Upgrade of Payment Provider to v4

Introduction

This document explains how to upgrade a NBStore v3 payment provider to the OpenStore v4 solution.

Before you start you might want to consider starting from a new VS template as explain in this document

https://github.com/openstore-ecommerce/Developer-Documents/blob/master/Create-Payment-Provider.pdf

I found this to take much the same time and I think offers cleaner result. But giving this document a quick read will still help you to understand the changes.

Begin

The NSB v4 payment module have been upgraded to use razor and ajax. This is NOT backward compatible with the old tag token system which was used before. This document explains the steps needed to upgrade a payment provider. Because payment providers can be different from each other, take this document and a basic guide to the conversion. You will probably need to expand on this guide to match your payment provider.

Keep in mind the quickest way to upgrade might be to create a new project and copy your code into it. There is VS Template available which will give you a starting structure if you want to do that.

You can use the NBrightPayBox or NBrightSystemPay projects as examples.

https://github.com/nbrightproject/NBrightPayBox

https://github.com/nbrightproject/NBrightSystemPay

Ajax Provider

The new method uses ajax to trigger the payment, you need to create an interface to the AjaxInterface and put it into you project

Example:
https://github.com/nbrightproject/NBrightPayBox/blob/master/Components/AjaxProvider.cs

Add a reference to the “RazorEngine.dll” in the DNN bin folder.

Payment.ascx

The “Payment.ascx” has been altered to ajax, so you need to change the code.

Example:

https://github.com/nbrightproject/NBrightPayBox/blob/master/Payment.ascx

The “Payment.ascx.cs” has been altered to ajax, so you need to change the code.

Example:

https://github.com/nbrightproject/NBrightPayBox/blob/master/Payment.ascx.cs

Settings.html

The file “settings.html” needs to be converted to razor, rename the file to “settingsfields.cshtml” and then edit the tag tokens to be razor tokens. Then created a “settings.cshtml” file which injects the “settingsfields.cshtml”

Some fields have can be added to deal with messages, like in the example.

Example:

https://github.com/nbrightproject/NBrightPayBox/blob/master/Themes/config/default/settings.cshtml

https://github.com/nbrightproject/NBrightPayBox/blob/master/Themes/config/default/settingsfields.cshtml

NOTE: the JS file will need to be added to the “settings.cshtml”

Payment.ascx.resx

If you have a resx file called “Payment.ascx.resx”, think about renaming it to stop any clashes.

Create JS file

A JS file will need to be created to work with ajax. Created a “js” folder under /theme/config and create/copy the required js file.

Change the ajax commands to match the AjaxProvider.cs.

Example:

https://github.com/nbrightproject/NBrightPayBox/blob/master/Themes/config/js/paybox.js

NOTE: update “settings.cshtml”

methodselection.html

Rename “methodselection.html” to “methodselection.cshtml” and convert to razor.

Example:

https://github.com/nbrightproject/NBrightPayBox/blob/master/Themes/config/default/methodselection.cshtml

Change the provider function “GetTemplate” to use this new cshtml template.

Example:

https://github.com/nbrightproject/NBrightPayBox/blob/master/Components/PaymentProvider.cs

payment_ok.cshtml and payment_fail.cshtml

These 2 templates are the default templates for the payment return message. It's optional if you want to overwrite these by placing a copy of the template into your payment provider. (It is recommended you do)

If you want to use the default then make the provider method "`ProcessPaymentReturn`" return empty string. If you want to use your payment provider templates return the payment provider template.

Example:

https://github.com/nbrightproject/NBrightPayBox/blob/master/Components/PaymentProvider.cs