自主创建MvcMusicStore控制台项目

1. 创建一个Asp.net mvc5控制台项目

①项目模板-选择为空

②视图引擎-选择 Razor

③使用 HTML5 语义标记的检查

2. 添加一个类命名为HomeController

①在HomeController类中创建一个HomeController.cs类

②更改此方法返回一个字符串而不是 ActionResult

③更改要返回"Hello 从主页"的返回语句

### ④运行代码查看是否

3.再增添一个类命名为StoreController写入代码

将使用实现音乐应用商店的浏览功能。 存储控制器将支持以下三种方案：

①在我们的音乐应用商店音乐流派列表页

②列出所有特定流派音乐 album 的浏览页面

③显示有关特定音乐唱片集信息的详细信息页

重新生成和运行该项目，并浏览以下 Url:

①/ 存储

②/ Store/浏览

③/ 存储/详细信息

## 4、添加视图模板

将更改 HomeController Index 方法，以返回一个 ActionResult，并使其返回 View()

添加视图Index，此视图使用 Razor 语法

## 5、更新了样式表

空项目模板包含一个非常简化的 CSS 文件只包括用来显示验证消息的样式。 我们的设计器提供了一些额外的 CSS 和图像，因此我们将添加中现在定义我们的站点，外观和感觉。

覆盖现有 Site.css 文件

## 6、使用模型将信息传递到视图

返回一个 ActionResult 控制器操作方法可以将模型对象传递给视图。

改 StoreController 若要使用它显示我们的模型中的动态信息的视图

将更新的详细信息控制器操作，以便它返回 ActionResult 而不是一个字符串

让我们现在创建一个使用我们的唱片集来生成 HTML 响应的视图模板

创建新的视图模板像 HomeController 与之前执行。

## 7、页之间添加链接

我们浏览页面的链接需要传递一个参数，不过，因此我们将使用另一个重载采用三个参数的 Html.ActionLink 方法：

* 链接文本，将显示类型名称
* 控制器操作名称 （浏览）
* 路由参数值，指定的名称 （类型） 和值 （类型名称）将它们组合在一起此处向存储区索引视图，我们就将编写这些链接：

## 8、模型和数据访问

#### 添加艺术家模型类

#### 更新唱片集类

对流派类进行更新

将应用添加\_数据文件夹

在web.config文件钟创建的链接字符串

添加上下文类

添加存储目录数据

## 9、编辑窗体和模板化

创建名为的新控制器**StoreManagerController**

* 它使用实体框架局部变量创建新 StoreManagerController
* 它将 StoreManager 文件夹添加到项目的 Views 文件夹
* 它将添加 Create.cshtml、 Delete.cshtml、 Details.cshtml、 Edit.cshtml 和 Index.cshtml 视图中，强类型化为唱片集类

修改基类生成的视图

### 10、创建操作方法

* 读取窗体值
* 检查窗体值是否传递的任何验证规则
* 如果表单提交有效，将数据保存并显示更新的列表
* 如果提交窗体不是有效的重新显示具有验证错误的窗体

修改增删查改

## 11、向我们唱片集窗体中添加验证

我们将使用以下数据注释属性：

* **所需**– 指示该属性是必填的字段
* **DisplayName** – 定义要我们在窗体字段和验证消息使用的文本
* **StringLength** – 定义字符串字段的最大长度
* **范围**– 为数字字段提供最大和最小值
* **将绑定**– 列出要排除或包括当参数或窗体值绑定到模型属性的字段
* **ScaffoldColumn** – 允许隐藏编辑器窗体中的字段

## 12、购物车与Ajax更新

添加购物车、顺序和OrderDetail模型类

更新MusicStoreEntities类

## 13、管理购物车业务逻辑

**AddToCart**采用唱片集作为参数并将其添加到用户的购物车。 购物车表会跟踪每个唱片集的数量，因为它包括逻辑来根据需要创建一个新行或只是递增数量，如果用户具有已排序的唱片集的一个副本。

**RemoveFromCart**采用唱片集 ID，并将其从用户的购物车中删除。 如果用户仅在其购物车中有一份唱片集，删除的行。

**EmptyCart**从用户的购物车中移除所有项。

**GetCartItems**检索以便进行显示或处理 CartItems 的列表。

**GetCount**检索用户具有在其购物车中的唱片集的总数。

**GetTotal**计算购物车中的所有项的总成本。

**CreateOrder**签出阶段将购物车转换为订单。

**GetCart**是一种静态方法，可允许我们控制器，若要获取的购物车对象。 它使用**GetCartId**方法以处理从用户的会话读取 CartId。 GetCartId 方法，以便它可以从用户的会话中读取用户的 CartId 需要 HttpContextBase。

## 14、创建一个ViewModels文件夹

在 Viewmodel 文件夹中添加 ShoppingCartViewModel 类

添加 ShoppingCartRemoveViewModel ViewModels 文件夹中

使用空的控制器模板向项目添加一个新的购物车控制器

删除和 CartSummary 控制器操作处理两种特殊情况

## 15、使用 jQuery Ajax 更新

创建强类型化为 ShoppingCartViewModel 并使用列表视图模板使用与之前相同的方法的购物车索引页。

* 从列表中移除已删除的唱片集
* 更新标头中的购物车计数
* 为用户显示一则更新消息
* 更新购物车的总价

运行输出查看

## 16、注册和结账

创建 CheckoutController 会收集购物者的地址和付款信息。

## 17、迁移购物车

添加 using 语句引用 MvcMusicStore.Models，然后添加 MigrateShoppingCart 方法

## 18、创建 CheckoutController

添加控制器类声明，以要求用户在签出之前注册的上面 Authorize 属性

CheckoutController 将具有以下控制器操作：

**AddressAndPayment （GET 方法）** 将显示一个窗体以允许用户输入他们的信息。

**AddressAndPayment （POST 方法）** 将验证输入和处理订单。