J.P. MORGAN CHASE BANK MERCHANT ID REQUEST PROCESS

UNIVERSITY of HOUSTON

A CARNEGIE-DESIGNATED TIER ONE PUBLIC RESEARCH UNIVERSITY

Table of Contents

Introduction

Welcome to the University of Houston's Office of the Treasurer comprehensive guide designed to assist you in navigating the essential processes for setting up and managing your payment solutions. This manual provides clear, step-by-step instructions on the following key topics:

- **How to request a checkout and pay by link merchant ID:** Learn the process to obtain your unique merchant ID, essential for initiating transactions.
- **How to create a Certificate Signing Request (CSR):** Understand how to generate a CSR, a critical step in securing your payment transactions.
- **Setting up the Payment Page for Checkout:** Follow detailed instructions to configure your payment page for seamless checkout experiences.
- **Setting up the Payment Page for Pay by Link:** Get guidance on setting up a payment page specifically for pay by link transactions.
- **How to Create a Pay by Link for a Customer:** Discover how to create and send a pay by link to your customers, making payments easier and more convenient.

By the end of this guide, you will have the knowledge and tools necessary to efficiently manage your payment processes, ensuring a smooth and secure transaction experience for both you and your customers. Let's get started!

Merchant On Boarding Procedures

The merchant onboarding process involves several steps to establish a working relationship between a business and a Primary Service Provider (“PSP”).

The following steps provides an overview of the merchant onboarding process:

1. Pre-onboarding preparation

Gather necessary information and documentation, as outlined in this guide. Treasury will provide assistance to research PSPs and payment gateways to find the best fit for your specific needs, however, the department or college is also able to identify a service provider.

2. Merchant application and review

Apply to the PSP or payment gateway, providing information about your business, its owners, and anticipated transaction volumes. The PSP or payment gateway will review the application to ensure your business meets the necessary requirements and is a suitable candidate for their services.

3. Compliance and risk assessment

The PSP or acquiring bank conducts due diligence to assess the risk associated with your business, considering factors such as the type of business, transaction volume, chargeback history, and compliance with relevant regulations and industry standards. This process may involve background checks, credit checks, and verification of the information you provide them.

4. Account setup and integration

Once approved, set up your account with the PSP. This involves connecting your online store, POS system, or mobile app with the PSP’s payment gateway, for frictionless payment processing. The integration may require technical support from the PSP, as well as testing to ensure everything works as expected.

5. Training and support

Most PSPs will provide your team with training, resources, and ongoing support to ensure a smooth and successful partnership. This may include guidance on using the payment platform, managing transactions, handling disputes and chargebacks, and adhering to security and compliance requirements.

6. Ongoing monitoring and optimization

After the onboarding process is complete, continually monitor and optimize your payment processing systems to ensure maximum efficiency, security, and compliance. This may involve updating software, implementing new fraud prevention measures, or adjusting transaction limits based on changing business needs.

Treasury staff are available to assist with the process and provide on-going training as needed.

Contact: Louis W. Edwards
Treasurer
(713) 743-5670
Lwedwar2@central.uh.edu

Thank you.

Merchant ID Overview

What is a merchant ID?
A merchant ID (MID) is a code that identifies a merchant's account and is used in credit and debit card payment processing.

Why do I need a merchant ID?
A merchant ID is an important business identifier that tells the banks and credit card processors where the money is going and is used to ensure your money is sent and received correctly.

How many merchant IDs do I need to set up a credit card payment process?
There are two merchant IDs used for the checkout payment set up.

A test sandbox merchant ID is used for testing purposes ONLY.

Merchant ID # 1

Test Sandbox

Another new merchant ID is required to push the web application into production.

Merchant ID # 2

Production

Only one merchant ID is required for the Pay by Link payment set up.

Merchant ID

Production

Requesting a Merchant ID

Merchant ID (test sandbox)

To request a merchant ID for testing purposes:

1. Create a **Certificate Signing Request (CSR) Key**.
2. Send an email to the Treasury department at Treasury@central.uh.edu and include the following:
    - Subject line: New Merchant ID request for <enter department name>
    - In the body have the following:
        - A request for a new Merchant ID for **testing purposes**
        - Include the Certificate Signing Request (CSR) Key
        - Request access to Chase's Checkout Settings page for the developer doing the configuration
            - Name
            - Email
3. Fill out the spreadsheet provided by the Treasury department and send it back.

Merchant ID (production)

To request a merchant ID for production:

1. Create a **Certificate Signing Request (CSR) Key** for a Checkout payment set up. You don't need to create one for a Pay by Link request.
2. Send an email to the Treasury department at Treasury@central.uh.edu and include the following:
    - Subject line: New Merchant ID request for <enter department name> for a Checkout or Pay by Link set up.
    - In the body have the following:
        - A request for a new Merchant ID for **production**
        - Include the Certificate Signing Request (CSR) Key if needed
        - Request access for the developer and department business administrators/etc.
            - Name
            - Email
        - What kind of access the user(s) need:
            - Reporting – Yes/No
            - Transactions – Yes/No
            - Refunds – Yes/No
            - Pay by Link – Yes/No
            - Chase Checkout settings – Yes/No
3. Fill out the spreadsheet provided by the Treasury department and send it back.

Merchant ID Request Check Lists

Merchant ID (Sandbox) request check list

This merchant ID is used for the testing phase.

#	Check List Item	Task Completed?
1	Create a Certificate Signing Request (CSR) Key for the new merchant ID you're going to do the development and testing on.	Yes ☐ No ☐
2	Send an email to the Treasury department with a request for a new merchant ID for testing purposes. Make sure to include: • The Certificate Signing Request (CSR) Key • Add the name(s) of the developer(s) that need access	Yes ☐ No ☐
3	Test the merchant ID sandbox.	Yes ☐ No ☐
4	Send an email to J.P. Morgan Chase bank that testing is complete.	Yes ☐ No ☐

Merchant ID (Production) request check list

This merchant ID is used to push the web application into production.

#	Check List Item	Task Completed?
1	Create a Certificate Signing Request (CSR) Key for the new merchant ID you're going to use to push the application into production.	Yes No

2	Send an email to the Treasury department with a request for a new merchant ID. Let them know you are ready to push the web application into production. Make sure to include:
	• A Certificate Signing Request (CSR) Key
	• Names and emails of the users who will need access to it
	• A list of the access each user will need:
	    ▪ Reporting – Yes/No
	    ▪ Transactions – Yes/No
	    ▪ Refunds – Yes/No
	    ▪ Pay by Link – Yes/No
	    ▪ Chase Checkout settings – Yes/No
		Yes No

3	Test all the testing scenarios J.P. Morgan Chase bank provides and wait for confirmation.	Yes No
4	A new merchant ID (production) is assigned. Begin to reconfigure the settings used in the development site.	Yes No
5	Perform a live test with a credit card.	Yes No

Merchant ID (Pay by Link) request check list

#	Check List Item	Task Completed?

1	Send an email to the Treasury department with a request for a new merchant ID for Pay by Link. Make sure to include:
	• Names and emails of the users who will need access to it
	• A list of the access each user will need:
	    • Reporting – Yes/No
	    • Transactions – Yes/No
	    • Refunds – Yes/No
	    • Pay by Link – Yes/No
	    • Chase Checkout settings – Yes/No
		Yes No

2	Fill out the spreadsheet provided by Treasury and send it back to them.	Yes No
3	When the new merchant ID has been assigned make sure to test that all users have appropriate access.	Yes No
4	Perform a live test with a credit card.	Yes No

Instructions on how to create a Certificate Signing Request (CSR) Key

To create a Certificate Signing Request (CSR) Key for Production:

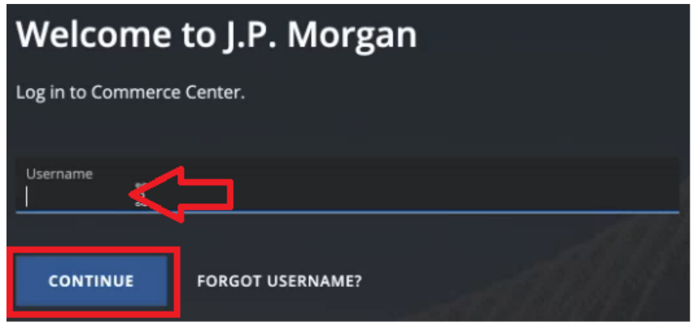
1. openssl req -new -newkey rsa:2048 -nodes -out **<filename>.txt** -keyout **<filename>.key** -subj "/C=<Country>/ST=<State>/L=<City>/O=<Organization>/OU=<Organization Unit>/CN=<Common Name>"
2. **Filename.txt**: This file contains your CSR that needs uploaded into CSR.
3. **Filename.key**: This file contains your private key. **DO NOT Send to J.P. Morgan**.
4. **Country**: must be two letter code.
5. **Common Name - CN *Must be unique PROD***

Note: You will need to do this if you’re not on uh.edu.

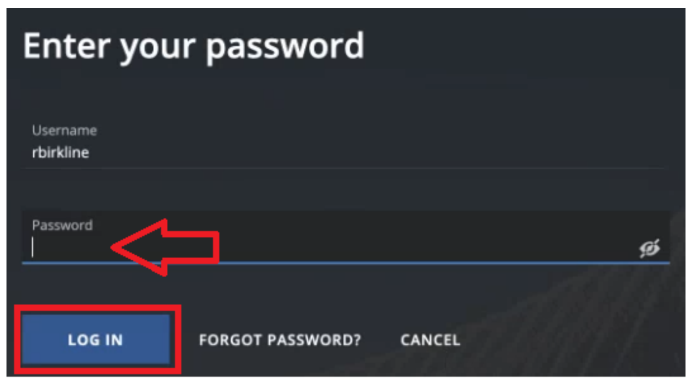
Instructions on Setting up the Payment Page for Checkout

To set up the payment page for Checkout:

1. Sign into the Commerce Center here: https://jpmorgan.com/commerce-center.
2. Enter your **username**.
3. Click **Continue**.

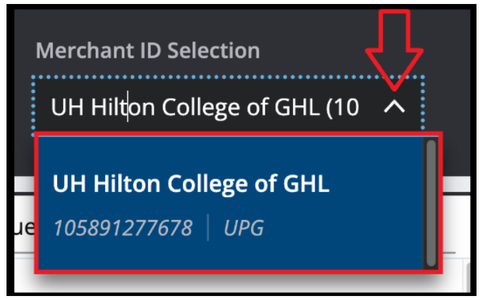


4. Enter your **password**.
5. Click **Log in**.

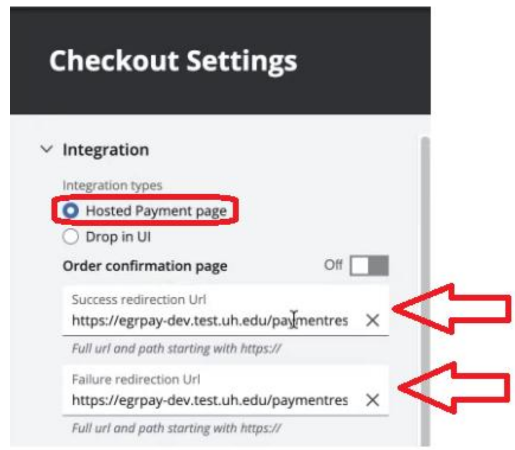


6. Click the drop-down arrow for **Settings**.
7. Select **Checkout**.

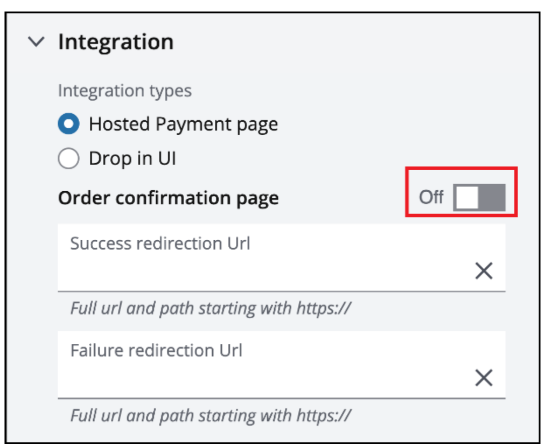
8. Click the drop-down arrow for **Merchant ID Selection**.
9. Select your **Merchant ID**.



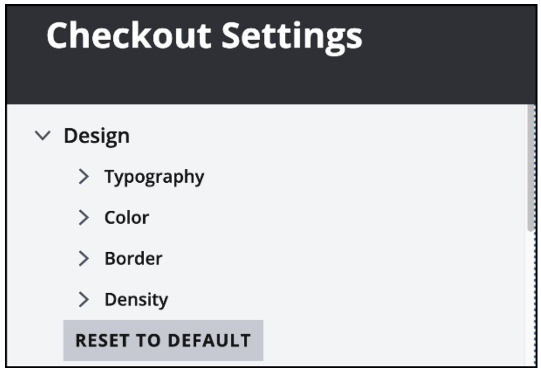
10. Select the Integration type as **Hosted Payment page**.
11. Add a redirect in the **Success redirection Url**.
12. Add a redirect in the **Failure redirection Url**.



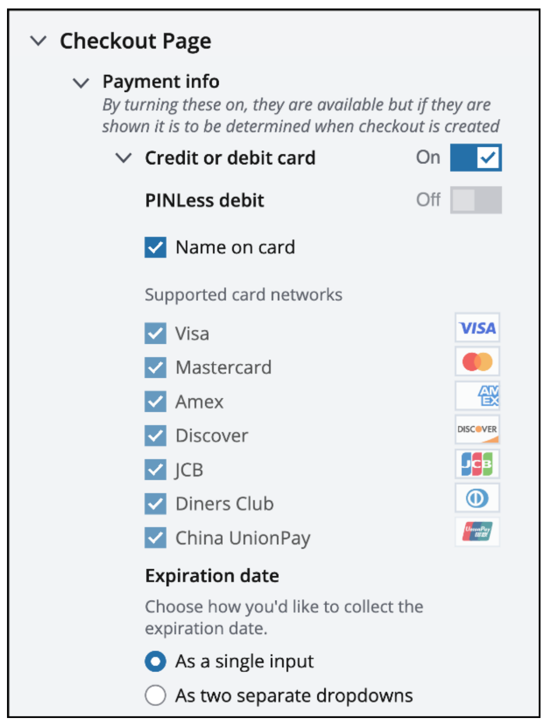
Note: If the text boxes for adding the Success and Failure redirection URLs are not visible, disable the **order confirmation page**.



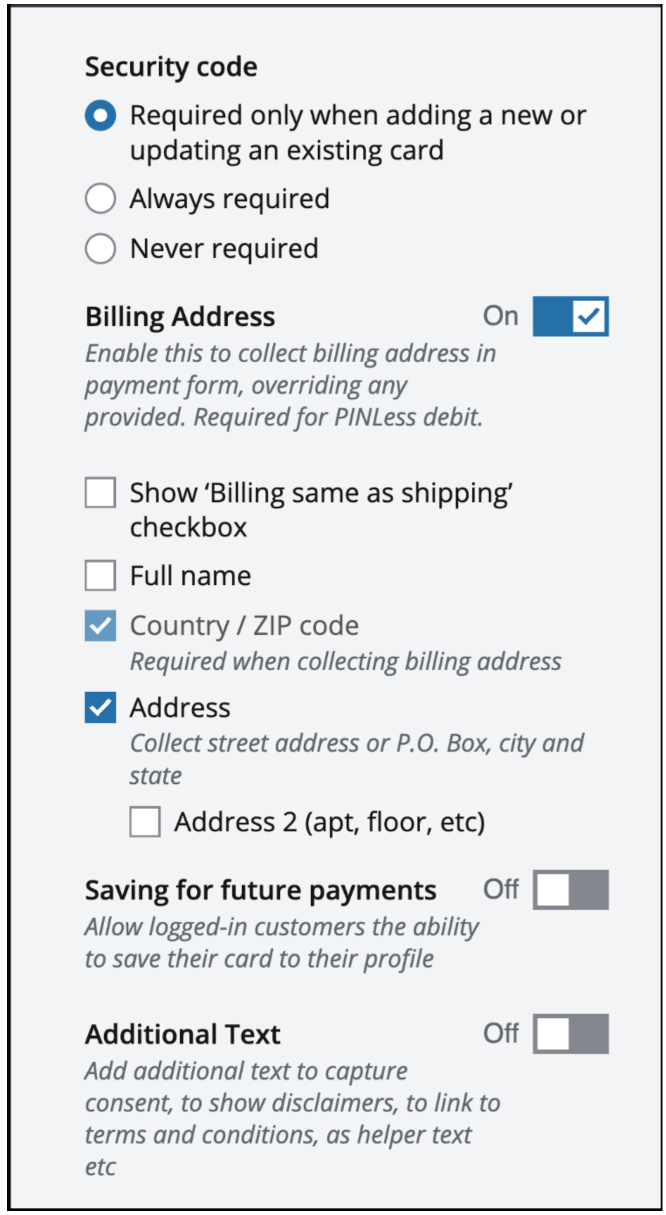
13. Select the **Design** settings.



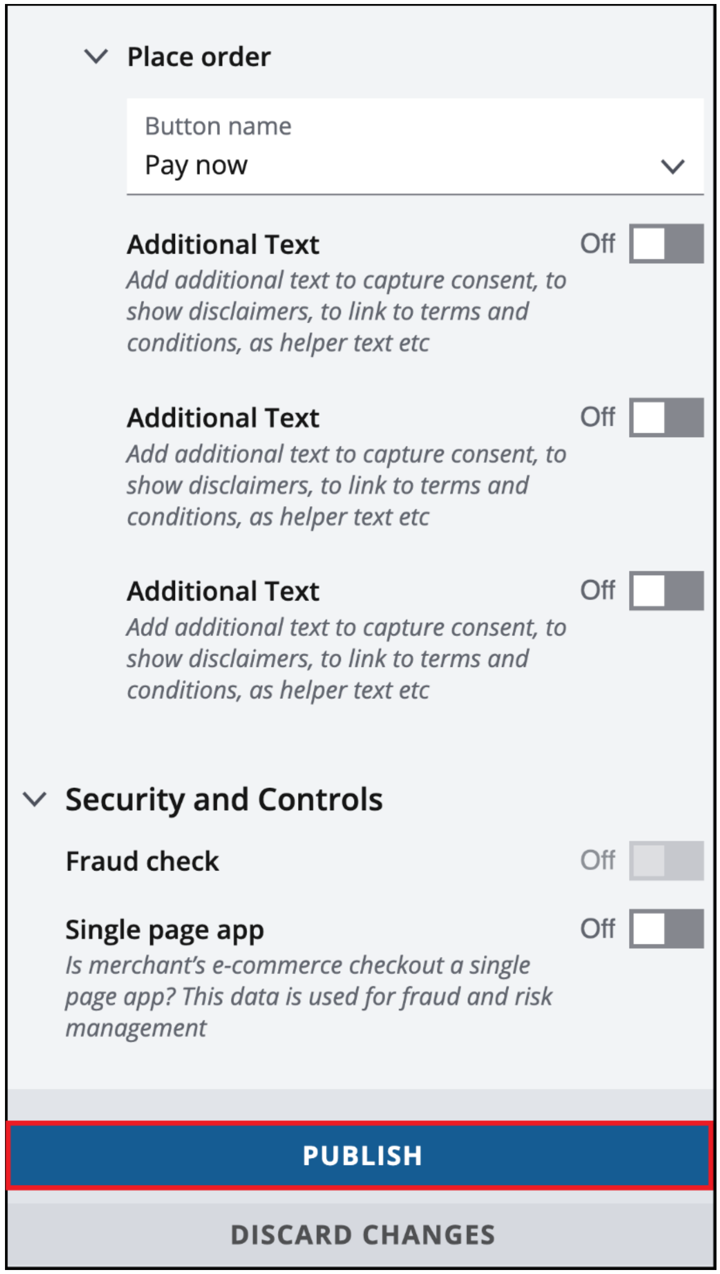
14. Select the **Payment info** settings.



15. Select the settings for:
- **Security Code**
- **Billing Address**
- **Saving for future payments**
- **Additional Text**



16. Select the settings for:
    - **Place Order**
    - **Security and Controls**
17. Click **PUBLISH**.

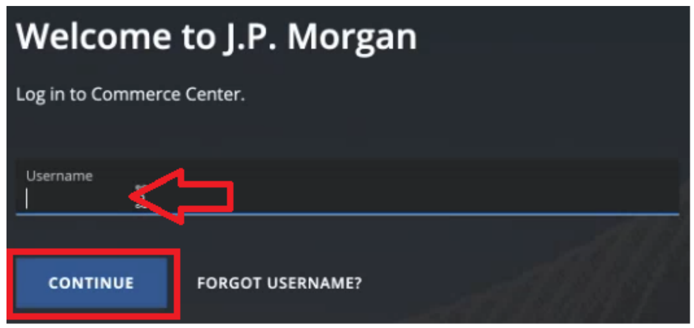


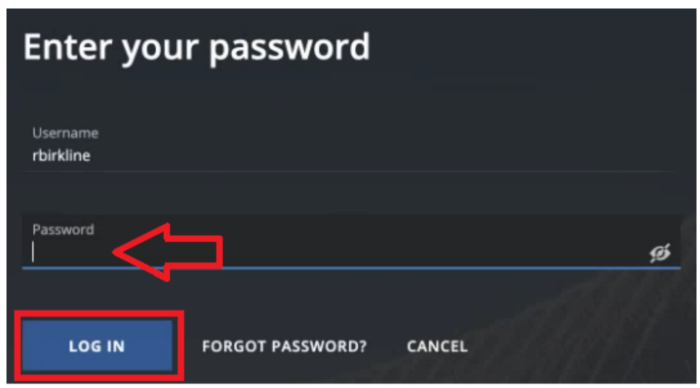
Instructions on Setting up the Payment Page for Pay by Link

This section provides step-by-step instructions for completing the **one-time** setup required to enable the Payment Page for Pay by Link. It covers configuration details necessary to activate and customize the payment experience for end users.

To set up the payment page for Pay by Links:

1. Sign into the Commerce Center here: https://jpmorgan.com/commerce-center.
2. Enter your **username**.
3. Click **Continue**.

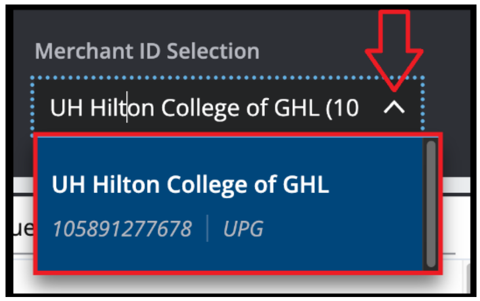




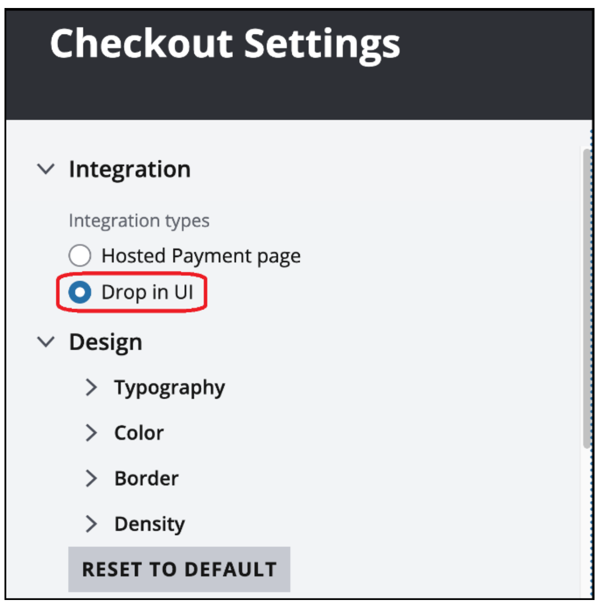
4. Enter your **password**.
5. Click **Log in**.

6. Click the drop-down arrow for **Settings**.
7. Select **Checkout**.

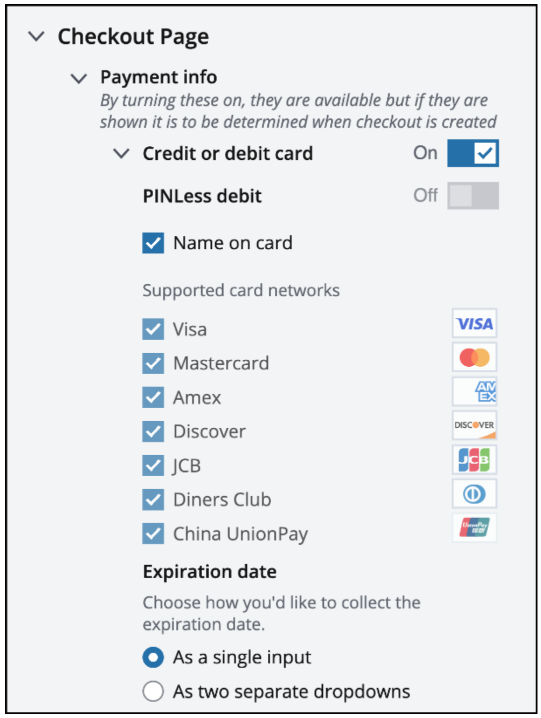
8. Click the drop-down arrow for **Merchant ID Selection**.
9. Select your **Merchant ID**.



10. Select **Drop in UI** as the Integration type.
11. Select the **Design** Settings.

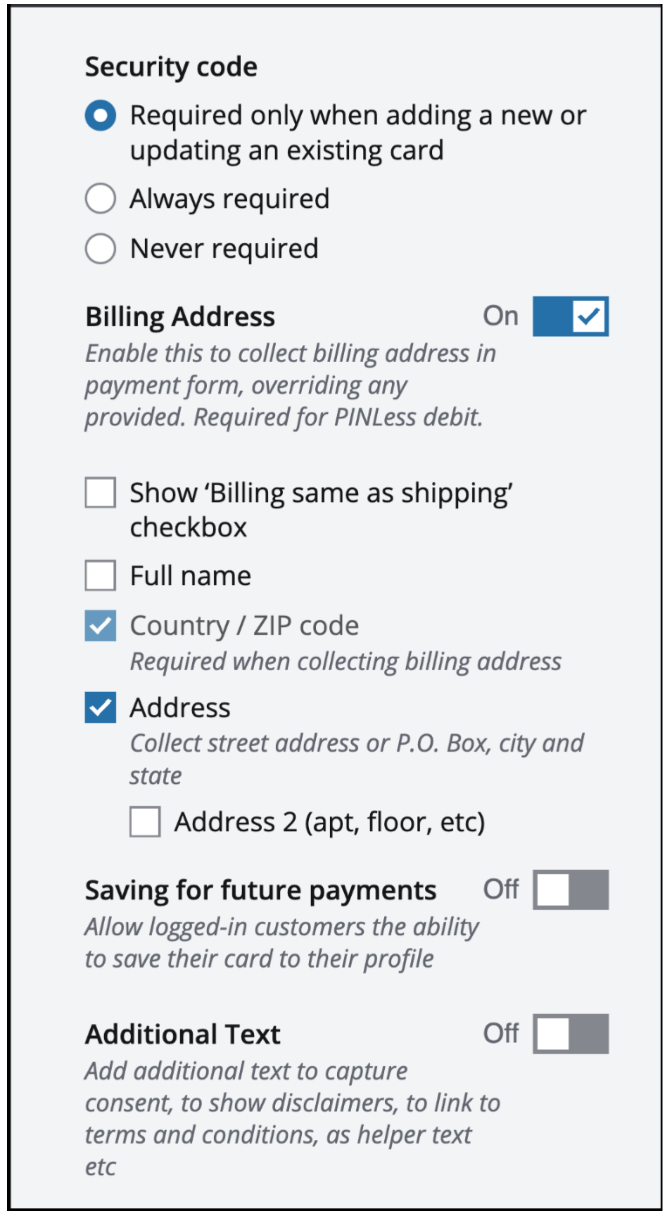


12. Select the **payment info** settings.

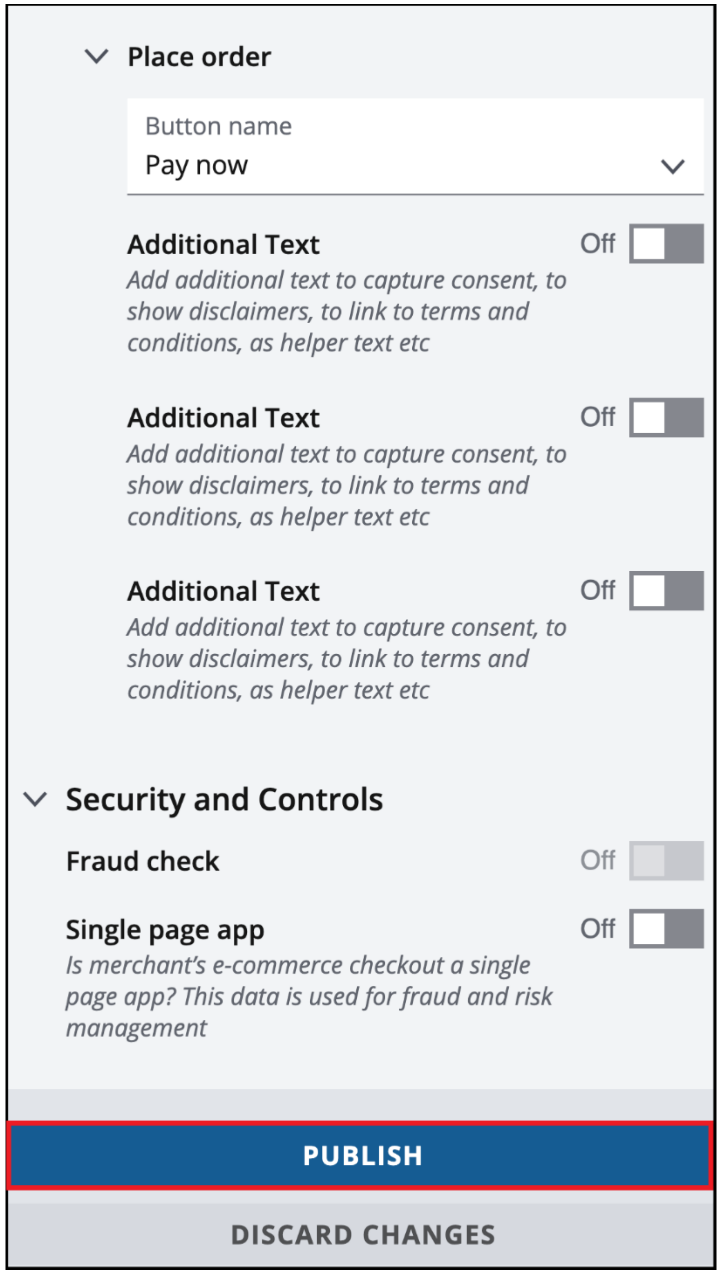


13. Select the settings for:

- **Security Code**
- **Billing Address**
- **Saving for future payments**
- **Additional Text**



14. Select the settings for:
    - **Place order**
    - **Security and Controls**
15. Click **PUBLISH**.

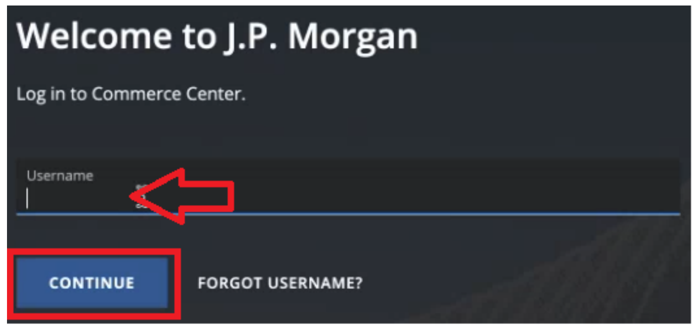


Sending Payment Links to Customers

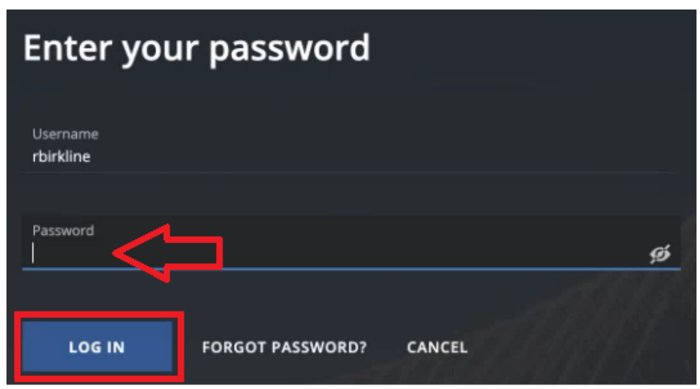
This section outlines the steps for sending a Pay by Link to a customer, including how to generate the link and deliver it securely for payment processing.

Sending a Pay by Link to a customer:

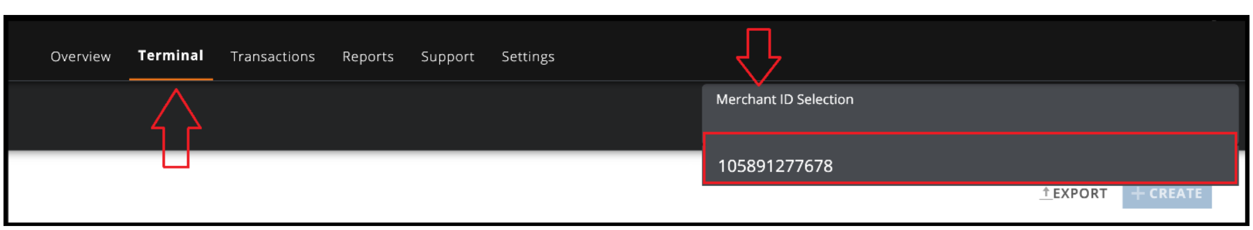
1. Sign into the Commerce Center here: https://jpmorgan.com/commerce-center.
2. Enter your **username**.
3. Click **Continue**.



4. Enter your **password**.
5. Click **Log in**.

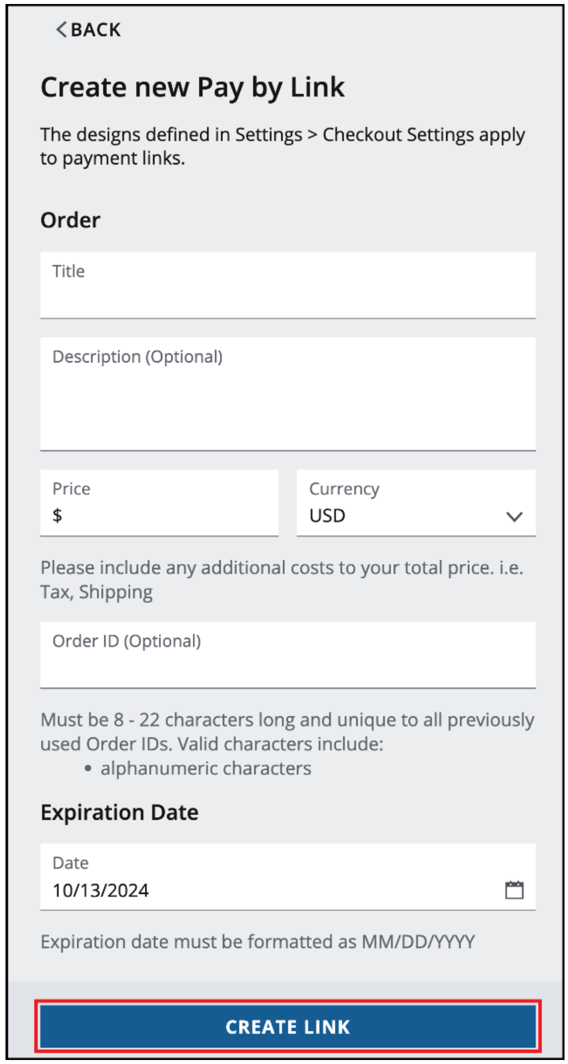


6. Click **Terminal**.
7. Click the drop-down arrow for **Merchant ID Selection**.
8. Select your **Merchant ID**.



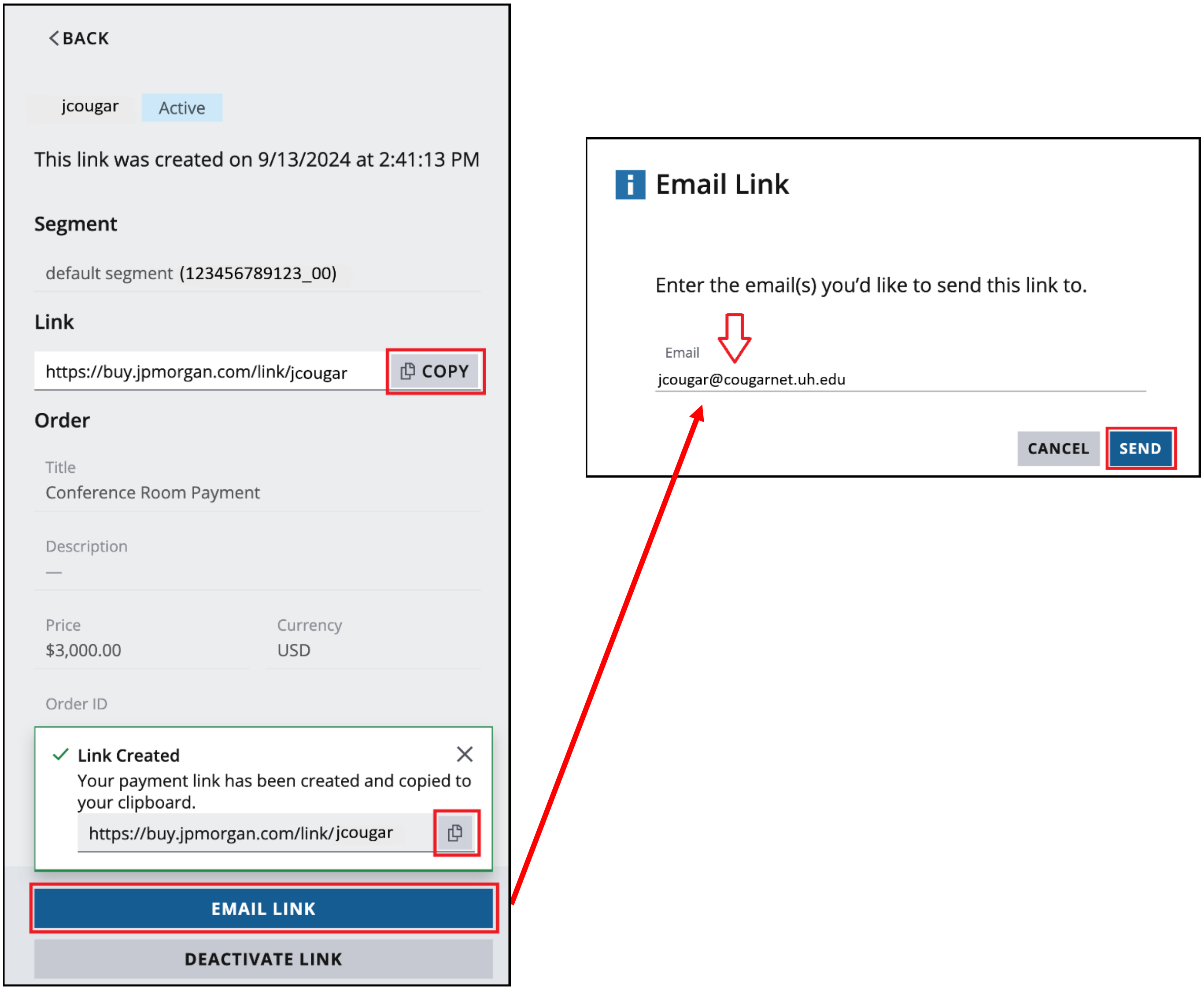
9. Click **Create**.

10. Enter the following:
    - Title
    - Description
    - Price
    - Order ID
    - Expiration Date
11. Click **CREATE LINK**.

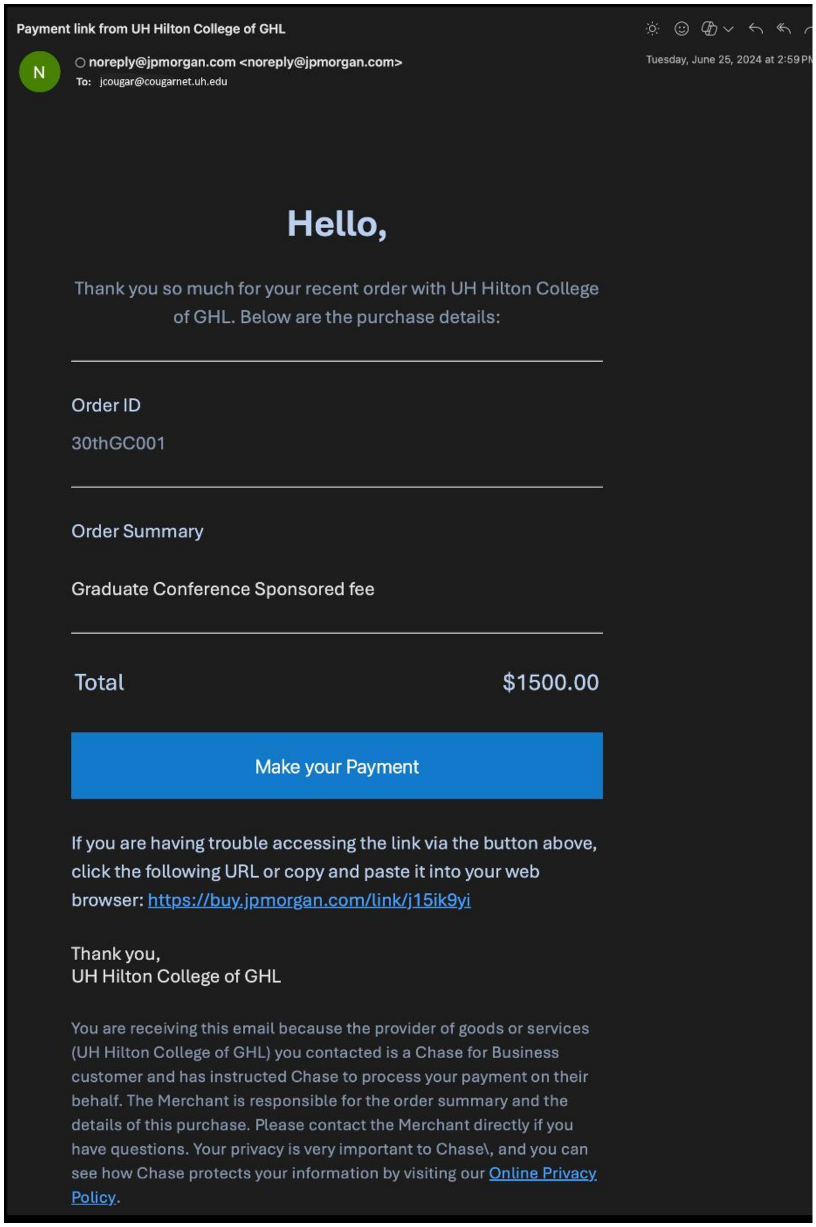


12. There are two ways you can send the Pay by Link:

- **__Method #1:__** Email the link from Outlook - Click **Copy** > Open a new **email** > Paste the **link** in the body of the email > click **Send**.
- **__Method #2:__** Send the link from Commerce Center - Click **EMAIL LINK** > Add the person's **email address** > click **Send**.



Note: this is a sample of what a customer sees after you send the Pay by Link.



Payment link from UH Hilton College of GHL

N

noreply@jpmorgan.com <noreply@jpmorgan.com>

To: jcougar@cougarnet.uh.edu

Tuesday, June 25, 2024 at 2:59 PM

Hello,

Thank you so much for your recent order with UH Hilton College of GHL. Below are the purchase details:

Order ID

30thGC001

Order Summary

Graduate Conference Sponsored fee

Total	$1500.00

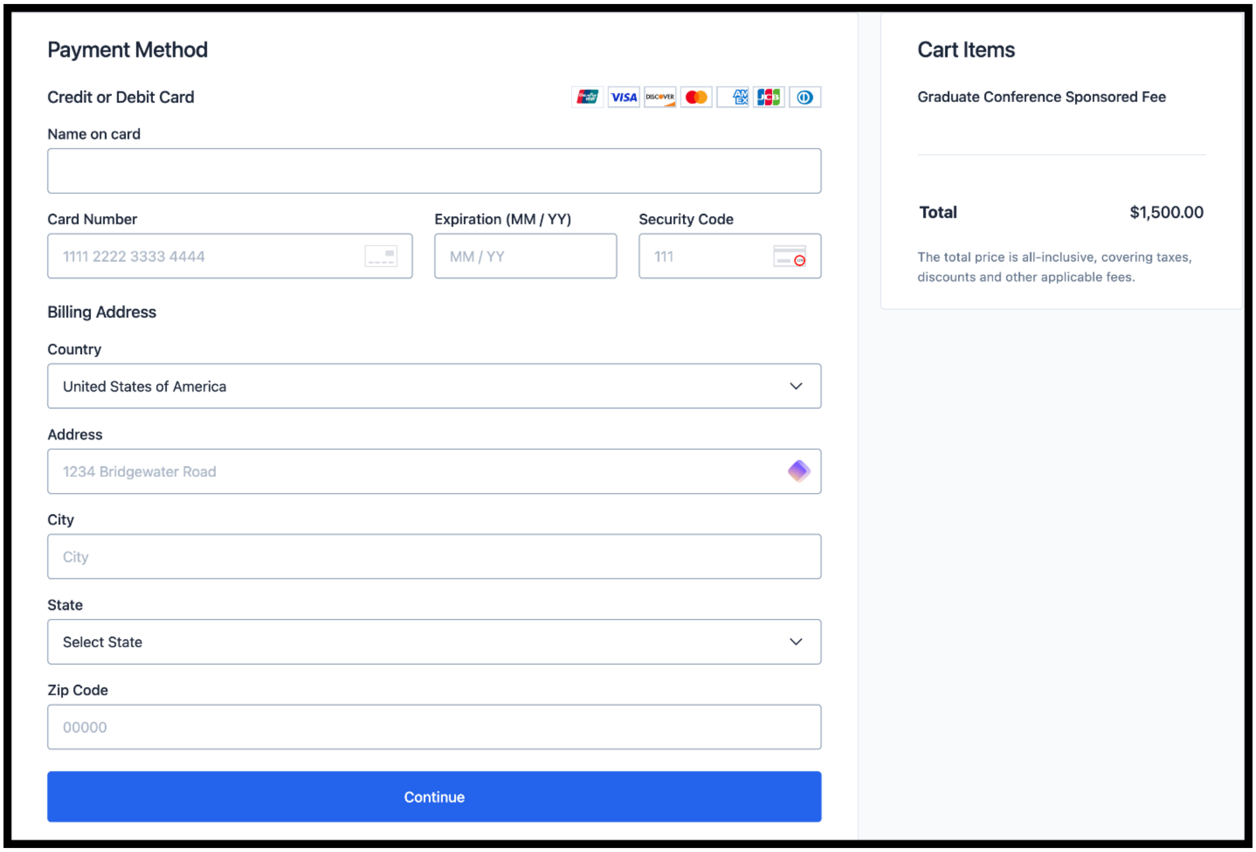
Make your Payment

If you are having trouble accessing the link via the button above, click the following URL or copy and paste it into your web browser: https://buy.jpmorgan.com/link/j15ik9yi

Thank you,
UH Hilton College of GHL

You are receiving this email because the provider of goods or services (UH Hilton College of GHL) you contacted is a Chase for Business customer and has instructed Chase to process your payment on their behalf. The Merchant is responsible for the order summary and the details of this purchase. Please contact the Merchant directly if you have questions. Your privacy is very important to Chase\, and you can see how Chase protects your information by visiting our Online Privacy Policy.

This is a sample of what a customer sees when they click the Make your Payment button.



Support Contacts

If you need help, please contact one of the following team members:

UH Treasury Support Contact

Email: Treasury@central.uh.edu

J.P. Morgan Chase Bank Technical Support Contact

Phone: 1-800-228-7782 (option 1)

Email: MerchantServicesTechOps.External@JPMorgan.Com

Frequently Asked Questions

Question
Can I use the merchant ID I already have to perform my development testing to migrate to the new J.P. Morgan Chase bank system?

Answer
No, if you need to add the J.P. Morgan Chase bank payment system to a web application a new merchant ID is required for testing and pushing the application into production.

Question
Can I get access to the J.P. Morgan Chase bank Commerce Center without requesting a merchant ID?

Answer
No, all UH departments need to request a merchant ID before getting access to the J.P. Morgan Chase bank Commerce Center.

Question
How do I request a merchant ID for my team?

Answer
Go to the Requesting a Merchant ID section in this manual for instructions on how submit a request for a new merchant ID.

Question
Why can't I sign into the J.P. Morgan Chase bank Commerce Center?

Answer
Make sure to try the following:

- Try clearing the cookies, close the browser, open it and try again
- Try using another browser (Example: Firefox, Chrome, Safari or Edge)
- Try resetting the password

If you're still unable to sign in to J.P. Morgan Chase banks Commerce Center contact JPMC technical support at 1-800-228-7782 (option 1) or send an email to MerchantServicesTechOps.External@JPMorgan.Com.

Question
What is a Certificate Signing Request (CSR) key?

Answer
A Certificate Signing Request (CSR) Key is an encryption key pair.

Question
Do I really need a Certificate Signing Request (CSR) key?

Answer
Yes, the Certificate Signing Request (CSR) Key is required for requesting a merchant ID for checkout. It is not needed for a pay by link set up.

Question
Who creates the Certificate Signing Request (CSR) key?

Answer
If you are on the www.uh.edu webservers the Web Technology Team will create the CSR and will also provide a path to those files. If you are not on www.uh.edu then your developer is responsible for creating the CSR for your domain. For assistance with creating it, please submit a request at webtech@uh.edu.

Question
How do I create a Certificate Signing Request (CSR) key for my merchant ID request?

Answer
Go to the Instructions on how to create a Certificate Signing Request (CSR) Key section in this manual to create one.

Question
Who do I send the Certificate Signing Request (CSR) Key to?

Answer
The UH department web developer will submit the CSR Key to the implementation manager with J.P. Morgan Chase bank.

Question
Who do I contact after I finish the development and testing phase of the web application?

Answer
After completing all testing, send an email to the implementation manager at J.P. Morgan Chase Bank.

Question
How long does it take to get the web application out into production?

Answer
It takes between 3 to 7 days to hear back from J.P. Morgan Chase bank. If you still haven't heard from them after that time frame, please contact the Treasury department at Treasury@central.uh.edu.

Question
How do I set up the checkout payment page in the Commerce Center?

Answer
Go to the Instructions on setting up the Payment Page for Checkout section in this manual to set it up.

Question
How do I set up the Pay by Link payment page in the Commerce Center?

Answer
Go to the Instructions on Setting up the Payment Page for Pay by Link section in this manual to set it up.

Question
Can a Department Business Administrator set up a Pay by Link for a customer?

Answer
Yes, go to the Sending a Pay by Link to a Customer section in this manual to create and send it.

Question
How do I test the Pay by Link set up?

Answer
You can send a Pay by Link to yourself, make the payment and then verify the transaction in the Commerce Center. Once verified, refund the transaction.

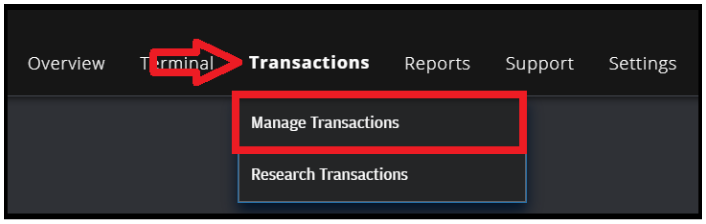
Question
Is there a knowledge base or how to guides from J.P. Morgan Chase bank I could use?

Answer
Yes, go into the Commerce Center click **Support** > **Learning Center**. A listing of guides and instructions appears on that page.

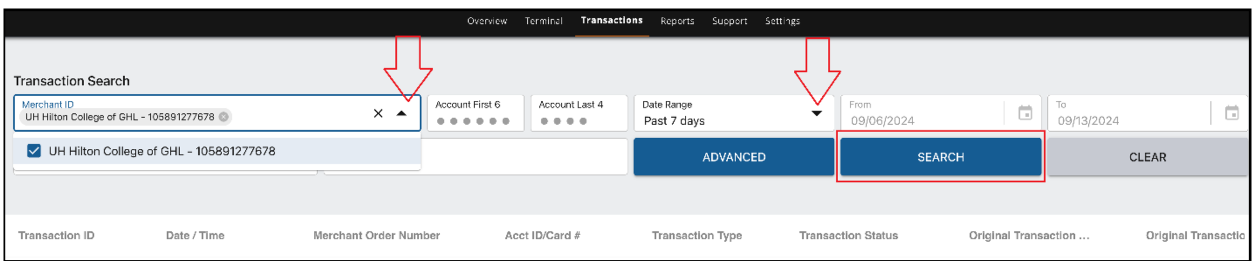
Question
Where can I verify credit card transactions in Commerce Center?

Answer
To verify credit card transactions in Commerce Center:

1. Click **Transactions**.
2. Select **Manage Transactions**.



3. Click the drop-down list for **Merchant ID.**
4. Select a **Date Range**.
5. Click **Search**.



Question
We've been on the new J.P. Morgan Chase bank checkout system for 6 months and are now running into issues. Who do we contact about getting support for this?

Answer
If you're encountering issues with the new J.P. Morgan Chase bank checkout system, please go to the Support Contact section in this manual for assistance.

Question
No one has contacted me back from J.P. Morgan Chase bank IT support. Who else can I contact in that group?

Answer
If you haven't received a response from J.P. Morgan Chase bank IT support, please contact the Treasury department at Treasury@central.uh.edu.

Appendices

Appendix A - Chase Validation Example for Visa

1. Once the transaction has been processed or attempted, fill in the results and other relevant information in the blue portion of the table.

Note: Live card numbers should never be processed through the test system. Please make certain to use only test card data for validation testing.

2. When finished, send the completed sheet back to: ms.pit.team@jpmorgan.com

Merchant ID	Provided by Treasury

E-Commerce - VISA

Test Case #		Amount	Request-Id	verificationValue	Expected Result
1	SALE	100.00		111	Approved
2	SALE	754.00		411	Declined

Test Case#		Amount	transactionReferenceId	Expected Result
3	REFUND (Test Case 1) - Send transactionReferenceId	N/A		Approved

Results

Result	transactionId (Returned in Response)	approvalCode	Comments		
Approved					
Declined					

Results

Result	transactionId (Returned in Response)	Comments
		Refund via Commerce Center

Appendix B - Chase Validation Example for MasterCard

1. Once the transaction has been processed or attempted, fill in the results and other relevant information in the blue portion of the table.

Note: Live card numbers should never be processed through the test system. Please make certain to use only test card data for validation testing.

2. When finished, send the completed sheet back to: ms.pit.team@jpmorgan.com

Merchant ID	Provided by Treasury

E-Commerce - MASTERCARD

Test Case #		Amount	Request-Id	verificationValue	Expected Result
1	SALE	100.00		111	Approved
2	SALE	303.00		512	Declined

Test Case#		Amount	transactionReferenceId	Expected Result
3	REFUND (Test Case 1) - Send transactionReferenceId	N/A		Approved

Results

Result	transactionId (Returned in Response)	approvalCode	Comments

Results

Result	transactionId (Returned in Response)	Comments
		Refund via Commerce Center

Appendix C - Chase Validation Example for Discover

1. Once the transaction has been processed or attempted, fill in the results and other relevant information in the blue portion of the table.

Note: Live card numbers should never be processed through the test system. Please make certain to use only test card data for validation testing.

2. When finished, send the completed sheet back to: ms.pit.team@jpmorgan.com

Merchant ID	Provided by Treasury

E-Commerce - DISCOVER

Test Case #		Amount	Request-Id	verificationValue	Expected Result
1	SALE	100.00		111	Approved
2	SALE	501.00		611	Declined

Test Case#		Amount	transactionReferenceId	Expected Result
3	REFUND (Test Case 1) - Send transactionReferenceId	N/A		Approved

Results Result	transactionId (Returned in Response)	approvalCode	Comments

Results Result	transactionId (Returned in Response)	Comments
		Refund via Commerce Center

Appendix D – Chase Validation Example for Features and Functions

1. Once the transaction has been processed or attempted, fill in the results and other relevant information in the blue portion of the table.

Note: Live card numbers should never be processed through the test system. Please make certain to use only test card data for validation testing.

2. When finished, send the completed sheet back to: ms.pit.team@jpmorgan.com

Merchant ID	Provided by Treasury

E-Commerce - Negative Testing

Test Case #		Amount	Request-Id	verificationValue	Expected Result
1	Auth/Sale - INVALID_ACCOUNT	304.00			Denied
2	Auth/Sale - INSUFFICIENT_FUNDS	521.00			Denied
3	Auth/Sale - TIMEOUT	301.00			Error
4	Auth/Sale - NEW_CARD_ISSUED	595.00			Denied
5	Auth/Sale - CARD_EXPIRED	522.00			Denied

Result	transactionId (Returned in Response)		

Appendix E – J.P.morgan PHP library readme files

J.P. Morgan Checkout PHP Library

Class information

This document provides instructions on how to use the `checkout` PHP class for processing transactions using the J.P. Morgan Chase API. The guide covers configuration, adding items, setting customer and shipping details, generating tokens, and processing checkout requests.

Requirements

- PHP 7.2+
- Composer
- Dependencies: `firebase/php-jwt`

JSON configuration files

Before using the `checkout` class, you need to create a configuration file (e.g., `config.json`) with the following structure:

```
{
"privateKey": "path/to/private/key",
"certPath": "path/to/cert",
"client_id": "your_client_id",
"resource_id": "your_resource_id",
"merchant_id": "your_merchant_id"
}
```

In addition you must have the public and private key to use this class.

Using the class

Initializing the Checkout Class

Initialize the class by passing the path to your configuration file:

```
require_once 'uh/vendors/jpmorgan/checkout.php';
```

$configPath = 'path/to/config.json';

$checkout = new checkout($configPath);

Adding Configuration Items

While you don't need to add any additional items to the Configuration it is helpful to add values to use a single source.

Return a single value of what the key contains

$checkout->addToConfig('new_key', 'new_value');

Getting Confirmation Items

To retreive a confirmation item you can use

$checkout->getConfig('key');

// to get all keys as an array

$checkout->getConfig();

Generating a JWT key

This is not nessesary because the class makes care of this for you. If you are needing the JWT token you can generate it like this.

Returns a JWT String

$jwtToken = $checkout->generateToken();

Generating an accessToken

This is not nessesary but if you are needing an access token to use the API you can generate one as follows.

$accessToken = $checkout->getAccessToken();

Returns

return [

"access_token" => "",

"token_type" => "error",

"expires_in" => "",

"bearer_token" => $this->config['jwtToken']

];

How to do a checkout

To do a checkout you need to do the following

1. Create your cart as you normally would.

2. Go to a page that makes the checkout intent.

1. Your intent must have at least one item.

2. Optional Customer information.

3. Optional Shipping information.

3. Generate the url that you will use to redirect them to the hosted checkout page.

Example:

require_once 'uh/vendors/jpmorgan/checkout.php';

$jpmorgan = new checkout('/publish/lib/auth/jp_morgan_payment_configs/test-certs/config1.json');

$jpmorgan->addItem([

"id"=>"ITEM$lp",

"name"=>"Item $lp",

"description"=>"this is an item desc of $lp",

"quantity"=>"1",

"unitPrice"=>"$x"

```
"imageUrl" => "<optional> can be url or base64 encoded image"
]);

/* OPTIONAL */
$jpmorgan->setCustomer([
"email"=>"rbirkline@uh.edu",
"phone"=>"26493",
"name"=>"Robert Birkline",
"address1"=>"my address",
"address2"=>"my address 2",
"city"=>"Houston",
"state"=>"Texas",
"country"=>"US",
"postalCode"=>"77204"
]);

/* OPTIONAL */
$jpmorgan->setShippingAddress([
"name"=>"Shipping",
"address1"=>"Ship 1",
"address2"=>"Ship 2",
"city"=>"Houston 2",
"state"=>"some state",
"country"=>"US",
"postalCode"=>"11111"
]);
```

```
$request = $jpmorgan->generateCheckoutRequest();
```

$request returns

```
[
"statusCode" => $info['http_code'],
"responseStatus" => "ok",
"responseMessage" => "normal redirect to payment page",
"requestID" => $requestId,
"orderNumber" => $orderNumber,
"merchantID" => $this->config['merchant_id'],
];
```

You will need to save the **orderNumber**. This is used to retrieve the details of the transaction later.

Suggestion something like this that would store your accessToken and order number so that you can use it later

```
setcookie('aToken', $jpmorgan->getConfig('accessToken'), (time()+60*15), '/', '.uh.edu'
setcookie('tOrderNumber', $request['orderNumber'], (time()+60*15), '/', '.uh.edu', true
```

Then you can redirect the user to the hosted checkout page

```
if ($request['redirectedUrl']){
header("Location: ".$request['redirectedUrl']);
}else{
print "<pre>";
print_r($request);
```

```
print "</pre>";
}
```

To retrieve information about the request

This page MUST be publically accessible for J.P. Morgan to redirect the user to.

An example of how you would do this is as follows

```
require_once 'uh/vendors/jpmorgan/checkout.php';
$p = new checkout('/publish/lib/auth/jp_morgan_payment_configs/test-certs/config1.json'
$p->addToConfig('accessToken',$_COOKIE['aToken']);
$p->addToConfig('orderNumber',$_COOKIE['tOrderNumber']);

print "<pre>";
print_r($p->getNotification('-2 min',$_COOKIE['tOrderNumber']));
print "</pre>";
```

J.P. Morgan function descriptions

This document describes what functions are public and private with descriptions and return values.

J.P. Morgan documentation can be found at: https://developer.payments.jpmorgan.com/api/commerce/onlinepayments/checkout#/

Public Functions

addConfig($key, $value)

used to add a key to the configruation set. There are several required values for the whole class to work correct.

It is also help full to add values if you need to store a state. You can retrieve those values using

getConfig($key=null).

{

"merchant_name":"",

"merchant_id": "",

"client_id": "",

"certPath": "",

"privateKey": "",

"resource_id": ""

}

Returns: True

getConfig($key=null)

returns either one config item if $key is passed or the whole configuration set if no key is set.

When key is set returns: value of configuration item.

If key can not be found returns: false

When no key returns: array of values

generateToken()

This function is used to generate the token retured to get tha access token. This function is only public because if you need the token for other purposes like postman you can create one.

Returns: string (token)

getAccessToken()

This function generates an access token. Access tokens are used for accessing the API and is normally passed in an API request. This function does not need to be public but is in case you need to create an access for requests like postman.

Returns: Array

```
[
"access_token" => "",
"token_type" => "error",
"expires_in" => "",
"bearer_token" => $this->config['jwtToken']
];
```

createUid($prefix=null)

This function can be used to create unique ids. It used to create a unique id for merchant Order numbers and request ids in generateCheckoutRequest()

Adding a prefix will generate an uid and will look like $prefix-<UID>

Returns: string (UID)

setCustomer($config=array)

Must pass and array to set customer function looks like if a value is not set it is not added to the checkout intent.

```
[
"phone" => "",
"email" => "",
"recipientFullName" => "",
"address1" => "",
"address2" => "",
"city" => "",
"state" => "",
"postalCode" => "",
"country" => ""
]
```

Returns: True (If customer items where set)

Returns: False (when config is not correct)

setShippingAddress($config=array)

Must pass and array to set customer function looks like if a value is not set it is not added to the checkout intent.

[

"recipientFullName" => "",

"address1" => "",

"address2" => "",

"city" => "",

"state" => "",

"postalCode" => "",

"country" => ""

]

Returns: True (If customer items where set)

Returns: False (when config is not correct)

addItem($config=array)

Adds an item to the cart. Config is an array, all fields are required except imageUrl. If no imageUrl is passed then an empty base64 encoded image is added since it is required by J.P. morgan.

imageUrl can be a full url or browser supported url src like data:image/jpeg;base64,

Config array

[

"id"=>"",

"quantity"=>0,

"unitPrice"=>""

"name"=>"",

"description"=> "",

"imageUrl"=> "data:image/jpeg;base64,"

]

generateCheckoutRequest($ordernumberprefix=null){

This function will create a checkout request to the hosted payment page. This function gets the access tokens, etc to complete the task. a unique request id and order number will be created for the request. If you pass $ordernumberprefix then the prefix is added to both the reference number and order number.

*** this function does not redirect the user.**

**** If an error is returned there will be not redirectUr**

Based on J.P. Morgan API
https://developer.payments.jpmorgan.com/api/commerce/onlinepayments/checkout#/paths/checkout-intent/post

Returns: array

[

"statusCode" =>"",

"responseStatus" => "ok | error",

"responseMessage" => "normal | <error message from JPmorgan>",

"requestID" => "",

"orderNumber" => "",

"merchantID" => "",

"checkoutSessionToken" => "",

"redirectUrl" =>

]

getNotification($date, $merchantOrderNumber=null)

Will return notifiction for a given date range. Recommend to use a $date to say when to start looking from.

* Be aware that a 30 day period of time will be used.

** Only most recent or when merchant order number is passed, limited to 1 notifiction

Returns: array

```
[
"status"=>"",
"totalAmount"=>"",
"transactionReference"=>"",
"checkoutReference"=>"",
"merchantOrderNumber"=>"",
"requestId"=>"",
"paymentMethod"=>[
"last4CardNumber"=>"",
"expiry"=>"",
"cardTypeName"=>"",
],
"billingAddress"=>"",
"responseCode"=>"",
"responseMessage"=>"",
"deviceIp"=>"",
```

"userAgent"=>"",

"acceptedTimestamp"=>"",

"transactionTimestamp"=>"",

"raw"=>[]

];

If you are wanting test instructions you can use...

testdeclineinfo()

returns array that can be used to help users know how to simulate a decline. Includes card number.

testapprovalinfo()

returns array that can be used to help users know how to simulate an approveal. Includes card number.

test_instructions()

This function will create an HTML version of decline card numbers and approval numbers that can be used for testing purposes.

Private Functions

_construct($configPath=null)

This the main contructor and used to initialize the class when calling new checkout.

configruatioon path can remain blank if you have the values to add to config set using addConfig($key,$value).

Returns: Class object

getCertThumbPrint($pem)

This is an internal function. Passing the PEM cert path to this function the pem is read and generates a fingerprint that is used by generateToken().

Returns: string (sha1 finger print)

generateCheckoutIntent($orderNumber)

Used internally by generateCheckoutRequest to create the checkout intent. This function uses information provided in addItem, setCustomer, and setShippingAddress

More about intent can be found at:
https://developer.payments.jpmorgan.com/api/commerce/onlinepayments/checkout#/paths/checkout-intent/post

Returns: transaction intent for J.P. Morgan

callGetNotifications($start,$end,$merchantOrderNumber=null)

Used to get the notifications from J.P. morgan over a period of time. Max is 30 days and is computed when calling public function getNotification().

Based on J.P. Morgan Documentation:
https://developer.payments.jpmorgan.com/api/commerce/onlinepayments/checkout#/paths/checkout-notifications/get

Returns: (Only most recent or when merchant order number is passed, limited to 1 notifiction)

```
[
"status"=>"",
"totalAmount"=>"",
"transactionReference"=>"",
"checkoutReference"=>"",
"merchantOrderNumber"=>"",
```

"requestId"=>"",

"paymentMethod"=>[

"last4CardNumber"=>"",

"expiry"=>"",

"cardTypeName"=>"",

],

"billingAddress"=>"",

"responseCode"=>"",

"responseMessage"=>"",

"deviceIp"=>"",

"userAgent"=>"",

"acceptedTimestamp"=>"",

"transactionTimestamp"=>"",

"raw"=>[]

];

prettifyNotification($rawNotification)

Used by callGetNotifications to reformat dollar amounts and return information that is useful because the raw response has so much information. If you want the raw response it is already returned in "raw".

[

"status"=>"",

"totalAmount"=>"",

"transactionReference"=>"",

"checkoutReference"=>"",

"merchantOrderNumber"=>"",

"requestId"=>"",

```
"paymentMethod"=>[
"last4CardNumber"=>"",
"expiry"=>"",
"cardTypeName"=>"",
],
"billingAddress"=>"",
"responseCode"=>"",
"responseMessage"=>"",
"deviceIp"=>"",
"userAgent"=>"",
"acceptedTimestamp"=>"",
"transactionTimestamp"=>"",
"raw"=>[]
];
```