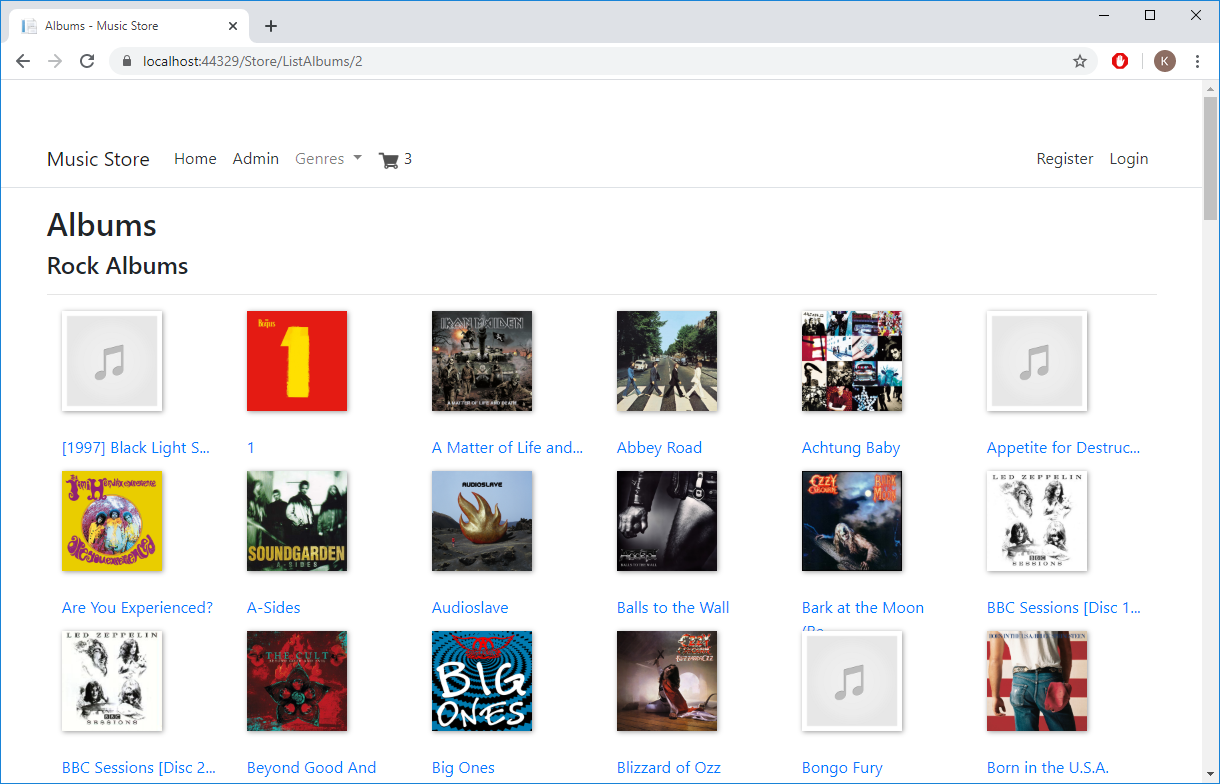
***Music Store – part 4***

***1. MyHelper***

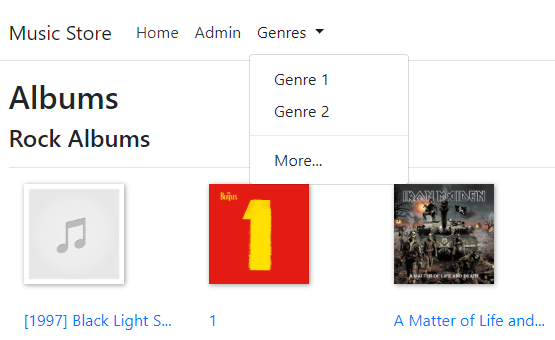
Make sure the titles in the *ListAlbums* page are also abbreviated to a maximum of 20 characters:



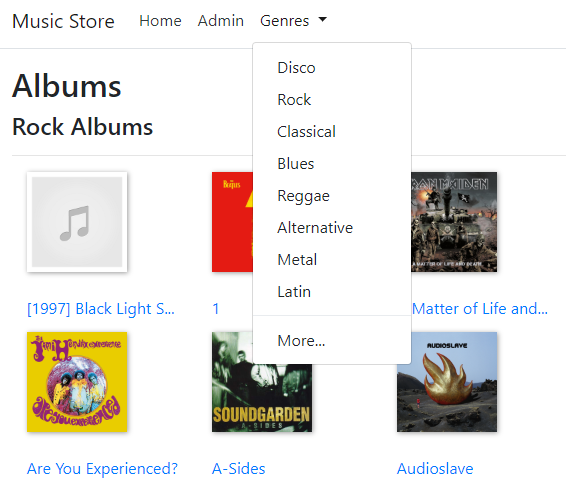
***2. GenreMenuComponent***

Create an additional component *GenreMenuComponent.cs*. In the *InvokeAsync* method, fetch the first 8 genres and then call the view. Pass the genres as parameters.

Then copy the view from the resources to the right place in your project. In the *\_Layout* page, replace the *Genres* *menu item* with the new component. For now, you get:



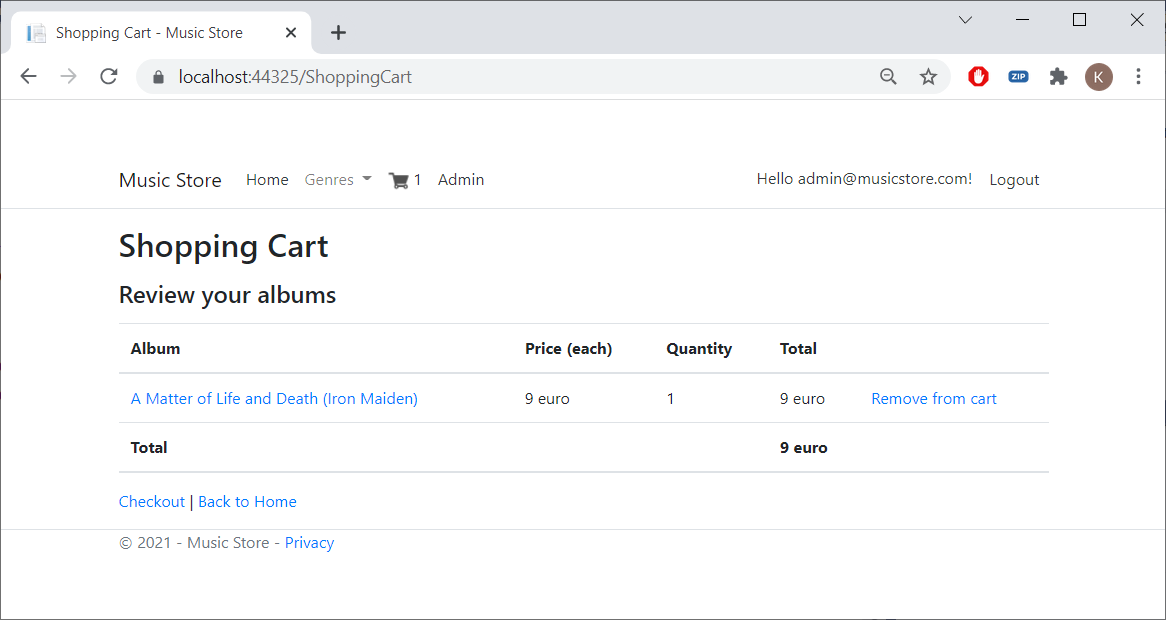
Now adjust the view of the component so that you get:



If you click on *More...,* you are taken to the page with all genres.

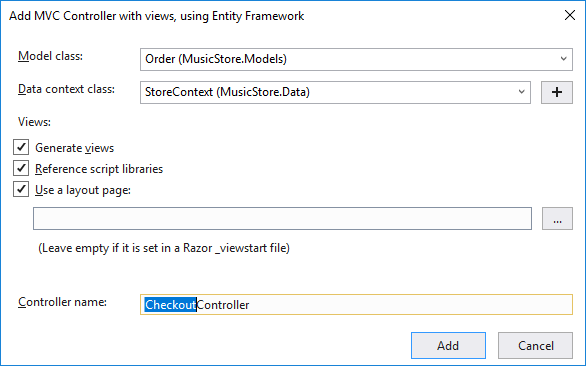
***3. CheckoutController***

Add an additional *Checkout* hyperlink to the shopping cart view:



If you click on it, you will be taken to the *Create* method of the *Checkout* controller.

We will now add the last controller to our application: *CheckoutController.cs*. Use scaffolding to generate the code and the views:

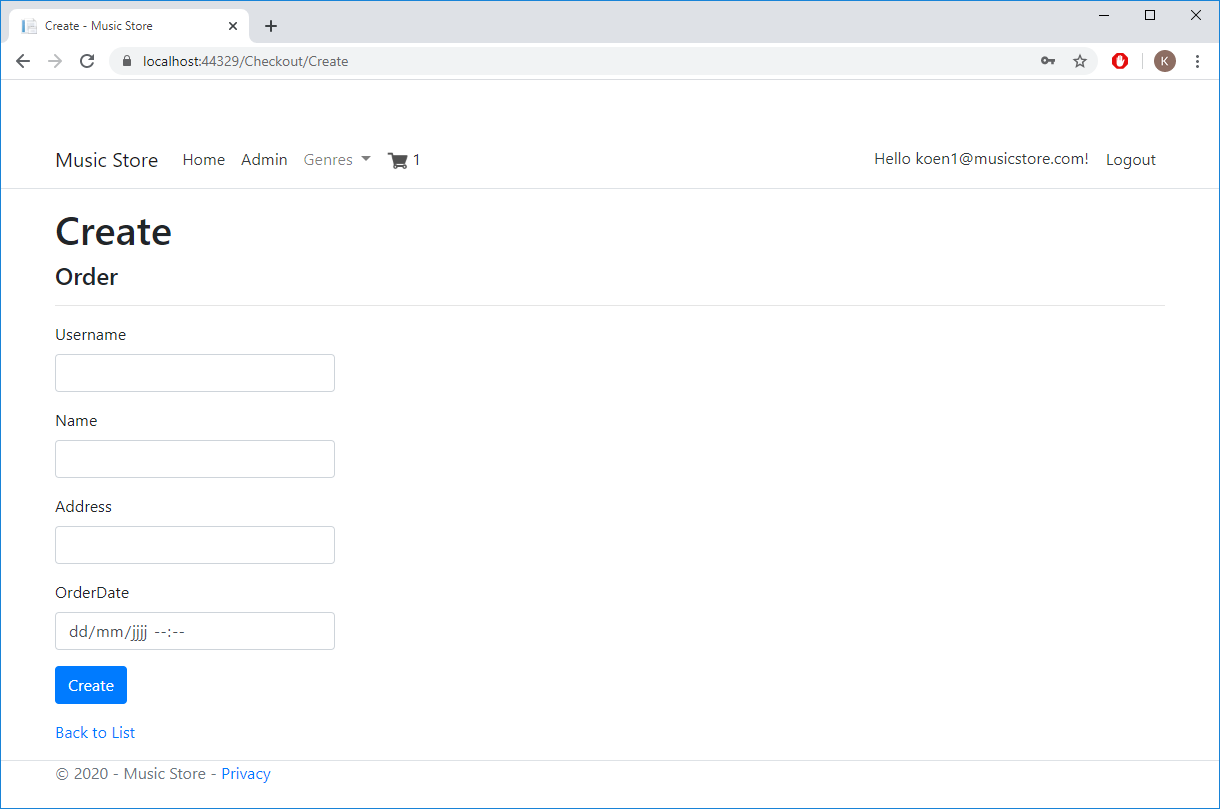


We only need to be able to add orders. Therefore, we only need the *Create* method. All other methods and associated views can be removed.

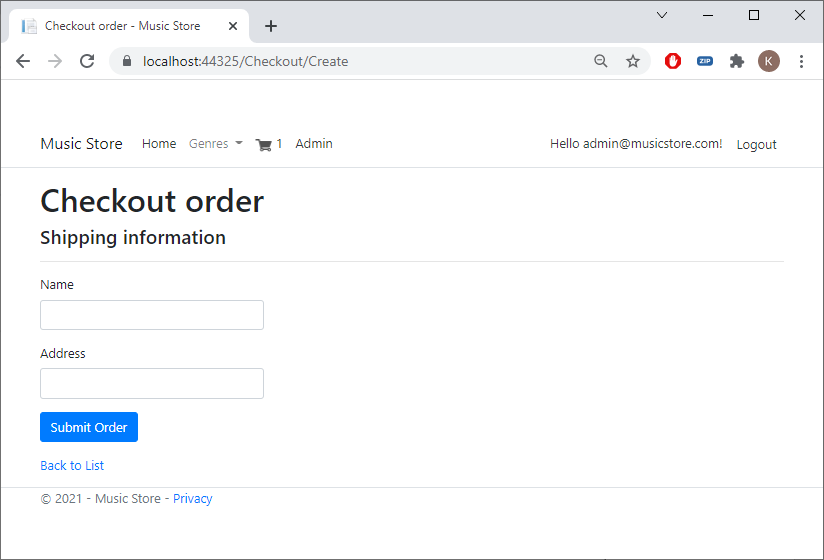
Since you can only checkout if you are logged in, add the following attribute just before the definition of the *Checkout Controller* class:

[Authorize]

You are now automatically redirected to the login page when you try to checkout. Once you are logged in (or after you have registered yourself), you will automatically return to the page where you can enter the order details:



Now modify the view above. *OrderDate* and *Username* can be omitted. We can fill in these data ourselves from the programme code. Also change the titles. You get:



Now make sure that the order and the order items are saved correctly. This can be done by modifying the *HttpPost-Create* method as follows (take a closer look at the programme code):

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("OrderID,Address,Name,OrderDate,Username")] Order order)

{

if (ModelState.IsValid)

{

order.Username = HttpContext.User.Identity.Name;

order.OrderDate = DateTime.Now;

\_context.Add(order);

\_context.SaveChanges();

var cart = new ShoppingCart(HttpContext, \_context);

cart.CreateOrderItems(order);

await \_context.SaveChangesAsync();

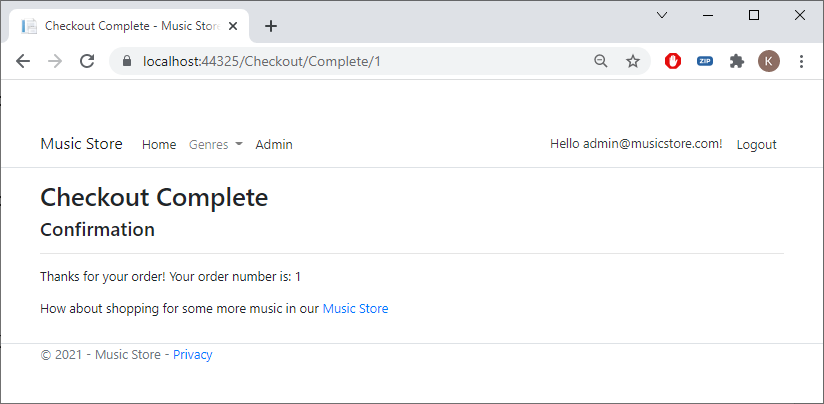
return RedirectToAction("Complete", new { id = order.OrderID });

}

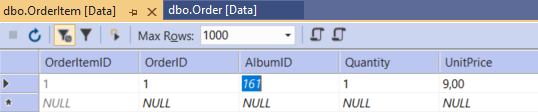
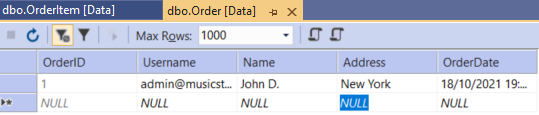
return View(order);

}

Now add the last method *Complete* with corresponding view so that you get the following confirmation after your order has been registered:



Test everything and check in the database whether both the order and the order lines (from the shopping cart) have been added:



Well done!