

When using [discounts](#) or [free shipping offers](#) on your MVC site, you may want to display the discounted prices or display banners that show how much more customers need to spend to obtain free shipping. This page describes how to use properties and methods of the [Kentico.Ecommerce integration package](#), used for [building MVC sites](#), to display such information.

Supported discounts

The **Kentico.Ecommerce** integration package currently supports [catalog discounts](#), free shipping offers and [product coupons](#) on MVC sites.

Other types of discounts also work. However, the integration package does not offer API for displaying information about those discounts.

Displaying catalog discounts

In a controller's action, use the **CalculatePrice** method from the **PricingService** in the **Kentico.Ecommerce** integration package. The method's calculations automatically include any catalog discounts that apply for the current customer.

The method returns a **ProductPrice** object that contains the **Discount** property. In the *Discount* property, you can find the amount reduced by catalog discounts.

```
// Initializes the needed services
ShoppingService shoppingService = new ShoppingService();
PricingService pricingService = new PricingService();

// Gets the current shopping cart
ShoppingCart shoppingCart = shoppingService.GetCurrentShoppingCart();

// Calculates prices for the specified product
ProductPrice price = pricingService.CalculatePrice(sku, shoppingCart);

// Gets the catalog discount
decimal catalogDiscount = price.Discount;
```



We recommend using a [dependency injection container](#) to initialize service instances. When configuring the lifetime scope for *ShoppingService* and *PricingService*, create a shared instance (singleton) for all requests.

You can then use the discount in your views.

Displaying free shipping offers

In a controller's action, use the **CalculateRemainingAmountForFreeShipping** method of the **PricingService** in the **Kentico.Ecommerce** integration package.

The method returns the remaining amount of money for free shipping. If the cart is null, if there is no valid discount, or if free shipping was already applied, the method returns 0.

The *CalculateRemainingAmountForFreeShipping* method has a shopping cart instance as a parameter. The *ShoppingCart* cart parameter specifies a shopping cart for which the remaining amount is calculated.



```
// Initializes the needed services
ShoppingService shoppingService = new ShoppingService();
PricingService pricingService = new PricingService();

// Gets the current shopping cart
ShoppingCart shoppingCart = shoppingService.GetCurrentShoppingCart();

// Gets the remaining amount for free shipping
decimal remainingFreeShipping = pricingService.
CalculateRemainingAmountForFreeShipping(shoppingCart);
```



We recommend using a [dependency injection container](#) to initialize service instances. When configuring the lifetime scope for *ShoppingService* and *PricingService*, create a shared instance (singleton) for all requests.

You can then use the remaining amount in your views.

Displaying product coupons

See [Adding coupon code fields to MVC checkout processes](#) to learn more about adding a coupon code field to your website.