

# Paidy payment processing SFRA

---

Version<22.1.1>



# Table of contents

## Details

Table of contents	2
1. Summary	4
2. Component summary	5
2-1. Features overview	5
2-2. Restrictions and limitations	5
2-3. Use cases	6
2-3-1. Standard payments	6
2-3-2. Subscription payments	9
2-4. Privacy when paying	13
3. Implementation guide	14
3-1. Uploading cartridges (UX Studio)	14
3-2. BusinessManager setup	14
3-2-1. Cartridge assignment	14
3-2-2. Importing metadata	16
3-3. Modifying cartridges on the merchant	20
3-3-1. Add payment method	20
3-3-2. CheckoutJS installation	25
3-3-3. Added paidy.js loading	25
3-3-4. Add attribute on order confirmation button	26
3-3-5. CheckoutServices.js modifications	26
3-3-6. Add payment method to email template	30
3-4. Advice	31
4. Cartridge deployment testing at merchant	33
4-1. Before beginning a test	33
4-1-1. Paidy merchant dashboard screen	33
4-1-2. Commerce Cloud Business Manager	34
4-2. PaidyCheckout test data	34
4-3. Browser compatibility	35
4-4. Test cases	35
4-4-1. Paidy standard payments	35
4-4-2. Paidy subscription payments	36
5. Webhook usage examples	37
5-1. Webhook usage	37
5-2. Detecting inconsistent payments	37
5-3. Handling inconsistencies	38
5-3-1. Processing example 1	38
5-3-2. Processing example 2	41
5-4. Webhook settings	41
6. What to do when the service is down	42
7. Revision history	43

# 1. Summary

This implementation guide supports Store Front Reference Architecture (SFRA 6.1.0) . Compatibility Mode up to 22.7 is supported.

Cartridges allow a Commerce Cloud store to make use of Paidy payments.

The developer must follow this text to install the cartridges and set them up on the online store.

Because Paidy allows for creating test payments by making use of a test API key, the user can test the settings before switching over to a live environment. For details on data needed for testing, see [4-1. Before beginning a test](#).

A cartridges must be implemented in order to begin using Paidy payments.

- "int\_paidy\_sfra" is the core cartridge that enables use of Paidy payments on the store.

\*Other cartridges are used on SiteGenesis based cartridges.

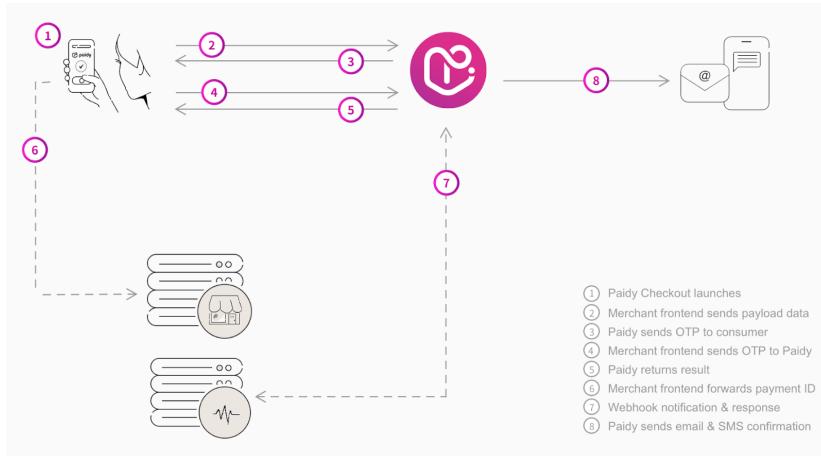
## 2.Component summary

### 2-1. Features overview

The process flow for Paidy is explained in the diagram below.

Paidy cartridges correspond to point 1-6 in this workflow.

In step 1-6, the results of authentication with Paidy are saved on the site.



### 2-2. Restrictions and limitations

- Only a “creation” process is in place for payment. Capture, Refund, Update, and Close must be implemented by the merchant.
- For first-time subscription payments, Commerce Cloud runs a creation process.  
Subsequent payments must be configured at the merchant, but the payment process can be implemented by executing the functions found in the Paidy cartridge.
- Subscription payments are designed to only work when the user is logged in.
- Confirm notifications from the Webhook on OMS; if the payment process results are incorrect, you can choose to set the order status to shippable in SFCC, or cancel the payment on Paidy. For details, please see [5. Webhook usage examples](#).
- This cartridge can only be used on Japanese language sites. (language: Japanese; currency: JPY)

### 2-3. Use cases

#### 2-3-1. Standard payments

Paidy Checkout launches when a user clicks a payment purchase button on a site.

Consumers enter the requisite information and make a Paidy payment.

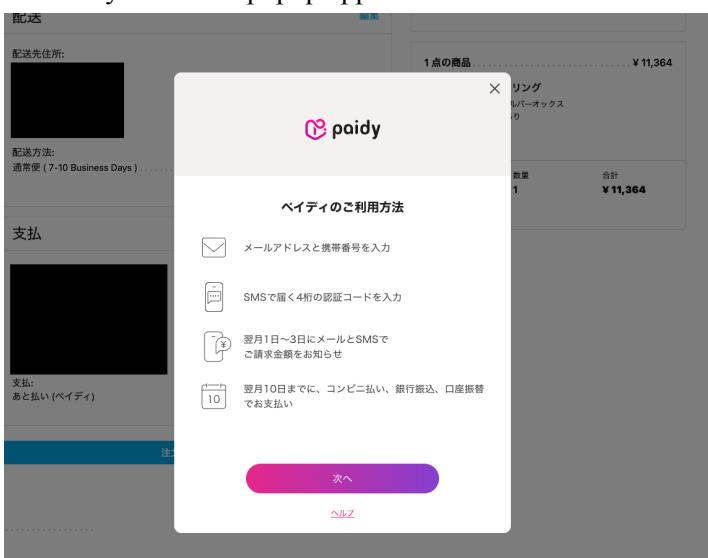
If the payment is successful, the process transitions to the product purchase confirmation screen.

#### - Payment flow

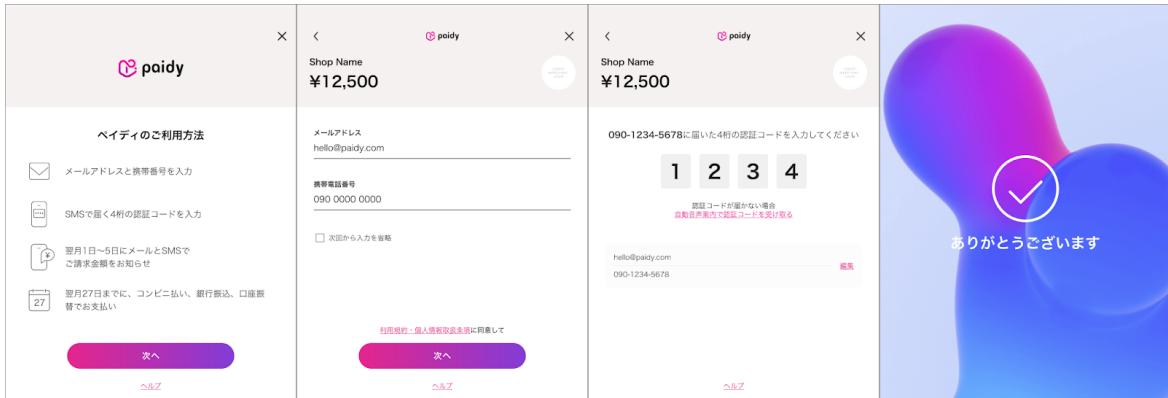
Select Paidy standard payments when selecting a payment method when checking out and proceed to confirming the order.



The Paidy Checkout popup appears.



The requisite information is filled in and Paidy payment is processed.



1 2 3 4

Shop Name  
¥12,500

Shop Name  
¥12,500

メールアドレス  
hello@paidy.com

携帯電話番号  
090 0000 0000

認証コードを入力してください

確認コードが届かない場合  
自動販売機内で認証コードを受け取る

hello@paidy.com  
090-1234-5678

ありがとうございます

If the payment succeeds, the order is complete.



Order data created on Commerce Cloud

全般	属性	支払	メモ	履歴			
<h2>注文 '00001508' の詳細</h2>							
情報:	1 件の配送先への 1 項目を含む。合計金額は ¥ 12,500,00 です。						
受取日:	11/15/22 2:32:51 AM Etc/UTC						
サイト:	Shop Name						
作成者:	顧客						
顧客:	[REDACTED]						
顧客ナンバー:	00006001	✓					
IP アドレス:	該当なし						
Eメール:	[REDACTED]						
電話:	[REDACTED]						
注文ステータス:	未処理	確認ステータス:	確認済み				
配達ステータス:	未配達	エクスポートステータス:	エクスポート可能				
配送 00002006							
個数	商品 ID	名前	製造元	税率	小売単価	課税基準	項目合計
1	013742000344M	シルバーボタンイヤリング		10.00 %	¥ 11,364.00	¥ 11,364.00	¥ 11,364.00
荷物配送料:					¥ 0.00		
合計配送料 (001):					¥ 0.00		
配送合計:					¥ 0.00		
税金合計:					¥ 1,136.00		
合計:					¥ 12,500,00		
<a href="#">Eメールの送信</a>				<a href="#">注文の印刷</a>			

Payment data created on Paidy

決済ID	顧客名	ユーザーID	金額	オーダリ日時	ステータス	キャプチャ期限
pay_Y3L2YWiaADgAG-oU TEST	[REDACTED]		¥12,500	2022-11-15 11:16:01	AUTHORIZED 取引ID: 00001506	2022-12-15 11:16:05
pay_Y23116YAAKUAkwyT TEST	[REDACTED]		¥2,024	2022-11-11 16:12:23	AUTHORIZED 取引ID: 00000702	2022-12-11 16:12:27

If the payment fails, Paidy Checkout displays a payment error message and order failure data is created on Commerce Cloud.

Payment data will not be created on Paidy.

 <p><b>ペイディのご利用方法</b></p> <p> メールアドレスと携帯番号を入力</p> <p> SMSで届く4桁の認証コードを入力</p> <p> 翌月1日～5日にメールとSMSでご請求金額をお知らせ</p> <p> 翌月27日までに、コンビニ払い、銀行振込、口座振替でお支払い</p> <p><a href="#">次へ</a></p> <p><a href="#">戻る</a></p>	 <p>Shop Name ¥12,500</p> <hr/> <p>メールアドレス hello@paddy.com</p> <hr/> <p>携帯電話番号 090 0000 0000</p> <hr/> <p><input type="checkbox"/> 次回から入力を省略</p> <p><a href="#">利用規約...個人情報収集・使用に同意して</a></p> <p><a href="#">次へ</a></p> <p><a href="#">戻る</a></p>	 <p>Shop Name ¥12,500</p> <hr/> <p>090-1234-5678に届いた4桁の認証コードを入力してください</p> <p><b>1 2 3 4</b></p> <p><small>認証コードが届かない場合 お問い合わせ窓口までお問い合わせください</small></p> <p>hello@paddy.com</p> <p>090-1234-5678</p> <p><a href="#">編集</a></p> <p><a href="#">戻る</a></p>	 <p>Shop Name ¥12,500</p> <p>今回の支払いは承認されませんでした</p> <p>恐れ入りますが、他の支払い方法を選びください。</p> <p>今回承認とならなかった理由の詳細はお伝えいたしかねますが、ご了承ください。</p> <p><a href="#">他の支払い方法を選ぶ</a></p> <p><a href="#">戻る</a></p>
--	--	--	---

\*Note that if the top-right X button is clicked on the popup, order failure data is created. Payment data will not be created on Paidy.

## Order data created on Commerce Cloud (order failure)

[全般](#) 属性 支払 メモ 履歴

### 注文 '00001509' の詳細

情報:	1 件の配送先への 1 項目を含む。合計金額は ¥ 12,500,00 です。																									
受取日:	11/15/22 2:36:57 AM Etc/UTC																									
サイト:	Shop Name																									
作成者:	顧客																									
顧客:	[REDACTED]																									
顧客ナンバー:	00008601																									
IP アドレス:	該当なし																									
Eメール:	[REDACTED]																									
電話:	[REDACTED]																									
注文ステータス:	失敗																									
配送ステータス:	未配達																									
配送	<table border="1"> <thead> <tr> <th>個数</th><th>商品 ID</th><th>名前</th><th>ステータス</th><th>製造元</th><th>税率</th><th>小売単価</th><th>課税基準</th><th>項目合計</th></tr> </thead> <tbody> <tr> <td>1</td><td>013742000344M</td><td>シルバーポタンイヤリング</td><td>CANCELLED</td><td></td><td>10.00 %</td><td>¥ 11,364.00</td><td>¥ 11,364.00</td><td>¥ 11,364.00</td></tr> </tbody> </table>								個数	商品 ID	名前	ステータス	製造元	税率	小売単価	課税基準	項目合計	1	013742000344M	シルバーポタンイヤリング	CANCELLED		10.00 %	¥ 11,364.00	¥ 11,364.00	¥ 11,364.00
個数	商品 ID	名前	ステータス	製造元	税率	小売単価	課税基準	項目合計																		
1	013742000344M	シルバーポタンイヤリング	CANCELLED		10.00 %	¥ 11,364.00	¥ 11,364.00	¥ 11,364.00																		
	<table border="1"> <tr> <td>荷物配送料: CANCELLED</td><td>¥ 0.00</td></tr> <tr> <td>合計配送料 (001):</td><td>¥ 0.00</td></tr> <tr> <td>配送合計:</td><td>¥ 0.00</td></tr> <tr> <td>税金合計:</td><td>¥ 1,136.00</td></tr> <tr> <td>合計:</td><td>¥ 12,500.00</td></tr> </table>								荷物配送料: CANCELLED	¥ 0.00	合計配送料 (001):	¥ 0.00	配送合計:	¥ 0.00	税金合計:	¥ 1,136.00	合計:	¥ 12,500.00								
荷物配送料: CANCELLED	¥ 0.00																									
合計配送料 (001):	¥ 0.00																									
配送合計:	¥ 0.00																									
税金合計:	¥ 1,136.00																									
合計:	¥ 12,500.00																									
	<a href="#">Eメールの送信</a> <a href="#">注文の印刷</a>																									

## 2-3-2. Subscription payments

Only usable by logged in users.

When clicking on the order confirmation button, Paidy Checkout launches. Consumers enter the requisite information and authenticate with Paidy. If the authentication completes correctly, a token is issued and is assigned to the order.

### - Payment flow

Select Paidy subscription payments when selecting a payment method when checking out and proceed to confirming the order.



An order confirmation screen is displayed and the order is confirmed.

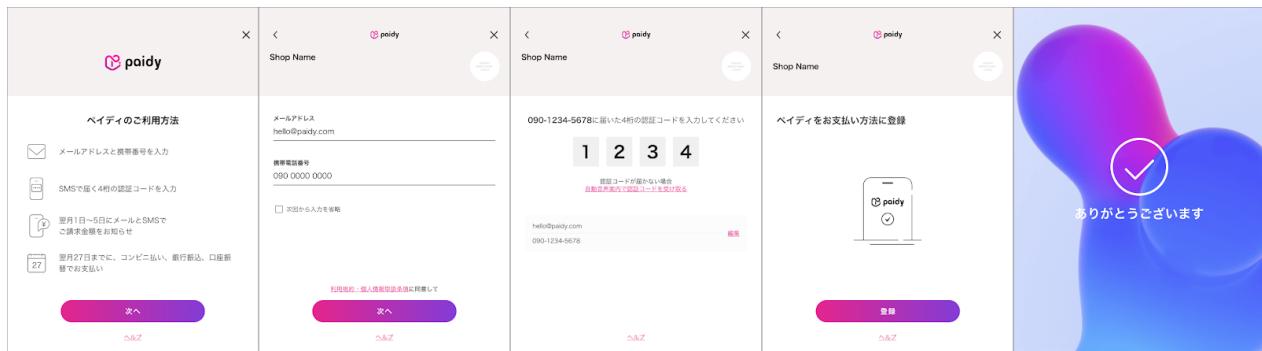
配送方法: 通常便 ( 7-10 Business Days ) .....	¥ 16	
		
単価 ¥ 2,653	数量 1	合計 <b>¥ 2,653</b>
<b>支払</b>	<a href="#">編集</a>	
請求先住所:		
支払: あと払い (ペイディ) *定期購入		
<a href="#">注文を確定する</a>		

The Paidy Checkout popup appears.



Enter the requisite information on Paidy Checkout and a token is issued.

\*If the popup is closed during the process by the user clicking the X button on the top right, the popup will no longer display, but a token can be reissued and the payment processed.



After issuing a token, the process moves to payment processing. If the payment succeeds, we transition to the order confirmation screen.



#### Order data created on Commerce Cloud

全般		属性		支払		メモ		履歴									
<strong>注文 '00001510' の詳細</strong>																	
情報: 1 件の配送先への 1 項目を含む。合計金額は ¥ 12,500,00 です。																	
受取日: 11/15/22 2:40:53 AM Etc/UTC																	
サイト: Shop Name																	
作成者: 顧客																	
顧客: 顧客ナシバード: 000006001																	
IP アドレス: 該当なし																	
Eメール: [REDACTED]																	
電話: [REDACTED]																	
注文ステータス: 未処理		確認ステータス: 確認済み		エクスポートステータス: エクスポート可能													
配送 00002007																	
個数	商品 ID	名前	製造元	税率	小売単価	課税基準	項目合計										
1	013742000344M	シルバーボタンイヤリング		10,00 %	¥ 11,364,00	¥ 11,364,00	¥ 11,364,00										
荷物配送料: ¥ 0,00																	
合計配送料 (001): ¥ 0,00																	
配達合計: ¥ 0,00																	
税金合計: ¥ 1,136,00																	
合計: ¥ 12,500,00																	
<a href="#">Eメールの送信</a> <a href="#">注文の印刷</a>																	

#### Payment data created on Paidy

支払管理							ログアウト
決済ID	顧客名	ユーザーID	金額	オーバーリ日時	ステータス	キャプチャ一覧	
pay_Y3LBNmIAAAHAAG_YI TEST	[REDACTED]		¥12,500	2022-11-15 11:40:54	AUTHORIZED 取引ID: 00001510	2022-12-15 11:40:54	-
pay_Y3L6YWIAAF4AG_JO TEST	[REDACTED]		¥12,500	2022-11-15 11:33:05	AUTHORIZED 取引ID: 00001508	2022-12-15 11:33:11	-

If payment fails, a payment error will be displayed on the order confirmation page and order failure data is created on Commerce Cloud.

Payment data will not be created on Paidy.

The screenshot shows the Salesforce Commerce Cloud Order Status page. At the top, there's a blue header bar with the text "注文手続き". Below it, the Salesforce logo and "commerce cloud" are displayed. A message box contains a message about order failure due to temporary connection errors. To the right, there's a summary table for the order, followed by sections for "Customer" and "配送".

注文の概要		
小計	.....	¥ 11,364
配送	.....	¥ 0
合計	.....	¥ 12,500

1点の商品 .....		
<b>シルバー ボタンイヤリング</b> 色: シルバーオックス 在庫あり		
単価	数量	合計
¥ 11,364	1	¥ 11,364

Customer	
Customer Information: [REDACTED]	

配送	編集
配送先住所: [REDACTED]	

## Order data created on Commerce Cloud (order failure)

全般	属性	支払	メモ	履歴
<b>注文 '000001511' の詳細</b>				
情報:	1 件の配達先への 1 項目を含む。合計金額は ¥ 12,500.00 です。			
受取日:	11/15/22 2:46:43 AM Etc/UTC			
サイト:	Shop Name			
作成者:	[REDACTED]			
顧客:	[REDACTED]			
顧客ナンバー:	0000000001			
IP アドレス:	該当なし			
Eメール:	[REDACTED]			
電話:	[REDACTED]			
注文ステータス:	失敗	確認ステータス:	未確認	
配達ステータス:	未配達	エクスポートステータス:	未エクスポート	
<b>配送</b>				
個数	商品 ID	名前	ステータス	製造元
1	013742000344M	シルバー ボタンイヤリング	CANCELLED	
				税率
				10.00 %
				小売単価
				¥ 11,364.00
				課税基準
				¥ 11,364.00
				項目合計
				¥ 11,364.00
				荷物配達料: CANCELLED
				¥ 0.00
				合計配達料 (0):
				¥ 0.00
				配達合計:
				¥ 1,136.00
				税金合計:
				¥ 12,500.00
				合計:
				¥ 12,500.00
				Eメールの送信
				注文の印刷

## 2-4. Privacy when paying

Paidy payment authentication requires entering a telephone number and e-mail address, but these are not saved on Commerce Cloud.

However, information entered upon registration and shipment and billing details are saved on Commerce Cloud.

### 3. Implementation guide

#### 3-1. Uploading cartridges (UX Studio)

The cartridge is uploaded to Commerce Cloud.

- int\_paidy
- int\_paidy\_sfra

The cartridge is for Store Front Reference Architecture (SFRA).

\*If your website is based SiteGenesis, please refer to SiteGenesis implementation guide.

Use Commerce Cloud UX-studio to upload the int\_paidy and int\_paidy\_sfra cartridges to Commerce Cloud.

#### 3-2. BusinessManager setup

Paidy payment settings are configured on the Commerce Cloud management screen (Business Manager).

##### 3-2-1. Cartridge assignment

A Paidy cartridge is assigned to the site.

Management > Sites > Site Management > Sites Under Management: Settings

## RefArch - 設定

適用をクリックして詳細を保存します。前回保存された状態に戻すには、リセットをクリックします。

インスタンスタイプ:

廃止予定。HTTP と HTTPS のホスト名の構成では、サイトのエイリアス構成の新しい機能を使用することをお薦めします ("SEO > エイリアス構成")。指定する HTTP/HTTPS ホスト名の値は、エイリアス構成でホスト名が定義されていない場合に、古い設定スタイルをサポートする目的でのみ使用されま

HTTP ホスト名:

HTTPS ホスト名:

インスタンスタイプ: すべて

カートリッジ:

有効なカートリッジのパス:

- int\_paidy\_sfra
- int\_paidy
- app\_storefront\_base
- plugin\_apple\_pay
- plugin\_facebook
- plugin\_payments
- plugin\_pinterest\_commerce
- plugin\_web\_payments
- bc\_content
- core

int\_paidy\_sfra:int\_paidy:merchant cartridge

### 3-2-2. Importing metadata

Meta\_data\_22.1.0.zip is provided in the package.

By importing this zip file, you can add services, add payment methods / payment processors, change system objects and custom objects to use Paidy payment. Details will be described later.

Administration > Site Development > Site Import & Export

Click [Choose File].



Select meta\_data\_22.1.0.zip in the package and click the [Upload] button.



After uploading the zip file, check meta\_data\_22.1.0.zip and click [Import].

選択	名前	場所	削除可能?	ファイルサイズ	最終更新日時
<input checked="" type="radio"/>	instance/meta_data_22.1.0.zip	ローカル	はい	3,48 KB	6/10/22 2:49:15 AM
<input type="radio"/>	SiteGenesis デモサイト				
<input type="radio"/>	Storefront Reference Architecture デモサイト				

Click [OK].



-Configuration when importing service metadata

Describes the service configuration when importing metadata.

Note that for services added after import

Please do not change the information described.

< Service Credentials >

Name : paidy.http.payment  
URL : https://api.paidy.com

< Service Profiles >

Name : paidy.api.prof

< Services >

Name : paidy.api.payment  
Type : HTTP  
Enabled : check on  
Service Mode : Live  
Profile : paidy.api.prof  
Credentials : paidy.http.payment

· Payment method · Configuration when importing payment processor metadata

Describes the payment method and payment processor configuration when metadata is imported.

As a point of caution, for payment methods and payment processors added after import

Please do not change the information described.

<Payment Processors >

ID : PAIDY\_STANDARD

<Payment Processors >

ID : PAIDY\_SUBSCRIPTION

< Payment Methods >

ID : PAIDY\_STANDARD

Name : あと払い（ペイディ）

Enabled : Yes

< Payment Methods >

ID : PAIDY\_SUBSCRIPTION

Name : あと払い（ペイディ）※定期購入

Enabled : Yes

- Object layout for metadata importing

Here we include instructions on the layout of objects when importing metadata.

One word of caution is that system and custom objects added after importing should not be modified.

<Order>

Management > Site Development > System Object Types > Order - Attribute Definitions

System objects: Order is an object that corresponds to order data

Order is given the below definitions as custom attributes.

paidyPaymentId: saves the ID issued when making a Paidy payment.

paidyToken: saves the token value used when using subscription payments.

すべて選択	ID	名前
<input type="checkbox"/>	<a href="#">paidyPaymentId</a>	
<input type="checkbox"/>	<a href="#">paidyToken</a>	

#### <Site Preferences>

Management > Site Development > System Object Types > Site Preferences > Attribute Definitions  
System object: Site Preferences is an object that corresponds to site settings.

The below is defined as a custom attribute in Site Preferences.

paidy\_api\_key: public key data used to connect to the Paidy API.

This can be confirmed on the Paidy merchant management screen.

For a test, save the public key used for testing purposes.

paidy\_enabled: Set to use or not use Paidy cartridges.

If turn it off, paidy payments will not be available as a payment method.

paidy\_logo\_url: the URL of a logo image to be displayed on the Paidy Checkout application.

If nothing is set, the Paidy logo will be shown.

paidy\_secret\_key: secret key data used to connect to the Paidy API.

This can be confirmed on the Paidy merchant management screen.

When testing, save the secret key used for testing purposes.

- Paidy merchant management screen (details found in section 4-1-1)

本番用APIキー	
パブリックキー	<input type="text" value="pk_live_"/>
シークレットキー	<input type="text" value="sk_live_"/>
テスト用APIキー	
パブリックキー	<input type="text" value="pk_test_"/>
シークレットキー	<input type="text" value="sk_test_"/>

paidy\_service\_name: key data used to call the information for communicating with Paidy as registered in BM.

This is fixed to paidy.api.payment.

paidy\_store\_name: the shop name displayed on Paidy Checkout application headers, MyPaidy, and the merchant dashboard.

paidy\_token\_description (optional): specifies descriptions of tokens defined by the merchant.

<input type="checkbox"/>	<a href="#"><u>paidy_api_key</u></a>
<input type="checkbox"/>	<a href="#"><u>paidy_enabled</u></a>
<input type="checkbox"/>	<a href="#"><u>paidy_logo_url</u></a>
<input type="checkbox"/>	<a href="#"><u>paidy_secret_key</u></a>
<input type="checkbox"/>	<a href="#"><u>paidy_service_name</u></a>
<input type="checkbox"/>	<a href="#"><u>paidy_store_name</u></a>
<input type="checkbox"/>	<a href="#"><u>paidy_token_description</u></a>

#### <Customer Profile>

Management > Site Development > System Object Types > Customer Profile > Attribute Definitions  
 System object: Customer Profile is an object relating to customer data.

The below is defined as a custom attribute on the Customer Profile.

paidyToken: token data used for subscription payments. Saved when issuing a token.

すべて選択	ID	名前
<input type="checkbox"/>	<a href="#"><u>paidyToken</u></a>	

#### <Custom site environment settings>

Merchant Tools > Site Environment Settings > Custom Site Environment Settings Group

Once metadata is successfully imported, a group with the Paidy ID will be created in the custom environment settings group.

ID	名前	説明	環境設定	サイト全体で表示
Paidy	Paidy	6	6	<a href="#">表示</a>

Clicking Paidy displays configuration screens for each value. Enter the requisite information and click the Save button at the top right.

The screenshot shows the Salesforce Marketing Cloud RefArch interface for configuring a custom site environment. The top navigation bar includes links for Sandbox (サンドボックス), Cache, Search Indexes, Code Versions, Cartridge Path, and Toolkit. The main content area is titled 'Paidy' and contains a table of configuration parameters:

名前	値	デフォルト値
paidy_enabled (paidy_enabled)	はい	はい
paidy_api_key (paidy_api_key) (文字列)	pk_test_XXXXXXXXXX	サイト全体で編集
paidy_logo_url (paidy_logo_url) (文字列)	https://paidy.com/dist/paidy-logo-noshadow.svg	サイト全体で編集
paidy_secret_key (paidy_secret_key) (文字列)	sk_test_XXXXXXXXXX	サイト全体で編集
paidy_store_name (paidy_store_name) (文字列)	PaidyDemoStore	サイト全体で編集
paidy_service_name (paidy_service_name) (文字列)	paidy.api.payment	サイト全体で編集
paidy_token_description (paidy_token_description) (文字列)		サイト全体で編集

At the bottom of the page, there are copyright notices: © 2020 salesforce.com, inc. All Rights Reserved. and RefArch タイムゾーン: 協定世界時 | インスタンスのタイムゾーン: アメリカ東部夏時間 | バージョン: 20.5 最終更新日時: 5月 5, 2020 (互換モード: 19.10).

### 3-3. Modifying cartridges on the merchant

#### 3-3-1. Add payment method

In order to display Paidy by payment method, write it in the template of the payment page.

By writing this script, you can:

- Selection tab added to payment methods
- The explanation for consumers is displayed when Paidy payment is selected
- The type of Paidy payment selected by the payment method is displayed on the order confirmation screen when selecting Paidy payment.

To embed a template, modify the following file.

- [app\_storefront\_base]/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsTabs.isml
- Add the below after </isif>.

```
.....  

<isif condition="${paymentOption.ID === 'PAIDY_STANDARD'}">  

  <isinclud template="checkout/billing/paymentOptions/paidyStandardTab" />  

</isif>  

<isif condition="${paymentOption.ID === 'PAIDY_SUBSCRIPTION'}">  

  <isinclud template="checkout/billing/paymentOptions/paidySubscriptionTab" />  

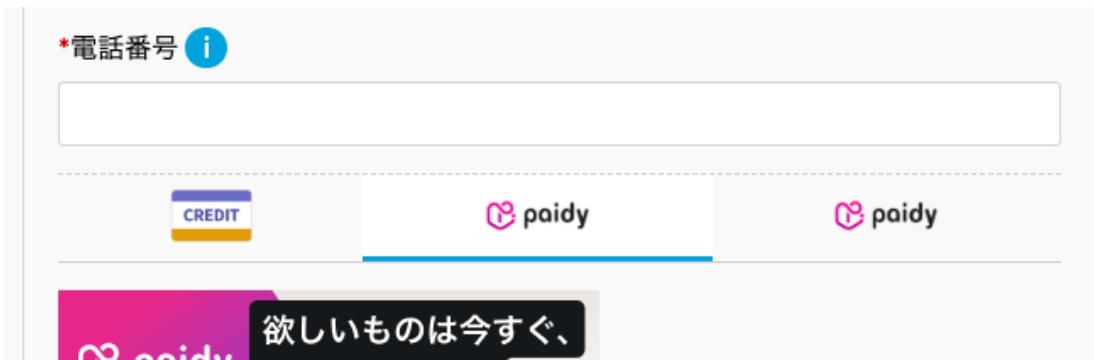
</isif>  

.....
```

```

<isloop items="${pdict.order.billing.payment.applicablePaymentMethods}" var="paymentOption">
  <isif condition="${paymentOption.ID === 'CREDIT_CARD'}">
    |   <isinclude template="checkout/billing/paymentOptions/creditCardTab" />
  </isif>
  <isif condition="${paymentOption.ID === 'PAIDY_STANDARD'}">
    |   <isinclude template="checkout/billing/paymentOptions/paidyStandardTab" />
  </isif>
  <isif condition="${paymentOption.ID === 'PAIDY_SUBSCRIPTION'}">
    |   <isinclude template="checkout/billing/paymentOptions/paidySubscriptionTab" />
  </isif>
</isloop>

```



- [app\_storefront\_base]/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsContent.isml  
Add the below after </isif>.

```

<isif condition="${paymentOption.ID === 'PAIDY_STANDARD'}">
  <isinclude template="checkout/billing/paymentOptions/paidyStandardContent" />
</isif>
<isif condition="${paymentOption.ID === 'PAIDY_SUBSCRIPTION'}">
  <isinclude template="checkout/billing/paymentOptions/paidySubscriptionContent" />
</isif>

```

```

<isloop items="${pdict.order.billing.payment.applicablePaymentMethods}" var="paymentOption">
  <isif condition="${paymentOption.ID === 'CREDIT_CARD'}">
    |   <isinclude template="checkout/billing/paymentOptions/creditCardContent" />
  </isif>
  <isif condition="${paymentOption.ID === 'PAIDY_STANDARD'}">
    |   <isinclude template="checkout/billing/paymentOptions/paidyStandardContent" />
  </isif>
  <isif condition="${paymentOption.ID === 'PAIDY_SUBSCRIPTION'}">
    |   <isinclude template="checkout/billing/paymentOptions/paidySubscriptionContent" />
  </isif>
</isloop>

```



• [app\_storefront\_base]/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsSummary.issml  
Add the below after </isif> and </div>.

(omission)

```
<isif condition="${payment.paymentMethod === 'CREDIT_CARD'}">
  <isinclude template="checkout/billing/paymentOptions/creditCardSummary" />
</isif>
<isif condition="${payment.paymentMethod === 'PAIDY_STANDARD'}">
  <isinclude template="checkout/billing/paymentOptions/paidyStandardSummary" />
</isif>
<isif condition="${payment.paymentMethod === 'PAIDY_SUBSCRIPTION'}">
  <isinclude template="checkout/billing/paymentOptions/paidySubscriptionSummary" />
</isif>
</isloop>
</div>
<div class="payment-details-summary paidy-standard" style="display: none;">
  <isinclude template="checkout/billing/paymentOptions/paidyStandardSummary" />
</div>
<div class="payment-details-summary paidy-subscription" style="display: none;">
  <isinclude template="checkout/billing/paymentOptions/paidySubscriptionSummary" />
</div>
```

```

<div class="payment-details">
  <isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
    <isif condition="${payment.paymentMethod === 'CREDIT_CARD'}">
      |   <isinclude template="checkout/billing/paymentOptions/creditCardSummary" />
    </isif>
    <isif condition="${payment.paymentMethod === 'PAIDY_STANDARD'}">
      |   <isinclude template="checkout/billing/paymentOptions/paidyStandardSummary" />
    </isif>
    <isif condition="${payment.paymentMethod === 'PAIDY_SUBSCRIPTION'}">
      |   <isinclude template="checkout/billing/paymentOptions/paidySubscriptionSummary" />
    </isif>
  </isloop>
</div>
<div class="payment-details-summary paidy-standard" style="display: none;">
  <isinclude template="checkout/billing/paymentOptions/paidyStandardSummary" />
</div>
<div class="payment-details-summary paidy-subscription" style="display: none;">
  <isinclude template="checkout/billing/paymentOptions/paidySubscriptionSummary" />
</div>

```

支払

請求先住所:

支払:  
あと払い (ペイディ)

編集

注文を確定する

- [app\_storefront\_base]/cartridge/scripts/checkout/checkoutHelpers.js
- Add the below function.

```

.....
/***
 * Selected payment instrument
 * @param {dw.order.Order} order - The order object to be placed
 * @returns {string|null} selected payment instrument
 */
function selectedPaymentInstrument(order) {
  if (order.billing.payment && order.billing.payment.selectedPaymentInstruments
    && order.billing.payment.selectedPaymentInstruments.length > 0) {
    return order.billing.payment.selectedPaymentInstruments[0].paymentMethod;
  } else {
    return null;
}

```

```

        }
    }
(omission)
setGift: setGift,
selectedPaymentInstrument: selectedPaymentInstrument
};

```

```

728@/** ↓
729 * Selected payment instrument ↓
730 * @param {dw.order.Order} order - The order object to be placed ↓
731 * @returns {string|null} selected payment instrument ↓
732 */
733@function selectedPaymentInstrument(order) { ↓
734     if (order.billing.payment && order.billing.payment.selectedPaymentInstruments ↓
735         && order.billing.payment.selectedPaymentInstruments.length > 0) { ↓
736         return order.billing.payment.selectedPaymentInstruments[0].paymentMethod; ↓
737     } else { ↓
738         return null; ↓
739     } ↓
740 } ↓
741 ↓
742 module.exports = {
743     getFirstNonDefaultShipmentWithProductLineItems: getFirstNonDefaultShipmentWithProductLineItems,
744     ensureNoEmptyShipments: ensureNoEmptyShipments, ↓
745     getProductLineItem: getProductLineItem, ↓
746     isShippingAddressInitialized: isShippingAddressInitialized, ↓
747     prepareCustomerForm: prepareCustomerForm, ↓
748     prepareShippingForm: prepareShippingForm, ↓
749     prepareBillingForm: prepareBillingForm, ↓
750     copyCustomerAddressToShipment: copyCustomerAddressToShipment, ↓
751     copyCustomerAddressToBilling: copyCustomerAddressToBilling, ↓
752     copyShippingAddressToShipment: copyShippingAddressToShipment, ↓
753     copyBillingAddressToBasket: copyBillingAddressToBasket, ↓
754     validateFields: validateFields, ↓
755     validateCustomerForm: validateCustomerForm, ↓
756     validateShippingForm: validateShippingForm, ↓
757     validateBillingForm: validateBillingForm, ↓
758     validatePayment: validatePayment, ↓
759     validateCreditCard: validateCreditCard, ↓
760     calculatePaymentTransaction: calculatePaymentTransaction, ↓
761     recalculateBasket: recalculateBasket, ↓
762     handlePayments: handlePayments, ↓
763     createOrder: createOrder, ↓
764     placeOrder: placeOrder, ↓
765     savePaymentInstrumentToWallet: savePaymentInstrumentToWallet, ↓
766     getRenderedPaymentInstruments: getRenderedPaymentInstruments, ↓
767     sendConfirmationEmail: sendConfirmationEmail, ↓
768     ensureValidShipments: ensureValidShipments, ↓
769     setGift: setGift, ↓
770     selectedPaymentInstrument: selectedPaymentInstrument ↓
771 }; ↓
772

```

• [app\_storefront\_base]/cartridge/templates/default/checkout/billing/paymentOptions.isml  
Add the below the first line of the file (after div.form-nav) and attribute in div.form-nav

```

<iscript>
    <isset name="paymentMethod"
value="${require('*!/cartridge/scripts/checkout/checkoutHelpers').selectedPaymentInstrument(pdct.order)}"
scope="page"/>
</iscript>
<div class="form-nav billing-nav payment-information"
    data-payment-method-id="CREDIT_CARD"
    data-selected-payment-method="${paymentMethod}"

```

```

data-is-new-payment="${pdict.customer.registeredUser && pdict.customer.customerPaymentInstruments.length ? false : true}"
>

.....
1 <isset name="paymentMethod" value="${require('*/*cartridge/scripts/checkout/checkoutHelpers').selectedPaymentInstrument(pdict.order)}" scope="page"/> ↓
2 <div class="form-nav billing-nav payment-information" ↓
3   data-payment-method-id="CREDIT_CARD" ↓
4   data-selected-payment-method="${paymentMethod}" ↓
5   data-is-new-payment="${pdict.customer.registeredUser && pdict.customer.customerPaymentInstruments.length ? false : true}" ↓
6 > ↓
7   <ul class="nav nav-tabs nav-fill payment-options" role="tablist"> ↓
8     <isinclude template="checkout/billing/paymentOptions/paymentOptionsTabs" /> ↓
9   </ul> ↓
10 </div> ↓
11 <div class="credit-card-selection-new" > ↓
12   <div class="tab-content"> ↓
13     <isinclude template="checkout/billing/paymentOptions/paymentOptionsContent" /> ↓
14   </div> ↓
15 </div> ↓

```

### 3-3-2. CheckoutJS installation

In order to process payment, the Checkout JS script is embedded in the payment page template. Including this script will call Paidy Checkout.

To embed the template, modify the following file.

- [app\_storefront\_base]/cartridge/templates/default/common/layout/checkout.isml

Add the below after <head>

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <script type="text/javascript" src="https://apps.paidy.com/"></script>
    <!--[if gt IE 9]><!-->
      <!--[if gt IE 9]><!-->
        <isinclude sf-toolkit="off" template="/common/scripts" /> ↓

```

### 3-3-3. Added paidy.js loading

Add the paidy.js required for the payment process below.

- [app\_storefront\_base]/cartridge/templates/default/checkout/checkout.isml

Add the below after assets.addJs('/js/checkout.js');

```

<isscript>
  var assets = require('*/*cartridge/scripts/assets.js');
  assets.addJs('/js/checkout.js');
  assets.addJs('/js/paidy.js');
  assets.addCss('/css/checkout/checkout.css');
</isscript>

```

```
<isdecorate template="common/layout/checkout">↓
```

```
<!-- Load Static Assets -->↓  
<!-->↓  
<script>↓  
  var assets = require('*cartridge/scripts/assets.js');↓  
  assets.addJs('/js/checkout.js');↓  
  assets.addJs('/js/paidy.js');↓  
  assets.addCss('/css/checkout/checkout.css');↓  
</script>↓
```

### 3-3-4. Add attribute on order confirmation button

Define the information required for the payment process as the attribute of the settlement confirmation button.

- [app\_storefront\_base]/cartridge/templates/default/checkout/checkout.isml

Add the below in button.place-order

```
.....  
<button class="btn btn-primary btn-block place-order"  
data-action="${URLUtils.url('CheckoutServices-PlaceOrder')}"  
  data-get-paidy-standard-config="${URLUtils.url('PaidyStandard-GetPaidyConfig')}"  
  data-paidy-standard-fail-order="${URLUtils.url('PaidyStandard-FailOrder')}"  
  data-paidy-standard-place-order="${URLUtils.url('PaidyStandard-PlaceOrder')}"  
  data-get-paidy-subscription-config="${URLUtils.url('PaidySubscription-GetPaidyConfig')}"  
  data-paidy-subscription-set-token="${URLUtils.url('PaidySubscription-SetPaidyToken')}" type="submit"  
name="submit" value="place-order">>${Resource.msg('button.place.order', 'checkout', null)}  
</button>  
.....
```

```
<button class="btn btn-primary btn-block submit-payment" type="submit" name="submit" value="submit-payment">  
  ${Resource.msg('button.next.place.order', 'checkout', null)}  
</button>
```

```
<button class="btn btn-primary btn-block place-order" data-action="${URLUtils.url('CheckoutServices-PlaceOrder')}"  
  data-get-paidy-standard-config="${URLUtils.url('PaidyStandard-GetPaidyConfig')}"  
  data-paidy-standard-fail-order="${URLUtils.url('PaidyStandard-FailOrder')}"  
  data-paidy-standard-place-order="${URLUtils.url('PaidyStandard-PlaceOrder')}"  
  data-get-paidy-subscription-config="${URLUtils.url('PaidySubscription-GetPaidyConfig')}"  
  data-paidy-subscription-set-token="${URLUtils.url('PaidySubscription-SetPaidyToken')}"  
  type="submit" name="submit" value="place-order">>${Resource.msg('button.place.order', 'checkout', null)}
```

```
</button>
```

```
</div>
```

```
</div>
```

### 3-3-5. CheckoutServices.js modifications

Select Paidy as the payment method, and do not process credit card payment in Ajax processing when transitioning to the order confirmation screen.

- [app\_storefront\_base]/cartridge/controllers/CheckoutServices.js

Add the below.

```
.....
```

```
server.post(
```

```
  'SubmitPayment',
```

```
  server.middleware.https,
```

```
  csrfProtection.validateAjaxRequest,
```

```
  function (req, res, next) {
```

(omission)

```
var validationHelpers = require('*cartridge/scripts/helpers/basketValidationHelpers');
var cartHelpers = require('*cartridge/scripts/cart/cartHelpers');
var paidyHelpers = require('*cartridge/scripts/paidy/paidyHelpers');
var currentBasket = BasketMgr.getCurrentBasket();
(omission)
}
paidyHelpers.resetPaymentForms(currentBasket);
var billingAddress = currentBasket.billingAddress;
var billingForm = server.forms.getForm('billing');
```

.....

```
40  /**↓
41  * Handle Ajax payment (and billing) form submit↓
42  */↓
43 server.post(↓
44   'SubmitPayment', ↓
45   server.middleware.https, ↓
46   csrfProtection.validateAjaxRequest, ↓
47   function (req, res, next) {↓
48     var PaymentManager = require('dw/order/PaymentMgr');↓
49     var HookManager = require('dw/system/HookMgr');↓
50     var Resource = require('dw/web/Resource');↓
51     var COHelpers = require('*cartridge/scripts/checkout/checkoutHelpers');↓
52     (中略)
141     var validationHelpers = require('*cartridge/scripts/helpers/basketValidationHelpers');↓
142     var cartHelpers = require('*cartridge/scripts/cart/cartHelpers');↓
143     var paidyHelpers = require('*cartridge/scripts/paidy/paidyHelpers');↓
144     ↓
145     var currentBasket = BasketMgr.getCurrentBasket();↓
146     var validatedProducts = validationHelpers.validateProducts(currentBasket);↓
147     ↓
148     var billingData = res.getViewData();↓
149     ↓
150     if (!currentBasket || validatedProducts.error) {↓
151       delete billingData.paymentInformation;↓
152     ↓
153       res.json({↓
154         error: true,↓
155         cartError: true,↓
156         fieldErrors: [],↓
157         serverErrors: [],↓
158         redirectUrl: URLUtils.url('Cart>Show').toString()↓
159       });↓
160       return;↓
161     }↓
162     ↓
163     paidyHelpers.resetPaymentForms(currentBasket);↓
164     ↓
165     var billingAddress = currentBasket.billingAddress;↓
166     var billingForm = server.forms.getForm('billing');↓
167     var paymentMethodID = billingData.paymentMethod.value;↓
```

Add a process to be interrupted before proceeding the payment process when creating an order from Paidy with Ajax  
Add the below.

.....

```
server.post('PlaceOrder', server.middleware.https, function (req, res, next) {
  var BasketMgr = require('dw/order/BasketMgr');
(omission)
  var addressHelpers = require('*cartridge/scripts/helpers/addressHelpers');
var paidyHelpers = require('*cartridge/scripts/paidy/paidyHelpers');
  var currentBasket = BasketMgr.getCurrentBasket();
(omission)
  if (handlePaymentResult.error) {
    res.json({
      error: true,
      errorMessage: Resource.msg('error.technical', 'checkout', null)
```

```
    });
    return next();
}
if (paidyHelpers.isPaidyStandard(currentBasket.paymentInstrument.paymentMethod)){
    var responseJson = require('*/*cartridge/scripts/paidy/standard/authorize')
        .getConfirmationPaidyJSON(currentBasket.paymentInstrument.paymentMethod, order.getCustomer(),
order);
    res.json(responseJson);
    return next();
} else if (paidyHelpers.isPaidySubscription(currentBasket.paymentInstrument.paymentMethod)){
    var paidySubscriptionAuthorize = require(
        '*/*cartridge/scripts/paidy/subscription/authorize')
        .Authorize(order, req.querystring.paidyToken);
    if (paidySubscriptionAuthorize.error) {
        Transaction.wrap(function () { OrderMgr.failOrder(order); });
        res.json({
            error: true,
            errorMessage: Resource.msg('error.technical', 'checkout', null)
        });
        return next();
    }
}
var fraudDetectionStatus = hooksHelper('app.fraud.detection', 'fraudDetection', currentBasket,
require('*/*cartridge/scripts/hooks/fraudDetection').fraudDetection);
.....
```

```

40 server.post('PlaceOrder', server.middleware.https, function (req, res, next) {↓
41   var BasketMgr = require('dw/order/BasketMgr');↓
42   var OrderMgr = require('dw/order/OrderMgr');↓
43   var Resource = require('dw/web/Resource');↓
44   var Transaction = require('dw/system/Transaction');↓
45   var URLUtils = require('dw/web/URLUtils');↓
46   var basketCalculationHelpers = require('*cartridge/scripts/helpers/basketCalculationHelpers');↓
47   var hooksHelper = require('*cartridge/scripts/helpers/hooks');↓
48   var COHelpers = require('*cartridge/scripts/checkout/checkoutHelpers');↓
49   var validationHelpers = require('*cartridge/scripts/helpers/basketValidationHelpers');↓
50   var addressHelpers = require('*cartridge/scripts/helpers/addressHelpers');↓
51   ↓
52   var paidyHelpers = require('*cartridge/scripts/paidy/paidyHelpers');↓
53   ↓
54   var currentBasket = BasketMgr.getCurrentBasket();↓
55   ↓

```

(中略)

```

81     errorMessage: Resource.msg('error.technical', 'checkout', null)↓
82   );↓
83   return next();↓
84 }↓
85 ↓
86 if (paidyHelpers.isPaidyNormal(currentBasket.paymentInstrument.paymentMethod)) {↓
87   var responseJson = require('*cartridge/scripts/paidy/normal/authorize')↓
88     .getConfirmationPaidyJSON(currentBasket.paymentInstrument.paymentMethod, order.getCustomer(), order);↓
89   res.json(responseJson);↓
90   return next();↓
91 } else if (paidyHelpers.isPaidyRegular(currentBasket.paymentInstrument.paymentMethod)) {↓
92   var paidyRegularAuthorize = require('*cartridge/scripts/paidy/regular/authorize')↓
93     .Authorize(order, req.querystring.paidyToken);↓
94   if (paidyRegularAuthorize.error) {↓
95     Transaction.wrap(function () { OrderMgr.failOrder(order); });↓
96     res.json({↓
97       error: true,↓
98       errorMessage: Resource.msg('error.technical', 'checkout', null)↓
99     });↓
100    return next();↓
101  }↓
102 }↓
103 ↓
104 var fraudDetectionStatus = hooksHelper('app.fraud.detection', 'fraudDetection', currentBasket, require('*car
105 if (fraudDetectionStatus.status === 'fail') {↓

```

- [app\_storefront\_base]/cartridge/scripts/cart/cartHelpers.js

Add the below function.

.....  
(omission)

```

/***
 * Deletes multiple payment instruments.
 *
 * @transactional
 * @alias module:models/CartModel~CartModel/removePaymentInstruments
 * @param {dw.order.Basket} basket - the target Basket object
 * @param {dw.util.Collection} paymentInstruments - The payment instruments to remove.
 */

```

```
function removePaymentInstruments(currentBasket, paymentInstruments) {

```

```
  collections.forEach(paymentInstruments, function(item) {
    currentBasket.removePaymentInstrument(item);
  });
}
```

(omission)

updateBundleProducts: updateBundleProducts,

**removePaymentInstruments: removePaymentInstruments**

```

};

.....

    /**
     * Deletes multiple payment instruments.
     */
    @transactional
    @Alias("models/CartModel~CartModel/removePaymentInstruments")
    @Param("dw.order.Basket") basket - the target Basket object
    @Param("dw.util.Collection") paymentInstruments - The payment instruments to remove.
 */
function removePaymentInstruments(currentBasket, paymentInstruments) {
    collections.forEach(paymentInstruments, function(item) {
        currentBasket.removePaymentInstrument(item);
    });
}

module.exports = {
    addLineItem,
    addProductToCart,
    checkBundledProductCanBeAdded: checkBundledProductCanBeAdded,
    ensureAllShipmentsHaveMethods: ensureAllShipmentsHaveMethods,
    getQtyAlreadyInCart: getQtyAlreadyInCart,
    getNewBonusDiscountLineItem: getNewBonusDiscountLineItem,
    getExistingProductLineItemInCart: getExistingProductLineItemInCart,
    getExistingProductLineItemsInCart: getExistingProductLineItemsInCart,
    getMatchingProducts: getMatchingProducts,
    allBundleItemsSame: allBundleItemsSame,
    hasSameOptions: hasSameOptions,
    BONUS_PRODUCTS_PAGE_SIZE: BONUS_PRODUCTS_PAGE_SIZE,
    updateBundleProducts: updateBundleProducts,
    removePaymentInstruments
};


```

### 3-3-6. Add payment method to email template

Add a payment method to the email sent when payment is completed.

- [app\_storefront\_base]/cartridge/templates/default/checkout/confirmation/confirmationEmail.isml

Add the below after </isif> in isloop of Payment information

(omission)

```

<span>${Resource.msg('msg.card.type.ending', 'confirmation', null)}
${payment.expirationMonth}/${payment.expirationYear}</span>
</div>
</isif>
<isif condition="${payment.paymentMethod === 'PAIDY_STANDARD'}">
    <isinclude template="checkout/billing/paymentOptions/paidyStandardSummary" />
</isif>
<isif condition="${payment.paymentMethod === 'PAIDY_SUBSCRIPTION'}">
    <isinclude template="checkout/billing/paymentOptions/paidySubscriptionSummary" />
</isif>
</isloop>
</div>
</div>

```

```

<isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
  <isif condition="${payment.paymentMethod === 'CREDIT_CARD'}">
    <div>
      <span>${Resource.msg('msg.payment.type.credit', 'confirmation', null)}<br>
        ${payment.type}</span>
    </div>
    <div>
      ${payment.maskedCreditCardNumber}
    </div>
    <div>
      <span>${Resource.msg('msg.card.type.ending', 'confirmation', null)} ${payment.expirationMonth}/${payment.year}</span>
    </div>
  </isif>
  <isif condition="${payment.paymentMethod === 'PAIDY_STANDARD'}">
    <isinclude template="checkout/billing/paymentOptions/paidyStandardSummary" />
  </isif>
  <isif condition="${payment.paymentMethod === 'PAIDY_SUBSCRIPTION'}">
    <isinclude template="checkout/billing/paymentOptions/paidySubscriptionSummary" />
  </isif>
</isloop>

```

### 3-4. Advice

- Notes on use of subscription purchases

If you followed the steps to add a payment method as indicated in section 3-3-1, by default, you will have two separate entries -- standard payments and subscription payments.



If there are means to detect whether subscription payments have been selected when transitioning to the payment selection screen, CSS or other means can be used to restrict the display to only show 「あと払い(ペイディ)」.

- Notes on tokens

int\_paidy\_sfra does not use PaidyToken for customer information.

Tokens can be checked via order data.

Merchant Tools > Order > Order > Order Numbers: Attribute

## PaidyPayments

paidyToken:

paidyPaymentId:

- Order confirmation e-mails for subscription purchases

int\_paidy\_sfra cartridge is designed such that order confirmation e-mails are sent in the same manner as standard e-mails even for subscription purchases. To customize the e-mail sent, you can modify the int\_paidy\_sfra cartridge.

- Error message when a subscription payment is rejected

Please refer to the following wording for the message when a subscription payment is rejected.

---

今回の決済は承認されませんでした

申し訳ございませんが、オンラインショップが提供する他の支払方法をご利用ください。  
なお、審査結果の詳細につきましては開示できませんのであらかじめご了承ください。

Paidyに関するお問い合わせ

0120-971-918

---

## 4. Cartridge deployment testing at merchant

### 4-1. Before beginning a test

#### 4-1-1. Paidy merchant dashboard screen

- Confirming the test API key

Check the public and secret keys used for the test API; these will be entered into the custom site environment settings screen.

本番用APIキー	
パブリックキー	pk_live_
シークレットキー	sk_live_
テスト用APIキー	
パブリックキー	pk_test_
シークレットキー	sk_test_

## 4-1-2.Commerce Cloud Business Manager

### - API key settings

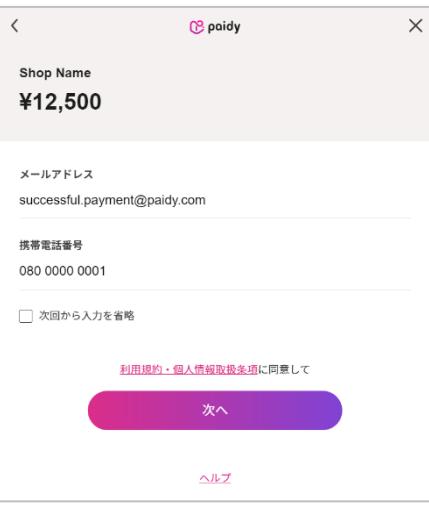
From the above settings screen, enter the test API keys you found for paidy\_api\_key and paidy\_secret\_key in section 4-1-1. Paidy merchant dashboard screen.

名前	値	デフ
paidy_api_key	pk_test_	
paidy_logo_url		
paidy_secret_key	sk_test_	
paidy_store_name		
paidy_service_name		
paidy_token_description		

## 4-2. PaidyCheckout test data

-mail address and phone number

E-mail address and telephone usable for test cases

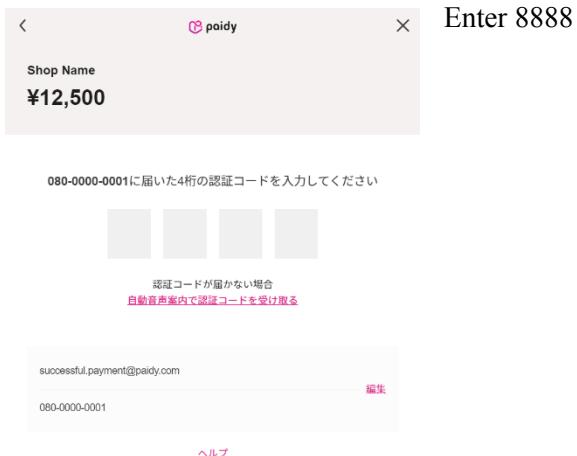


Test cases	mail address	phone number
Success	successful.payment@paidy.com	080-0000-0001
Failure	rejected.payment@paidy.com	

\*Note that the e-mail address and phone number for the above member data and shipping and billing address can differ.



-Authentication number



## 4-3. Browser compatibility

This cartridge supports the following OSes and browsers.

Device - OS	Browser
PC - Windows	Edge: latest version
	Chrome: latest version
	FireFox: latest version
PC - MacOS	Chrome: latest version
	Safari: latest version
Smartphone - iOS	iPhone (latest version of iOS) Safari iPhone (latest version of iOS) Chrome
Smartphone - Android	Android4.4 and later, Chrome

## 4-4. Test cases

Test cases to run after configuring the cartridge are listed below.

### 4-4-1. Paidy standard payments

Standard payment is tested using a guest and a member.

#### 1. Successful order as guest

- If not logging in and selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a successful e-mail address and phone number will display the product purchase confirmation screen.
- The Commerce Cloud order number created on the Paidy merchant site should be displayed as a transaction ID
- The order status for the corresponding order number on Business Manager on Commerce Cloud will be new

#### 2. Order success as member

- After logging in, when selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a successful e-mail address and phone number will display the product purchase confirmation screen.
- The Commerce Cloud order number created on the Paidy merchant site should be displayed as a transaction ID
- The order status for the corresponding order number on Business Manager on Commerce Cloud will be new

### 3. Failed order as guest

- If not logging in and selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a failing e-mail address and phone number will display an error message.
- The order status for the corresponding order number on Business Manager on Commerce Cloud will have failed

### 4. Order failure as member

- After logging in, when selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a failing e-mail address and phone number will display an error message.
- The order status for the corresponding order number on Business Manager on Commerce Cloud will have failed

## 4-4-2. Paidy subscription payments

Test as a member.

### 1. Success as a member

- When logging in as a member and selecting a payment method for Paidy subscription payments, the Confirm Order button is clicked and the PaidyCheckout popup appears. Entering a successful e-mail address and phone number and proceeding to processing the payment will display a product purchase confirmation screen
- The Commerce Cloud order number created on the Paidy merchant site should be displayed as a transaction ID
- The order status for the corresponding order number on Business Manager on Commerce Cloud will be new

### 2. Failure as a member

- When selecting a payment method for Paidy subscription payments and clicking the Confirm Order button, the PaidyCheckout popup appears. Entering a failing e-mail address and phone number and proceeding to processing the payment will display an error message
- The order status for the corresponding order number on Business Manager on Commerce Cloud will have failed

## 5. Webhook usage examples

### 5-1. Webhook usage

Paidy payments run credit holds for a purchase by means of communication directly between a consumer and Paidy over JavaScript. Therefore, one limitation of the system is that if the consumer bounces or the connection is cut, the e-commerce site may fail to obtain the results of payment confirmation and the stale payment data will remain on Paidy.

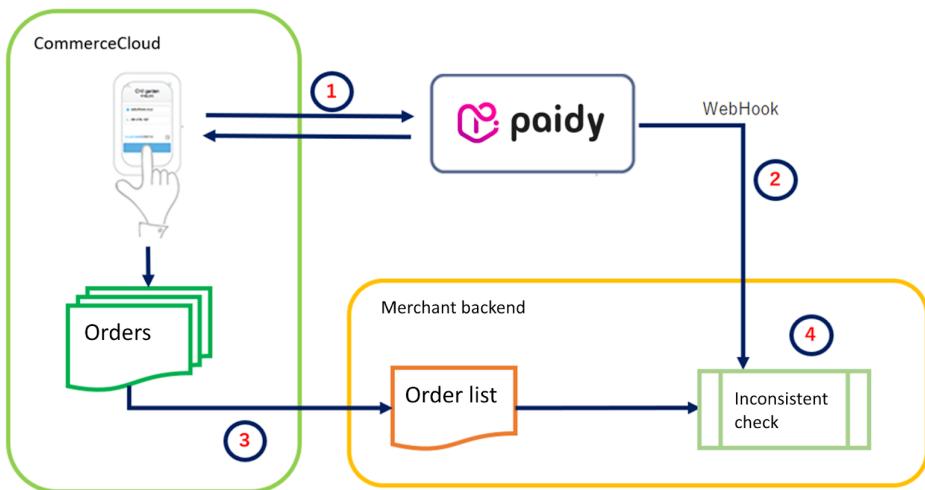
In the event of an inconsistency like the above, there are two problems.

1. The shop loses the opportunity to complete the order.
2. If the order does not have to be completed, the consumer has a credit hold placed against their Paidy account.

To accommodate these cases, a Webhook is provided.

The merchant can use the Webhook to detect inconsistent payments.

## 5-2. Detecting inconsistent payments



## 1. Payment

Payment is made via Commerce Cloud using Paidy. A callback from Paidy's payment results returns one of the three order statuses below.

- Payment success: New Outputted to the order list. Payment possible status
  - Payment failure: Failed Not outputted to the order list. Payment not possible status
  - Processing: Created Not outputted to the order list. When in a payment not possible status, updating the payment transitions to a payable status.

## 2. Webhook

Setting Webhook endpoints on the merchant backend allows for receiving notifications from Paidy of payment completion.

### 3. Order list

A list of successful orders processed through batch processing on Commerce Cloud is sent as an xml file to the merchant's backend.

#### 4. Inconsistent payment check processing

Checks for payments where Webhook data (2) is found, but no payment data (3) exists on Commerce Cloud.

### 5-3. Handling inconsistencies

Inconsistent payments that have been detected can be handled as follows, allowing for resolving the above issues.

### 5-3-1. Processing example 1

In this solution, a list of inconsistent/invalid payments is outputted and can be checked by the merchant. After a check is performed, the necessary steps are taken (approving or cancelling payment).

## 1. Establishing payment

Use the Order function on Business Manager on Commerce Cloud to update the status.

- #### - General tab

General Attributes Payment Notes History

## Details for Order 'WBSC00005100'

Information:	Contains 1 line item to 1 shipping location The total price is ¥14,506.00.		
Date Received:	11/24/17 6:27:23 am Etc/UTC		
Site:	Paidy_Master		
Created By:	Customer		
Customer:	[REDACTED]		
Customer No.:	00033017		
IP Address:	n/a		
Email:	[REDACTED]		
Phone:	[REDACTED]		
<b>Order Status:</b>	Open	<b>Confirmation Status:</b>	Confirmed
<b>Shipping Status:</b>	Not Shipped	<b>Export Status:</b>	Ready for Export

The status of the above three points is changed per below

Order Status : Created→Open

Confirmation Status : Not Confirm→Confirmed

Export Status:Not Exported→Ready for Export

• Attribute tab

The screenshot shows the Salesforce Merchant Tools interface for managing an order. The top navigation bar includes links for Sandbox, Merchant Tools, Administration, and Store. Below this, the breadcrumb navigation shows: Merchant Tools > Ordering > Orders > Order: WBSC00005100(Paidy\_Master). The main content area is titled "Attributes for Order 'WBSC00005100'". A note at the top states: "On this page you can edit the attributes of the order. Fields with a red asterisk (\*) are mandatory. Click Apply to save changes. Click Reset to cancel changes." The form contains several input fields:

- deliveryDate: [Input field]
- deliveryTimeSpan: [Input field]
- paymentCode: [Input field]
- Channel Type: [- None -] [Dropdown menu]
- Replacement Order Number: [Input field]
- Cancel Code: [- None -] [Dropdown menu]
- Shipping Status: \* 0 (NOTSHIPPED) [Dropdown menu]
- Customer Order Reference: [Input field]
  - Replaced Order No (replacedOrderNo): [Input field] (Redacted)
  - Replaced Order No: [Input field]
  - Replace Code: [- None -] [Dropdown menu]
  - External Order Status: [Input field]
  - Payment Status: \* 0 (NOTPAID) [Dropdown menu]
- PaidyPayments:
  - paidyToken: [Input field]
  - paidyPaymentId: pay\_20171124\_001 [Input field] (This field is circled in red)

Set the paidyPaymentID that arrived via the Webhook for paidyPaymentId

\*In order to e-mail an order confirmation e-mail as you would with StoreFront, it must be customized, as the format differs (see below).

- Order confirmation e-mail (concept)
- SiteGenesis StoreFront e-mail (normally used e-mail)



注文日: 24-11-2017  
注文番号: WBSC00005102

支払情報	
<b>支払方法</b> あと払い(ペイディ) 金額: ¥ 2,492	<b>支払金額合計</b> 小計      ¥ 1,808 配送 通常配達 ¥ 500

#### ■ E-mails sent from SendMail button on bottom of General tab on BusinessManager > Orders

<b>SiteGenesis</b> <b>Demo SiteGenesis</b> 5 Wall Street Burlington, MA 01803 <u><a href="#">SiteGenesis</a></u> Phone: +1.888.553.9216													
<b>Order Confirmation:</b> Order Number: WBSC00005101 - Thank you for your order!													
<p>This email confirms we received your order at <a href="#">SiteGenesis</a>. We will email your shipping confirmation with your tracking number when your items have been shipped. Your order will generally be shipped within 24 hours after we've received it. You can check your order's status on your account pages at <a href="#">SiteGenesis</a></p>													
<b>Thank you for shopping with us!</b> <b>Please print and save a copy for reference.</b> Please review the order below and verify that everything is correct. <b>PLEASE NOTE:</b> Since we begin processing your order shortly after you submit it on our web site, if any changes are necessary you MUST contact us via PHONE ONLY, at <b>+1.888.553.9216</b>													
<b>HIPMENT 1</b> <table border="1"> <thead> <tr> <th>PRODUCT DETAILS</th> <th>QTY.</th> <th>TOTAL</th> <th>SHIPPING DETAILS</th> </tr> </thead> <tbody> <tr> <td>           Samsung Series 6 22" LCD High Definition Television            Item Number: samsung-ln22a650            全品セール中         </td> <td>1</td> <td>           ¥ 66,318            ¥ - 33,159         </td> <td style="background-color: black;"></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>PAYMENT METHOD</th> <th>AMOUNT</th> </tr> </thead> <tbody> <tr> <td>Payment Total</td> <td>¥ 36,351</td> </tr> </tbody> </table>		PRODUCT DETAILS	QTY.	TOTAL	SHIPPING DETAILS	Samsung Series 6 22" LCD High Definition Television Item Number: samsung-ln22a650 全品セール中	1	¥ 66,318 ¥ - 33,159		PAYMENT METHOD	AMOUNT	Payment Total	¥ 36,351
PRODUCT DETAILS	QTY.	TOTAL	SHIPPING DETAILS										
Samsung Series 6 22" LCD High Definition Television Item Number: samsung-ln22a650 全品セール中	1	¥ 66,318 ¥ - 33,159											
PAYMENT METHOD	AMOUNT												
Payment Total	¥ 36,351												

## 2. Cancelling payments

Displaying the transaction in question on the Paidy merchant management screen and Canceling it (do not capture and close).

The screenshot shows the Paidy merchant management interface. On the left is a sidebar with icons for Home, Payment Management, Reports, Settings, and Search. The main area has a title '支払詳細' (Payment Details). It displays the following information:

ステータス	AUTHORIZED
取引ID	WBSC00005640
Authorized金額	¥ 5,487 (注文金額: ¥4,581 税金: ¥406 配送: ¥500)
キャプチャ一金額	¥ 0 (注文金額: ¥0 税金: ¥0 配送: ¥0 リファンド: ¥0)
日付	2017-12-01 14:44:40
キャプチャ期限	2017-12-31 14:44:41
ストア名	[REDACTED]
決済ID	pay_WiDsSEoAAFYAdAYR TEST

Below this is a section for '顧客情報' (Customer Information) with fields for Name (姓: 山田, 名: 太郎, フリガナ: ヤマダ タロウ), Email (メールアドレス: successful.payment@paidy.com), and Address (住所: [REDACTED]).

Finally, there is an 'オーダー内容' (Order Content) table:

商品番号	商品名	価格	数量	小計
701642843610	ワイドウエストベンシルスカート	¥4,581	1	¥4,581

In the 'Order Content' table, the last two columns ('キャプチャ一せずにクローズ' and 'キャプチャ') are highlighted with colored boxes.

### 5-3-2. Processing example 2

In this example, the system makes a determination and automatically cancels inconsistent payments. If, after referencing the order data and Webhook data on Commerce Cloud, the payment does not complete on Commerce Cloud, and authorization data is found only on Paidy, the close API is requested and the payment is closed.

<https://paidy.com/docs/jp/payments.html#close>

### 5-4. Webhook settings

In order to receive notifications from Paidy on the backend, the Webhook URL must be saved to the Paidy merchant management screen settings.

URL: <https://merchant.paidy.com>

You can enable these settings for a live environment by setting the merchant's live backend URL in the Webhooks: Live API Key and Old Version API Access Key sections.

In order to check connectivity with test payments before configuring the live environment, set the merchant's backend test URL in Webhooks: Test API Key.

Paidy merchant management screen

Webhooks - 「本番用APIキー」および「旧バージョンAPIアクセスキー」

Webhookの設定 (※オフにすると現在設定されているWebhook URLは削除されます)

Webhook URL

現在設定されているWebhook URL

Webhooks - 「テスト用APIキー」

テスト用Webhookの設定 (※オフにすると現在設定されているWebhook URLは削除されます)

テスト用Webhook URL

現在設定されているテスト用 Webhook URL

## 6. What to do when the service is down

- what happens when the service is down

If you select Paidy payment, you will get an error screen or the pop-up will stop working.

- what the merchant should do to fix it

To get started, please contact Paidy Technical Support.

Paidy Technical Support

Mail: [tech-support@paidy.com](mailto:tech-support@paidy.com)

TEL : 03-5545-5099 (weekdays 10:00-18:00)

If it takes a time to recover, please set the enable flag for paidy\_enabled to "いいえ(No)" in the Custom Preferences.

インスタンスタイプ  
Sandbox (サンドボックス)

IDで検索...  / 7

名前	値	デフォルト値
paidy_enabled (paidy_enabled)	いいえ <input type="text" value="いいえ"/>	はい

サイト全体で編集

## 7. Revision history

Version	Date	Revisions
1.1.0	2019/12/06	First edition
20.1.0	2020/01/15	3-2-2. Importing metadata

		-Modify 「meta_data_19.1.0.zip」 to 「meta_data_20.1.0.zip」
20.1.0	2020/4/5	<ul style="list-style-type: none"> <li>-SFRAのバージョン情報を追記。</li> <li>-Business Managerの互換モードを記載。</li> <li>-Add Version of SFRA and Compatibility Mode</li> <li>-Add 6. What to do when the service is down</li> </ul>
21.1.0	2021/6/21	<ul style="list-style-type: none"> <li>-Update 1. Cartridges version update</li> <li>-Update 4.SFRA support version update</li> <li>-Update 16-17. int_paidy postscript</li> <li>-Update 18. Import meta file name update</li> <li>Deleted description related to SiteGenesis and common metafile</li> <li>-Update 25. Corrected the description of function selected Payment Instrument</li> <li>- Update 31. Embedded code fix</li> <li>-Removed 35. Deleted the description of package.json</li> </ul>
22.1.0	2022/6/21	<p>p.1 update version in cover</p> <p>all pages Modified normal payment ID and subscription ID, as follows.</p> <ul style="list-style-type: none"> <li>- normal -&gt; standard</li> <li>- regular -&gt; subscription</li> </ul> <p>p.4 updated supported store Front Reference Architecture and compatibility mode version</p> <p>p.17 update import meta file name</p> <p>p.37 4-3. Removed IE11 from Browser compatibility</p>
22.1.1	2022/11/24	All pages

		<p>The following modifications were made due to the change in the name of Paidy standard and subscription payment service.</p> <ul style="list-style-type: none"><li>● standard payment あと払い(ペイディ)</li><li>● subscription payment あと払い(ペイディ)※定期購入</li><li>● 支払方法説明<ul style="list-style-type: none"><li>・クレジットカード、事前登録不要。</li><li>・メールアドレスと携帯番号だけで、今すぐお買い物。</li><li>・1か月に何度もお買い物しても、お支払いは翌月まとめて1回でOK。</li><li>・お支払いは翌月10日までに、コンビニ払い・銀行振込・口座振替で。</li></ul></li></ul> <p>ペイディについて詳しくはこちら。</p> <p>Capture was changed to reflect the above changes.</p>
--	--	--