی دیدیم که پارامتری به نام routeValues دریافت میکرد این پارامتر برای دریافت مقادیر داخل route است. نوع داده این پارامتر از نوع object است و عموما به آن یک anonymous object به عنوان مقدار میدهیم ، خصوصیات این anonymous object باید همنام با کلید های تعریف شده داخل route باشند.

new { Name = “John”, Family = “Doe” }

ctx.Entry<نوع داده>(نام شی).State = System.Data.Entity.EntityState.Modified;

برای جلوگیری از اعمال تغییرات بر روی یک خصوصیت خاص در پایگاه داده

ctx.Entry<نوع داده>

(نام شی).Property("نام خصوصیت")

.IsModified = false;

اعمال تنظیمات در مدل Code-First

برای اعمال تنظیمات در مدل کد فرست دو روش کلی وجود دارد

1. استفاده از Data Annotation ها: در این روش تنظیمات به صورت صفاتی در بالای کلاس های entity و خصوصیات آنها نوشته میشود. این صفات در عملیات Validation در ساختار MVC هم شناخته میشوند و کاربرد دارند
2. استفاده از Fluent API :‌ در این روش تنظیمات بوسیله فراخوانی توابعی در بازنویسی تابع OnModelCreating در کلاس Context انجام میشود.این روش از نظر قدرت اعمال تنظیمات در پایگاه کامل تر از روش اول است اما هیچ کاربرد Validation در MVCندارد.

HTMLHelpers

این متد ها ابزار های کمکی برای تولید تگ های کلاینت ساید در View ها هستند

<form>

</form>

@using (Html.BeginForm())

{

}

پارامتر htmlAttributes

این پارامتر صفات HTML برای تگ جاری را به صورت یک Anonymous Object از ما دریافت میکند

User Input Validation در MVC

در MVC سیستم ModelBinder علاوه بر صرفا قرار دادن مقادیر ورودی در پارامتر ها تابع Action توانایی تطبیق مقادیر با قوانین اعتبار سنجی (Validation)‌را هم دارد.

اصلی ترین منبع این قوانین اعتبار سنجی همان DataAnnotation ها هستند

در ActionMethod ها خصوصیتی به نام

ModelState.IsValid

وجود دارد که در صورت وجود خطای اعتبار سنجی مقدار آن false میشود

اگر قبل از انجام عملیات اصلی این مقدار را بررسی کنیم میتوانیم تشخیص دهیم که آیا مدل ما خطای اعتبارسنجی دارد یا خیر

Scaffold

اسکلت بندی

Scaffolder

ابزار اسکلت بندی

این ابزار در MVC با این هدف تولید شده که اسکلت بندی اولیه کدی را برای شما تولید کند