



PayPal Add-on for Hybris 2205-2211

USER GUIDE



SAP Hybris (v)

Table of Contents

1	Introduction	4
2	About this Document	5
2.1	Acronyms and Abbreviations	5
3	Delivered Package Description	6
4	Supported Versions, Configurations, Features	7
4.1	Supported Hybris versions and PayPal features.....	7
4.2	PayPal API calls.....	7
5	Solution Installation	9
5.1	Prerequisites	9
5.1.1	Software	9
5.1.2	OS-level file “hosts”	9
5.2	Package Installation.....	9
5.2.1	B2C Installation	9
5.2.2	B2B Installation	11
5.3	Package Installation step-by-step with a hybris.zip archive	11
5.4	Package Installation step-by-step with a working project.....	12
5.5	Upgrade from version 1.9.1.0 to version 1.10.0.0 (Optional)	13
5.6	Upgrade from version < 1.12.0.0 to version 1.12.0.0 (Optional)	13
5.7	Hybris Out of the Box Applications.....	14
6	Solution Configuration.....	15
6.1	Essential configuration	15
6.2	Plugin Configuration	15
6.2.1	PayPalGeneralConfiguration.....	16
6.2.2	PayPalConfiguration	17
6.2.3	VenmoConfiguration	20
6.2.4	LocalPaymentsConfiguration	20
6.2.5	HostedFieldsConfiguration	22
6.2.6	ConnectWithPayPalConfiguration	23
6.2.7	ApplePayConfiguration.....	23
6.3	Plugin configuration (Located in properties files)	24
6.4	Backend configuration, HAC (Non-Persisted)	25
6.5	StylesConfiguration	25
6.5.1	PayPalStylesConfiguration	26
6.5.2	ApplePayConfiguration	27
6.5.3	HostedFieldsStylesConfiguration	27
6.6	Connect with PayPal Button Configuration	28
6.6.1	Overview.....	28
6.6.2	Configuration	28

6.6.3	Button styles config	29
6.6.4	Password setup.....	29
6.6.5	Additional Security Layer Connect with PayPal	30
6.6.6	Unlink Connect with PayPal.....	30
6.7	Checkout buttons configuration	30
7	<i>Solution Functionalities</i>	32
7.1	Session Payments	32
7.1.1	Conditions.....	32
7.1.2	Restriction.....	32
7.2	Replenishment flow for B2B	33
7.2.1	General flow	33
7.2.2	Restrictions	34
7.3	PayPal Vault flow	34
7.3.1	Save PayPal	34
7.3.2	Change Payment Button.....	35
7.4	Apple Pay	35
7.4.1	Set up sandbox account to accept Apple Pay	35
7.4.2	Before using the sandbox/live environment	36
7.5	Notifications.....	37
7.5.1	Connect with PayPal notifications	37
7.5.2	Hosted fields and 3DS notifications.....	38
7.5.3	Other notifications.....	38
7.6	Level 2 and Level 3 data for transactions	39
7.7	Image URL for Purchase Units.....	39
7.8	Tracking API.....	40
7.9	Transaction Logs.....	43
8	<i>Order\Transaction Managements in Backoffice.....</i>	46
8.1	Overview	46
8.2	Backoffice transaction statuses.....	46
8.2.1	Custom Actions with Intent Authorize.....	47
8.2.2	Custom Actions with Intent Capture	48
8.2.3	Custom Actions with Intent Save Order	49
9	<i>Move to Production</i>	51
10	<i>Solutions Limitation.....</i>	52

Copyright Information

© 2024 PayPal, Inc. All rights reserved.

PayPal is a registered trademark of PayPal, Inc. The PayPal logo is a trademark of PayPal, Inc. Other trademarks and brands are the property of their respective owners. The information in this document belongs to PayPal, Inc. It may not be used, reproduced, or disclosed without the written approval of PayPal, Inc.

Copyright © PayPal. All rights reserved. PayPal (Europe) S.à r.l. et Cie, S.C.A., Société en Commandite par Actions. Registered office: 22-24 Boulevard Royal, L-2449, Luxembourg, R.C.S. Luxembourg B 118 349

Consumer advisory: The PayPal™ payment service is regarded as a stored value facility under Singapore law. As such, it does not require the approval of the Monetary Authority of Singapore. You are advised to read the terms and conditions carefully.

Notice of Non-Liability

PayPal, Inc. is providing the information in this document to you “AS-IS” with all faults. PayPal, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. PayPal, Inc. assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. PayPal, Inc. reserves the right to make changes to any information herein without further notice.

1 Introduction

This document contains the functional description and installation guideline of the PayPal Checkout Add-on. The primary goal is to help you integrate a complete PayPal flow with Hybris Commerce Suite versions 2205-2211. This document will also provide some details around the implemented features (such as PayPal Smart Payment Buttons) and how they need to be configured. The PayPal functionality described here is intended to be a generic business use case and may require a previously customized checkout experience to be modified further to support the PayPal Add-on.

Note: It is recommended that the installation and integration of this extension will be done by a “SAP Certified Development Professional”.

2 About this Document

2.1 Acronyms and Abbreviations

Acronym	Description
B2C	Business-to-Customer e-commerce Channel.
B2B	Business-to-Business e-commerce Channel.
ACC	Accelerator
SPA	Spartacus
Hybris Platform initialization	Initialization drops existing type definitions from the database. The entire type system is created from scratch.
HAC	Hybris Administration Console. Used by system administrators to manage/configure the Hybris engine.
BO	Backoffice. Successor or replacement for HMC. Since Hybris-6.7, HMC is not available.
CMS	Content Management System.
Responsive design	Application design that adapts the presentation to the device it is rendered on.
REST	Architectural style for providing standards between computer systems on the web.
Merchant's PayPal account	Merchant's PayPal business account.
PayPal Smart Payment Buttons	Smart Payment Buttons (SPB) provide a set of payment instruments that can be used by buyers and processed through PayPal. The Smart Payment Button code is integrated through PayPal's JS SDK.
CwPP/ LiPP	Connect with PayPal/Login with PayPal

Before installing and integrating a solution into your merchant code base, it's recommended that you get a sample store working with the help of this guide.

3 Delivered Package Description

The package includes the following folders:

Folder	Description
docs_release	Contains documentation for the release, including this guide.
extensions	Contains solution source code.
samples	Contains sample Bash-script and properties files. Bash-script shows how to rollout solution from scratch with Hybris B2C and B2B sample stores (Electronics for B2C, PowerTools for B2B).

4 Supported Versions, Configurations, Features

4.1 Supported Hybris versions and PayPal features

The table below shows supported versions and features.

Feature \ Version	Hybris B2C	Hybris B2B
Guest checkout	+	N/A
Logged in checkout	+	+
Express Checkout Flow	+	+
Multi-step checkout	+	+
Replenishment order	N/A	+
PayPal Smart Payment Buttons configuration (HAC)	+	+
Backoffice extension	+	+
PayPal vault flow	+	+
PayPal Change Payment Button	+	+
PayPal Credit Banner\Messages	+	+
Local Payment Methods	+	+
PayPal Pay Later Offers	+	+
Venmo Payment Method	+	+
Connect with PayPal feature	+	+
Webhooks (Authorize\capture\void webhooks)	+	+
Save Order flow	+	+
PayPal FraudNet integration	+	+
PayPal advanced Credit Card integration	+	+
Tracking API	+	+
Apple Pay	+	+
Transaction Logs in BO	+	+
Saving Limitations on PMs	+	+

4.2 PayPal API calls

API call \ Solution	Hybris B2C	Hybris B2B
Render button	+	+
Create PayPal Order	+	+
Update Order	+	+
Get Order Details	+	+
Create Setup Token	+	+
Create Payment Token	+	+
Authorization	+	+

Reauthorization	+	+
Authorization Void	+	+
Capture	+	+
Refund	+	+
Delete Payment Token	+	+
Send Tracking Info	+	+
Convert Billing Agreement to Payment Token	+	+

This solution uses PayPal's JS SDK for the UI layer and JAVA SDK and direct API requests for backend calls.

5 Solution Installation

This section reviews setting up the solution in a development environment.

5.1 Prerequisites

5.1.1 Software

Before continuing, make sure you have configured your environment as follows:

- Linux-system, 64-bit
- Oracle JDK 17, 64-bit (For Hybris 2205-2211)
- Configured JAVA_HOME environment variable
- Configured Ant-1.9.1+
- Hybris package, with supported version

5.1.2 OS-level file “hosts”

Add the following two lines to update your local OS-level “hosts” file:

```
127.0.0.1 localhost
127.0.0.1 electronics.local powertools.local
```

5.2 Package Installation

5.2.1 B2C Installation

Required files to set up 2211 sample store with PayPal.

File	Description
CXCOMM221100P_XXXX.ZIP	Hybris 2211 package
paypal_rest_1.X.X.X.zip	Delivered package

1. Copy the files to the directory you want to use as your “working” directory.
2. Unzip “paypal_rest_1.X.X.X.zip”.
 - a. Rename the unzipped folder “paypal_rest_1.X.X.X” to “plugin”
3. Copy “/plugin/samples/prepare.sh” into the “working” directory.

The “working” directory should now contain four items:

File	Description
prepare.sh	Extracted Bash-script from delivered package
CXCOMM221100P_XXXX.ZIP	Hybris 2211 package
paypal_rest_1.X.X.X.zip	Delivered package
plugin	Unzipped folder containing contents of paypal_rest_1.X.X.X.zip

4. Configure prepare.sh
 - a. Enter “working” directory and open console and run command:
 - ./prepare.sh
 - b. Enter path to HYBRIS.ZIP

- c. Enter the name of the folder where the hybrid project will be located (default - hybris)
- d. Choose yes if you agree with the version of hybris (this version of hybris is taken from the archive that you specified)
- e. Choose accelerator (for B2C accelerator choose b2c)
- f. Choose yes if you will install Spartacus in future
- g. This results in a working test store for merchant developers, who will see how the solution was rolled out and how it works with the default store.
- h. It is recommended that you review the "prepare.sh" script to see how it creates the environment and builds the platform.

Errors when executing "prepare.sh"

- Unzipping the Hybris package may display a "cannot create" error for a ".html" file.
 - The installation will continue and should not be impacted.
- "-bash: ant: command not found" error
 - May require running "setantenv.sh" before "prepare.sh" can be run.
 - It is recommended to open "prepare.sh" and run the lines individually from the "working" directory, starting with "cd hybris-2211/hybris/bin/platform".
- "file not found" errors
 - Double-check that the "working" directory structure and "prepare.sh" contents match up.
- This installation can take 15 minutes to run, possibly more.
- "sudo" may be required to run the "ant" commands.
- To restart the test store, run "./hybrisserver.sh" from the "hybris-2211/hybris/bin/platform" folder.
- Hybris test stores are valid for 30 days.
 - To re-initialize, run "ant clean all initialize" from the "hybris-2211/hybris/bin/platform" folder.

Once you got access to the test store, you can continue with SolutionConfiguration.

5.2.1.1 URLs, credentials

Application	URL, Default Login / Password
Electronics site	https://electronics.local:9002/yacceleratorstorefront Guest access allowed. acceptanceuser@test.com / 12341234

5.2.2 B2B Installation

Please use the installation steps listed in section 5.2.1, B2C Installation, updating step 5.d:

1. Choose accelerator (for B2B accelerator choose b2b)

Once you got access to the test store, you can continue with Solution Configuration.

5.2.2.1 URLs, credentials

Application	URL, Default Login / Password
PowerTools site	https://powertools.local:9002/yb2bacceleratorstorefront Guest access not allowed. axel.krause@rustic-hw.com / 12341234 ulf.becker@rustic-hw.com / 12341234

5.3 Package Installation step-by-step with a hybris.zip archive

The instructions below are general for all 2205-2211 Hybris versions.

Use this guide if prepare.sh isn't working or you're using Windows OS.

Required files to set up 2211 sample store with PayPal.

File	Description
CXCOMM221100P_XXXX.ZIP	Hybris 2211 package
paypal_rest_1.X.X.X.zip	Delivered package

1. Copy the files to the directory you want to use as your “working” directory.
2. Unzip “paypal_rest_1.X.X.X.zip”.
 - a. Rename the unzipped folder “paypal_rest_1.X.X.X” to “plugin”.
3. Unzip “CXCOMM221100P_XXXX.zip”.
 - a. Rename the unzipped folder “CXCOMM221100P_XXXX” to “hybris”.

4. The “working” directory should now contain four items:

File	Description
hybris	Unzipped folder containing contents of CXCOMM221100P_XXXX.ZIP
CXCOMM221100P_XXXX.ZIP	Hybris 2211 package
paypal_rest_1.X.X.X.zip	Delivered package
plugin	Unzipped folder containing contents of paypal_rest_1.X.X.X.zip

5. Copy directory “extensions” from “/plugin” directory into the “/hybris/hybris/bin” directory and rename directory “extensions” into “/hybris/hybris/bin” folder to “paypal_custom”.

6. Copy directory paypal_recipe from “/plugin/sample/2211-b2c” directory into the “/hybris/installer/recipes” directory.

Decrypt the folder name:

- 2205-2211 - hybris version (This version should be the same as the archive version of hybris)
- b2c, b2b – accelerator version(B2C accelerator or B2B accelerator)
- spa - this folder contains settings for future installation of Spartacus.

7. Enter into directory “/hybris/installer” and open console and run command:

- For Linux (./install.sh -r paypal_recipe) or for Windows (install.bat -r paypal_recipe);

8. Close console.

9. Enter into directory “/hybris/hybris/bin/platform” and open console and run all commands :

- For Linux (./setantenv.sh) or for Windows (setantenv.bat);
- ant clean all initialize
- For Linux (./hybrisserver.sh) or for Windows (hybrisserver.bat);

10. Done.

This results in a working test store for merchant developers, who will see how the solution was rolled out and how it works with the default store.

11. URLs credentials are the same as:

- B2C accelerator – 5.2.1.1 section
- B2B accelerator – 5.2.2.1 section

5.4 Package Installation step-by-step with a working project

1. Unzip “paypal_rest_1.X.X.X.zip”.
2. Enter to the unzipped folder and copy the folder 'extensions' to your project in the folder ‘<project-name>/hybris/bin’ and you can rename this folder to ‘paypal_custom’.
3. Enter to the unzipped folder ‘/sample/your version hybris and storefront/paypal_recipe’ and open the file ‘build.gradle’ and copy all 'property' that is not in your file ‘<project-name>/hybris/config/local.property’ and insert into this file all copied properties.

Decrypt the folder name:

- 2205-2211 - hybris version (This version should be the same as a working project)
- b2c, b2b – accelerator version (B2C accelerator or B2B accelerator)
- spa - this folder contains settings if working project works with Spartacus.

4. Enter to the unzipped folder '/sample/your version hybris and storefront/paypal_recipe' and open the file 'build.gradle' and copy all custom extensions from '//Start PayPal custom extensions' comment to '//End PayPal custom extensions' comment and insert into the file 'localextensions.xml' in your project '<project-name>/hybris/config/localextension.xml'.
5. Enter to your project '<project-name>/hybris/bin/platform' folder open console and run commands:

- B2C accelerator

```
ant addoninstall -Daddonnames="paypaladdon" -  
DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"
```

- B2C accelerator + Spartacus

```
ant addoninstall -Daddonnames="paypaladdon" -  
DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"  
ant addoninstall -Daddonnames="paypaloccaddon" -  
DaddonStorefront.ycommercewebservices="ycommercewebservices"
```

- B2B accelerator

```
ant addoninstall -Daddonnames="paypalb2baddon" -  
DaddonStorefront.yacceleratorstorefront="yb2bacceleratorstorefront"
```

- B2B accelerator + Spartacus

```
ant addoninstall -Daddonnames="paypalb2baddon" -  
DaddonStorefront.yacceleratorstorefront="yb2bacceleratorstorefront"  
ant addoninstall -Daddonnames="paypalb2boccaddon" -  
DaddonStorefront.ycommercewebservices="ycommercewebservices"
```

6. Build, initialize, start

```
ant clean all initialize  
./hybrisserver.sh
```

* hybrisserver.bat for Windows OS

7. Done

This results in a working test store for merchant developers, who will see how the solution was rolled out and how it works with a working project.

5.5 Upgrade from version 1.9.1.0 to version 1.10.0.0 (Optional)

If you want to upgrade from version 1.9.1.0 to version 1.10.0.0, you need to run the next groovy script: extensions/common/paypalcore/resources/paypalcore/groovy/payment-info-update.groovy. This must be done because the SubscriptionId in version 1.10.0.0 stored in a PayPalOrderId field.

If you are building an addon from scratch, this step (5.5) is not necessary.

5.6 Upgrade from version < 1.12.0.0 to version 1.12.0.0 (Optional)

If you want to upgrade from version less than 1.12.0.0 to version 1.12.0.0, you need to run the next groovy script: `extensions/common/paypalcore/resources/paypalcore/groovy/migration-billing-token-to-vault.groovy`

This must be done because the Billing Agreement deprecated and removed in version 1.12.0.0, and it is replaced with PayPal Vault flow.

If you are building an addon from scratch, this step (5.6) is not necessary.

5.7 Hybris Out of the Box Applications

Hybris provides the following applications out of the box. URLs are common for Hybris B2C, B2B.

Application	URL, Default Login / Password
HAC	https://localhost:9002/hac admin / nimda or nimdar00t
Backoffice	https://localhost:9002/backoffice admin / nimda or nimdar00t
B2C storefront	https://electronics.local:9002/yacceleratorstorefront
B2B storefront	https://powertools.local:9002/yb2bacceleratorstorefront

6 Solution Configuration

6.1 Essential configuration

For each Hybris instance, please review and update the next properties in the `local.properties` file.

Solution connectivity properties:

Property	Comment
paypal.username	Merchant account API user name.
paypal.password	Merchant account API password.
paypal.signature	Merchant account API signature.
paypal.seller.email	Merchant account API e-mail.
paypal.environment.configuration.type	Profile name, case sensitive, either "sandbox" or "production"
paypal.client.id	Application Client ID
paypal.secret.key	Application Secret

6.2 Plugin Configuration

Plugin configuration stored under BaseStore model within `PayPalPluginConfiguration`.

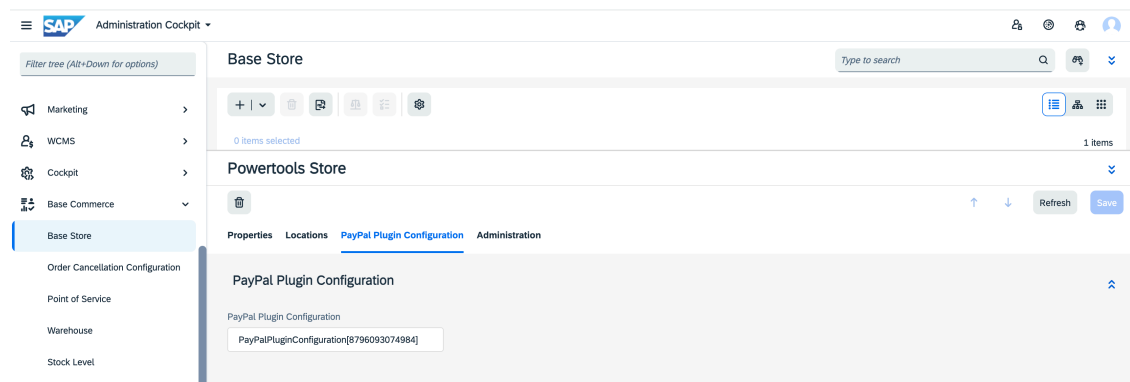


Figure 1 PayPal Plugin configuration in Backoffice under BaseStore

All configurations are separated in logical representation and next sub-configurations are available:

- `PayPalGeneralConfiguration` – contain general configuration of plugin.
- `PayPalConfiguration` – contain configuration of PayPal payment method.
- `VenmoConfiguration` – contain Venmo related configuration.
- `LocalPaymentsConfiguration` – contain LPM related configuration.
- `HostedFieldsConfiguration` – contain hosted fields related configuration.
- `ConnectWithPayPalConfiguration` – contain CwPP related configuration.
- `ApplePayConfiguration` – contain Apple Pay related configuration.

Default configuration of a plugin located in next impex essentialdata-paypal-plugin-configuration.impex which is can be found in one of the next extensions paypaladdon, paypalb2baddon, paypalocc, paypalb2bocc.

Configurations are available also as separate tree node in the Backoffice:

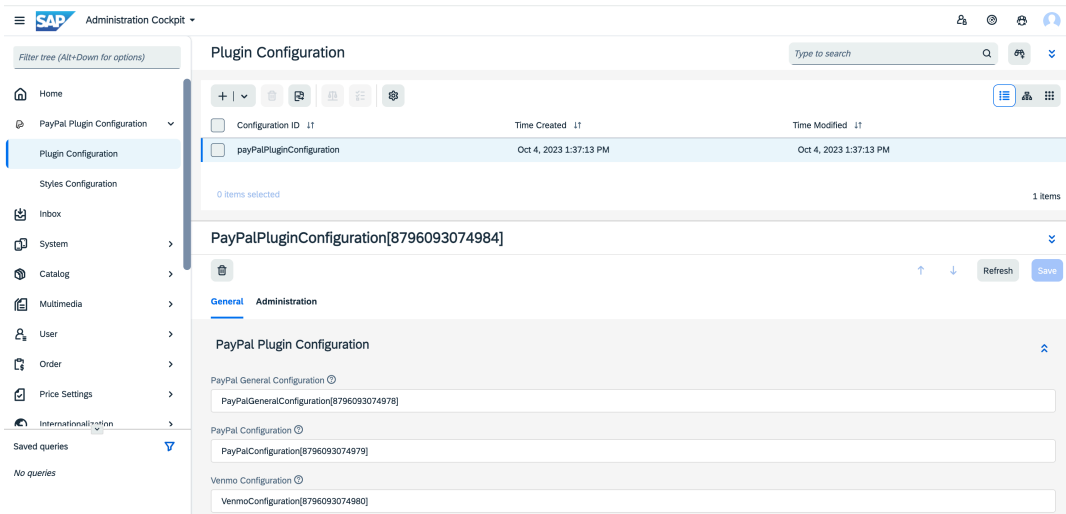


Figure 2 PayPal Plugin configuration in Backoffice

The Administration tab for the Plugin Configuration was left as a hybris out of the box one and stores metadata information about the model.

6.2.1 PayPalGeneralConfiguration

General configuration contains next available properties:

Property	Values supported	Description
Settle Transaction Immediately	true/false	true: capture transaction immediately; false: create Authorization that will be manually captured later.
Level 2 Level 3 Data Enabled	true/false	Enables Level 2 Level 3 data transfer while transaction data processing.
PayPal FraudNet Enabled for Sandbox	true/false	Enables PayPal FraudNet feature for sandbox enviorement
PayPal FraudNet Enabled	true/false	Enables PayPal FraudNet feature that collects buyers browser-based data from their browsers and passes it to PayPal
Buyer Country	DE (String)	Buyer country parameter for testing on the sandbox environment
Locale for Button Language	en_US	Allows rendering Smart Buttons in the language which is inherent to the entered locale
Session Payments Enabled	disabled - session payments is not available for both guests and logged-in buyers logged-in users only - session payments are	This option determines whether session payments should be enabled for guest or logged-in users or both or disabled

	available only for logged-in buyers guest and logged-in users - session payments are available for both guests and logged-in buyers	
--	--	--

When put the Settle Transaction Immediately property value to true, then the additional notification message appears - “Please note that in case you enable Local Payment Methods (LPM) they will not be available for buyers since LPM does not support the authorization payment model”.

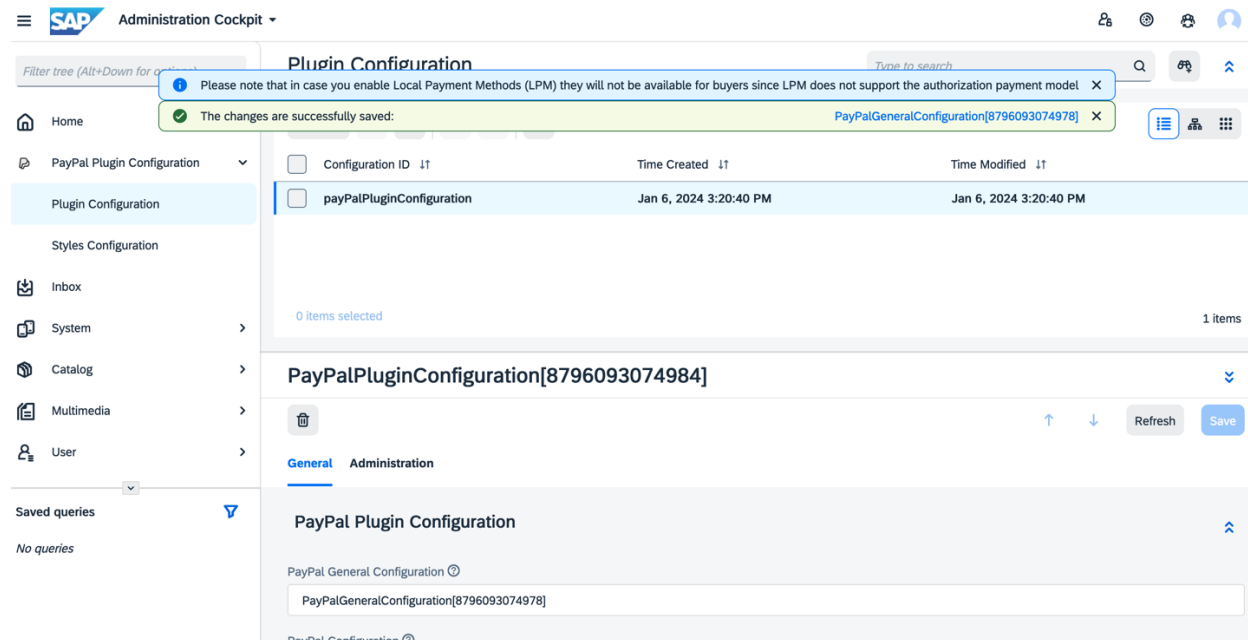


Figure 3 Notification for Settle Transaction Immediately property

6.2.2 PayPalConfiguration

PayPal configuration contains next available properties:

Property	Values supported	Description
PayPal Enabled	true/false	Activates PayPal buttons (PayPal, PayPal Debit/Credit, PayLater)
PayPal Vault Flow	true/false	Enables PayPal Vault flow which allows saving PayPal accounts for later use. Vaulted payments will also be used in replenishment orders (B2B only)
Change Payment Button	true/false	Activates redesigned PayPal button with returning customer experience, namely a saved PayPal account and the preferred funding source will be displayed on the PayPal button and it's possible to Check out without re-entering a payment method.
PayPal on Express Checkout Enabled	true/false	Activates PayPal button on Express Checkout.

PayPal Pay Later for non-US Countries Enabled	true/false	Allows using rendering PayLater buttons (EC and Mark Checkout) for non-US based buyers
PayPal Save Order Flow Enabled	true/false	Enables PayPal Save Order flow which can be triggered via the PayPal and Venmo buttons. Note: In case you use PayPal Hosted Fields, then authorized transaction will be created instead of PayPal Save Order (O-Order).
PayPal Banners\Messages Enabled	true/false	Activates PayPal banners/messages on Cart, PDP, PLP, PGP pages
PayPal Funding Source on Express Checkout	PayPal - activates PayPal button on the Cart\Mini-cart page. Credit or debit cards value - activates PayPal Debit Credit button on the Cart\Mini-cart page if the Change Payment Button is disabled. Pay Later value – (depending on the eligibility) activates PayLater button on the Cart\Mini-cart page. Empty list - hide all the PayPal-related funding sources on the Cart\Mini-cart page.	Activates\Deactivates additional PayPal-related funding sources (buttons) on Express Checkout. Note: Button rendering depends on the merchant\buyer eligibility.
PayPal Funding Source on Checkout	PayPal - activates PayPal button on the Billing page. Credit or debit cards value - activates PayPal Debit Credit button on the Billing page if the Change Payment Button is disabled. Pay Later value - (depending on the eligibility) activates PayLater button on the Billing page. Empty list - hide all the PayPal-related funding sources on the Billing page.	Activates\Deactivates additional PayPal-related funding sources (buttons) on the Billing page. Note: Button rendering depends on the merchant\buyer eligibility.
Limit of saved PayPal accounts	One/Two/Three/Four/Five/Unlimited	One-Five – limits customers to save PayPal accounts. Unlimited – disables this limit. Note: this property is available only when Vault Flow for PayPal is enabled.

When put the PayPal Enabled property value to false, then the additional notification message appears - “Please note that PayPal payment method is disabled, thus the properties related to this payment method are deactivated”.

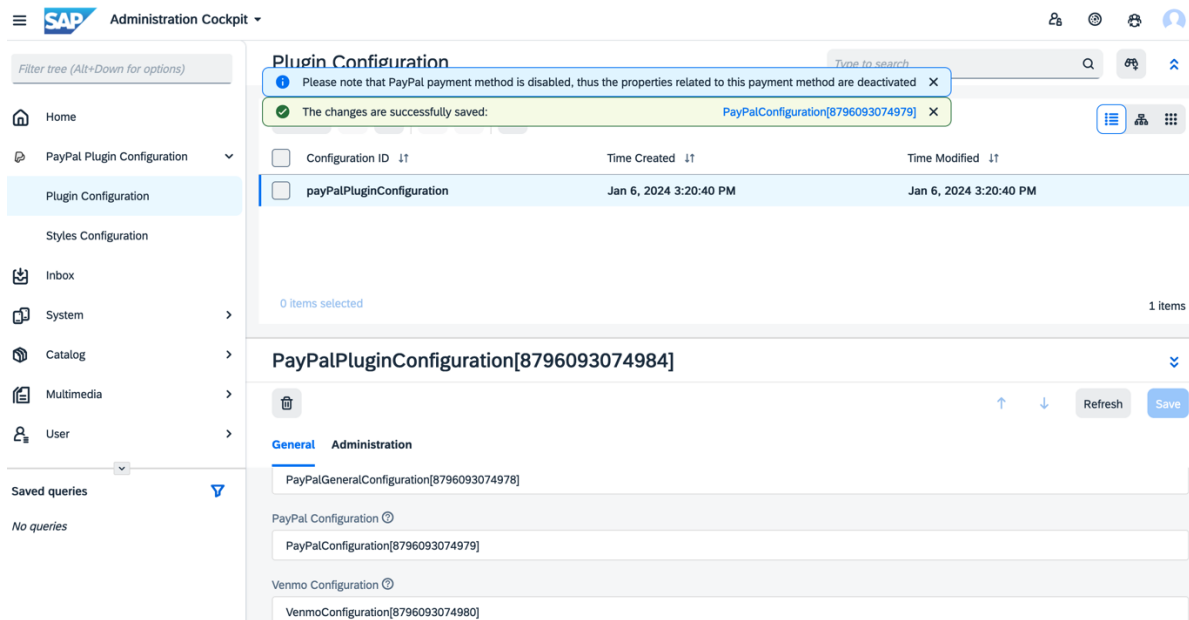


Figure 4 Notification for PayPal Enabled property

When put the Limit of saved PayPal accounts property value to more than ONE, then the additional notification message appears - “Please note that setting the limit on saving PayPal accounts to a value greater than one may cause a problem with payment by saved accounts”.

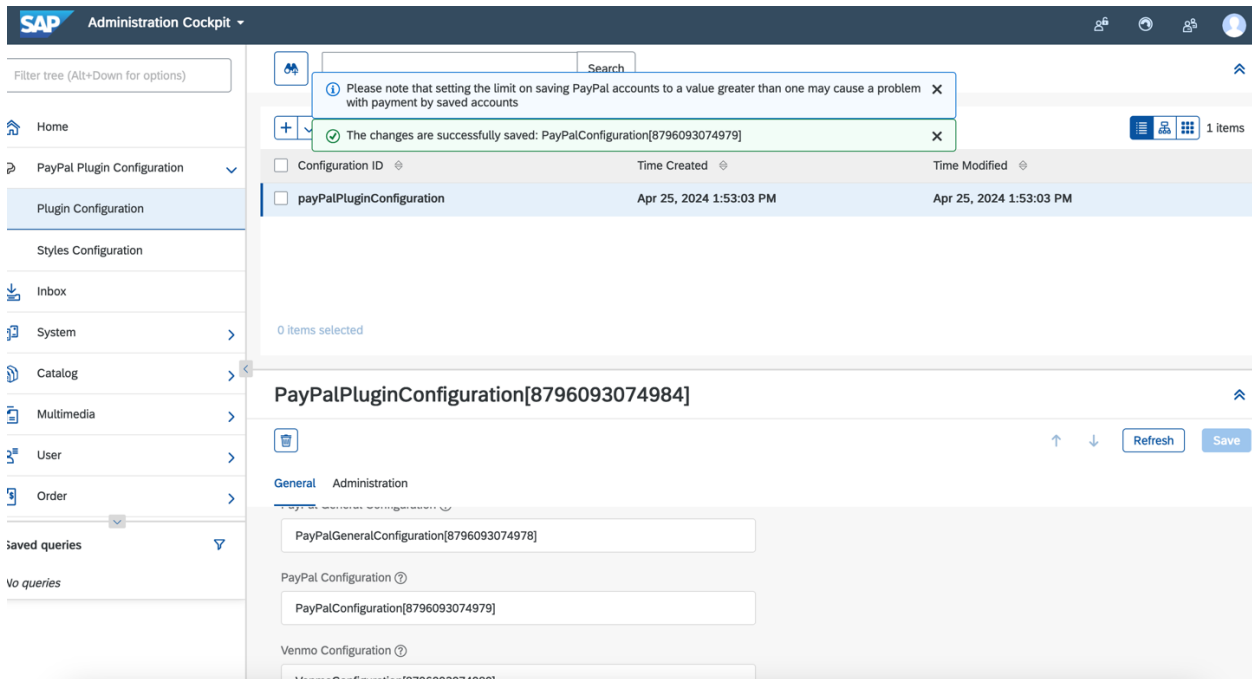


Figure 5 Notification in the pop-up for Limit of saved PayPal account

And the warning message in the pop-up – “Setting the limit on saving PayPal accounts to a value greater than one may cause a problem with payment by saved accounts”.

Edit item PayPalConfiguration[8796093074979]
✕

Setting the limit on saving PayPal accounts to a value greater than one may cause a problem with payment by saved accounts
✕

Refresh Save

General
PayPal Button
Administration

PayPal Save Order Flow Enabled ?
☐ True ☒ False

PayPal Funding Source on Checkout ?

PAYPAL

...

PayPal Funding Source on Express Checkout ?

PAYPAL

...

Limit of saved PayPal accounts ?

FOUR

▼

Figure 6 Notification in the pop-up for Limit of saved PayPal accounts

6.2.3 VenmoConfiguration

Venmo configuration contains next available properties:

Property	Values supported	Description
Venmo Enabled	true/false	Activates Venmo button

6.2.4 LocalPaymentsConfiguration

Local Payments configuration contains next available properties:

Property	Values supported	Description
LPM Enabled	true/false	Enables LPM payment methods on your storefront
LPM Funding Source on Checkout	'Bancontact' value - activates the respective LPM button on the Billing page. 'Eps' value - activates the respective LPM button on the Billing page. 'iDEAL' value - activates the respective LPM	Activates\Deactivates additional Local(Alternative) Payment Method-related funding sources (buttons) on the Billing page. Note: Button rendering depends on the merchant\buyer eligibility.

	<p>button on the Billing page.</p> <p>'MyBank' value - activates the respective LPM button on the Billing page.</p> <p>'Przelewy24' value - activates the respective LPM button on the Billing page.</p> <p>'Blik' value - activates the respective LPM button on the Billing page.</p> <p>Empty list - hide all the PayPal-related funding sources on the Billing page.</p>	
--	--	--

When put the LPM Enabled property value to false, then the additional notification message appears - “Please note that Local Payments are disabled, thus the properties related to this functionality are deactivated”.

The screenshot displays the SAP Administration Cockpit interface for the PayPal Plugin Configuration. A notification banner at the top states: "Please note that Local Payments are disabled, thus the properties related to this functionality are deactivated". Below it, a success message indicates: "The changes are successfully saved: LocalPaymentsConfiguration[8796093074981]". The configuration table shows the following entry:

Configuration ID	Time Created	Time Modified
paypalPluginConfiguration	Jan 6, 2024 3:20:40 PM	Jan 6, 2024 3:20:40 PM

Below the table, the configuration details for "PayPalPluginConfiguration[8796093074984]" are shown. The configuration includes the following sections:

- General**: VenmoConfiguration[8796093074980]
- Local Payments Configuration**: LocalPaymentsConfiguration[8796093074981]
- Hosted Fields Configuration**

Figure 7 Notification for LPM Enabled property

6.2.5 HostedFieldsConfiguration

Hosted Fields configuration contains next available properties:

Property	Values supported	Description
PayPal Hosted Fields Enabled	true/false	Activates PayPal Hosted Fields for accepting credit cards instead of Hybris mocked Hosted Fields
Vault Flow for Hosted Fields	true/false	true: save the payment method to the PayPal vault when transaction is successful false: never save the payment method to the PayPal vault. Please note it's not possible to save credit cards for logged-in buyers
3D Secure Verification Enabled	DISABLE, AUTO, ALWAYS Default value: AUTO	Enables 3D secure verification for credit cards Note: 3D Secure Verification will not be triggered for saved cards
Credit Card Expire Between for Adding New CC Enabled	true/false	Enables validation of the cards expiring soon when adding a new credit card to the wallet.
Credit Card Expire Between Enabled	1 any value as int	Enables the time period to perform validation of the expired cards. If 0, it will only look for already expired cards, if 1, it will look for the cards which will expire in 1 month, etc.
Credit Card Verification on My Account Page	true/false	Enables verification while saving credit card on the My Account page
Limit of saved credit cards	One/Two/Three/Four/Five/Unlimited	One-Five – limits customers to save Credit Cards to their wallets. Unlimited – disables this limit. Note: this property is available only when Vault Flow for Hosted Fields is enabled.

When put the PayPal Hosted Fields Enabled property value to false, then the additional notification message appears - “Please note that PayPal Hosted Fields are disabled, thus the properties related to this functionality are deactivated”.

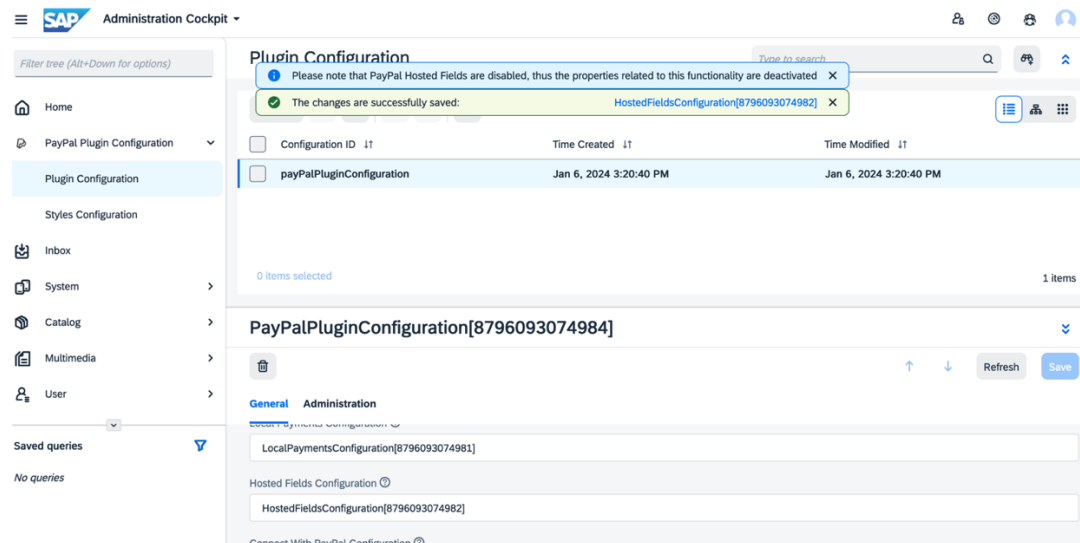


Figure 8 Notification for PayPal Hosted Fields Enabled

6.2.6 ConnectWithPayPalConfiguration

CwPP configuration contains next available properties:

Property	Values supported	Description
Automatic Payment Method Adding Flow for Connect with PayPal	true/false	Enables the Automatic PM Adding Flow which allows logged-in users to save their PayPal accounts and the addresses from these accounts to their accounts immediately after the successful login via Connect with PayPal button
Automatic User Account Linking for Connect with PayPal	true/false	Allows users with PayPal account to login into their existing user account if the primary PayPal account email matches the user account email. Note: we strongly don't recommend enabling this property if your website allows user account registration without email confirmation.
Connect with PayPal Enabled	true/false	Enables Connect with PayPal button on your website. Note: we strongly don't recommend enabling this property if your website allows user account registration without email confirmation.
Additional security for Connect with PayPal	true/false	Enables additional security during paypal connect process. If true - the buyer will be requested to enter credentials of existing account before linking paypal account. Page with password verification will be shown. Note: we strongly recommend enabling this property.

When put the Automatic Payment Method Adding Flow for Connect with PayPal property value to false, then the additional notification message appears - “Please note that Connect with PayPal is disabled, thus the properties related to this functionality are deactivated”.

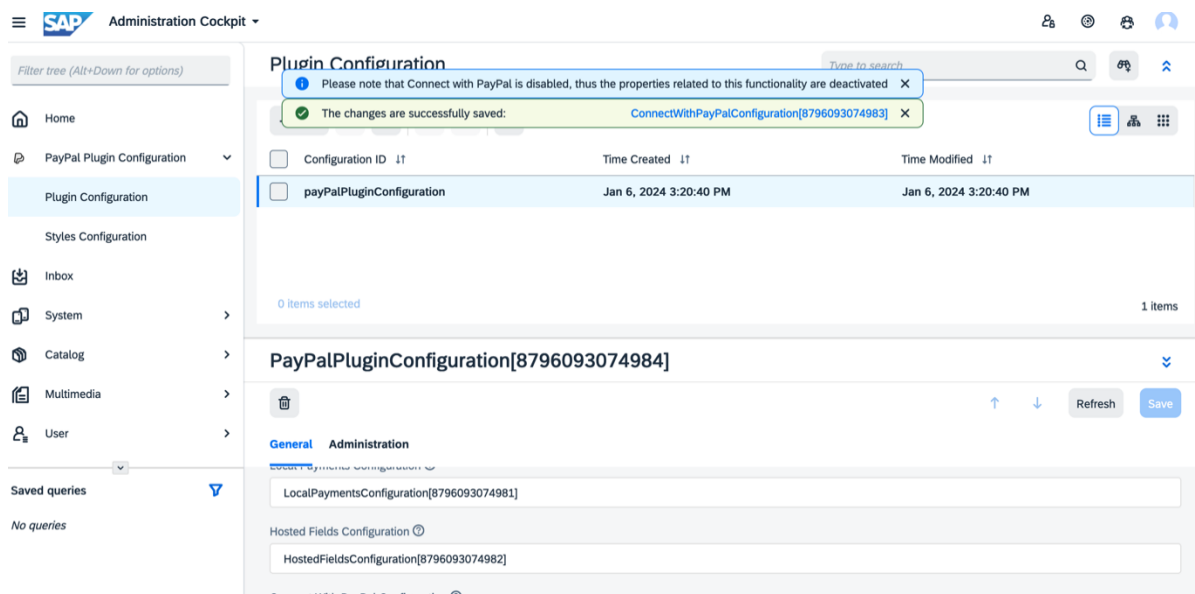


Figure 9 Notification for Automatic Payment Method Adding Flow for Connect with PayPal property

6.2.7 ApplePayConfiguration

Apple Pay configuration contains next available properties:

Property	Values supported	Description
Apple Pay Enabled	true/false	Activates the Apple Pay payment method
Apple Pay on Express Checkout Enabled	true/false	Activates Apple Pay buttons on Express checkout

6.3 Plugin configuration (Located in properties files)

Property	Values supported	Comment
paypal.fraudnet.billing.id	HYBRIS-PP- PLUGIN-BILLING String	Source website identifier for billing page
paypal.fraudnet.cart.id	HYBRIS-PP- PLUGIN-CART String	Source website identifier for cart page
paypal.fraudnet.add.to.cart.popup.id	HYBRIS-PP- PLUGIN- ADDTOCART- POPUP String	Source website identifier for add to cart popup page
paypal.fraudnet.cart.popup.id	HYBRIS-PP- PLUGIN-CART- POPUP String	Source website identifier for cart popup page
paypal.webhook.id.payment.authorization. voided	WEBHOOK_ID Should be replaced with webhook id which will be generated in PayPal	Used for validation incoming request for corresponding event
paypal.webhook.id.payment.capture.comp leted	WEBHOOK_ID Should be replaced with webhook id which will be generated in PayPal	Used for validation incoming request for corresponding event
paypal.webhook.id.payment.capture.refun ded	WEBHOOK_ID Should be replaced with webhook id which will be generated in PayPal	Used for validation incoming request for corresponding event
paypal.webhook.id.checkout.order.approv ed	WEBHOOK_ID Should be replaced with webhook id which will be generated in PayPal	Used for validation incoming request for corresponding event
paypal.webhook.id.vault.payment- token.deleted	WEBHOOK_ID Should be replaced with webhook id which will be generated in PayPal	Used for validation incoming request for corresponding event

paypal.webhook.id.vault.payment-token.created	WEBHOOK_ID Should be replaced with webhook id which will be generated in PayPal	Used for validation incoming request for corresponding event
---	--	--

NOTE: See Disable Funding for more Info

6.4 Backend configuration, HAC (Non-Persisted)

Using HAC, it is possible to change properties at runtime, however, the next time Hybrisis restarted, they will be returned to persisted values.

See a sample screenshot of PayPal properties below:

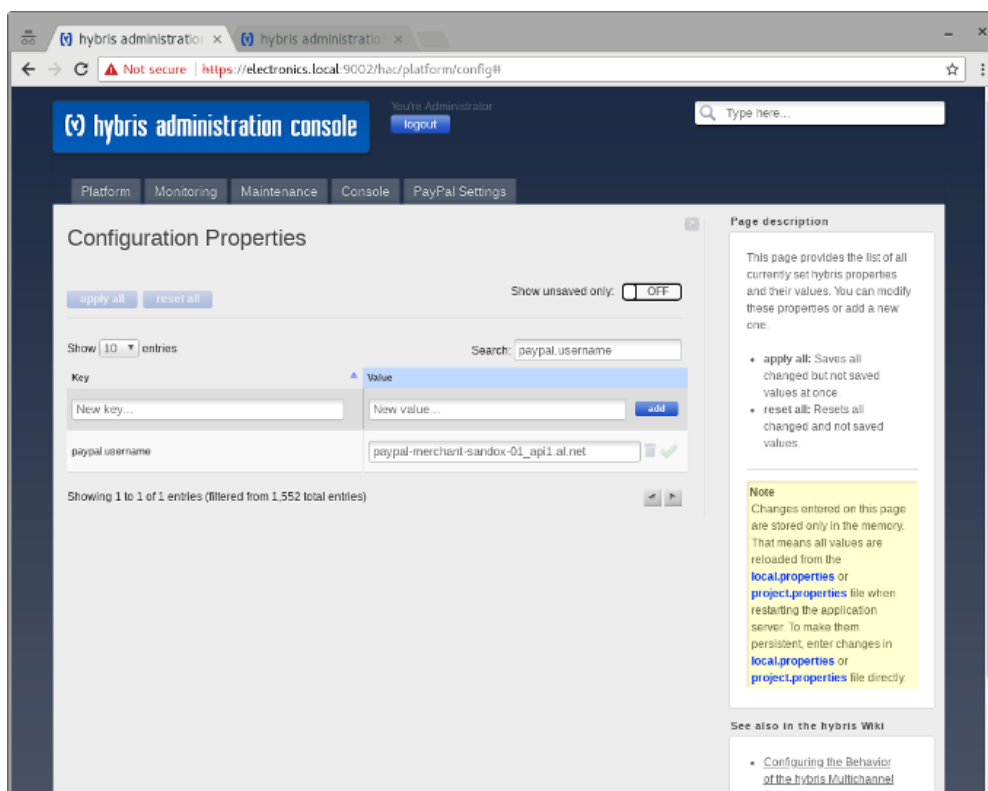


Figure 10 Configuration Properties

6.5 StylesConfiguration

For the PayPal, Apple Pay and Hosted fields we have a possibility to configure the styles of the buttons or the fields. Each styles configuration is related to the specific location – Mini cart, mark and cart.

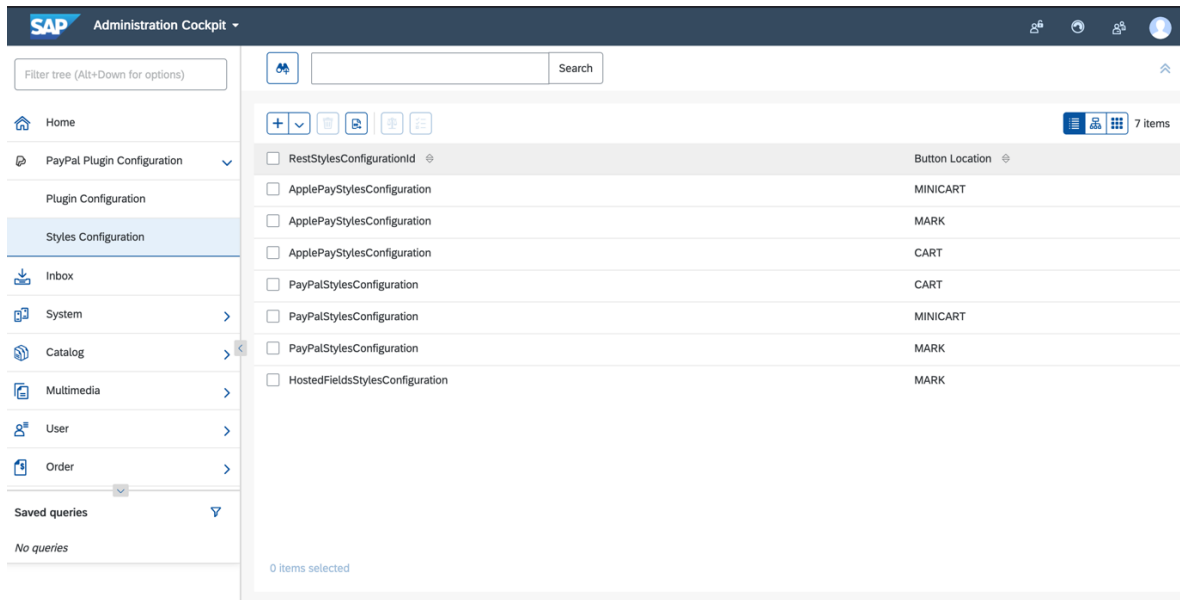


Figure 11 Styles configuration in BO

For the PayPal button and Apple Pay button there are a button preview in the Backoffice which allows to change the styles and see the result after the saving:

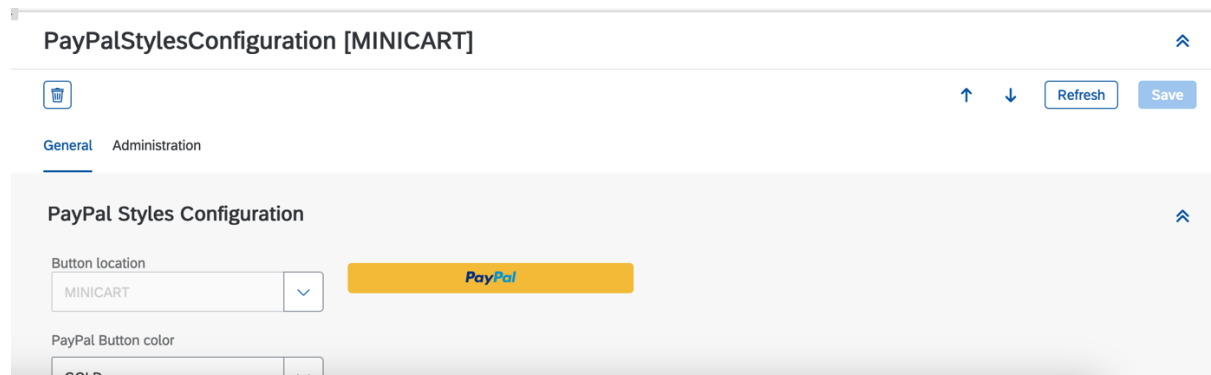


Figure 12 Button styles preview in BO

6.5.1 PayPalStylesConfiguration

Property	Values supported	Description
Button Location	MINICART/MARK/CART	On the different pages you can have the different styles of the PayPal button.
PayPal Button color	BLACK/BLUE/GOLD/SILVER	Used for the specifying the PayPal button color.
PayPal Button shape	RECT/PILL/SHARP	Used for the specifying the PayPal button shape.
PayPal Button height	Number	Used for the specifying the PayPal button height. Please enter a property value in the range of 25 – 55.
PayPal Border radius	Number	Used for the specifying the PayPal button border radius. Please enter a property value of 0 or greater. Note that any entered value except 0 will overwrite PayPal Button shape option.
PayPal Button label	BUYNOW/CHECKOUT/PAY/PAYPAL	The label which displays in the middle of the button. Close to the PayPal label

6.5.2 ApplePayConfiguration

Note that the button preview in the Backoffice is available only in the Safari browser.

Property	Values supported	Description
Button Location	MINICART/MARK/CART	On the different pages you can have the different styles of the Apple Pay button.
Apple Pay Button color	BLACK/WHITE/WHITE_OUTLINE	Used for the specifying the Apple Pay button color.
Apple Pay Button type	ADD_MONEY/BOOK/BUY/CHECK_OUT/CONTINUE/CONTRIBUTE/DONATE/ORDER/PAY/PLAIN/RELOAD/RENT/SET_UP/SUBSCRIBE/SUPPORT/TIP/TOP_UP	Used for the specifying the Apple Pay button shape.

6.5.3 HostedFieldsStylesConfiguration

Property	Values supported	Description
Hosted fields styles configuration id	Text field	Used for specify the configuration id.
Button Location	MARK	Styles for hosted fields are configurable only for the Mark page.
Hosted fields input field color	BLACK/GRAY/GREEN/RED	Used for the specifying the Hosted fields input field color.
Hosted fields invalid field color	BLACK/GRAY/GREEN/RED	Used for the specifying the Hosted fields invalid field color.
Hosted fields valid field color	BLACK/GRAY/GREEN/RED	Used for the specifying the Hosted fields valid field color.
Font size	Number field	Used for the specifying the Hosted fields input text size.

6.5 PayPal Webhooks Configuration

Solution supports next webhooks events with their corresponding handler in paypalwebservices extension:

Payment authorization voided	/paypalwebservices/paypal/webhook/payments
Payment capture completed	/paypalwebservices/paypal/webhook/payments
Payment capture refunded	/paypalwebservices/paypal/webhook/payments
Checkout order approved	/paypalwebservices/paypal/webhook/order/approved
Vault payment-token deleted	/paypalwebservices/paypal/webhook/payments/deleted
Vault payment-token created	/paypalwebservices/paypal/webhook/vault/created

Each event will process order or transaction or payment corresponding to what exactly happens to it on PayPal side.

For **Payment** type of events (Payment authorization voided, Payment capture completed, Payment capture refunded): The transaction will be marked as voided, captured or refunded and order will be processed as well to its corresponding status after such action.

For **Checkout** type of event (Checkout order approved): Used for completing order with Local Payment Methods when after confirmation user leaves site.

For **Vault** type of events (Vault payment-token deleted, Vault payment-token created): The Credit card or PayPal payment methods will be created or deleted dependent on event type.

After webhook is configured in PayPal dashboard it assigns it a webhook id which should be stored in corresponding property `paypal.webhook.id.<webhook_name>`. List of all available properties are in solution configuration section of this document. This needed to achieve event verification before processing it. All implementation of webhook listeners are available here:
`com.paypal.hybris.addon.controllers.webhook.WebhookListener`

6.6 Connect with PayPal Button Configuration

6.6.1 Overview

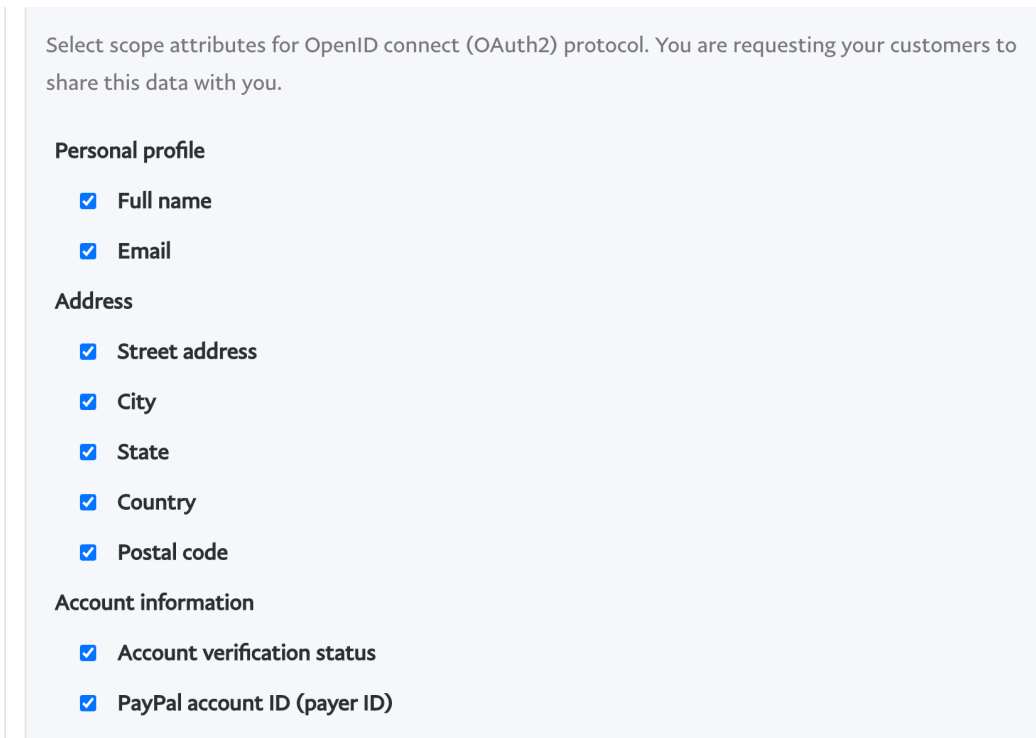
Connect with PayPal enables a customer to log into your website or app using their PayPal credentials or they can link their PayPal account to your website or app so that basic information can be shared. Solution only use the primary email address which is retrieved from PayPal and only allows to proceed with log in if that address is confirmed.

6.6.2 Configuration

This functionality is available on the next pages: Account Login, Page before Checkout (for guest buyers).

More about this feature you may read on [this](#) page.

To use this feature, you have to go through [these](#) steps. During configuration on PayPal side, select all fields to successfully create an account on storefront:



Select scope attributes for OpenID connect (OAuth2) protocol. You are requesting your customers to share this data with you.

Personal profile

- ☒ Full name
- ☒ Email

Address

- ☒ Street address
- ☒ City
- ☒ State
- ☒ Country
- ☒ Postal code

Account information

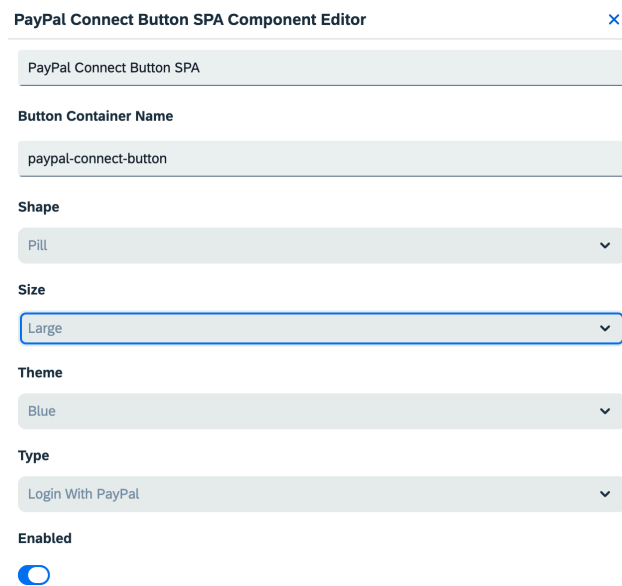
- ☒ Account verification status
- ☒ PayPal account ID (payer ID)

Figure 13 Connect with PayPal configurations

After user clicked on the “Connect with PayPal” button and went through auth PayPal form successfully an account on storefront will be created with data from PayPal buyer account.

1. Recommendations before moving to production: Go to [PayPal Sandbox configuration](#):
 - enter your credentials for access PayPal Dev Console;
 - go to the My Apps&Credentials;
 - select your application;
 - go to SANDBOX APP SETTINGS;
 - click Advanced Option (below Log in with PayPal checkbox);
 - unselect the 'Enable customers who have not yet confirmed their email with PayPal to log in to your app' checkbox.
2. Ensure that you have configured email confirmation for user account registration;

6.6.3 Button styles config



The screenshot shows the 'PayPal Connect Button SPA Component Editor' window. It contains several configuration fields: 'PayPal Connect Button SPA' (text input), 'Button Container Name' (text input with value 'paypal-connect-button'), 'Shape' (dropdown menu with 'Pill' selected), 'Size' (dropdown menu with 'Large' selected), 'Theme' (dropdown menu with 'Blue' selected), 'Type' (dropdown menu with 'Login With PayPal' selected), and an 'Enabled' toggle switch which is currently turned on.

Figure 14 Connect with PayPal Button Configuration

It is possible to configure the following:

- Button type (Connect with PayPal or Login With PayPal);
- Button shape (pill or rectangle);
- Button size (small, medium, large);
- Button theme (blue or neutral);

Also, you can hide\unhide Connect with PayPal button with the help of the ‘Enable Connect With PayPal’ checkbox.

NOTE: In case of having used the shape ‘rectangle’, it is advised to run:

paypalcore/resources/paypalcore/import/set-proper-buttonShape.impex to avoid potential issues.

6.6.4 Password setup

Users created with connect with PayPal feature are able to setup his own password on MyAccount to login on storefront with his PayPal email and new created password.

If you already using Connect with PayPal feature you should consider to run script located in `paypalcore/resources/paypalcore/groovy/set-up-property-for-single-auth.groovy` this will allow existing customers to use this feature.

6.6.5 Additional Security Layer Connect with PayPal

Enables additional security during PayPal connect process, namely buyer will be requested to enter credentials of existing account before linking PayPal account. Page with password verification will be shown.

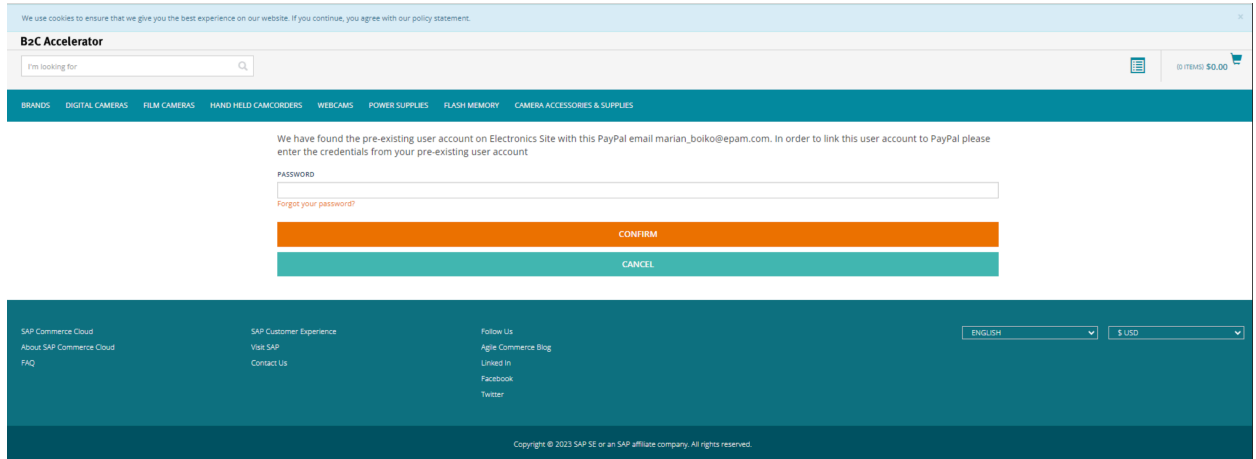


Figure 15 Additional security page

6.6.6 Unlink Connect with PayPal

Provides ability for customer to unlink his previously linked PayPal account from storefront account. When they will try to login back with PayPal after unlinking was performed, they will be forced to perform all actions as it first login in the system (ex: additional security layer, add payment method and shipping address).

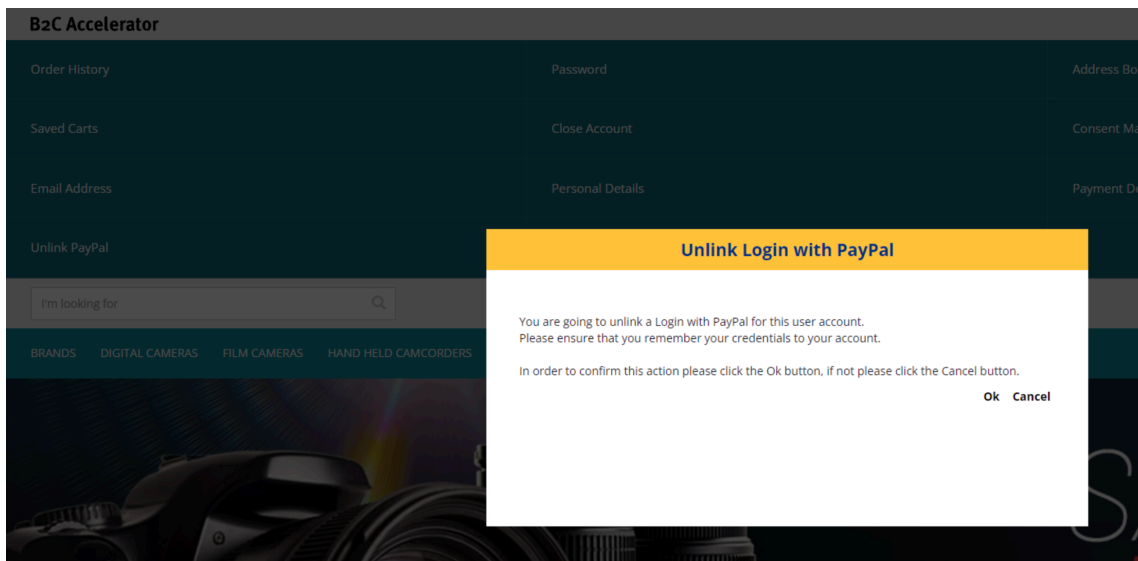


Figure 16 Unlink confirmation pop-up

6.7 Checkout buttons configuration

Plugin provides functionality to configure available payment options independent for two places: payment step of checkout, and express checkout. Additional button should be eligible from PayPal side in order to be rendered.

On payment step of checkout is possible to enable next options:
PayPal buttons (include PayPal, Pay Later, and Credit/Debit Buttons) can be configured by using 'PayPal Funding Source on Checkout' property from PayPalConfiguration object.

Local Payment Buttons (include Bancontact, Eps, iDEAL, MyBank, Przelewy24, Blik) can be configured by using 'LPM Funding Source on Checkout' property from LocalPaymentsConfiguration object.

Venmo Button can be configured by turning on 'Enable' property from VenmoConfiguration. Also, Hosted Fields (Card payment) can be enabled by turning on 'Enable' property from HostedFieldsConfiguration. Also, additional required to be enabled on PayPal side.

On express checkout is possible to enable next options:
PayPal buttons (include PayPal, Pay Later, and Credit/Debit Buttons) can be configured by using 'PayPal Funding Source on Express Checkout' property from PayPalConfiguration object.

Venmo Button can be configured by turning on 'Enable' property from VenmoConfiguration.

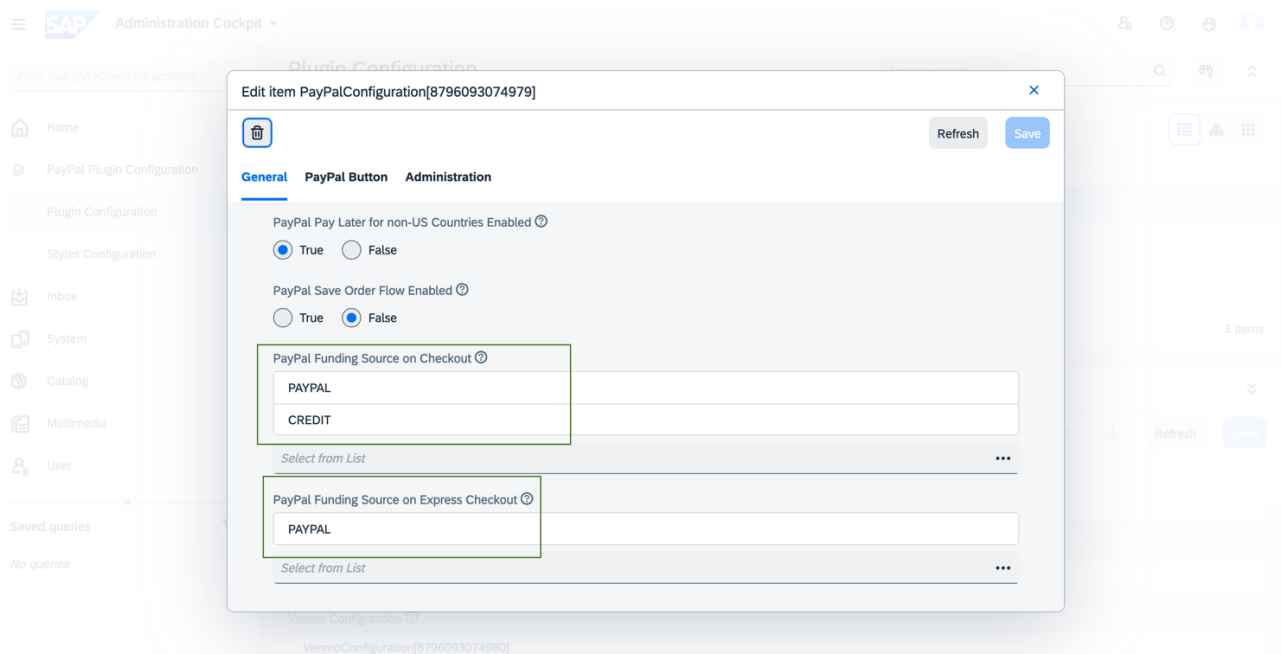


Figure 17 PayPal Funding Sources

7 Solution Functionalities


7.1 Session Payments


The session payment method feature allows guests and logged-in users to keep the previously entered payment method in the session on checkout. Users can simply select it on the billing page instead of re-entering credentials for the payment method. Applicable for the PayPal, Venmo, Credit Card, and Express Checkout.

The Session payment will only be displayed on the Payment section during the checkout with the 'Session Payment' label at the top of it.

3. Payment & Billing Address


Payment & Billing Address

☒  WHAT IS PAYPAL?

 Pay Later

☐ PAY WITH CARD

Session Payment

PAYMENT TYPE	BILLING ADDRESS
 PayPal Account I.VinogradovVN@gmail.com	Ivan C Vinogradov 2253 Hudson Street Denver Colorado United States 80207

USE THE PREVIOUSLY ENTERED PAYMENT METHOD

4. Final Review

Figure 18 Session Payment label

7.1.1 Conditions

The session payment functionality should be available depending on its configuration which is described in the tab PayPalGeneralConfiguration. However, some scenarios when the session payment will disappear are described below:

- 1) The user has placed the order.
- 2) The lifetime of session payment exceeds 15 minutes.
- 3) The user selects the other payment method.
- 4) The 3ds functionality for hosted fields is turned on and the order amount was changed.
- 5) The user logs in or logs out to/from the storefront.
- 6) PayPal account which is marked for saving shouldn't appear as session payment.

7.1.2 Restriction

On the accelerator for the guest user, it is now possible to keep the session payment method only in the checkout flow scope, and if you leave the checkout flow, the session payment method will be deleted. In order for the session payment method not to be deleted and to be available in the scope of the entire site, it is necessary to override the `DefaultGuestCheckoutCartCleanStrategy.cleanGuestCart()` method and remove the null setting in `setPaymentInfo()` and this restriction will disappear. And after the guest leaves the checkout flow and returns to the billing page, the session payment method will be available to him, and he will be able to pay for the purchase without re-entering the credentials of the payment method.

7.2 Replenishment flow for B2B

Replenishment scheduling is the feature which allows customers to define schedules for recurring orders. Available by default on the Billing Page and only for the B2B.

Applicable for PayPal, Google Pay, Apple Pay, Venmo. With hosted fields works only with saved Credit Cards.

7.2.1 General flow

Logged-in buyer on the review page can:

- 1) Set the start date of replenishment.
- 2) Set replenishment frequency:
 - daily replenishment (the day of replenishment can be selected)
 - weekly (the day of replenishment can be selected and a week frequency)
 - monthly (the day of replenishment can be selected)

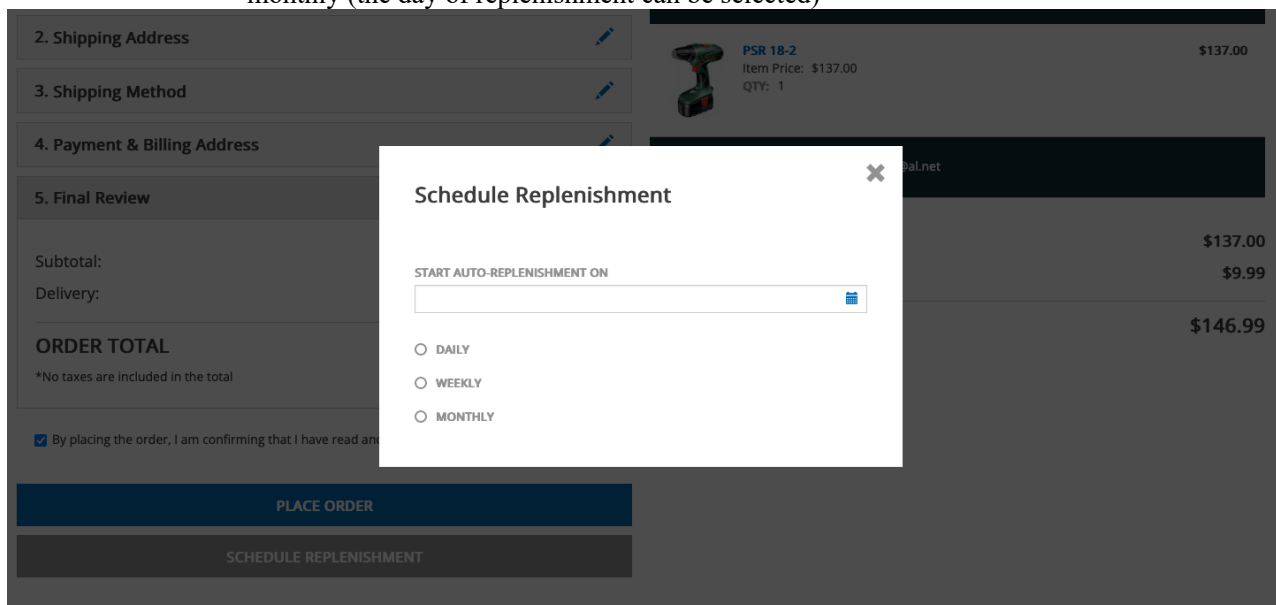


Figure 19 Schedule replenishment pop-up

On the order confirmation page, system displays:

- 1) That it is replenishment order.
- 2) Status of the replenishment (active).
- 3) Start date, frequency and the next order date.

In BackOffice (Administration Cockpit->Cronjobs) the cronjob creates.

THANK YOU FOR YOUR SCHEDULED REPLENISHMENT ORDER!

Confirmation Number: 000002BD
A copy of your order details has been sent to ulf.becker@rustic-hw.com

Order Summary		
REPLENISHMENT NO 000002BD	START Dec 4, 2023	NEXT ORDER DATE Dec 4, 2023
P.O. NO	FREQUENCY Every Monday at 00:00:00	TOTAL \$88.99
ORDER PLACED BY Mr Ulf Becker		

Figure 20 Replenishment Order Confirmation page

B2B Accelerator

WELCOME ULF
MY ACCOUNT
SIGN OUT

I'm looking for

(0 ITEMS) \$0.00

POWER DRILLS
ANGLE GRINDERS
SCREWDRIVERS
SANDERS
MEASURING & LAYOUT TOOLS
SAFETY
HAND TOOLS

HOME / MANAGE REPLENISHMENT SCHEDULE

Replenishment Orders

SORT BY:

DATE CREATED

1 Replenishments found

REPLENISHMENT NO	P.O. NO	START	SCHEDULE	NEXT ORDER DATE	DATE CREATED	EST. TOTAL	CANCEL
000002BD		Dec 4, 2023	Every Monday at 00:00:00	Dec 4, 2023	Nov 28, 2023	\$88.99	X

SORT BY:

DATE CREATED

Figure 21 Replenishment Orders page on My Account

7.2.2 Restrictions

Scenario	Outcome
Merchant deletes payment method which was linked to the replenishment order via the PayPal side	Replenishment order which was linked to the respective payment method account will be removed from the storefront and on PayPal side. Thus, Replenishment order which was linked to the respective payment method account will be canceled
Logged-in buyer removes payment method from the storefront (My Account page)	Order replenishment will be canceled because of the vault payment token deletes on the PayPal side too
Payment Intents (Authorize or Capture) will not have an impact on the replenishment flow.	
Replenishment doesn't work with PayLater.	
Replenishment doesn't work with 'Save Order' flow	

7.3 PayPal Vault flow

7.3.1 Save PayPal

After customers save their PayPal Wallet, they can select it for faster checkout. Customers won't have to enter payment details for future transactions.

To save PayPal Wallets, payers need to log in to your site, make a purchase, and remain on your site when transactions take place or save on the My Account page.

Businesses save payment methods if they want customers to check out without re-entering a payment method.

7.3.2 Change Payment Button

The following is an example of what a payer sees after they save their PayPal Wallet on your site.

When you save the PayPal payment method and enable the appropriate property in BO, the customer will be able to see the redesigned PayPal button with returning customer experience.

ORDER SUMMARY

Subtotal after
discounts: \$541.34

Shipping: \$8.99

The order total includes tax of \$26.21.

Total: \$550.33



Figure 22 Change Payment Button on the Cart Page

Also, customer can pay with a different funding source and different account.

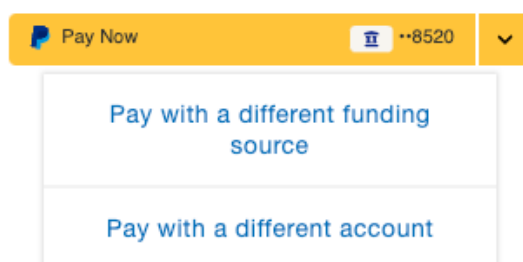


Figure 23 Pay with a different funding source and different account

Note: If the customer doesn't have a default shipping address in the address book during the express checkout, he must choose a shipping address on the PayPal pop-up and approve the order.

7.4 Apple Pay

Apple Pay works on Safari browsers and the following versions of iOS devices:

- macOS 10.14.1 and later.
- iOS 12.1 and later.

7.4.1 Set up sandbox account to accept Apple Pay

Before using the Apple Pay on your website verify that your sandbox business account supports Apple Pay:

1. Log into the [PayPal Developer Dashboard](#) and go to your sandbox account.
2. Go to Apps & Credentials.
3. Make sure you are in the PayPal sandbox environment by selecting Sandbox at the top.
4. Select or create an app.
5. Scroll down to Features and check if Apple Pay is enabled. If Apple Pay isn't enabled, select the Apple Pay checkbox and select the "Save" link to enable Apple Pay.

7.4.2 Before using the sandbox/live environment

Enable Apple Pay for your live account:

1. Log into the PayPal Developer Dashboard with your live PayPal account.
2. Select the Sandbox/Live toggle so it shows Live.
3. Go to Apps & Credentials.
4. Scroll down to Features.
5. Select the Apple Pay checkbox and select the Save link.

Create an app:

1. Log into the PayPal Developer Dashboard with your live PayPal account.
2. Select the Sandbox/Live toggle so it shows Live.
3. Go to Apps & Credentials.
4. Create an app similar to what you created in the sandbox. You don't need to log into a separate test account.

Verify any domain names in your live environment that will show an Apple Pay button. Apple Pay transactions only work on a domain and site registered to you.

Download and host live domain association file:

Host a domain association file for each high-level domain and subdomain that show the Apple Pay button.

1. Download the domain association file for your live environment. The file you can find by the [link](#).
2. Host the file on your live site for each domain and subdomain you want to register, at **/.well-known/apple-developer-merchantid-domain-association**. For example:
 - <https://example.com/.well-known/apple-developer-merchantid-domain-association>
 - <https://subdomain.example.com/.well-known/apple-developer-merchantid-domain-association>

Register your live domain on PayPal:

Add all high-level domains that show the Apple Pay button.

1. Log into the PayPal Developer Dashboard with your live PayPal account.
2. Select the Sandbox/Live toggle so it shows Live.
3. Go to Apps & Credentials.
4. Select your app.
5. Scroll down to Features > Accept payments > Advanced Credit and Debit Card Payments.
6. Check if Apple Pay is enabled. If Apple Pay isn't enabled, select the Apple Pay checkbox and select the Save link to enable Apple Pay.
7. Select the Manage link in the Apple Pay section.
8. Select Add Domain and enter your domain name.
9. Select Register Domain. If registration fails, check that the domain association file is live and saved to the right place on your live site.

After your domain is registered:

- The domain appears under Domains registered with Apple Pay.
- Buyers can make payments using the Apple Pay button on the registered website.

7.5 Notifications

Some features of the addon includes notifications. They are of three types: **Information**, **Warning**, and **Error**.

7.5.1 Connect with PayPal notifications

Notification type	Message	Description
Error	Please use another login or confirm email address	Appears when a user tries to create an account via Connect with PayPal using an unconfirmed email or phone number.
Error	You have already created a user account with this email via Connect with PayPal button. Please Log in via PayPal credentials and go to the Password section on the My Account page so that you can set the password for login with this email via Single Authentication login	Will be shown after a second attempt to authenticate via Connect with PayPal using the same email.
Error	Sorry, but your account registration is not confirmed by the merchant yet. Please wait for a confirmation email and try again. If you do not receive an email, please contact us.	Applicable for B2B only. Shown when a customer tries to log in with PayPal before account approval.
Information	Thank you! Your request was submitted to the merchant. After the merchant's confirmation, you will receive an email and will be able to log in via Connect with PayPal.	Applicable for B2B only. Customer can see after creating an account via Connect with PayPal and fill specific B2B registration form.
Information	Do you want to add your PayPal account and the address from this account to your storefront?	The customer sees this notification during the account linking flow.
Information	Do you want to add the address from this PayPal account to your storefront?	The customer sees this notification during the account linking flow when he already has a saved PayPal account.
Information	The address was successfully added	Notifies the customer that he successfully added address from PayPal account.
Information	Do you want to add the PayPal account to your storefront?	The customer sees this notification during the account linking flow when he already has a saved address.
Information	The payment method was successfully added	Notifies the customer that he successfully added payment method from PayPal account.
Information	Please note that you are logged in to your pre-existing user account with the same email.	Will be shown after the successfully linking if customer already has account with this email before linking.

Information	You can set up a password for access to this user account via Single Authentication by using the existing email and the created password. Note: In case you set up this password, you will be able to have access to your user account from both Connect with PayPal button and Single Authentication login	Registered with PayPal user who created account via Connect with PayPal button wants set up the password for login in with this email via Single Authentication.
Information	Password for access to this user account via Single Authentication has been added successfully	Registered with PayPal user who created account via Connect with PayPal button successfully set up the password.

7.5.2 Hosted fields and 3DS notifications

Notification type	Message	Description
Error	Invalid payment details	Appears when the customer tries to save invalid CC or try to place an order via invalid CC.
Error	Credit card verification failed	Will be shown after unsuccessfully 3DS verification.
Error	Something went wrong, please try again	Will be shown after trying to place an order with unsuccessfully 3DS verification.
Error	This card is unsecured. You should use another payment method.	Will be shown after unsuccessfully 3DS verification on the My Account page.
Warning	Saved account usage was disabled by merchant	Appears on the My Account page when a customer saves a card payment method and then the Vault flow is disabled by the merchant. Applicable for PayPal payment methods as well.
Information	Please note that you have a credit card in your wallet that is going to expire soon. Please check it.	Notifies user that he has card which will be expired soon.
Information	Please note that you have a credit card in your wallet that is expired. You will not be able to place the order with this credit card	Notifies user that he has expired card.
Information	The payment method was successfully added	The payment method was successfully added to the wallet.

7.5.3 Other notifications

Notification type	Message	Description
Information	This is the address provided by John Doe's Test Store. You can change it on the John Doe's Test Store site.	Appears when customer tries to change billing address from the PayPal pop-up.
Information	Sorry, MyBank can't be used for this purchase. You can still	Business user/guest/registered user goes through mark checkout

	complete your payment using a debit or credit card.	and select country which differs from "buyer.country" property
Information	You have 3 more authorization attempts left. Exceeded amount specified in the order. Authorization is successful. Order was captured. You have 2 more authorization attempts left. The transaction was successfully refunded.	Agent performs authorize/capture/refund of orders with Venmo payment method in backoffice.
Information	Payment method removed successfully	User removes payment method from My Account
Information	Verification successful. Payment method will be added to the wallet after the order is placed.	User saves the PayPal account during checkout
Warning	You reached the maximum number of saved PayPal accounts. In order to save a new one please remove the already saved one.	Shows if the number of saved PayPal Accounts reaches the limit (according to the property value)
Warning	You reached the maximum number of saved credit cards. In order to save a new credit card please remove the already saved one.	Shows if the number of saved credit cards reaches the limit (according to the property value)

7.6 Level 2 and Level 3 data for transactions

Card payment processing has 3 levels: Level 1, Level 2, and Level 3. The levels differ in the amount of information required to complete the payment as a Level 1, Level 2, or Level 3 eligible transaction. Level 1 requires less information and therefore generally incurs higher interchange fees compared to transactions processed with Level 2 and Level 3 data. Most businesses can operate at Level 1, however, when processing corporate and purchase credit cards, IC++ merchants can benefit from reduced interchange fees by providing additional transaction information called Level 2 and Level 3 data. You can find all available information about L2 L3 data [here](#).

L2 data:

- Invoice ID
- Total tax

L3 data:

- Postal code of address from where the product was shipped
- Line items (products):
 - Name
 - Quantity
 - Description
 - SKU
 - Unit amount
 - Tax
 - Unit of measure
 - Discount amount
 - Total amount
 - Shipping amount
 - Discount amount
 - Shipping address

7.7 Image URL for Purchase Units

The image URL for purchased units is generated dynamically for each of the item bought. In case the merchant uses different but local media provider (for example Amazon S3), they ought to override the attribute handler bean named as 'paypalPrettyImgUrlHandler'.

7.8 Tracking API

Customer Support employees can pass tracking data to the PayPal side. The new button is added to the customer support Backoffice view and available when capture is completed, order contains at least 1 consignment in status *Shipped* for which tracking information has not sent yet to the PayPal side.

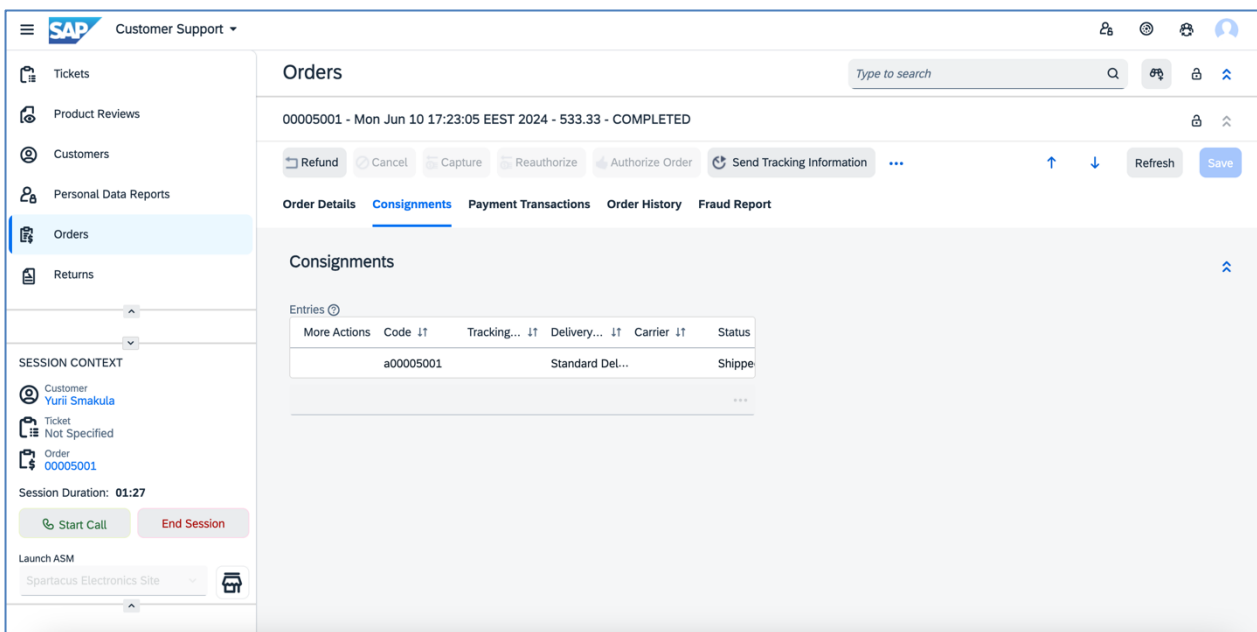


Figure 24 Send Tracking Info button in BO

For each available consignment (status=Shipped) employee can choose a transaction ID, carrier and tracking number. All available carriers are stored in our DB. Carrier contains three fields: code, name, location.

'name': used for displaying available options in Backoffice.

'code': used in request to the PayPal.

'location': can be used to filter the carriers.

The screenshot shows a 'Tracking information' pop-up form. It has four input fields: 'Consignment ID' with value 'a00005001', 'Capture transaction ID' with value '9GT53255AN0970714 - \$533.', 'Carrier' with value 'AB Custom Group', and 'Tracking Number' with value '1234-5678-9012'. A blue 'Save' button is located at the bottom right of the form.

Figure 25 Tracking Info pop-up in BO

After submitting popup carrier and trakingId are saved for corresponding consignment, the request to PayPal is sent with tracking information, success message if all data processed successful: "Shipping information sent successfully". If an error occurs during passing data: "Something went wrong, please try again later." message is shown.

Consignment is updated with Tracking ID:

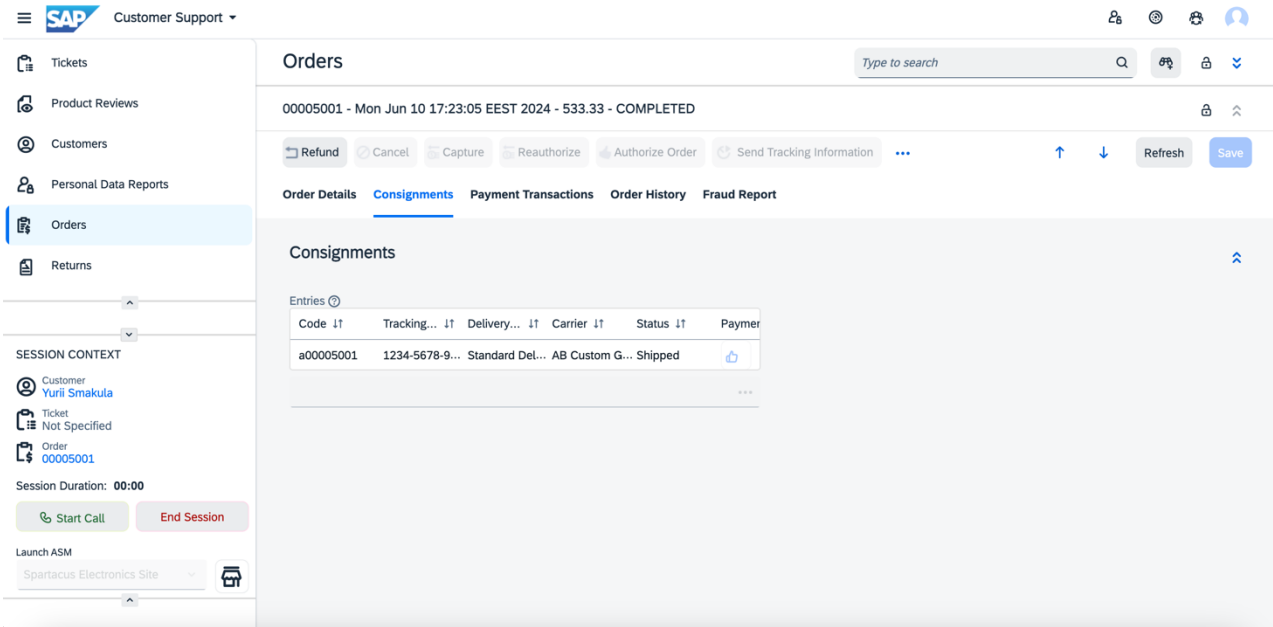


Figure 26 Tracking info attached to consignment

Items information was added to the PayPal order request and is updated if order amount is changed during checkout. Item information is sent within amount breakdown property of create/update order request and is present in the response of get order details request.

Request Response

```

"amount": {
  "currency_code": "EUR",
  "value": "10006.29",
  "breakdown": {
    "item_total": {
      "currency_code": "EUR",
      "value": "9523.81"
    },
    "shipping": {
      "currency_code": "EUR",
      "value": "5.99"
    },
    "tax_total": {
      "currency_code": "EUR",
      "value": "476.49"
    }
  }
}

```

Breakdown provides details such as total item amount, total tax amount, shipping, and discounts, if any.

Item_total	The subtotal for all items. Required if the request includes purchase_units[].items[].unit_amount. Must equal the sum of (items[].unit_amount * items[].quantity) for all items. item_total.value can not be a negative number.
shipping	The shipping fee for all items within a given purchase_unit. shipping.value can not be a negative number.

tax_total	The total tax for all items. Required if the request includes purchase_units.items.tax. Must equal the sum of (items[].tax * items[].quantity) for all items. tax_total.value can not be a negative number.
discount	The discount for all items within a given purchase_unit. discount.value can not be a negative number.

Additional information on tax configuration in Backoffice:

There are two types of taxes in Hybris - Net and Gross.

Gross - taxes included in the amount of the product

Net - taxes are separate from the amount of the product

If Gross type is enabled (set the value of Net property in Backoffice to false), then Cart total price excludes taxes as all taxes are already included in the price of each product and this amount (total price of cart) will be sent to PayPal

If Net type is enabled (set the value of Net property in Backoffice to true), then Cart's total price includes taxes, and the total amount (total price of cart + total taxes sum) will be sent to PayPal

To calculate the total price including taxes (with enabled Net property) and send it to the PayPal side additionally, the OOTB 'External Tax Calculation' property should be enabled.

Consider that when the 'External Tax Calculation' property is disabled, then the total price excluding taxes will be sent to the PayPal side.

To configure the 'Net' property and the 'External Tax Calculation' property Hybris Backoffice: Log in to Backoffice → Select Base Commerce → Basestore → Click on the store name.

Here is the example of tax calculation for every variation of each property enabled/disabled:

Net	External Tax	Sandbox / order amount captured	tax_total sent to PayPal
Net = true	EXT = true	Gross amount	Sales tax
Net = true	EXT = false	Net amount	0
Net = false	EXT = true	Gross amount	0
Net = false	EXT = false	Gross amount	0

The example of orders with same product for each scenario:

Net	External Tax	Total Amount Captured	Product Price Storefront	Tax amount	Delivery
Net = true	EXT = true	230.99	200.00	21.00	9.99
Net = true	EXT = false	209.99	200.00	0	9.99

Net = false	EXT = true	228.99	220.00	0	8.99
Net = false	EXT = false	228.99	220.00	0	8.99

Note: for different properties configuration – different delivery costs applied. Could be found in Backoffice under Price Settings → Delivery Costs.

Delivery Costs

<input type="checkbox"/>	Identifier ↓↑	Name ↓↑	Is ... ↓↑
<input type="checkbox"/>	standard-net	Standard Delivery	true
<input type="checkbox"/>	standard-gross	Standard Delivery	true
<input type="checkbox"/>	premium-net	Premium Delivery	true
<input type="checkbox"/>	premium-gross	Premium Delivery	true
<input type="checkbox"/>	free-standard-shipping		true

Figure 27 Type of Payment methods

7.9 Transaction Logs

The transaction logs were added to the transaction entry (PaymentTransactionEntryModel) pop-up for both successful and unsuccessful transaction entries.

After performing any action on order(authorize/reauthorize/capture/cancel/refund) the transaction entry is populated with additional information:

- Request field – populated for each entry
- Response field – populated for each entry
- Expiration Time – only relevant for authorization transaction entries
- Create and update time - populated for each entry
- Debug id - populated for each entry
- Failure reason – populated only for failed transactions

For Reauthorization transaction Debug Id, Request Id, Update time, Request and Response fields will be updated. Expiration time will remain the same.

To check PaymentTransactionEntry information go to Administration Cockpit -> Orders -> choose order -> Administration tab -> PaymentTransaction -> Entries

Edit item PaymentTransaction[8796290943028] ×

🔒

Refresh

Save

Unbound ⬆

Documents

+ Create Output Document

...

Code*

testuser14@test.com-ae302b94-3d78-430a-9eee-ed898983cc!

Currency ?

...

Payment Info ?

PAYPAL - ***** - 0 - 0 - Test User - 2253 Hudson ...

Payment provider

PayPal

Assigned Cockpit Item Templates

Select from List

...

Comments

Select from List

...

Entries

PaymentTransactionEntry[8796389247029] ×

+ Create Payment transaction entry

PaymentTransactionEntry[8796389247029]

...


Order

00007040 - Fri Jun 07 14:25:39 EEST 2024 - 399.99 - PA...

Planned amount ?

399.99000000

Figure 30 Transaction entry

 PayPal

Copyright ©2024 PayPal Inc. All rights reserved.

Solutions Limitation | 45

8 Order\Transaction Managements in Backoffice

8.1 Overview

PayPal plugin has custom actions which are embedded in the Backoffice Order Management flow. These actions allow managing transactions (create additional authorization, capture, void, refund and update payment statuses) created withing PayPal plugin.

Custom actions are available for CSA (Customer Support Agent) role in Backoffice and can be active depending on the payment model which was set up by merchants and transaction statuses.

The screenshot displays the SAP Backoffice interface for order management. The top navigation bar includes 'SAP Customer Support' and a search bar. The left sidebar shows navigation options: Customers, Personal Data Reports, Orders (selected), and Returns. The main content area shows the 'Order Details' tab for order '00000007' created on 'Tue Aug 15 20:23:33 UTC 2023' with a status of 'COMPLETED'. Custom actions like 'Authorize Order', 'Void', 'Multicapture', 'Partial Refund', and 'Refresh Payment Status' are visible. The 'Order Information' section shows customer details, delivery mode, and addresses. The 'Line Details' section shows a table with one entry: Product Code 300938, Product Name Photosmart E31..., Currency USD, Item Price 0.23, QTY 1, and Total Price 0.23. The 'Order Totals' section shows an order total of 8.22, shipping of 7.99, and taxes of 0.39.

Figure 31 Custom actions embedded to Backoffice (Backoffice->Orders->select respective order)

8.2 Backoffice transaction statuses

Each order in Backoffice contains a transaction status. You can review the transaction status by following the next steps: Go to Backoffice->Orders->select order->go to Payment Transaction tab->review the payment status in the Payment Status column of the Payment Transaction section.

The screenshot shows the 'Orders' list in the SAP Backoffice. The table has columns: Customer, Date/Time Created, Delivery Mode, Status, and Payment Status. The 'Payment Status' column is highlighted with a green border. The table contains 7 rows of data.

Customer	Date/Time Created	Delivery Mode	Status	Payment Status
John Smith - hybrisuser@test.com	Nov 2, 2023 3:13:59 AM	Standard Delivery[standard-gross]	Payment Captured	Refunded
John Smith - hybrisuser@test.com	Nov 2, 2023 3:19:45 AM		Cancelled	Cancelled
John Smith - hybrisuser@test.com	Nov 2, 2023 3:19:45 AM	Standard Delivery[standard-gross]	Completed	Refunded
John Smith - hybrisuser@test.com	Nov 2, 2023 3:24:25 AM	Standard Delivery[standard-gross]	Completed	Completed
John Smith - hybrisuser@test.com	Nov 2, 2023 3:26:56 AM	Standard Delivery[standard-gross]	Payment Not Captured	Pending
John Smith - hybrisuser@test.com	Nov 2, 2023 3:28:56 AM	Standard Delivery[standard-gross]	Completed	Completed
John Smith - hybrisuser@test.com	Nov 2, 2023 3:30:34 AM	Standard Delivery[standard-gross]	Completed	Completed
guest - 7e145688-024c-4f3f-a17...	Nov 2, 2023 10:11:51 AM	Standard Delivery[standard-gross]	Completed	Completed

Figure 32 Column for review the transaction statuses

A Backoffice logged-in merchant is able to see the updated payment statuses of the created transactions so that he can avoid checking these statuses in the PayPal console and can speed up managing transactions. The following statuses can be set to the transactions:

Pending: This status is updated after the authorization process is performed. It indicates that the payment is pending confirmation.

Cancelled: The status is updated to "Cancelled" after a void operation is performed. This signifies that the transaction has been canceled and the payment won't be processed.

Completed: After a successful capture operation, the status is updated to "Completed." This indicates that the payment has been processed and completed successfully.

Refunded: When a refund is performed, the status is updated to "Refunded." This shows that a refund has been initiated for the transaction.

The key to effective transaction management in Backoffice is linking these statuses with your specific orders and ensuring that the Backoffice transaction statuses align with the corresponding PayPal statuses. This integration allows merchants to have a unified view of transaction statuses, making it more convenient to track and manage orders without having to check the PayPal console separately.

8.2.1 Custom Actions with Intent Authorize

The Authorize Intent allows authorizing a payment and place funds on hold after the customer makes a payment. Authorized payments are guaranteed for up to three days but are available to capture (manually in the Hybris Backoffice) for up to 29 days. After the three-day honor period, the original authorized payment expires, and you must re-authorize the payment. A separate request should be made to capture payments on demand.

Important notes:

- This intent is not supported when there is more than one `purchase_unit` within the order.
- This intent is not supported by Local Payment Methods (LPM)\Alternative Payment Methods.

To enable the Authorize Intent please set the 'Settle Transaction Immediately' property to 'False' value. The following custom actions are available:

Scenario 1: Create order with intent Authorize:

Action	Status	Payment Status	Available actions
1. Create order	Payment Not Captured	Pending	Cancel, Capture
2. Partial Capture	Payment Not Captured	Partial Capture	Refund, Capture
3. Full Capture	Payment Captured	Completed	Refund

In case when the user puts the wrong amount to be capture, the system will show the error:

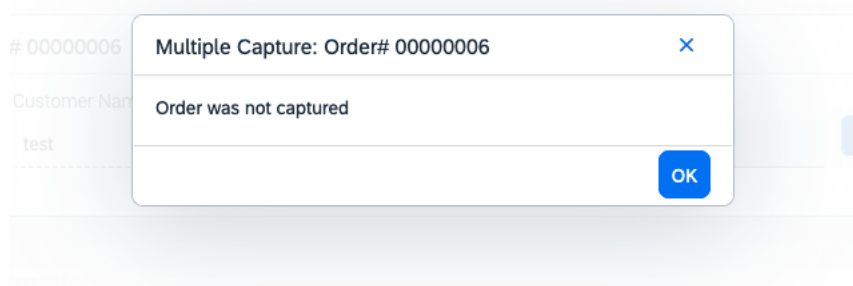


Figure 33 Capture error notification

Scenario 2: Create order with intent Authorize and make a refund:

Action	Status	Payment Status	Available actions
1. Create order	Payment Not Captured	Pending	Cancel, Capture
2. Partial Capture	Payment Not Captured	Partial Capture	Refund, Capture
3. Full Capture	Payment Captured	Completed	Refund
4. Refund first transaction	Order split	Partial Refund	Refund
5. Refund second transaction	Order split	Refunded	

In case when the user puts the wrong amount to be refunded, the system will show the error:

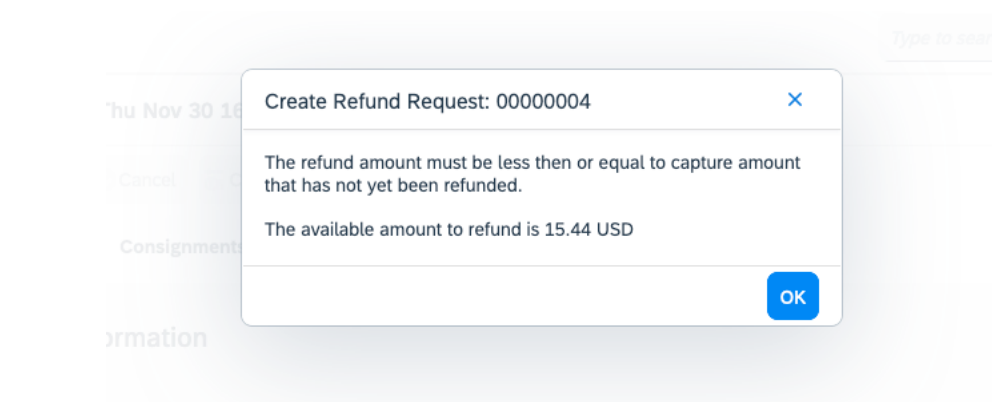


Figure 34 Refund error notification

Scenario 3: Create order with intent Authorize and cancel it:

Action	Status	Payment Status	Available actions
1. Create order	Payment Not Captured	Pending	Cancel, Capture
2. Cancel	Cancelled	Cancelled	

Scenario 4: Create order with intent Authorize and make a several refunds:

Action	Status	Payment Status	Available actions
1. Create order	Payment Not Captured	Pending	Cancel, Capture
2. Full Capture	Payment Captured	Completed	Refund
3. Partial Refund	Order Split	Partial Refund	Refund
4. Full Refund	Completed	Refunded	

Note: In case the Authorization token was expired (after 29 days), the Reauthorize button is available in BO.

8.2.2 Custom Actions with Intent Capture

The Capture Intent allows capturing payment immediately after the customer makes a payment. To enable the Capture Intent please set the 'Settle Transaction Immediately' property to 'True' value. The following custom actions are available:

Scenario 1: Main flow of order creation

Action	Status	Payment Status	Available actions
1. Create order	Order Split	Completed	Refund
Wait some time	Completed	Completed	Refund

Scenario 2: Full refund

Action	Status	Payment Status	Available actions
1. Create order	Order Split	Completed	Refund
Wait some time	Completed	Completed	Refund
2. Full Refund	Completed	Refunded	

Scenario 3: Partial refund

Action	Status	Payment Status	Available actions
1. Create order	Order Split	Completed	Refund
After some time	Completed	Completed	Refund
2. Partial Refund	Completed	Partial Refund	Refund
3. Full Refund	Completed	Refunded	

8.2.3 Custom Actions with Intent Save Order

Save Order flow allows to have the possibility to authorize transactions during a long period (up to 365 days). for merchants with a complex delivery flow which requires to have the ability to extend the period for authorization or sell orders that are not in stock

To use Save Order flow the following property should be set:en

1. set the 'PayPal Save Order Flow Enabled' property to 'True'
2. set the 'Settle Transaction Immediately' property to 'False'
3. set the 'Vault' property to 'False'. (Save Order flow doesn't work with enabled Vault)

Scenario 1: Make 2 authorizations of the order and capture them:

Action	Status	Payment Status	Available actions
1. Create order	Payment Not Authorized	O-Order Created	Authorize Order
2. Authorize Order (not full amount)	Payment Not Captured	Pending	Cancel, Capture, Authorize Order
3. Authorize Order (rest amount)	Payment Not Captured	Pending	Cancel, Capture
4. Capture first order	Payment Not Captured	Pending	Cancel, Capture
5. Capture second order	Payment Captured	Completed	Refund

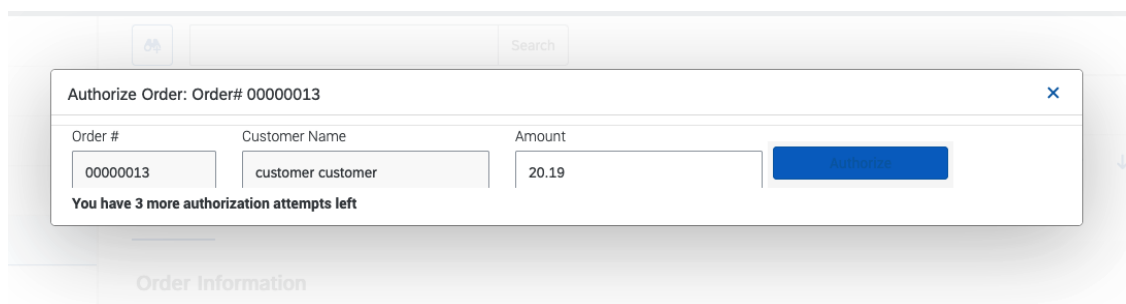


Figure 35 Three attempts to authorize order

Multiple Capture: Order# 00000013

Order #	Customer Name	Transactions	Amount	Payment Status
00000013	customer customer	0U5164102T0705813	10.00	Pending
00000009	customer customer - ok@o...			Pending
00000011	customer customer - ok@o...			Pending
00000013	customer customer - ok@o...			Pending

Figure 36 Created authorizations

Scenario 2: Make a cancelation of authorized order:

Action	Status	Payment Status	Available actions
1. Create order	Payment Not Authorized	O-Order Created	Authorize Order
2. Authorize Order (full amount)	Payment Not Captured	Pending	Cancel
3. Cancel	Cancelled	Cancelled	

9 Move to Production

NOTE: Before moving the solution to your Production system, it is highly recommended that you test and validate the solution in a test / sandbox environment.

To run the solution in production, update the following properties:

Property	Comment
paypal.username	Merchant account API user name.
paypal.password	Merchant account API password.
paypal.signature	Merchant account API signature.
paypal.seller.email	Merchant account API e-mail.
paypal.environment.configuration.type	Profile name, case sensitive, shall be set to "production" value.
paypal.client.id	Application Client ID.
paypal.secret.key	Application Secret.

10 Solutions Limitation

Limitation	Comment
“paypal.intent” property runtime configuration	The property can be changed on runtime, but placed and not processed orders will be processed with the Intent they had while being placed. To be on the safe side, it is recommended to configure it once and use that value across the cluster environment, making sure that the same and correct value is deployed for any single server instance.
Single merchant per Hybris Setup	The solution supports one PayPal merchant account per Hybris Setup.
Actions from PayPal	Any action made directly on the PayPal site will not be reflected in the Backoffice.
Billing agreement token to Vault token migration	Payment method may not convert if it was already vaulted.
Change Payment Button single-agreement issue	When we create a payment token with an already vaulted PayPal account, we receive an existing payment token. It directly influences the Change Payment button because, despite the Hybris user having a PM on his wallet on the Hybris side, he may not have this PM on the wallet from the PayPal side attached to his vault customer ID. To check if the current Payment Method is attached to the wallet of the current user or to the wallet of other user you can look and compare the properties on the Customer level (customerVaultId) and on the PayPalCreditCardPaymentInfo (PMCustomerVaultId).
Some data is missing during the displaying the vaulted PayPal Payment Method	During the vaulting the PayPal Payment Method at some cases we are not receiving some of the fields from the PayPal API -line2, city, region and cardholdername. So, until the issue won't be fixed, we are not displaying these fields at all.
The Discover credit cards are not saving to the wallet	The issue occurs only for our sandbox environment and does not affect production.
PayPal Credit/Debit Card pop-up/fields during the LPM flow	In the sandbox during the emulation of LPM payment, we are facing the behavior of appearing of PayPal Credit/Debit Card pop-up/fields time to time. It does not block the customer flow but can be unexpected.
IDEAL LPM temporary is not eligible in the sandbox	Reproducible when the currency is EUR and country of customer is NL.
Some carts are causing the error in the sandbox if use them on My Account for saving hosted fields with 3ds turned on	One of such testing cart is 4111 1111 1111 1111. The error message – “Things don't appear to be working at the moment. Please try again later”.
Customer cannot place order using Change Payment button when billing (payer) address is not coming fully from the PayPal	In some cases when the customer is going through checkout and adding the PM, the billing address can be fulfilled not fully (only country code) so that when customer is placing the order, he faced with an Internal Server Error.