

```
1  @{
2      ViewBag.Title = "Inventory";
3  }
4  @section css
5  {
6      <link href="~/Content/WebGrid.css" rel="stylesheet" />
7      <link href="~/Content/CartGrid.css" rel="stylesheet" />
8  }
9  <h2>Halloween Superstore Inventory</h2>
10 @{
11     using (Html.BeginForm("Index", "Inventory", FormMethod.Post, new { id =
12         "FrmInventory" })))
13     {
14         <div>
15             @{
16                 WebGrid grid = new WebGrid(Model);
17             }
18             @grid.GetHtml(
19                 tableStyle: "webgrid-table",
20                 headerStyle: "webgrid-header",
21                 footerStyle: "webgrid-footer",
22                 alternatingRowStyle: "webgrid-alternating-row",
23                 columns: grid.Columns(
24                     grid.Column("ProductID", header: "Product ID"),
25                     grid.Column("Name"),
26                     grid.Column("ShortDescription", header: "Description"),
27                     grid.Column("UnitPrice", header: "Price", format:
28                         @<text>@item.UnitPrice.ToString("$0.00")</text>),
29                     grid.Column("OnHand", header: "Available"),
30                     grid.Column("LongCategoryDescription", header: "Category")
31                 )
32             </div>
33         }
34     }
```

```
1
2 @{
3     ViewBag.Title = "Halloween";
4 }
5 @section css
6 {
7     <link href="~/Content/Home.css" rel="stylesheet" />
8 }
9
10 <h1>@ViewBag.HeaderText</h1>
11
12 <div>
13     Thanks for visiting us on this beautiful @DateTime.Today.DayOfWeek!
14 </div>
15
16 @section footer
17 {
18     <p>@ViewData["FooterText"]</p>
19 }
20
21
```

```
1 @model Halloween_MVC_1.Models.ViewModels.CartViewModel
2 @{
3     ViewBag.Title = "Index";
4 }
5 @section css
6 {
7     <link href="~/Content/WebGrid.css" rel="stylesheet"/>
8     <link href="~/Content/CartGrid.css" rel="stylesheet"/>
9 }
10 <h2>Confirm Your Order</h2>
11 @{
12     if (Model.Cart.Count == 0)
13     {
14         <p>There is nothing in your cart to confirm.</p>
15     }
16     else
17     {
18         using(Html.BeginForm("Confirmed", "Home", FormMethod.Post, new {id =
19             "frmConfirmed"}))
20         {
21             <div>
22                 @{
23                     WebGrid grid = new WebGrid(Model);
24                 }
25                 @grid.GetHtml(
26                     tableStyle: "webgrid-table",
27                     headerStyle: "webgrid-header",
28                     footerStyle: "webgrid-footer",
29                     alternatingRowStyle: "webgrid-alternating-row",
30                     columns: grid.Columns(
31                         grid.Column("ProductID", header: "Product ID"),
32                         grid.Column("Name"),
33                         grid.Column("ShortDescription", header: "Description"),
34                         grid.Column("UnitPrice", header: "Price", format:
35                             @<text>@item.UnitPrice.ToString("$0.00")</text>),
36                         grid.Column("Quantity")
37                     )
38                 </div>
39                 <br/>
40                 <input type="submit" id="btnConfirm" value="Confirm Order"
41                     class="button"/>
42             }
43         }
44     }
45 }
```

```
1 using Halloween_MVC_1.Models;
2 using Halloween_MVC_1.Models.ViewModels;
3 using System.Collections.Generic;
4 using System.Web.Mvc;
5
6 namespace Halloween_MVC_1.Controllers
7 {
8     public class InventoryController : Controller
9     {
10         public ActionResult Index()
11         {
12             OrderModel model = new OrderModel();
13             List<ProductViewModel> prodList = model.GetProductsList();
14             return View(prodList);
15         }
16     }
17 }
```


```
1 @using Halloween_MVC_1.Models.ViewModels
2 @model Halloween_MVC_1.Models.ViewModels.CartViewModel
3
4 @{
5     ViewBag.Title = "List";
6 }
7
8 <h2>Your Shopping Cart</h2>
9 @if (Model.Cart.Count == 0)
10 {
11     <p>There is nothing in your cart to display.</p>
12 }
13 else
14 {
15     <select name="selectedProducts" multiple="multiple">
16         @foreach (ProductViewModel item in Model.Cart)
17         {
18             <option>@item.Display()</option>
19         }
20     </select>
21     <br/>
22 }
23
24
25
```

```
1 using Halloween_MVC_1.Models;
2 using Halloween_MVC_1.Models.ViewModels;
3 using System;
4 using System.Collections.Generic;
5 using System.Linq;
6 using System.Web;
7 using System.Web.Mvc;
8
9 namespace Halloween_MVC_1.Controllers
10 {
11     public class OrderController : Controller
12     {
13         private OrderModel order = new OrderModel();
14         // GET: Order
15         [HttpGet]
16         public ActionResult Index(string id)
17         {
18             //get list for drop down either from TempData or from order data
19             SelectList products = (SelectList)TempData["products"];
20             if (products == null)
21             {
22                 List<ProductViewModel> list = order.GetProductsList();
23                 products = new SelectList(list, "ProductId", "Name", id) ;
24             }
25             //if no URL parm, get first product from list and refresh
26             if (string.IsNullOrEmpty(id))
27             {
28                 id = products.ElementAt(0).Value;
29                 TempData["products"] = products;
30                 //redirect to action - cause the browser to make Get request to ↗
31                 //the specified action
32                 return RedirectToAction("Index", "Order", new { id });
33             }
34             else
35             {
36                 //get the selected product and return in view method
37                 OrderViewModel model = order.GetOrderInfo(id);
38                 model.ProductsList = products;
39                 model.SelectedProduct.Quantity = 1;
40                 return View(model);
41             }
42         }
43
44         [HttpPost]
45         //RedirectToAction - instead of making a post back to the server,
46         //cause the browser to make a GET request to the specified action
47         public RedirectToRouteResult Index(FormCollection collection)
48         {
```

```
49         string pID = collection["ddlProducts"];
50         return RedirectToAction("Index", "Order", new { id = pID });
51     }
52 }
53 }
```

```
1 using Halloween_MVC_1.Models.ViewModels;
2 using System.Collections.Generic;
3 using System.Linq;
4
5 namespace Halloween_MVC_1.Models
6 {
7     public class OrderModel
8     {
9         private List<ProductViewModel> products;
10
11         public List<Product> GetAllProductsFromDataStore()
12         {
13             HalloweenDataEntities dataContext = new HalloweenDataEntities();
14             return dataContext.Products.OrderBy(p => p.Name).ToList();
15         }
16         public Product GetProductByIDFromDataStore(string id)
17         {
18             HalloweenDataEntities dataContext = new HalloweenDataEntities();
19             return dataContext.Products.FirstOrDefault(p => p.ProductID == id);
20         }
21
22         public ProductViewModel ConvertToViewModel(Product p)
23         {
24             HalloweenDataEntities dataContext = new HalloweenDataEntities();
25             ProductViewModel model = new ProductViewModel
26             {
27                 ProductID = p.ProductID,
28                 ImageFile = p.ImageFile,
29                 LongDescription = p.LongDescription,
30                 ShortDescription = p.ShortDescription,
31                 Name = p.Name,
32                 UnitPrice = p.UnitPrice,
33                 LongCategoryDescription = dataContext.Categories.FirstOrDefault
34                     (c => c.CategoryID == p.CategoryID)
35                     .LongName.ToString();
36             };
37             return model;
38         }
39         public List<ProductViewModel> GetAllProducts()
40         {
41             List<ProductViewModel> productModels = new List<ProductViewModel>();
42             List<Product> productsList = GetAllProductsFromDataStore();
43             foreach (Product p in productsList)
44                 productModels.Add(ConvertToViewModel(p));
45             return productModels;
46         }
47
48         public List<ProductViewModel> GetProductsList()
```



```
49     {
50         if (products == null)
51             products = GetAllProducts();
52         return products;
53     }
54     public ProductViewModel GetSelectedProduct(string id)
55     {
56         if (products == null)
57             return ConvertToViewModel(GetProductByIDFromDataStore(id));
58         else
59             return products.FirstOrDefault(p => p.ProductID == id);
60     }
61     public OrderViewModel GetOrderInfo(string id)
62     {
63         OrderViewModel order = new OrderViewModel { SelectedProduct = 
64             GetSelectedProduct(id) };
65         return order;
66     }
67 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using Halloween_MVC_1.Models;
7 using Halloween_MVC_1.Models.ViewModels;
8
9 namespace Halloween_MVC_1.Controllers
10 {
11     public class CartController : Controller
12     {
13         private CartModel cart = new CartModel();
14         // GET: Cart
15         [HttpGet]
16         public RedirectToRouteResult Index()
17         {
18             return RedirectToAction("List");
19         }
20         [HttpGet]
21         public ViewResult List()
22         {
23             CartViewModel model = (CartViewModel) TempData["cart"];
24             if (model == null)
25                 model = cart.GetCart();
26             return View(model);
27         }
28
29         [HttpPost]
30         public RedirectToRouteResult List(OrderViewModel order)
31         {
32             CartViewModel model = cart.GetCart(order.SelectedProduct.ProductID);
33             model.AddedProduct.Quantity = order.SelectedProduct.Quantity;
34             cart.AddToCart(model);
35             TempData["cart"] = model;
36             return RedirectToAction("List", "Cart");
37         }
38     }
39 }
```

```
1 using Halloween_MVC_1.Models.ViewModels;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Web;
6
7 namespace Halloween_MVC_1.Models
8 {
9     public class CartModel
10    {
11        private List<ProductViewModel> GetCartFromDataStore()
12        {
13            List<ProductViewModel> cart;
14            cart = (List<ProductViewModel>)(HttpContext.Current.Session
15                ["cart"]);
16            if (cart == null)
17            {
18                cart = new List<ProductViewModel>();
19                HttpContext.Current.Session["cart"] = cart;
20            }
21            return cart;
22        }
23        private void AddItemToDataStore(CartViewModel model)
24        {
25            model.Cart.Add(model.AddedProduct);
26        }
27        private ProductViewModel GetSelectedProduct(string id)
28        {
29            OrderModel order = new OrderModel();
30            return order.GetSelectedProduct(id);
31        }
32
33        public CartViewModel GetCart(string id = "")
34        {
35            CartViewModel model = new CartViewModel();
36            model.Cart = GetCartFromDataStore();
37            if (!string.IsNullOrEmpty(id))
38                model.AddedProduct = GetSelectedProduct(id);
39
40            return model;
41        }
42        public void AddToCart(CartViewModel model)
43        {
44            if (model.AddedProduct.ProductID != null)
45            {
46                string id = model.AddedProduct.ProductID;
47                ProductViewModel inCart = model.Cart.FirstOrDefault(p =>
```

```
48         if (inCart == null)
49             AddItemToDataStore(model);
50         else
51             inCart.Quantity += model.AddedProduct.Quantity;
52     }
53 }
54
55 public void RemoveFromCart(string id = "")
56 {
57     if (string.IsNullOrEmpty(id)) return;
58
59     List<ProductViewModel> cart;
60     cart = (List<ProductViewModel>)(HttpContext.Current.Session ↗
61         ["cart"]);
62     if (cart == null) return;
63
64     foreach (ProductViewModel p in cart.Where(p => p.ProductID == id))
65     {
66         cart.Remove(p);
67     }
68 }
69 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using Halloween_MVC_1.Models;
7 using Halloween_MVC_1.Models.ViewModels;
8
9 namespace Halloween_MVC_1.Controllers
10 {
11     public class ConfirmController : Controller
12     {
13         private readonly CartModel cart = new CartModel();
14
15         // GET: Confirm
16         public ActionResult Index()
17         {
18             CartViewModel model = (CartViewModel) TempData["cart"];
19             if (model == null)
20                 model = cart.GetCart();
21             return View(model);
22         }
23     }
24 }
```

```
1
2 @{
3     ViewBag.Title = "Confirmed";
4 }
5
6 <h2>Your order was confirmed.</h2>
7
8
```

```
1 @model Halloween_MVC_1.Models.ContactModel
2
3 @{
4     ViewBag.Title = "Contact Us";
5 }
6 <h1>Contact Us</h1>
7
8 <p>Email: @Model.Email</p>
9 <p>Phone: @Model.Phone</p>
10 <p>Fax: @Model.Fax</p>
11 <p>Address: @Model.Address</p>
12 <p>
13     @foreach (string url in Model.SocialMediaUrls)
14     {
15         <a href="@url" target="_blank">@url</a> <br />
16     }
17 </p>
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5
6 namespace Halloween_MVC_1.Models
7 {
8     public class ContactModel
9     {
10         public string Email => "info@halloweensuperstore.com";
11         public string Phone => "1-800-123-4567";
12         public string Fax => "1-800-123-4568";
13         public string Address => "1234 Main Street, Anytown, USA";
14
15         public List<string> SocialMediaUrls
16         {
17             get
18             {
19                 List<string> urls = new List<string>();
20                 urls.Add("http://www.facebook.com/halloweensuperstore");
21                 urls.Add("http://www.twitter.com/halloweensuperstore");
22                 return urls;
23             }
24         }
25     }
26 }
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using Halloween_MVC_1.Models;
7
8 namespace Halloween_MVC_1.Controllers
9 {
10     public class HomeController : Controller
11     {
12         // GET: Home
13         public ActionResult Index()
14         {
15             ViewBag.HeaderText = "Welcome to the Halloween store.";
16             ViewData["FooterText"] = "Where every day is Halloween!";
17             return View();
18         }
19         public ActionResult Contact()
20         {
21             ContactModel model = new ContactModel();
22             return View(model);
23         }
24
25         public ActionResult Confirmed()
26         {
27             return View();
28         }
29     }
30 }
```

```
1 @model Halloween_MVC_1.Models.ViewModels.OrderViewModel
2
3 @{
4     ViewBag.Title = "Order";
5     var product = Model.SelectedProduct;
6 }
7 @section css
8 {
9     <link href="~/Content/Order.css" rel="stylesheet" />
10 }
11
12 <form id="frmDropDown" action="/Order/Index/" method="post">
13     <h2>Halloween Store Order Page</h2>
14     <label>Please select a product&nbsp;</label>
15     <select id="ddlProducts" name="ddlProducts" class="autoPostBack">
16         @foreach (var item in Model.ProductsList)
17         {
18             <option value="@item.Value" selected="@item.Selected">@item.Text</
19                 option>
20         }
21     </select>
22 </form>
23 @using (Html.BeginForm("List", "Cart", FormMethod.Post, new { id = "frmCart" }))
24 {
25     <div id="productData">
26         <label id="lblName">@product.Name</label>
27         @Html.Label("", @product.ShortDescription, new { id =
28             "lblShortDescription"})
29         <label id="lblLongDescription">@product.LongDescription</label>
30         <label id="lblUnitPrice">@product.UnitPrice.ToString("C2")</label>
31         <label id="lblQuantity">Quantity&nbsp;</label>
32         @*<input id="txtQuantity" name="txtQuantity" type="text"
33             value="@product.Quantity" />
34         <input type="hidden" name="hdnID" value="@product.ProductID" />*@
35         @Html.TextBoxFor(m => m.SelectedProduct.Quantity, new {id =
36             "txtQuantity"})
37         @Html.HiddenFor(m => m.SelectedProduct.ProductID)
38
39         <input type="submit" id="btnAdd" value="Add to Cart" class="button"/>
40     </div>
41     
42 }
```