Adyen Hybris V5 Plugin Implementation Guide

**Version 1.0**

# Revisions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Description | Author | Date | |
| 1.0 | Initial Draft | Kenneth Zhou | 2015-08-12 |
| 1.1 | Peer Review by Andy Cooper | Kenneth Zhou | 2015-08-14 |
| 1.2 | ADY-126 The order number should be the same as the cart number | Kenneth Zhou | 2015-11-09 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[Revisions 2](#_Toc428366411)

[Table of Contents 2](#_Toc428366412)

[Introduction 3](#_Toc428366413)

[Fundamental Concepts 4](#_Toc428366414)

[API with CSE 4](#_Toc428366415)

[HPP 4](#_Toc428366416)

[Prerequisites 4](#_Toc428366417)

[Audience 5](#_Toc428366418)

[Supported Environment 5](#_Toc428366419)

[Configuration with Adyen 5](#_Toc428366420)

[Overview 5](#_Toc428366421)

[High Level Plugin Anatomy 5](#_Toc428366422)

[Installing the Plugin into Hybris 6](#_Toc428366423)

[For Hybris 5.1 6](#_Toc428366424)

[For Hybris 5.5 7](#_Toc428366425)

[URLs 8](#_Toc428366426)

[Detailed Plugin Design 9](#_Toc428366427)

[AdyenServices Extension 9](#_Toc428366428)

[Adyen Payment Service 9](#_Toc428366429)

[Adyen Service 10](#_Toc428366430)

[Ayden Integration Data Models 10](#_Toc428366431)

[Adyen Notification Event 10](#_Toc428366432)

[AdyenCommerceServices Extension 11](#_Toc428366433)

[Basic Authentication 11](#_Toc428366434)

[Accelerator Extensions 12](#_Toc428366435)

[adyenAddon 12](#_Toc428366436)

[adyencockpits 13](#_Toc428366437)

[adyenstorefront 13](#_Toc428366438)

[adyenfulfilmentprocess 13](#_Toc428366439)

[adyencore 14](#_Toc428366440)

[adyenfacades 14](#_Toc428366441)

[adyeninitialdata 15](#_Toc428366442)

[Customized Types 15](#_Toc428366443)

[Configurations 15](#_Toc428366444)

[Adyen Site Attribute 16](#_Toc428366445)

[Local Properties Configuration 16](#_Toc428366446)

[Plugin Business Process 17](#_Toc428366447)

[Plugin Checkout Process 17](#_Toc428366448)

[Integration Type Selection Flow 17](#_Toc428366449)

[Direct API Payment Flow 18](#_Toc428366450)

[HPP Payment Flow 18](#_Toc428366451)

[Notification Processing 18](#_Toc428366452)

[Backoffice Process Flows 19](#_Toc428366453)

[Considerations for an Existing Application 20](#_Toc428366454)

[Suggested Further Reading 20](#_Toc428366455)

# Introduction

Hybris is a java based eCommerce framework built on the Spring MVC framework. The purpose of providing a plugin for Adyen is to aid integration of the Adyen payment gateway into a hybris implementation. This document describes the Adyen Hybris plugin, how it’s built, and the fundamental concepts to consider when integrating with a Hybris eCommerce site.

Because Hybris is built on the Spring framework this makes it highly customizable and extensible. The plugin also utilizes this framework so can also easily be extended to add specific behavior if required.

## Fundamental Concepts

Before describing more about the plugin some fundamental concepts will be considered.

The extend module has been created with the intent of keeping the hybris application out of PCI scope. Both of the integration methods described in this section are include in that scope.

### API with CSE

The plugin has been developed to support API (direct) payment model with Adyen with Client-Side Encryption (CSE) to encrypt card data inside the client.

Adyen provides three ways to communicate with it JSON, Restful and SOAP. The plugin is using JSON to transmit data between a server and web application.

#### Recurring

The plugin is able to allow shopper to re-use the payment details which previously used and saved to Adyen by shoppers themselves. It is important to note that no card details are stored in hybris as payment methods. They are only stored within Adyen.

### HPP

The plugin has been developed to support HPP (Hosted Payment Pages) payment model.

The HPP payment model enables a site to devolve PCI considerations down to a third party (in this case Adyen) via HPP. Adyen host the pages that capture and process the card details, all that is required of the Hybris is to redirect the user to these pages at the relevant point in their checkout flow and then process the authorization response returned from the hosted payment pages.

# Prerequisites

This section describes the dependencies and prerequisite actions that are required before integration is attempted.

## Audience

The expected audience for this document are technical resources actively involved in implementing Adyen into Hybris – they should have reasonable understanding of Enterprise Edition Java, Hybris eCommerce framework and payment models.

## Supported Environment

The Hybris framework and accelerator plugin has been developed against version 5.1.1.7 and version 5.5.0.0 of Hybris.

The plugin has been developed against version 4.1.0 of Adyen API and HPP.

See:

<https://docs.adyen.com/display/TD/API+manual+changelog> and

<https://docs.adyen.com/display/TD/HPP+manual+changelog>

## Configuration with Adyen

Configuration with Adyen is essential before attempting to integrate the plugin with Hybris. The following steps need to be completed with Adyen:

* Account needs to be setup with Adyen.
* Creation of merchant account.
  + Notification URLs must be specified together with basic authentication credentials.
* Web Service user must be created for Direct API integration.
* Skin for payment pages must be created for HPP integration.

# Overview

The plugin has been designed to provide quick and easy access into the Adyen payment gateway from a Hybris application. It is expected that implementers of the plugin will be in varied stages of their own development cycle and so the plugin is modularized to try to make integration as painless as possible.

## High Level Plugin Anatomy

The plugin is separated into two basic component layers:

* Hybris platform extensions layer provides API integration and Notification handling with the Adyen functions through standard extensions to the Hybris platform.
* Hybris accelerator extensions layer provides a reference implementation of the platform extensions. This can be used as a starting point for integrating Adyen into a storefront either by using this as the basis for storefront development, or simply using this as a reference to see how the underlying code can be implemented.

## Installing the Plugin into Hybris

First ensure that the version of Hybris being used is supported for the plugin.

The plugin is supplied as a zip file with several Hybris projects inside. Take the following steps to include the full plugin into your Hybris application:

### For Hybris 5.1

1. Unzip Hybris 5.1.1.7 to ${HYBRIS\_HOME} directory.
2. Run setantenv.bat (.sh when using Unix/Linux) from ${HYBRIS\_HOME}/bin/platform
3. Build Hybris by running “ant clean all” command from ${HYBRIS\_HOME}/bin/platform.
4. Unzip the plugin zip file.
5. Create ${HYBRIS\_HOME}/bin/custom
6. Create adyen directory under ${HYBRIS\_HOME}/bin/custom
7. Copy below extensions folders to the ${HYBRIS\_HOME}/bin/custom/adyen directory.
   * extensions/adyen5\_1/adyenAddon
   * extensions/adyen5\_1/adyencockpits
   * extensions/adyen5\_1/adyencore
   * extensions/adyen5\_1/adyenfacades
   * extensions/adyen5\_1/adyeninitialdata
   * extensions/adyen5\_1/adyenstorefront
   * extensions/adyen/adyenServices
   * extensions/adyen/adyenCommerceServices
   * extensions/adyen/adyenfulfilmentprocess
8. Copy the relevant <extension> elements from configuration/config/localextensions.5.1.xml to the file in your Hybris installation ${HYBRIS\_HOME}/config/localextensions.xml.
9. Comment out/remove existing yaccelerator extensions
   * <extension name="yacceleratorfulfilmentprocess" />
   * <extension name="yacceleratortest" />
   * <extension name="yacceleratorstorefront" />
   * <extension name="yacceleratorinitialdata" />
   * <extension name="yacceleratorcockpits" />
   * <extension name="yinstoreinitialdata" />
   * <extension name="yb2bacceleratorcore" />
   * <extension name="yb2bacceleratorfacades" />
   * <extension name="yb2bacceleratorinitialdata" />
   * <extension name="yb2bacceleratorstorefront" />
   * <extension name="yb2bacceleratortest" />
   * <extension name="yinstoreinitialdata" />
   * <extension name="b2ccheckoutaddon" />
10. Copy all configuration entries from configuration/config/local.properties to the file in your Hybris installation ${HYBRIS\_HOME}/config/local.properties.
11. Copy below tomcat configuration files into your Hybris installation and replace the default ones.
    * ${HYBRIS\_HOME}/config/tomcat/conf/server.xml
    * ${HYBRIS\_HOME}/config/tomat/conf/tomcat-users.xml
12. Re-build Hybris by running “ant clean all” command from ${HYBRIS\_HOME}/bin/platform.
13. Start Hybris server by running “hybrisserver.sh” from ${HYBRIS\_HOME}/bin/platform.
14. Initialize the platform from HAC.   
    Note: Full initialization will remove all data. If this is not acceptable an update can be done instead. However, there is data loaded during the initialization/update that is required for the integration to function correctly.

### For Hybris 5.5

1. Unzip Hybris 5.5.0.0 to ${HYBRIS\_HOME} directory.
2. Run setantenv.bat (.sh when using Unix/Linux) from ${HYBRIS\_HOME}/bin/platform
3. Build Hybris by running “ant clean all” command from ${HYBRIS\_HOME}/bin/platform.
4. Unzip the plugin zip file.
5. Create ${HYBRIS\_HOME}/bin/custom
6. Create adyen directory under ${HYBRIS\_HOME}/bin/custom
7. Copy below extensions folders to the ${HYBRIS\_HOME}/bin/custom/adyen directory.
   * extensions/adyen5\_5/adyenAddon
   * extensions/adyen5\_5/adyencockpits
   * extensions/adyen5\_5/adyencore
   * extensions/adyen5\_5/adyenfacades
   * extensions/adyen5\_5/adyeninitialdata
   * extensions/adyen5\_5/adyenstorefront
   * extensions/adyen/adyenServices
   * extensions/adyen/adyenCommerceServices
   * extensions/adyen/adyenfulfilmentprocess
8. Copy the relevant <extension> elements from configuration/config/localextensions.5.5.xml to the file in your Hybris installation ${HYBRIS\_HOME}/config/localextensions.xml.
9. Comment out/remove existing yaccelerator extensions from  
   ${HYBRIS\_HOME}/config/localextensions.xml.
   * <extension name="yacceleratorfulfilmentprocess" />
   * <extension name="yacceleratortest" />
   * <extension name="yacceleratorstorefront" />
   * <extension name="yacceleratorinitialdata" />
   * <extension name="yacceleratorcockpits" />
   * <extension name="yinstoreinitialdata" />
   * <extension name="yb2bacceleratorcore" />
   * <extension name="yb2bacceleratorfacades" />
   * <extension name="yb2bacceleratorinitialdata" />
   * <extension name="yb2bacceleratorstorefront" />
   * <extension name="yb2bacceleratortest" />
   * <extension name="yinstoreinitialdata" />
   * <extension name="b2ccheckoutaddon" />
10. Copy all configuration entries from configuration/config/local.properties to the file in your Hybris installation ${HYBRIS\_HOME}/config/local.properties.
11. Copy below tomcat configuration files into your Hybris installation and replace the default ones.
    * config/tomcat/conf/server.xml
    * config/tomat/conf/tomcat-users.xml
12. Re-build Hybris by running “ant clean all” command from ${HYBRIS\_HOME}/bin/platform.
13. Start Hybris server by running “hybrisserver.sh” from ${HYBRIS\_HOME}/bin/platform.
14. Initialize the platform from HAC.  
    Note: Full initialization will remove all data. If this is not acceptable an update can be done instead. However, there is data loaded during the initialization/update that is required for the integration to function correctly.

### URLs

If using the provided accelerator store and cscockpit they can be accessed on the following URLs:

Storefront: <https://apparel-uk.local:9002/adyenstorefront>

Customer Services: <https://apparel-uk.local:9002/cscockpit/login.zul>

In order for the storefront return URLs to resolve correctly on return from 3D secure pages or HPP the following must be added to your hosts file:

127.0.0.1 apparel-uk.local

# Detailed Plugin Design

Hybris Platform

B2C Accelerator Extensions

adyenServices extension

Adyen Payment Gateway

adyenCommerceServices extension

## AdyenServices Extension

AdyenService Extension is a Hybris standard extension which is designed to integrate with Adyen payment services and also provide the payment services to Hybris B2C accelerator store.

### Adyen Payment Service

#### com.adyen.services.impl.DefaultAdyenPaymentService

This Service is created as sub-class of DefaultPaymentServiceImpl to provide payment service for B2C accelerator store.

There are four important methods in this service

* authorize(BigDecimal, Currency, CartModel, AdyenPaymentInfoModel, String, String, String)  
  To provide authorize service for accelerator storefront and call Adyen service to invoke Adyen authorize JSON API.
* authorize3DSecure(CartModel, String, String, String, String, String)  
  To provide 3D secure authorize service for accelerator storefront and call Adyen service to invoke Adyen 3D secure authorize JSON API.
* cancelOrRefund(PaymentTransactionModel, String, CurrencyModel, Double)  
  To provide cancel payment service for accelerator cscockpit and call Adyen service to invoke Adyen cancel or refund JSON API.
* capture(PaymentTransactionModel, String)  
  To provide payment capture service for order fulfilment process and call Adyen service to invoke Adyen modification API.

### Adyen Service

#### com.adyen.services.integration.impl.DefaultAdyenService

This class is designed as the interface between Hybris and Adyen. It allows Hybris to make all authorization, capture, refund… calls through to the Adyen payment gateway.

There are four important methods in this service.

* requestRecurringPaymentDetails(AdyenListRecurringDetailsRequest)  
  Call Adyen listRecurringDetails API to retrieve saved payment details from Adyen.
* authorise(PaymentTransactionModel, AdyenPaymentRequest, boolean)  
  Call Adyen authorize or authorise3d API to make authorization for Order payment.
* cancelOrRefund(PaymentTransactionModel, AdyenModificationRequest)  
  Call Adyen cancelOrRefund and refund API to request payment cancel or refund.
* directory()  
  Call Adyen directory.shtml API to get HPP directories for HPP payment model.

### Ayden Integration Data Models

The Data Models used for integration with Adyen payment gateway API are created under package com.adyen.services.integration.data

All data object are designed based on Jackson JSON API. That makes it easy to serialize custom java objects to JSON string and deserialize JSON string back to java objects

### Adyen Notification Event

Adyen notification event is designed using Hybris standard event system.

When Hybris receive an Adyen notification request, an Adyen notification event will be published and Adyen notification event listener will react against the reception.

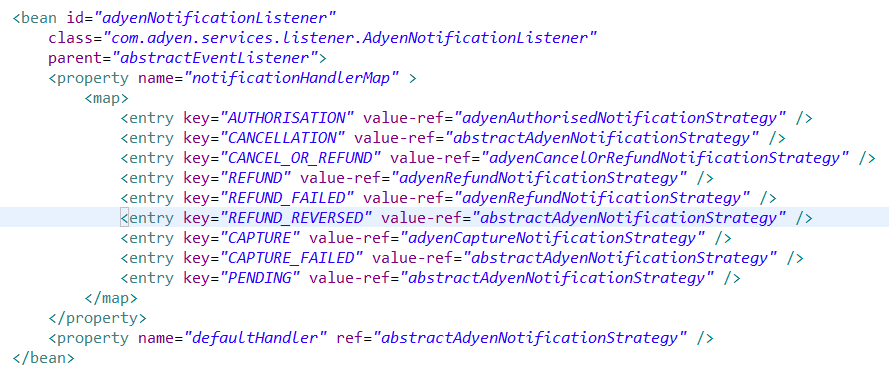
#### AdyenNotificationListener

Adyen Notification Listener will react against the notification event and distribute the event to corresponding notification handler to handle the notification.

notificationHandlerMap is configured in adyenservices-spring.xml file as a map.

The key of the map is Adyen notification event code.

The value of the map is configured to refer to notification handler instance.



Note: When Hybris get REFUND\_REVERSED, unsuccessful RRFUND and CAPTURE\_FAILED notification, the plugin will only create the payment transaction entry, but do nothing else. SI should customize their own business logic by implementing corresponding notification handler if it’s needed.

## AdyenCommerceServices Extension

AdyenCommerceServices Extension is a Hybris standard extension. It provides a Restful web service to handle the notification request from Adyen.

The restful API is implemented based on standard spring MVC controller.

com.adyen.commerceservices.rest.AdyenNotificationReceiver.onReceive(String) method will receive all post notification request from Adyen.

Adyen notification event will be published here and request body will be passed to notification handler along with event.

The method will always return “[accepted]” response to Adyen as required.

### Basic Authentication

When Adyen send notification to website, it requires the consumer provide the HTTP basic authentication. This requires a different approach to securing the service in hybris to the standard OAuth2 based Spring Security configuration of Omni Commerce Connect modules in hybris.

Below configuration changes provides the HTTP basic authentication support.

* /config/tomcat/conf/server.xml  
  UserDatabase data source is configured here for Basic Authentication.  
  UserDatabaseRealm is configured and using UserDatabase data source.
* /config/tomcat/conf/tomcat-users.xml  
  User and Password information are configured in this file.
* /adyenCommerceServices/web/webroot/WEB-INF/web.xml  
  security-constraint configuration is configured here to enable HTTP basic authentication for all coming requests to adyenCommerceServices extension.

## Accelerator Extensions

Accelerator Extensions includes adyenAddon, adyencockpits, adyencore, adyenfacades, adyeninitialdata and adyenstorefront extensions to provide B2C E-commerce service to the end shoppers and business users.

### adyenAddon

adyenAddon extensions contains most of the frontend changes.

The changes are made follow the standard Hybris implementation to provide Adyen integrated Multi-steps checkout flow.

#### Direct API payment

**com.adyen.storefront.controllers.pages.checkout.MultiStepCheckoutController**

This controller has been modified to support Adyen API and HPP payment flow.

It will check the Adyen configuration from current CMS site and prepare the response page accordingly. The listRecurringDetails API request will be sent from here if Use Saved Payment is enabled from HMC.

**/adyenAddon/acceleratoraddon/web/webroot/WEB-INF/views/desktop/pages/checkout/multi/addPaymentMethodPage.jsp**

The JSP file has been updated to display API relevant frontend elements such as credit card form with CSE, Boleto payment form etc.

#### HPP payment

**com.adyen.storefront.controllers.pages.checkout.MultiStepCheckoutController**

This controller has been modified to support Adyen HPP payment flow.

It will check the Adyen configuration from current CMS site and prepare the response page accordingly. The directory request will be sent from here if Lookup Directory is enabled from HMC.

**/adyenAddon/acceleratoraddon/web/webroot/WEB-INF/views/desktop/pages/checkout/multi/addPaymentMethodPage.jsp**

The JSP file has been updated to display HPP relevant frontend elements such as HPP post form and HPP redirect Pay button.

If Lookup Directory is enabled from HMC, The Adyen Payment Methods button list will also displayed here.

**Note:** For Hybris 5.5 the frontend control layer has been refactored. The controller for Hybris 5.5 is **com.adyen.storefront.controllers.pages.checkout.steps.PaymentMethodCheckoutStepController.**

### adyencockpits

The changes are made on adyencockpits to provide Adyen integrated cancel and refund functionalities.

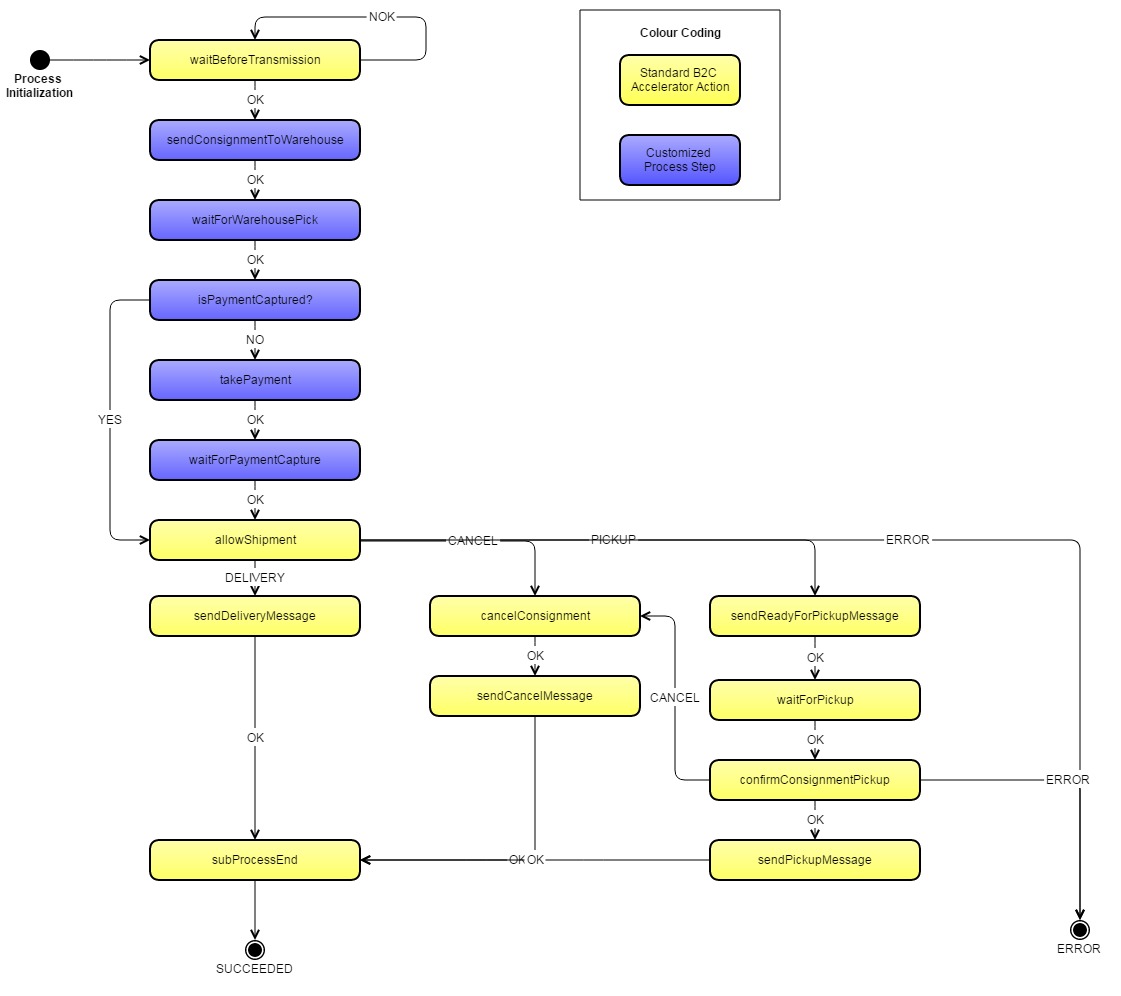
### adyenstorefront

Only some minor changes on adyenstorefront to show Adyen relevant payment information on Order Confirmation pages.

### adyenfulfilmentprocess

The adyenfulfilmentprocess extension extends the default Hybris yacceleratorfulfilmentprocess extension to customize the placeOrder business process. This whole place order (order-process.xml) business process can be customized, if needed.

The consignment process has been customized as below diagram shows as an example.



* The order process will be initialized after successful authorization notification sent to Hybris.
* The OOTB capture action is removed from main order process.
* New action is added for capture Adyen payment. com.adyen.fulfilmentprocess.actions.consignment.CapturePaymentAction
* After Warehouse pick the order, Hybris will send capture request to Adyen, the process will run into a Waiting node and wait for the capture notification.
* Once Hybris receives successful capture notification the consignment process will continue and order will be shipped.

### adyencore

Only some minor changes on adyencore extension to show Adyen relevant payment information on Order Confirmation email.

### adyenfacades

adyenfacades extension provide model to data conversion services to the store front.

No change made from accelerator store.

### adyeninitialdata

adyeninitialdata extension provide data initialization service during the system setup.

The data load into Hybris is the accelerator sample store apparel store.

## Customized Types

To support Adyen payment transaction some new types are created based on the Hybris standard type system.

All customized types are defined in adyenServices extension.

* PaymentTransactionType  
  Enum type to extend the OOTB payment transaction types.
* BoletoBrand  
  Enum type to define supported Boleto payment brands.
* OrderStatus  
  Enum type to extend the OOTB Order status.
* AdyenPaymentTransactionEntry  
  Extended payment transaction entry type for Adyen.
* AdyenPaymentInfo  
  Extended payment information type for Adyen.
* AdyenPaymentMethod  
  New type to define Adyen supported payment method and if capture is needed.
* CMSSite  
  This type is extended from OOTB CMSSite type. All Adyen configuration attributes are added here.

## Configurations

To support Adyen integration all configuration entry are configure in Hybris Site Attribute and local.properties.

Hybris provides a Multi-Site Ecommerce platform architecture. For those site specific configurations Site Attribute will serve better such as Merchant Code and Adyen HPP skin code.

The configuration can be changed by login to Hybris HMC and update on specific CMS site.

For those common configurations such as API endpoint URLs are configured in local.properties.

The configuration can be changed by login to Hybris HAC and update the configuration temporarily, or update the local.properties file and restart the Hybris server.

### Adyen Site Attribute

* adyenUseAPI - Determines whether use API integration.
  + adyenAPIAccount – API integration account ID.
  + adyenAPIPassword – API integration account password.
  + adyenCSEKey – Client side encryption public key.
  + adyenUseSavedPayment - Determines whether use saved payment.
  + adyenBoletoAvailable - Determines whether Boleto payment is available.
  + availableBoletoTypes – Available Boleto brands.
  + adyenEnableInstallment - Determines whether enable installment.
  + siteAdyenPaymentMethods – Adyen API supported payment methods.
* adyenUseHPP - Determines whether use HPP integration.
  + adyenUseDirectoryLookup – Determines whether lookup Adyen directory for HPP
  + adyenHmacKey – Adyen HPP payment HMAC key.
  + adyenSkinCode – Adyen HPP payment skin code.

### Local Properties Configuration

#Set PCI strategy to default to disable OOTB HOP and SOP checkout PCI options

site.pci.strategy=Default

#Adyen API integration base url, by default point to test env, should be updated when move to PROD.

integration.adyen.baseurl=https://pal-test.adyen.com/pal/servlet

#Adyen HPP integration base url, by default point to test env, should be updated when move to PROD.

integration.adyen.hpp.baseurl=https://test.adyen.com/hpp

#Adyen API HTTP client time out setting.

integration.adyen.http.readtimeout=10000

integration.adyen.http.connectiontimeout=10000

#Adyen listRecurringDetails API endpoint URL

integration.adyen.requestRecurringPaymentDetails.url=${integration.adyen.baseurl}/Recurring/v12/listRecurringDetails

#Adyen authorise API endpoint URL

integration.adyen.authorise.url=${integration.adyen.baseurl}/Payment/v12/authorise

#Adyen authorise3d API endpoint URL

integration.adyen.authorise.3d.url=${integration.adyen.baseurl}/Payment/v12/authorise3d

#Adyen cancelOrRefund API endpoint URL

integration.adyen.cancel.or.refund.url=${integration.adyen.baseurl}/Payment/v12/cancelOrRefund

#Adyen refund API endpoint URL

integration.adyen.refund.url=${integration.adyen.baseurl}/Payment/v12/refund

#Adyen capture API endpoint URL

integration.adyen.capture.url=${integration.adyen.baseurl}/Payment/v12/capture

#Adyen HPP pay request URL

integration.adyen.hpp.pay.url=${integration.adyen.hpp.baseurl}/pay.shtml

#Adyen HPP details request URL

integration.adyen.hpp.details.url=${integration.adyen.hpp.baseurl}/details.shtml

#Adyen HPP directory request URL

integration.adyen.hpp.directory.url=${integration.adyen.hpp.baseurl}/directory.shtml

#Hybris CSRF allowed url pattern with Adyen 3D secure return URL "/[^/]+(/[^?]\*)+(adyen-response)$"

csrf.allowed.url.patterns=/[^/]+(/[^?]\*)+(sop-response)$,/[^/]+(/[^?]\*)+(merchant\_callback)$,/[^/]+(/[^?]\*)+(hop-response)$,/[^/]+(/[^?]\*)+(adyen-response)$

# Plugin Business Process

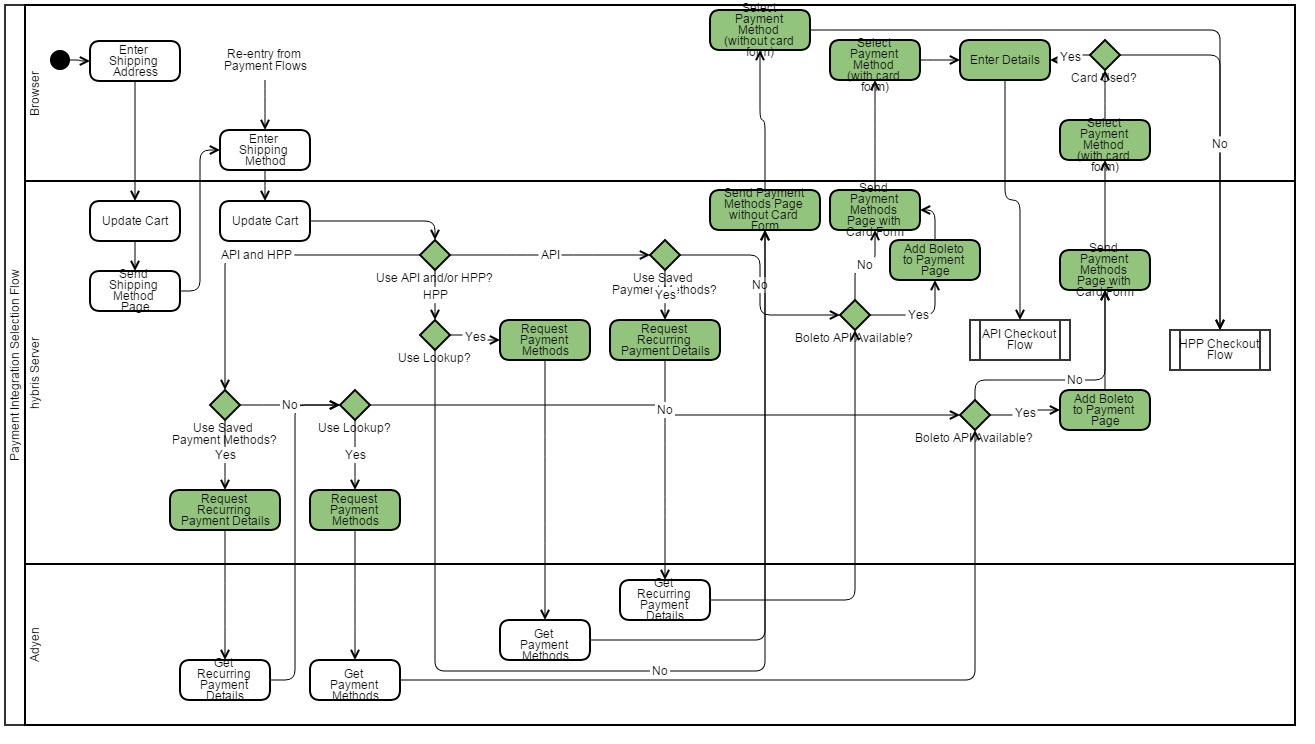
## Plugin Checkout Process

The plugin has made some customization again Hybris OOTB Multi-Steps checkout flow.

The plugin Multi-Steps checkout flow diagram are listed below.

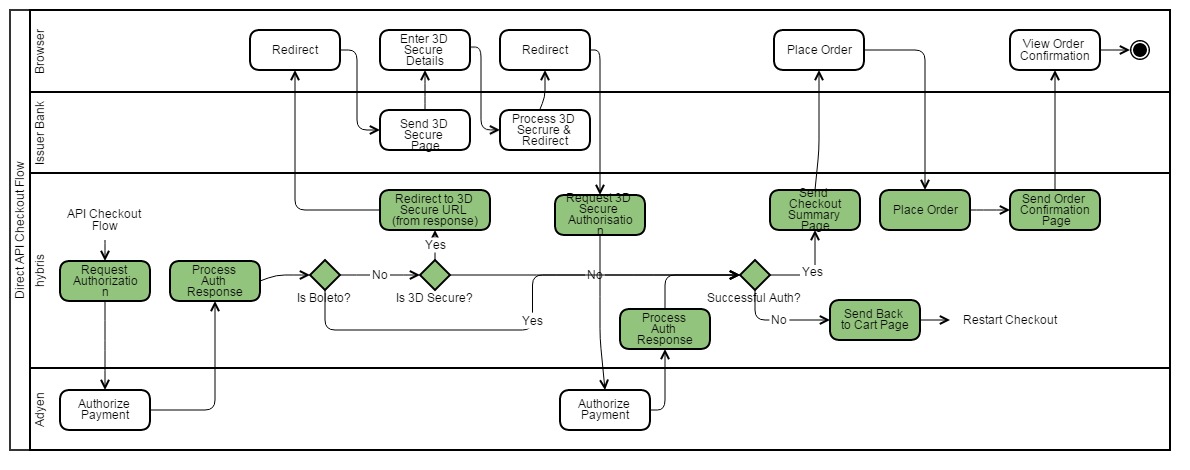
### Integration Type Selection Flow

According to the configured value of site attribute, Hybris will provide different payment details form for the shopper to continue the checkout process.



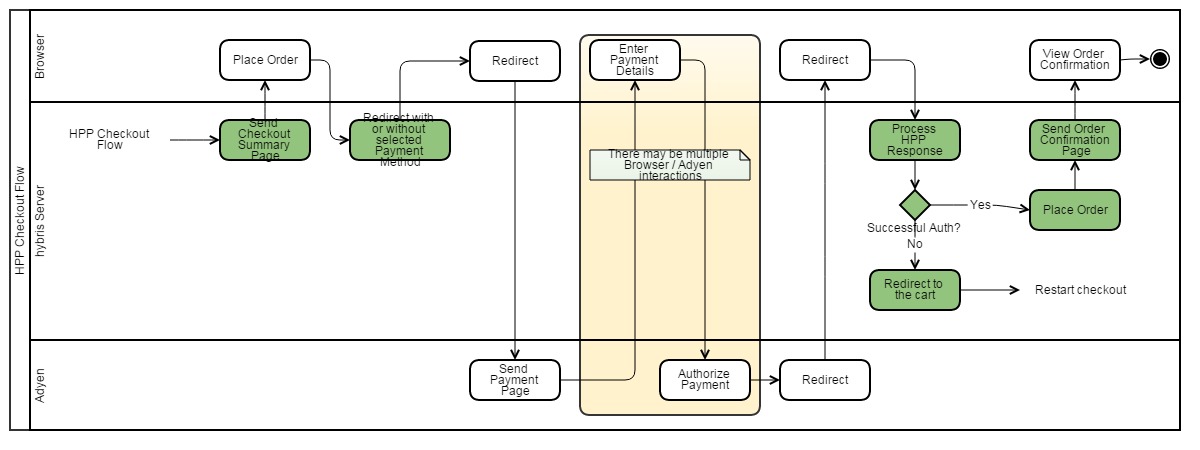
### Direct API Payment Flow

If Direct API payment model is selected, shopper will go through below checkout flow to authorize the payment from Adyen and then place order in Hybris.



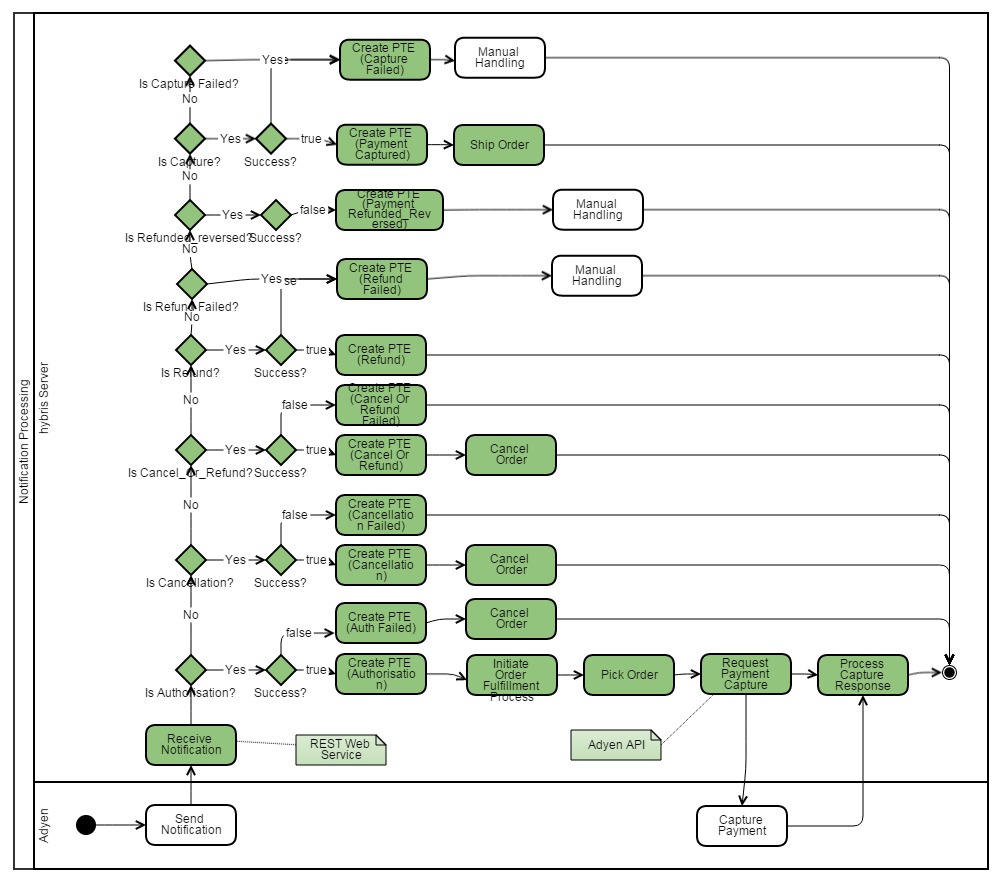
### HPP Payment Flow

If HPP payment model is selected, shopper will go through below flow to authorize the payment from Adyen and then place order in Hybris.



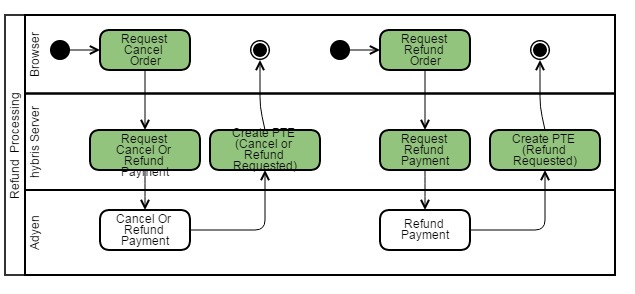
## Notification Processing

Adyen Notification handling will follow below process.



Note: When Hybris get REFUND\_REVERSED, unsuccessful RRFUND and CAPTURE\_FAILED notification, the plugin will only create the payment transaction entry, but do nothing else. SI should customize their own business logic by implementing corresponding notification strategy if it’s needed.

## Backoffice Process Flows



# Considerations for an Existing Application

There are several approaches that could be taken to implement the Adyen into an existing storefront. Deciding on the best approach would depend on how far the storefront has already been altered, which payment model was being included and which checkout flow.

If the Multi-steps checkout flow is being used then the easiest approach is probably to start at the adyenAddon MultiStepCheckoutController and copy code across to the existing CheckoutController. As you do this you will need to make sure adyenServices and adyenCommerceServices are included into the localextensions.xml. And your own checkout addon or storefront extensions are depend on them.

If a different checkout flow is being used then it may be easier to look through the spring configurations of the accelerator projects to find out the classes and related configuration that would require being copied. Again the adyenServices and adyenCommerceServices should be included into the localextensions.xml. And your own checkout addon or storefront extensions should depend on them.

# To Meet The Best Adyen Practice

## Same Code for Order and Cart

By Hybris OOTB behavior, the order number will be different from shopping cart number. The defined bean orderCodeGenerator is responsible for creating the order number.

Class de.hybris.platform.order.strategies.impl.DefaultCreateOrderFromCartStrategy will call orderCodeGenerator to get the new order code when creating order from shopping cart.

To override this behavior to have same code for both cart and order you can follow below steps.

1. Create a new ExtCreateOrderFromCartStrategy class which extends from de.hybris.platform.order.strategies.impl.DefaultCreateOrderFromCartStrategy.
2. Override protected method protected String generateOrderCode(final CartModel cart) to just simply cart.getCode()
3. Add spring bean definition of ExtCreateOrderFromCartStrategy in adyenServices-spring.xml and override the alias <alias alias="createOrderFromCartStrategy" name="extCreateOrderFromCartStrategy"/>

# Suggested Further Reading

For anyone that has access to the Hybris and Adyen wiki there is some documentation that might provide more insight to this implementation guide:

<https://wiki.hybris.com/display/release5/payment+Extension+-+Technical+Guide>

<https://wiki.hybris.com/display/release5/Payment+Integration+Overview>

<https://wiki.hybris.com/display/release5/Creating+Web+Applications>

<https://wiki.hybris.com/display/release5/Setting+Basic+Authentication+in+Core+Plus+Services>

<https://docs.adyen.com/display/TD/API+Manual>

<https://docs.adyen.com/display/TD/HPP+Manual>

https://docs.adyen.com/display/TD/Recurring+Manual