

Ucommerce Master Class

---

# Browse + Checkout Handouts

## Table of Contents

<b>Preparing for Master Class</b>	<b>4</b>
<b>Setup and pre-requirements</b>	<b>4</b>
Preinstalled software before the Master Class	4
Getting the Accelerator for Kentico	4
Setup a new Kentico Site prepared for the master class	5
Getting the Master Class Visual Studio Solution	5
<b>Solution overview</b>	<b>5</b>
The solution projects	6
The deploy tool	6
<b>Exercise 020: Browse - Category Navigation</b>	<b>7</b>
Intro	7
Relevant APIs	7
Hands-on	8
Bonus	8
<b>Exercise 030: Browse - Category Detail Page</b>	<b>9</b>
Intro	9
Relevant APIs	9
Hands-on	10
Bonus	10
<b>Exercise 040: Browse - Product Listing (Category Detail Page)</b>	<b>11</b>
Intro	11
Relevant APIs	11
Hands-on	12
Bonus	12
<b>Exercise 050: Browse - Prices and Simple Discounts (Product Listing)</b>	<b>13</b>
Intro	13
Relevant APIs	13
Hands-on	14
Bonus	14
<b>Exercise 060: Browse - Product Detail</b>	<b>15</b>
Intro	15
Relevant Concepts and APIs	15
Hands-on	16

Bonus	16
<b>Exercise 070: Browse - Add to Basket</b>	<b>17</b>
Intro	17
Relevant Concepts and APIs	17
Hands-on	17
Bonus 1	17
Bonus 2	17
<b>Exercise 210: Checkout - View Basket</b>	<b>18</b>
Intro	18
Relevant APIs	18
Hands-on	19
Relevant APIs	20
Hands-on	20
Bonus	20
<b>Exercise 220: Checkout - Billing/Shipping Information</b>	<b>21</b>
Intro	21
Relevant Concepts and APIs	21
<b>Exercise 230: Checkout - Shipping Method</b>	<b>22</b>
Intro	22
Relevant Concepts and APIs	22
Hands-on	23
<b>Exercise 235: Checkout - Update Selected Shipping Method</b>	<b>24</b>
Relevant APIs	24
Hands-on	24
<b>Exercise 240: Checkout - Payment Method</b>	<b>25</b>
Intro	25
Relevant Concepts and APIs	25
Hands-on	25
<b>Exercise 245: Checkout - Update Selected Payment Method</b>	<b>26</b>
Relevant APIs	26
Hands-on	26
<b>Exercise 250: Checkout - Order Preview</b>	<b>27</b>
Intro	27
Relevant Concepts and APIs	27
Hands-on	28

## Preparing for Master Class

Now that you've signed up for the Kentico + Ucommerce master class there are a few things you need to do in order to come prepared for the master class. Please read the following sections before the master class starts.

## Setup and pre-requirements

In this section you'll find a list of required software to attend the course. This section will help you get everything setup, so you're ready when the master class begins.

### Problems?

If you have any issues getting up and running please write to [support@ucommerce.net](mailto:support@ucommerce.net) and we will try to provide the help needed. Let us know that you're about to attend a Ucommerce for Kentico master class.

### Preinstalled software before the Master Class

These are the requirements for downloaded/installed software, so you can follow along on the master class:

- Visual Studio (Our solution is set to .NET 4.6)
- SQL Server (Express or Standard. We don't support SQL Server Compact).
- SQL Server Management Studio
- IIS 7
- Latest version of the Accelerator for Kentico installed (see below)
- The Master Class Visual Studio Solution (see below)

### Getting the Accelerator for Kentico

If you go to <http://www.ucommerce.net/download> you'll be asked to fill out a form to get access to our download page. Click on "Accelerator for Kentico" to download the accelerator. It's a rather big file as it includes a full Kentico 10 installation.

On the site from which you download the accelerator there's a link to a setup guide of the accelerator ("how to setup the accelerator on Kentico"). In there you can read the steps needed to setup the accelerator.

## Setup a new Kentico Site prepared for the master class

Once the accelerator is set up, you'll need to "stop" the site in Kentico, as we'll be building our own small webshop during the master class. We'll use the accelerator as inspiration and make use of the categories and products that already ships with it.

1. Go to the backend of Kentico, go to Sites, and choose to stop "Avenue Clothing".
2. Click "New site wizard".
3. On step 1 of the wizard, select "Create a new site using a wizard".
4. On step 2 we'll name the site "Masterclass". Domain name should be set to "localhost", and leave culture to default.
5. On step 3, leave all default settings.
6. On step 4, you'll have to wait until the importing process has finished.
7. On step 5, make sure to select "Blank master page" and click Next and then "Finish".
8. We now have our Master Class site up and running.

## Getting the Master Class Visual Studio Solution

During the master class we'll be working in a smaller Visual Studio solution than the one that ships with the accelerator. This helps us keep the accelerator and the master class material separated.

There's a script in the Master Class Visual Studio solution, which will make sure our changes is pushed to the directory structure of the site we created in the steps above. More about that later.

Download the Master Class Visual Studio Solution from the following link:  
[https://drive.google.com/open?id=17lJuXPD\\_miZXzbniaLxlt0zB8E2h4Tb7](https://drive.google.com/open?id=17lJuXPD_miZXzbniaLxlt0zB8E2h4Tb7)

## Solution overview

Now that you've installed a clean version of Kentico + Ucommerce + Avenue Clothing store and opened the Master Class solution you're ready to build your brand new store using the snippets provided in the solution. In this section we'll cover what's in the box before we start coding away. The solution is created so it resembles a "real life" Ucommerce project. Of course best practices varies and is a matter of personal preference so it may be slightly different from how you would choose to do it.

The solution will only contain the extensions we're building along with the configuration files we're modifying.

### The solution projects

The solution is built up from 3 different projects:

- UCommerce.MasterClass.BusinessLogic
  - This will be used on the deep-dive part of the course primarily to create different extensions we will deploy to the website but also on the integration project when exploring the query APIs.
- UCommerce.MasterClass.Integration
  - Console Application that can access the APIs and the Ucommerce data store. We will use this on the deep-dive part of the course when we explore the query APIs. This project has an assembly reference for the BusinessLogic project.
- UCommerce.MasterClass.Website
  - project where all resources for the site exists including:
    - Ucommerce configuration files to register custom components used for deep-dive
    - Webforms Pages and User controls to scaffold the webshop
    - Few javascript files and styles sheets including Bootstrap for bells and whistles 😊
  - The purpose of the project is to push extensions and modifications to the site.

### The deploy tool

The deploy tool is used to push over all binaries, javascript files, style sheets, configuration files and more. The deploy tool is created as a powershell script and will run every time you build the Website project. It's set up as a post-build event for the website project.

You need to be sure the deploy tool is copying files to the right folder, so you need to update a variable in the script. Open the provided Master Class Solution. **Make sure that visual studio runs in administrator mode.**

Modify the `$website_root` variable found in "Deploy-Local.ps1" under the Deploy folder in top of the solution explorer.

The variable needs to point to the root of your website (that's the CMS-folder of the site, where the web.config resides).

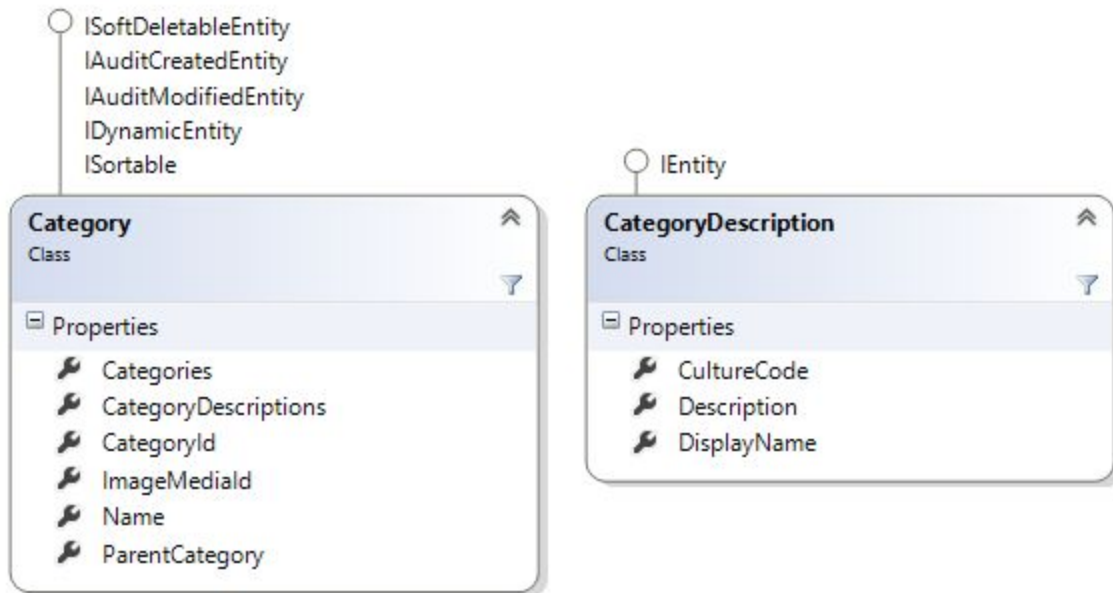
That's it! You're now ready to become a Ucommerce Rockstar 😊

## Exercise 020: Browse - Category Navigation

### Intro

Build a category listing for overall navigation of your store. You will gain knowledge of APIs relevant to loading categories and their related information along with an over-all understand of how to navigate the catalog structure.

### Relevant APIs



### UCommerce.Api

```
CatalogLibrary.GetRootCategories(ProductCatalog)
CatalogLibrary.GetCategories(Category)
CatalogLibrary.GetNiceUrlForCategory() (optional)
```

### UCommerce.Extensions

```
CategoryExtensions.DisplayName()
```

### Hands-on

Find the "CategoryNavigation.ascx.cs" under the UserControls folder in the website project.

On Page\_Load we build and renders the usercontrol "CategoryNavigation.ascx" as requested with the following line in "Main.Master"

```
<uc:CategoryNavigation id="UCCategoryNavigation" runat="server" />
```

Find categories and sub categories using the CatalogLibrary and map them into the CategoryModel list

Map the categories recursively by adding sub categories into the model as well using CatalogLibrary to find the children.

Add link to the CategoryModel.Url that points to '/category?category=categoryId'

### Bonus

Try creating a multiple stores in Kentico. Create a new Kentico Site. Create a new store in Ucommerce. Setup a new catalog and new categories under the store as well. Assign different domains to each of the stores. Create new content structure in the Pages application for your new store.

Visit the new site. Does the category navigation change?

More information can be found on the documentation site:

<http://docs.ucommerce.net/ucommerce/v6.8/getting-started/catalog-foundation/catalog-structure.html>

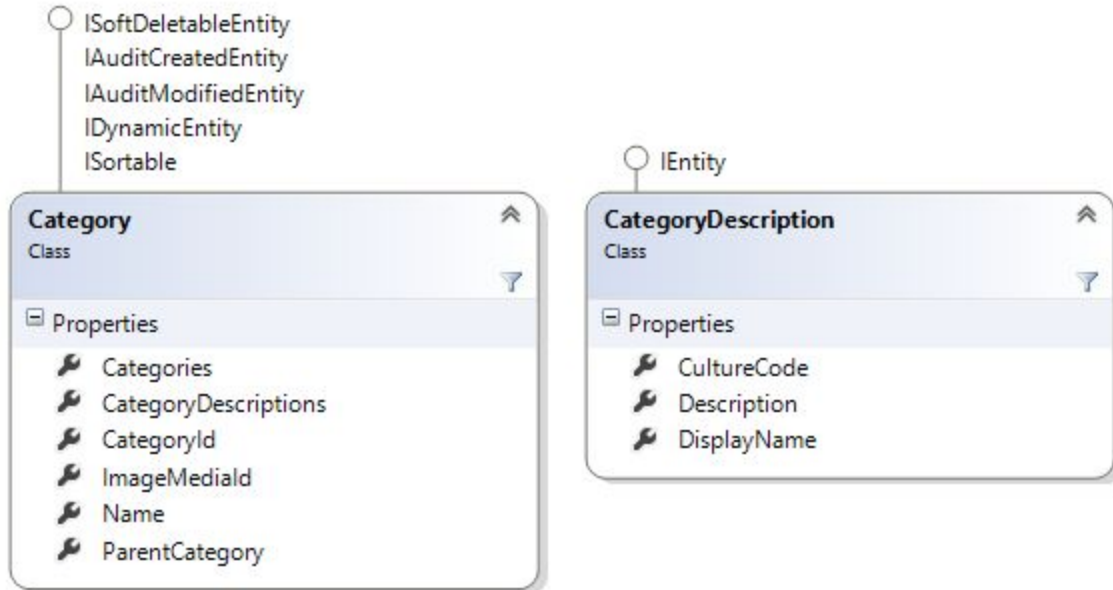


## Exercise 030: Browse - Category Detail Page

### Intro

Tease the contents of the category and get the customer excited.

### Relevant APIs



```
UCommerce.Runtime
    SiteContext.Current.CatalogContext
        CurrentCategory
```

```
UCommerce.EntitiesV2.Category
    .ImageId
```

```
UCommerce.Extensions
    Category.DynamicProperty()
    Category.DisplayName()
    Category.Description()
```

### Hands-on

Find the “Category.aspx.cs”.

The method “Page\_Load” needs to populate the CategoryProductModel, which is then used to bind data to the view Category.aspx.

Map “CurrentCategory” to the CategoryProductModel with

- Name
- Description

### Bonus

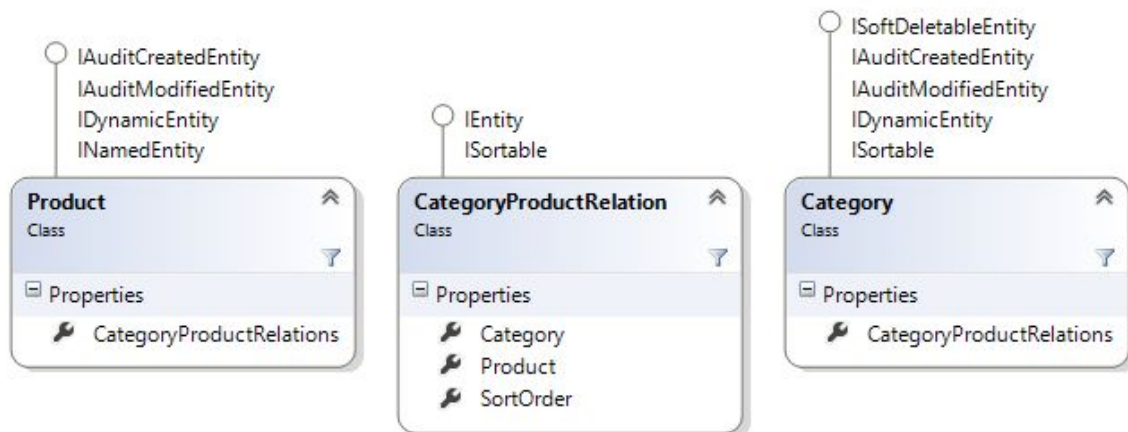
- Display category images using Kentico APIs.
  - Figure out if there’s a suitable property you can use, or you need to extend the CategoryProductModel with a property to hold the image.
  - You would need to make use of the ImageService.

## Exercise 040: Browse - Product Listing (Category Detail Page)

### Intro

Build a product listing based with products in a given category. You will gain knowledge of APIs relevant to loading products and categories efficiently from Ucommerce as well as dealing with prices and simple discounts.

### Relevant APIs



```
UCommerce.EntitiesV2.Product
    PrimaryImageMediaId
    ThumbnailImageMediaId
```

```
UCommerce.Runtime
    SiteContext.Current.CatalogContext
        CurrentCatalog
        CurrentCategory
```

```
UCommerce.Api
    CatalogLibrary.GetProducts(category)
```

```
UCommerce.Extensions
    Product.DynamicProperty()
    Product.DisplayName()
    Product.ShortDescription()
    Product.LongDescription()
```

```
UCommerce
    Money
```

### Hands-on

In `Category.aspx.cs` you need to map the `Products` property on the `CategoryProductModel` to hold the list of products in `CurrentCategory`.

### Bonus

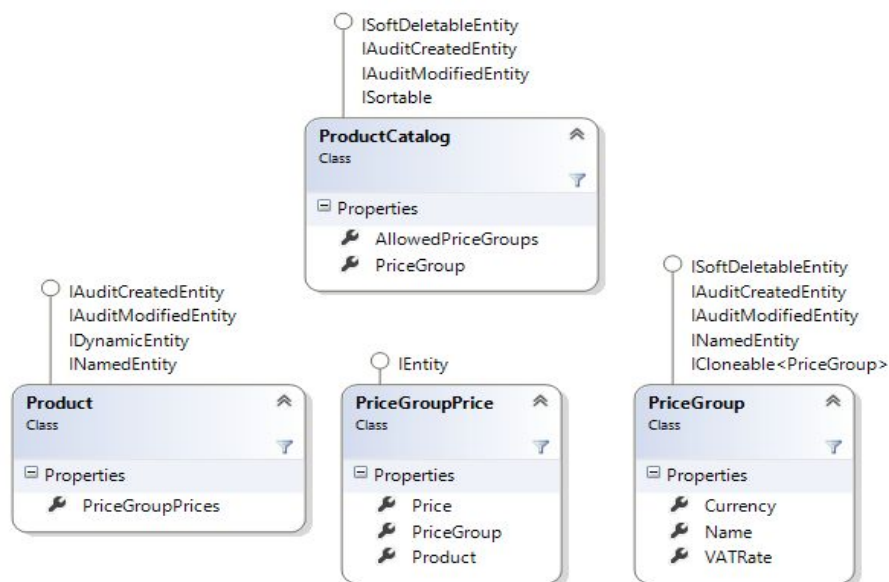
- Display product images using Kentico APIs.
  - Does the `ProductModel` contain enough fields to do so or do you need to add in extra fields?
  - Remember, to view images you should make use of `ImageService`.

## Exercise 050: Browse - Prices and Simple Discounts (Product Listing)

### Intro

Add price and tax information to your product pages. Learn how Ucommerce applies price information to your products based on your catalog configuration and how products can have multiple prices.

### Relevant APIs



### UCommerce.Api

```
CatalogLibrary.CalculatePrice(Product, ProductCatalog)
```

### PriceCalculation

```
YourPrice <-- Price incl simple discounts  
ListPrice  
Discount  
YourTax  
IsDiscounted
```

### Price

```
AmountExclTax  
Amount  
AmountInclTax
```

### UCommerce.EntitiesV2

Product  
ProductCatalog

### Hands-on

On the ProductModel you need to set the PriceCalculation coming from the API

### Bonus

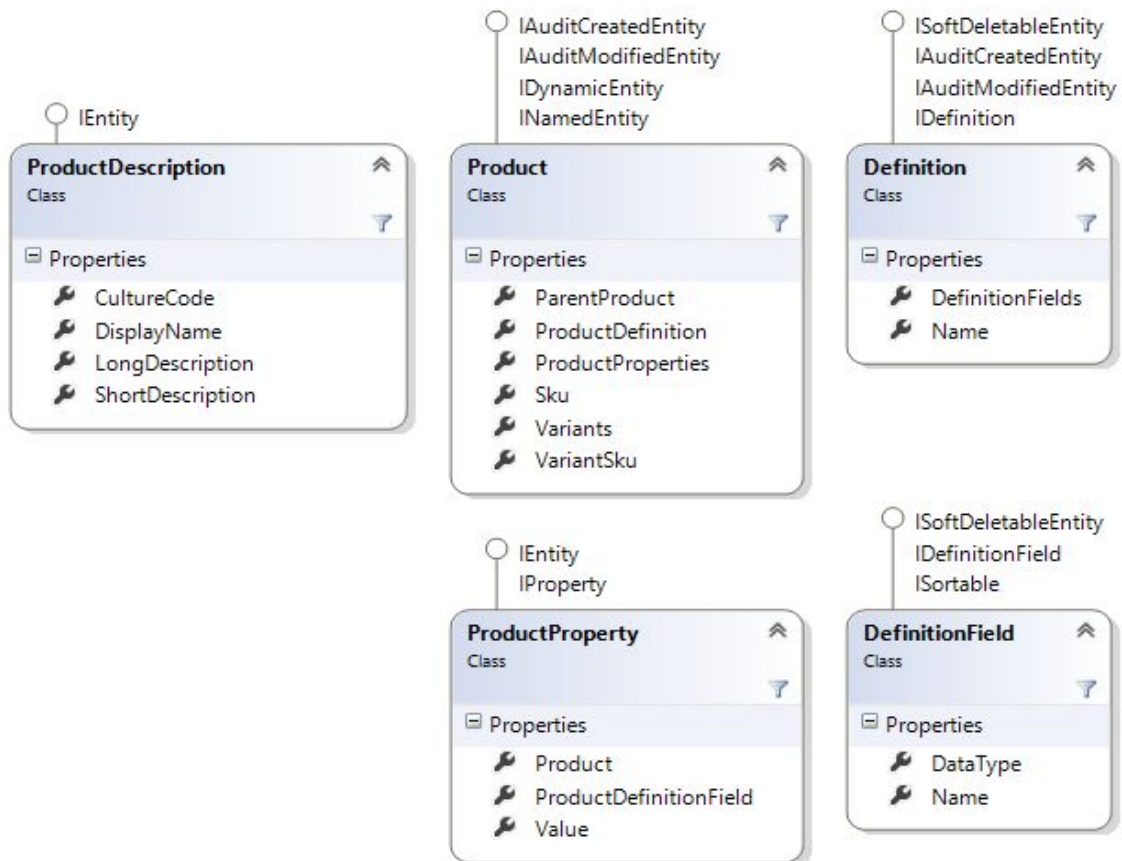
- Set up a unit discount and display the discount on the product listing.
- Add the original price with a dash through if the price is discounted

## Exercise 060: Browse - Product Detail

### Intro

Build a product detail page, which will delight customers using the server-side APIs. Learn how to display language specific content via dynamic properties. As a bonus we'll dive into the client-side APIs as well.

### Relevant Concepts and APIs



`UCommerce.Runtime`

```
SiteContext.Current.CatalogContext
    CurrentProduct
    CurrentCatalog
```

`UCommerce.Api`

```
CatalogLibrary.CalculatePrice(Product, Catalog)
CatalogLibrary.GetRelatedProducts(productId, relationType)
```

`UCommerce.Extensions`

```
DynamicEntityExtensions.DynamicProperty()
```

```
ProductExtensions.DisplayName()  
ProductExtensions.Description()
```

### Hands-on

Find the file "Product.aspx.cs".

The method "Page\_Load" loads data that will be binded to the page Product.aspx. We'll populate the ProductModel with data from Ucommerce.

Map "CurrentProduct" from the CatalogLibrary to the ProductModel with

- Name
- Description
- Sku
- VariantSku
- LongDescription
- Variants
- PriceCalculation

### Bonus

- Display product images using Kentico APIs (as earlier, remember to use ImageService).
- Explore related products on CatalogLibrary using the following line of code:
  - `CatalogLibrary.GetRelatedProducts(productId).SelectMany(x => x.Value)`



## Exercise 070: Browse - Add to Basket

### Intro

First step towards making an honest buck is getting customers to add items to the basket. Discover the server-side APIs to get it done and sneak a peek at the client-side APIs available in the Avenue Clothing Demo Store.

### Relevant Concepts and APIs

UCommerce.Api

```
TransactionLibrary.AddToBasket(  
    quantity,  
    sku,  
    variantSku = null,  
    addToExistingLine = true,  
    executeBasketPipeline = true,  
    catalogId = null)
```

### Hands-on

The sku and VariantSku from the page are posted back as form data to your "Product.aspx.cs". Use the API above to add the product to the basket.

### Bonus 1

Add another input field to your form, that takes a quantity. Modify AddToBasketButton\_OnClick to read that quantity and add it to the call to AddToBasket.

### Bonus 2

Use your javascript ninja-skills! Add a reference to JQuery to Main.Master and do an async AddToBasket call - so we'll not refresh the whole page. You need to collect both Sku, VariantSku and the Quantity.

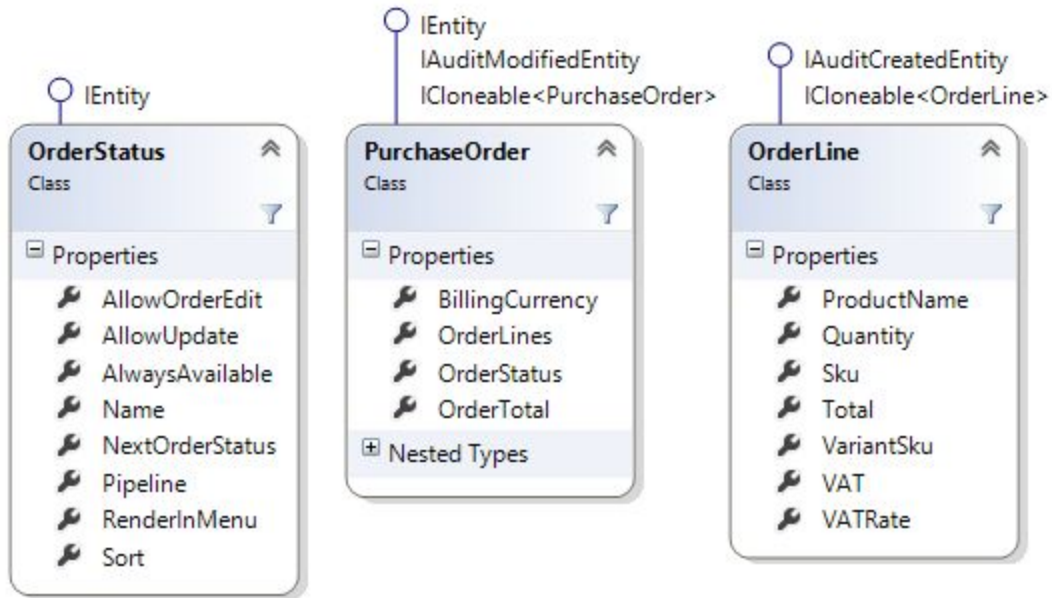
```
$.uCommerce.addToBasket(  
    {  
        sku: "",  
        variantSku: "",  
        quantity: 1,  
        addToExistingLine: true  
    },  
    onSuccess,  
    onError)
```

## Exercise 210: Checkout – View Basket

### Intro

Customers need an overview of what they're about to buy and also a way to manage the items they've put in their basket.

### Relevant APIs



```
UCommerce.Api  
    TransactionLibrary.GetBasket().PurchaseOrder
```

```
UCommerce.EntitiesV2  
    PurchaseOrder  
        OrderLines  
        Discounts  
    OrderLine  
    Discount
```

```
UCommerce  
    Money
```

### Hands-on

Navigate to the file “Basket.aspx.cs”

The method “Page\_Load” loads relevant data and populates the PurchaseOrderModel, which is then used to render Basket.aspx.

Map the customer’s basket into the PurchaseOrderModel by grabbing the order from the TransactionLibrary.

## Exercise 215: Checkout – Update Basket

### Relevant APIs

UCommerce.Api

```
TransactionLibrary.UpdateLineItem(orderLineId, quantity)
```

```
TransactionLibrary.ExecuteBasketPipeline()
```

```
MarketingLibrary.AddVoucherCode(voucherCode)
```

### Hands-on

When “UpdateBasketButton\_OnClick” is called on Basket.aspx.cs you need to update each line item with either the new quantity. If the user clicks Remove you’ll need to remove the order line in “RemoveOrderLineButton\_OnClick”.

You also need to execute the basket pipeline so the order is kept up-to-date. You can read more about pipelines here:

<http://docs.ucommerce.net/ucommerce/v6.8/getting-started/transaction-foundation/pipelines-explained.html>

### Bonus

Continue the javascript gig from before by updating the basket using a code snippet similar to the one below. You need to collect the orderLineId and the Quantity entered from the basket table.

```
$uCommerce.updateLineItem(  
    {  
        orderLineId: $(this).data("lineitemid"),  
        newQuantity: $(this).val()  
    },  
    function() {},  
    function() {}  
)
```

## Exercise 220: Checkout - Billing/Shipping Information

### Intro

When time comes to complete the order the customer need to supply their billing and shipping details so we know where to ship their items.

Multiple shipments supported (PurchaseOrder.Shipments).  
Order addresses are stored per order level (OrderAddress).  
Shared customer addresses stored per customer (Address).

### Relevant Concepts and APIs

UCommerce.Api

```
TransactionLibrary.GetBillingInformation(  
    firstName,  
    lastName,  
    emailAddress,  
    phoneNumber,  
    mobilePhoneNumber,  
    company,  
    line1,  
    line2,  
    postalCode,  
    city,  
    state,  
    attention, countryId)  
TransactionLibrary.GetShipmentInformation()  
TransactionLibrary.EditBillingInformation()  
TransactionLibrary.EditShipmentInformation()
```

UCommerce.EntitiesV2

```
PurchaseOrder  
    BillingAddress  
    Shipments
```

```
OrderLine  
    Shipment
```

```
OrderAddress
```

```
Shipment
```

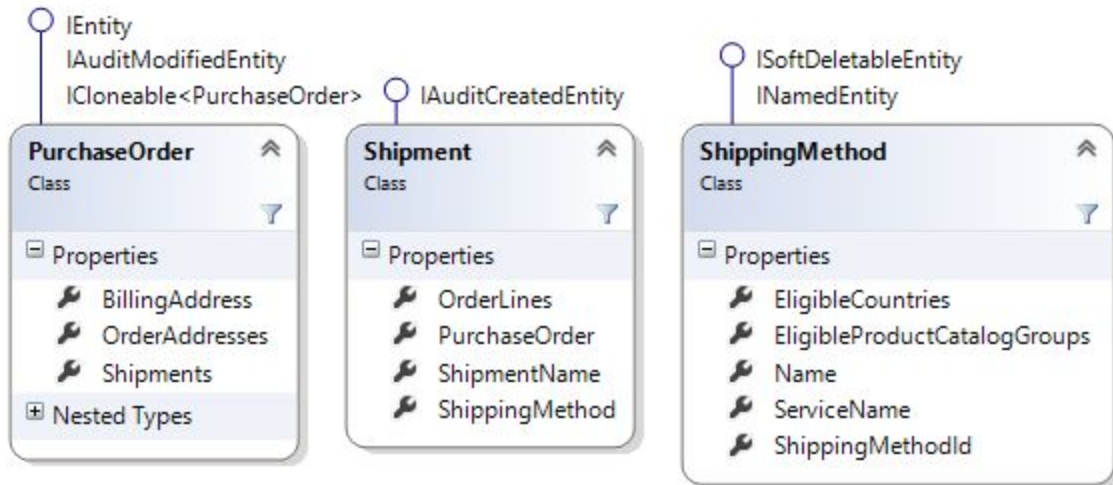
```
Country
```

## Exercise 230: Checkout - Shipping Method

### Intro

Display available shipping methods for the customer to select. Shipping methods can be filtered on store level and country level.

### Relevant Concepts and APIs



UCommerce.Api

```
TransactionLibrary.GetShippingInformation()
```

```
TransactionLibrary.GetShippingMethods(country = null)
```

```
TransactionLibrary.GetShippingMethod(shipmentName = null)
```

UCommerce.EntitiesV2

Shipment

ShippingMethod

```
GetPriceForCurrency(currency)
```

UCommerce

```
Money(amount, currency)
```

### Hands-on

Navigate to “Shipment.aspx.cs”.

The method Page\_Load loads the available shipment methods and populates Shipment.aspx.

## Exercise 235: Checkout – Update Selected Shipping Method

### Relevant APIs

UCommerce.Api

```
TransactionLibrary.CreateShipment(  
    shippingMehtodId,  
    addressName = null,  
    overwritingExisting = true)
```

```
TransactionLibrary.ExecuteBasketPipeline()
```

### Hands-on

In the eventhandler of the “Continue to payment”-button on the shipment page, you need to create a new shipment based on selected shippingmethod.

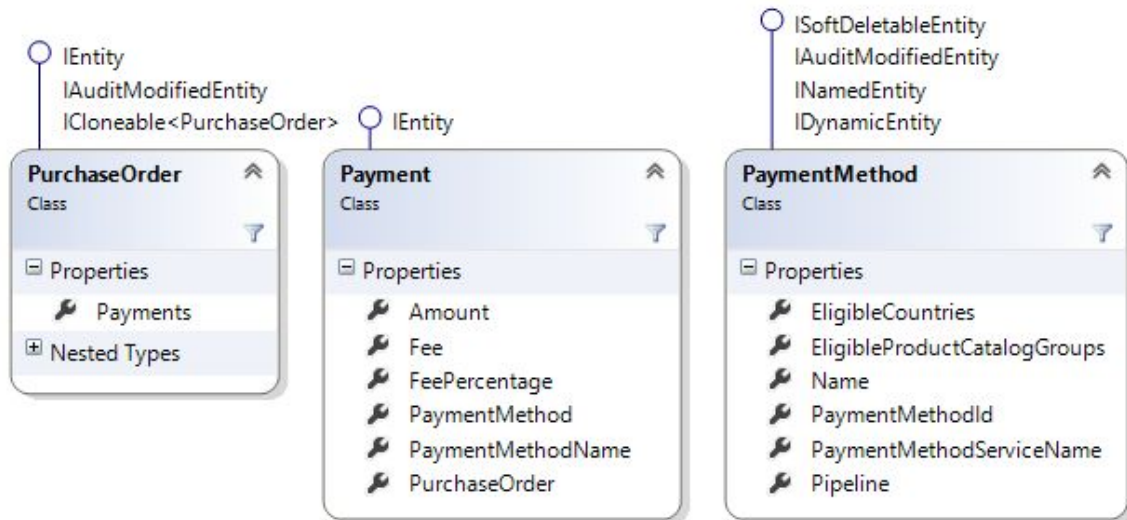


## Exercise 240: Checkout - Payment Method

### Intro

Display available payment methods for the customer to select. Can be filtered based on country.

### Relevant Concepts and APIs



`UCommerce.Api`

```
TransactionLibrary.GetBillingInformation()
```

```
TransactionLibrary.GetPaymentMethods(country = null)
```

`UCommerce.EntitiesV2`

```
PurchaseOrder
```

```
    Payments
```

```
Payment
```

```
PaymentMethod
```

`UCommerce`

```
Money(amount, currency)
```

### Hands-on

Navigate to "Payment.aspx.cs".

The method "Page\_Load" reads available payment methods from Ucommerce and populates the site. The payment methods should be filtered based on selected shipping country.

## Exercise 245: Checkout – Update Selected Payment Method

### Relevant APIs

```
TransactionLibrary.CreatePayment(  
    paymentMethodId,  
    amount = -1,  
    requestPayment = true,  
    overwriteExisting = true)
```

```
TransactionLibrary.ExecuteBasketPipeline()
```

### Hands-on

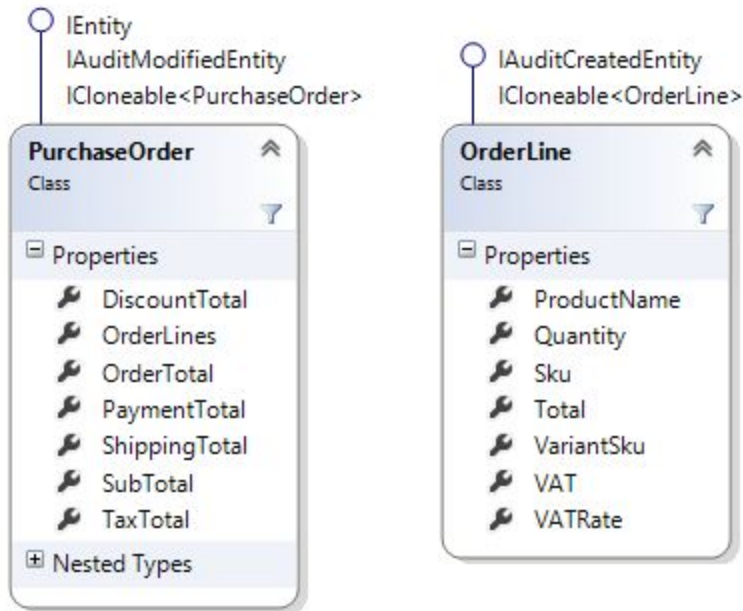
When the form is submitted to “SavePaymentAndGoToPreviewBtn\_OnClick” eventhandler, you need to update current payment by creating a new one that overrides the existing one. Use the TransactionLibrary to do so.

## Exercise 250: Checkout - Order Preview

### Intro

Many countries require online stores to show a complete picture of what the customer is buying and paying before completing an order. Discover the APIs and properties available to help you in this.

### Relevant Concepts and APIs



### UCommerce.Api

```
TransactionLibrary.GetBasket()
TransactionLibrary.RequestPayments()
```

### UCommerce.EntitiesV2

```
PurchaseOrder
    SubTotal
    TaxTotal
    Discount (discounts applied to the order itself)
    DiscountTotal (all discounts applied at any level)
    PaymentTotal
    ShippingTotal

    OrderLines
    Shipments
    Payments
```

OrderLine

    Sku  
    VariantSku  
    Quantity  
    Price  
    Discount  
    VAT  
    Total

Shipment

Payment

Discount

UCommerce

    Money

### Hands-on

Navigate to “Preview.aspx.cs”.

Map the basket into the PurchaseOrderModel.

When ready to purchase, the user will click checkout. This will cause the event “ConfirmAndContinueToPaymentBtn\_OnClick” to be fired. In here you’ll need to use TransactionLibrary to fulfill the payment.