

The Kentico E-commerce Solution can process coupon codes for several types of [discounts and offers](#):

- [Product coupons](#)
- [Gift cards](#)
- [Order discounts](#)
- [Buy X Get Y discounts](#)
- [Free shipping offers](#)

[Customers](#) who obtained a coupon code are eligible to receive a discount or redeem a gift card (based on the configuration of the discount or gift card).

When building a [checkout process on an MVC site](#), you may want to add an input text box that allows customers to enter coupon codes. The system validates whether a specified coupon code provides a discount. In Kentico MVC projects, the [Kentico Ecommerce integration package](#) provides an API for working with coupon codes.

To add a coupon code input to your MVC site:

1. Open your MVC project in Visual Studio.
2. Edit the controller class that you use for your checkout process (see [Using a shopping cart on MVC sites](#)).
3. Implement two POST actions – one for adding coupon codes to the current shopping card and another for removing coupon codes.
 - To add and validate coupon codes, call the **AddCouponCode** method of the *ShoppingCart* object representing the current shopping cart.



The **AddCouponCode** method returns a *true* value if the specified coupon code provides a valid discount for the current shopping cart and customer.

- To remove coupon codes, call the **RemoveCouponCode** method of the *ShoppingCart* object.



```
/// <summary>
/// Adds the specified coupon code to the shopping cart.
/// </summary>
[HttpPost]
public ActionResult AddCouponCode(string couponCode)
{
    // Gets the current user's shopping cart
    ShoppingCart currentCart = shoppingService.
GetCurrentShoppingCart();

    // Adds the coupon code to the shopping cart
    // The 'ShoppingCart.AddCouponCode' method automatically
recalculates the shopping cart,
    // so there is no need to call the Evaluate() method after
    if ((couponCode == "") || !currentCart.AddCouponCode
(couponCode))
    {
        // Adds an error message to the model state if the entered
coupon code is not valid
        ModelState.AddModelError("CouponCodeError", "The entered
coupon code is not valid.");
    }

    // Initializes the shopping cart model
    ShoppingCartViewModel model = new ShoppingCartViewModel
    {
        // Assigns the current shopping cart to the model
        Cart = currentCart,
        RemainingAmountForFreeShipping = pricingService.
CalculateRemainingAmountForFreeShipping(currentCart)
    };

    // Displays the shopping cart
    return View("ShoppingCart", model);
}

/// <summary>
/// Removes the specified coupon code from the shopping cart.
/// </summary>
[HttpPost]
public ActionResult RemoveCouponCode(string couponCode)
{
    // Gets the current user's shopping cart
    ShoppingCart currentCart = shoppingService.
GetCurrentShoppingCart();

    // Removes the specified coupon code
    // The 'ShoppingCart.RemoveCouponCode' method automatically
recalculates the shopping cart,
    // so there is no need to call the Evaluate() method after
    currentCart.RemoveCouponCode(couponCode);

    // Displays the shopping cart
    return RedirectToAction("ShoppingCart");
}
```

4. Perform the following changes in your shopping cart view:



- Create a form with a coupon code text input and an apply button (send POST requests to the action that adds coupon codes).
- Display applied coupon codes from the **AppliedCouponCodes** collection of the current shopping cart object.
- For each applied coupon code, add a form with a single button that allows customers to remove the given coupon code (send POST requests to the action that removes coupon codes).

```
@* Coupon code text box and apply button *@
@using (Html.BeginForm("AddCouponCode", "Checkout", FormMethod.Post))
{
    <input type="text" name="CouponCode" />
    <input type="submit" value="Apply" />
    @Html.ValidationMessage("CouponCodeError")
}
*@ List of applied coupon codes with remove buttons *@
<ul>
    *@ Loops through all applied coupon codes *@
    @foreach (string couponCode in Model.Cart.AppliedCouponCodes)
    {
        <li>
            @couponCode

            @using (Html.BeginForm("RemoveCouponCode", "Checkout", FormMethod.
Post))
            {
                <input type="hidden" name="CouponCode" value="@couponCode" />
                <input type="submit" value="Remove" />
            }
        </li>
    }
</ul>
```

The system now accepts coupon codes typed during the checkout process and customers can get their discounts.