



CASHU Merchant Integration Guide

(Getting Started)

Copyright Notice

All rights reserved © 2018 CASHU Inc.

No part of this document may be reproduced in any form or by any means or used to make any derivative, such as translation, transformation or adaptation, without a prior written permission from CASHU Inc.

Trademark Statement (*Statement of Use*)

CASHU is a registered trademark of CASHU Inc. Contents are subject to change without prior notice.

Whom to Contact for Queries

For technical support queries, please contact CASHU Merchant Support team:

Tel.: (962-6) 582-6097

Email: integration@cashu.com

Abraj Al Hejaz Amman

Amman, Jordan

WWW.CASHU.COM

You can also submit a ticket to our Merchant Support team through [this link](#).

Contents

1. About CASHU	4
2. About This Document	5
2.1 Intended Audience.....	5
2.2 Document Conventions	5
3. Glossary.....	7
4. CASHU Prepaid Payment Transaction Workflow	9
5. CASHU Prepaid Integration Steps	11
Appendix 1 – URLs	38
Appendix 2 – (<i>Important</i>) Security Measures	39

1. About CASHU

CASHU is an electronic payment solution developed specifically for the Arab speaking world with the aim to help any business become a successful online merchant in the region. It is available for almost everyone selling products or services without restrictions, setup fees or bureaucracy.

The CASHU Merchant Service offers you a way to accept payments through the CASHU Payment Gateway.

2. About This Document

This document provides systematic instructions to get started with CASHU Integration. It tackles the integration of CASHU Products and describes the features, services, parameters and the technical environment required by merchants and provided by CASHU to achieve a successful implementation.

CASHU supports the setup of payments in both test and production environments by providing merchants with two accounts; the first one is designated for testing (*Sandbox*) and the other one is for live transactions.

2.1 Intended Audience

This document is intended for software developers with knowledge and authority to integrate third-party payment solutions on a merchant's e-commerce platform. Since CASHU Payment Gateway does not require any software installation, the required skills are determined by technologies used on the merchant's website and the necessity to integrate the Payment Gateway with back-end systems.


The integration details in this document apply to all merchants regardless of the programming language.

2.2 Document Conventions

The following typographic conventions are used in this document:

Convention	Description
Bold	Used for headings, names of pages and menus, elements supplied by the system, names of commands and illustrated steps.
<u>Underlined</u>	Text in this typeface represents important resources and names of used programs, and it indicates terms that are defined in the text or appear in a glossary.
<u>Bold and Underlined</u>	Used for sub heading.
[]	Contains regular expression definitions.

The following types of notes are used in this document:

Type	Description
Note	<div>NOTE! <i>Notes with a blue background contain important information for you to keep in mind, as well as helpful tips.</i></div>

3. Glossary

Term	Definition
Merchant ID	A unique ID for each merchant. This ID will be chosen and created during the process and will be used for login purposes, as well as identifying the merchant's requests in the integration process. It is preferred to use a descriptive and an easy-to-remember ID.
Encryption Keyword	Any suitable Keyword selected and defined by the merchant, which will be used to verify the payment transaction request, by using its value in the <u>Token</u> , <u>verificationString</u> and <u>cashuToken</u> calculations.
Token	<p>An encrypted parameter sent by the merchant in every payment request and used to verify that the transaction is owned by this particular merchant. It is also sent after any successful payment transaction by CASHU to <u>Return URL</u> and <u>Notification URL</u>.</p> <p>The merchant should recalculate the original parameters and match them with this parameter to verify that the correct amount was collected successfully without any manipulation.</p> <p>The <u>Token calculation</u> is explained in a later section.</p>
verificationString	<p>A parameter sent after any successful payment transaction by CASHU to the <u>Return URL</u>. The merchant should recalculate the received parameters and match them with this parameter to verify that the transaction response was received from CASHU.</p> <p>The <u>verificationString calculation</u> is explained in a later section.</p>
Return URL	Refers to the URL where the customer will be redirected after a successful payment transaction, and it is <u>mandatory</u> .
cashuToken	<p>One of the parameters sent by CASHU to the merchant's <u>Notification URL</u>, where the merchant should calculate the received parameters and match them with this parameter to verify that the transaction response was received from CASHU.</p> <p>The <u>cashuToken calculation</u> is explained in a later section.</p>
Notification URL	Refers to the URL where the merchant will be receiving the XML Notification sent by CASHU immediately after a successful payment transaction. It is optional but <u>recommended</u> .
Merchant Display Name	The name that will appear to the customers on the payment page, which refers to the merchant's trade name.
Service Name	<p>A parameter the merchant should use when referring to <u>Sub Merchant Checkout</u> name that currently is under integration.</p> <p>CASHU allows the merchant to create multiple Merchant Checkout Pages under the same Merchant Account, in case a merchant has more than one Merchant Checkout Page.</p>

SDK Notification URL	Refers to the URL where the merchant will be receiving the XML Notifications sent by CASHU immediately after a successful payment transaction when using CASHU's Mobile SDK. Its mandatory.
-----------------------------	---

4. CASHU Prepaid Payment Transaction Workflow

A. Web-based Solution:

A payment transaction between the merchant and the customer on CASHU takes the following steps:

1. The customer visits the merchant website and selects to purchase a product/service.
2. The customer follows the merchant's purchasing flow and proceeds to the payment section (**checkout page**) of the merchant website.
3. The customer selects to pay with CASHU.
4. The customer is redirected to CASHU payment page, where the description of the product/service and the total price are presented.
5. The customer authorizes and completes the payment (*clicks pay*).
6. CASHU validates the payment. If the transaction is approved, it is executed in real-time (*directly*) and recorded in CASHU's system.
7. In case of any error, CASHU either displays a relevant error message on CASHU payment page, or if the merchant implements the Sorry URL; CASHU sends an error code to the merchant's Sorry URL (*it should be predefined*), and the merchant will be responsible for showing the relevant error description to the customer.
8. After the execution of a transaction, the customer is redirected to a predefined section of the merchant's website (Return URL) and CASHU will send a notification to the merchant's Notification URL (*if implemented*).

B. Mobile SDK Solution:

A payment transaction between the merchant and the customer on CASHU takes the following steps:

1. The customer visits the merchant's mobile application and selects to purchase a product/service.
2. The customer follows the merchant's purchasing flow and proceeds to the payment section (**checkout screen**) of the merchant mobile application.
3. The customer selects to pay with CASHU.
4. The customer is redirected to CASHU payment SDK login screen.
5. The customer logs in to their wallet.
6. The customer authorizes the payment (*clicks Pay Now!*).
7. CASHU validates the payment. If the transaction is approved, it is executed in real-time (*directly*) and recorded in CASHU's system.

8. In case of any error, either CASHU displays a relevant error message to the customer, or CASHU sends an error code to the merchant's mobile application (*it should be predefined*), and the merchant will be responsible for showing the relevant error message to the customer.
9. After the execution of a transaction, the customer will be redirected to the merchant's mobile application (with success status) and CASHU will send a notification to the merchant's SDK Notification URL.

5. CASHU Prepaid Integration Steps

Please apply the following steps to integrate CASHU Prepaid into your website:

5.1 Registering as a Merchant on CASHU's Testing Area (*Sandbox*)

- Go to <https://sandbox.cashu.com/Merchants/en/register>.
- On the main registration page, enter the following details:
 - A **Merchant ID** that will be used for login. Please use a descriptive and an easy-to-remember word, as you will use it in the Integration process.
 - A valid **Email Address** of yours, where the activation email and other further correspondence will be sent.
 - Your **Country**.
 - Your **Website Address(es)**.
 - A valid **Mobile Number** of yours.

**NOTE:**

You cannot register as a CASHU merchant unless you own a registered company and have a valid trade license/certificate of incorporation and a bank account under your company name.

- Once you complete the registration form, click "**I agree! Register Now**", and activate your account through the activation email sent to your registered email address.

5.2 Logging in to your Merchant Account on the Testing Area (*Sandbox*)

- From the following page <https://sandbox.cashu.com/Merchants/en/login>, log in to your testing merchant account using the **Merchant ID** you have chosen in the registration form, and the password that was sent to your email address.

**NOTE!**

*If you forgot your login password, you can use the **Password Reminder** tool located at this link:*

<https://sandbox.cashu.com/Merchants/en/login>

and a new password will be sent to your registered email address.

- Your account will indicate that you are in the "**Test it**" phase, as shown in Figure 1 below.

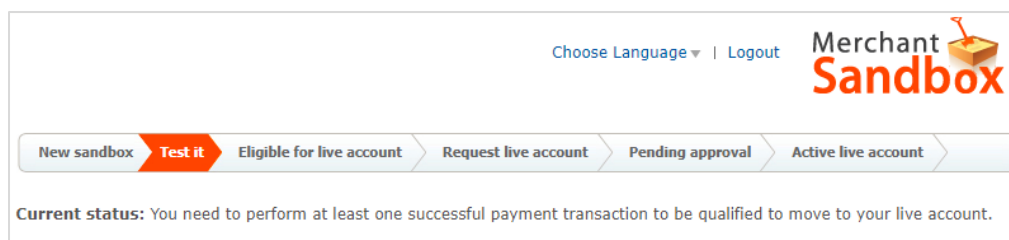
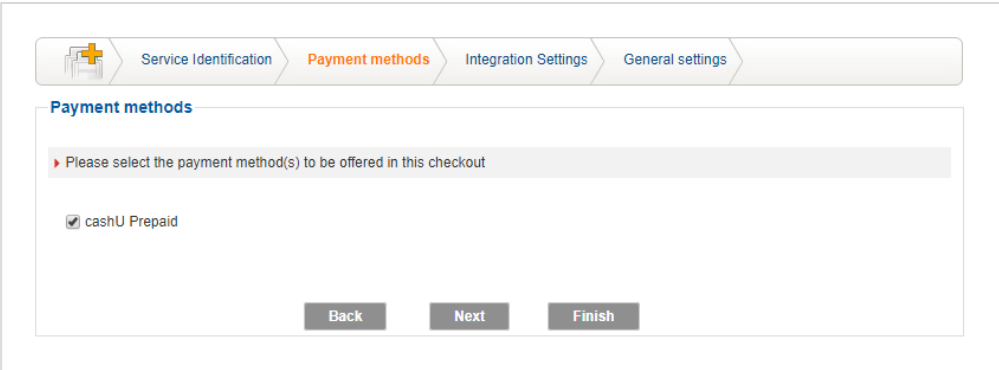


Figure 1: Current Status – Test It

- In this phase, you are requested to complete the setup for at least one **Merchant Checkout Page**. To do that you need to set your **Service Setup** as described below:
 - Click the **“Merchant Services”** tab, and choose **“Service Setup”** from the drop-down menu. The following page will appear to you.

Figure 2: Service Setup – First Step

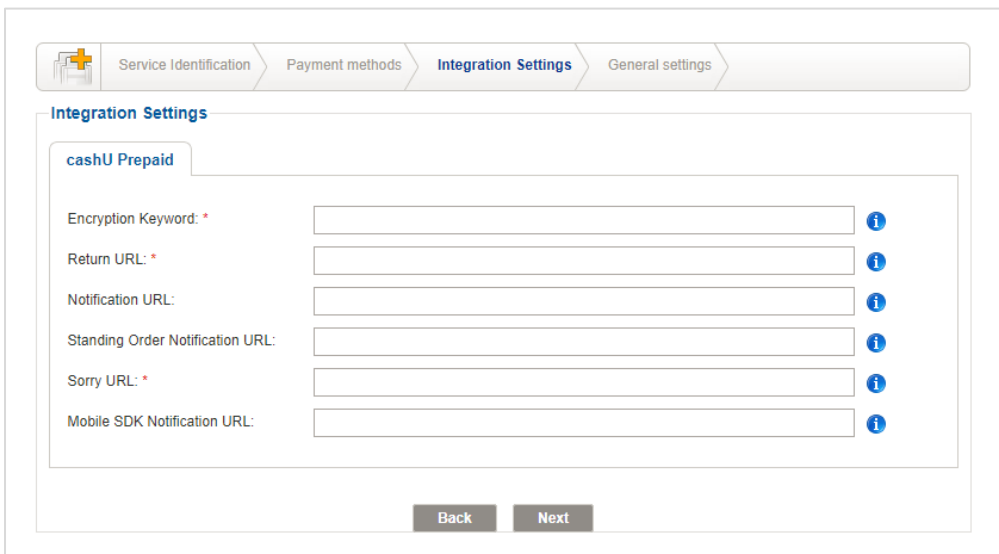
- As shown in the wizard above, and under the **“Merchant Checkout”** section, you already have a service called **“Default”**. The first step is editing this service setup. Click on the **“View Settings”** button in the table.
- Under the **“Service Identification”** section, you will see that **“Default”** is shown in the **“Service Name”** field. Enter a suitable **“Merchant Display Name”**, which will appear on your Checkout Page, and then click **“Next”**.
- The second step is choosing the **“Payment Methods”**, as shown in Figure 3 below. You will see that **“CASHU Prepaid”** is selected by default. Click **“Next”**.



The screenshot shows the 'Payment methods' step of the 'Service Setup' wizard. At the top, there is a navigation bar with four tabs: 'Service Identification', 'Payment methods' (which is highlighted in orange), 'Integration Settings', and 'General settings'. Below the navigation bar, the title 'Payment methods' is displayed. A message box says 'Please select the payment method(s) to be offered in this checkout'. Below this, there is a checkbox labeled 'cashU Prepaid' which is checked. At the bottom, there are three buttons: 'Back', 'Next', and 'Finish'.

Figure 3: Service Setup – Payment Methods

- Next comes the most important step, which is “**Integration Settings**”, as shown in Figure 4 below. This is where you enter your Encryption Keyword, Return URL, Notification URL and other fields. Each field that has a red star is mandatory and you should enter valid URLs in there, while the others are optional.



The screenshot shows the 'Integration Settings' step of the 'Service Setup' wizard. At the top, there is a navigation bar with four tabs: 'Service Identification', 'Payment methods', 'Integration Settings' (which is highlighted in blue), and 'General settings'. Below the navigation bar, the title 'Integration Settings' is displayed. A tab labeled 'cashU Prepaid' is selected. Below this, there are six input fields, each with a red star indicating it is mandatory. The fields are: 'Encryption Keyword: *', 'Return URL: *', 'Notification URL:', 'Standing Order Notification URL:', 'Sorry URL: *', and 'Mobile SDK Notification URL:'. Each field has a blue information icon to its right. At the bottom, there are two buttons: 'Back' and 'Next'.

Figure 4: Service Setup – Integration Settings

- o **Encryption Keyword** (*mandatory*): Enter any suitable keyword (*no one else should know it besides you*). The **Encryption Keyword** will be used in the **Token**, **VerificationString** and **CASHUToken** calculation to verify the payment transaction.
- o **Return URL** (*mandatory*): Enter the URL to where you wish to redirect your customer after any successful payment transaction. This URL should be as follows:
 - Must be entered using a full path. See the following example:
<http://www.yourwebsite.com/successPage.php>

- Special characters are not allowed. Even when entering a URL that includes special characters in the **Return URL** field, those characters will be automatically removed. Instead of that, the merchant can use the optional parameters (*txt2 to txt5*) that CASHU offers to pass any data that he wants to receive back on his Notification URL (*please check CASHU Parameters section*).
- Must be accessible and should accept server-to-server calls, as CASHU will use curl request to call that URL and invoke POST the success parameters to it. In the case that you are using the mobile web interface, CASHU will curl request to call the return URL and invoke GET the success parameters to it.
- .

**NOTE!**

All images, scripts and CSS style sheets that you are calling in your Return URL should be presented in full path, otherwise it will not be displayed correctly, and will look broken.

- **Notification URL** (*optional*): Enter the URL of the page where you wish to receive CASHU XML Notifications after any successful payment transaction. However, this URL should be as follows:
 - Must be hosted on a secure server (*HTTPS*) for the live mode.
 - Must be entered with a full path. See the following example:
<https://www.yourwebsite.com/NotificationPage.php>
 - Special characters are not allowed. Even when entering a URL that includes special characters in the Notification URL field, they will be automatically removed. Instead of that, the merchant can use the optional parameters (*txt2 to txt5*) that CASHU offers to pass any data that he wants to receive back on his Notification URL (*please check section CASHU Parameters*).
 - Must be accessible and should accept server-to-server calls, as CASHU will POST the success parameters to this URL.
- **Sorry URL** (*mandatory*): Enter the URL where you want CASHU to redirect the customer in case the payment transaction failed for any other reason, along with a list of parameters that identifies the reason of the transaction failure.
 - Must be entered with a full path, like:
<http://www.yourwebsite.com/SorryPage.php>
 - Special characters are not allowed. Even when entering a URL that includes special characters in the Sorry URL field, they will be automatically removed.
- **SDK Notification URL** (*mandatory if CASHU SDK is enabled on your merchant account*): Enter the URL where you want CASHU to send XML notifications for successful payments done through the SDK on your mobile app.
- Once you enter the URLs (*keeping in mind the format*), click "**Next**".

**NOTE!**

The parameters will be sent using HTTP POST. All URLs must be entered with full path, like: *(http://www.yourwebsite.com/SorryPage.php)* and they must be accessible.

URLs cannot be the same. You should enter a different URL in each field.

The URLs used in your Sandbox account cannot be used for your Live account. If you want to use the same URLs that you have set in Sandbox, you will need to change them first in your Sandbox account before using them in the live account.

The Sorry URL is now mandatory. Make sure to use a proper one, otherwise, you will not be able to complete the integration process.

- The last step is the “**General Settings**”. As shown in Figure 5 below, this step is for setting any payment restrictions. We usually recommend keeping the default settings without changing anything, unless you really want to restrict the payment to some countries, or only for the “Know Your Customer” (KYC) complied CASHU customers.

General settings

Payment Restrictions

Payment Settings

▶ Limit your Product/Service sale to the following countries.

- ☒ ALL
- ☒ GCC
- ☒ LEVANT
- ☒ NORTH AFRICA
- ☒ EU
- ☒ Others
- ☒ Iraq
- ☒ Iran
- ☒ Nigeria
- ☒ Yemen

KYC Limitation

▶ You can limit your Product/Service sales to only KYC complied Payment accounts.

☒ I agree to add this limitation to my transactions.

Note

1. If you do not select any country, then payments from all countries will be accepted.

Back **Finish**

Figure 5: Service Setup – Payment Restrictions

- After you click **“Finish”**, the below page will appear, meaning that the setup for this checkout page has been successfully completed.

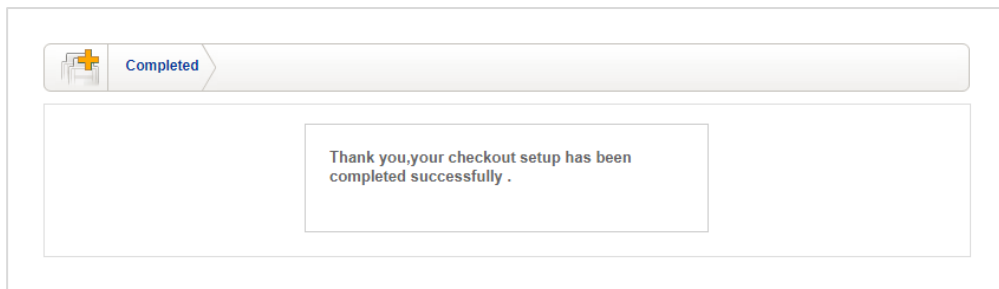


Figure 6: Service Setup Complete

**NOTE!**

CASHU allows the merchant to create as many checkout pages as needed, which can be managed from the same Merchant Account.

If you want to integrate more than one website under the same Merchant ID, you can create a sub checkout for each website. A new checkout will be created by clicking on the add button as appears below in Figure 3.

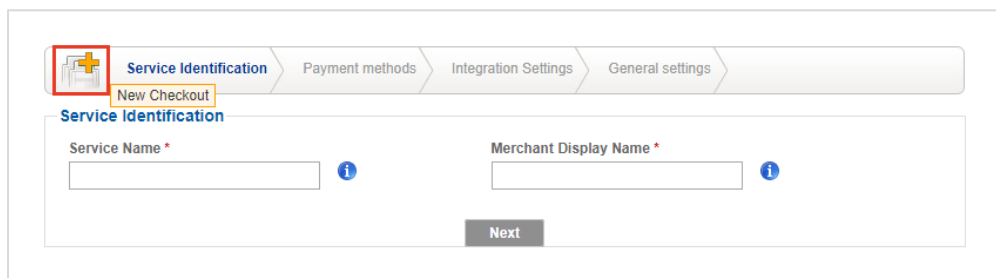


Figure 7: Create More Checkout Pages

- After completing the setup for at least one **Merchant Checkout Page**, you need to choose your **Encryption Type**. Click the **“Merchant Services”** tab, and choose **“Security Settings”** from the drop-down menu. The following page will appear to you:

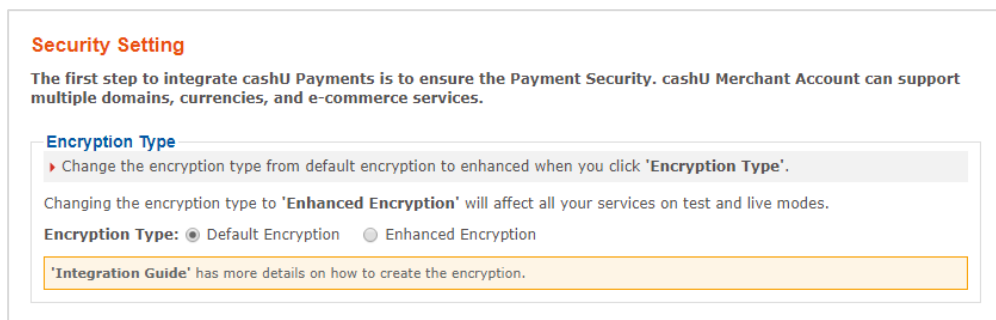


Figure 8: Security Settings

- CASHU offers two types of encryption as shown in figure 8 above: the **Default Encryption** (*which is already selected*), and the **Enhanced Encryption**. We recommend that you always select the **Enhanced Encryption**.

In the **Default Encryption**, the Token calculation will be as follows:
`MD5("merchant_id:amount:currency:encryption key")`

In the **Enhanced Encryption**, the Token calculation will be as follows:
`MD5("merchant_id:amount:currency:session_id:encryption key")`

As you can see, the only difference between both types is how you calculate the Token Value. For example, if you choose to use the **Default Encryption** type, then you will remove the **session_id** from the **Token** calculation (*which is not recommended*). While if you choose to use the **Enhanced Encryption**, you will add the **session_id** to the Token calculation.

- Once you are done setting up your **Encryption Type**, you can start building your **Integration Form**.

**NOTE!**

*Remember that the data entered in the above wizard will be used in the **Integration Form** Building.*

5.3 Start building your Integration Form:

CASHU Parameters:

Before you start, please have a look at CASHU Parameters and their descriptions. The below list of Parameters will be used in the integration process:

Parameter Name	Description	Value Required
merchant_id	The Merchant ID as entered during registration. Must be between 1 and 20 characters long with characters chosen from [A-Za-z_\.\-@0-9].	Mandatory
amount	A number identifying the value of the payment transaction. It must be presented in maximum 2 decimal points.	Mandatory

currency	<p>The currency in which the service/item is sold on the merchant's website. The accepted currencies are:</p> <p>USD (<i>United States Dollar</i>) AED (<i>Emirati Dirham</i>) EUR (<i>Euro</i>) JOD (<i>Jordanian Dinar</i>) EGP (<i>Egyptian Pound</i>) SAR (<i>Saudi Riyal</i>) DZD (<i>Algerian Dinar</i>) LBP (<i>Lebanese Pound</i>) MAD (<i>Moroccan Dirham</i>) QAR (<i>Qatari Riyal</i>) TRY (<i>Turkish Lira</i>) KWD (<i>Kuwaiti Dinar</i>) BHD (<i>Bahraini Dinar</i>) OMR (<i>Omani Riyal</i>) GBP (<i>British Pound Sterling</i>)</p>	Mandatory
language	<p>A string identifying the interface language of the payment page.</p> <p>Supported values are "en" for the English interface and "ar" for the Arabic interface.</p>	Mandatory
display_text	<p>English or Arabic description of the transaction. This is displayed to the customer on the payment page.</p> <p>Maximum length allowed is 250 characters.</p> <p>Allowed characters are [A-Za-z_\. \- @\`"0-9] as well as Arabic characters.</p> <p>This parameter will appear in the Statement of Account under column named "Item/Service".</p>	Mandatory
txt1	<p>One of the multi-use variables to enable the merchant to send any additional value that he wants to pass to his Return URL page.</p> <p>Maximum length allowed is 150 characters.</p> <p>Allowed characters are [A-Za-z_\. \- 0-9].</p>	Mandatory
test_mode	Must be '0'.	Mandatory

Token	<p>Must be a hex-encoded MD5 HASH (a 32-digit hexadecimal number) of a concatenation of the following parameter values separated by ":" (values should be in lower case), appended with the Encryption Keyword. The Encryption Keyword is selected by the merchant (please refer to step 1 for more details).</p> <p>The example below explains how to create the hash. The used values are fictitious.</p> <p>First, values should be converted to lower case:</p> <p>Parameter: merchant_id / value: test</p> <p>Parameter: amount / value: 15.25</p> <p>Parameter: currency / value: aed</p> <p>Then their values are concatenated:</p> <p>test:15.25:aed</p> <p>The Encryption Key is appended to the end of the string:</p> <p>test:15.25:aed:encryptionkey</p> <p>The MD5 function is called on the resulting string:</p> <p>MD5("test:15.25:aed:encryptionkey")</p> <p>The result of the function is the hash to be included in the request. This must be presented in 32 digit hexadecimal number.</p>	Mandatory
--------------	--	-----------

Below is a description of optional parameters that CASHU provides for merchants to include any data related to the payment transaction and needed to be passed to the Return URL:

Parameter Name	Description	Required
txt2	One of the multi-use variables to enable the merchant to send any additional value that he wants to pass to his success page. Maximum length allowed is 250 characters from [A-Za-z_\. \-@0-9].	Optional
txt3	One of the multi-use variables to enable the merchant to send any additional value that he wants to pass to his success page. Maximum length allowed is 250 characters from [A-Za-z_\. \-@0-9].	Optional
txt4	One of the multi-use variables to enable the merchant to send any additional value that he wants to pass to his success page. Maximum length allowed is 250 characters from [A-Za-z_\. \-@0-9].	Optional
txt5	One of the multi-use variables to enable the merchant to send any additional value that he wants to pass to his success page. Maximum length allowed is 250 characters from [A-Za-z_\. \-@0-9].	Optional

session_id	Unique reference to the transaction generated by the merchant. Maximum length allowed is 100 characters from [A-Za-z_\.@0-9]. This parameter will appear in the exported Statement of Account under column named " Merchant Transaction Reference ".	Mandatory
service_name	Name of the service or domain entered by the merchant that has multiple checkout pages. This parameter cannot be empty if the merchant uses multiple checkout pages. Maximum length allowed is 50 characters from [A-Za-z_\.@0-9].	Mandatory only If using Merchant Sub Checkout

The below Parameter is used in the **POST Request**:

Parameter Name	Description	Required
Transaction_Code	This is a unique reference to the transaction that is returned by CASHU, if parameters check was successful. Length is 40 characters from [a-z0-9].	Mandatory

CASHU Integration (Premier Integration):

A sample on Direct Integration PHP code [soap Request] on Testing Area (*Sandbox*)
(Please note that some URLs change in the Live Area.)

- Start with the **Items Amount** page, where you list down your items and their Amounts.

Items Amount page:

```
<html>
  <p>Please enter the amount that you want to pay:</p>
  <form action="www.merchantdomainname.com/IntegrationPage.php" method="post">
    <input type="text" name="amount" value="">
    <input type="submit" value="proceed">
  </form>
</html>
```

CASHU Integration Code page:

- In the first step, the interaction is based on a **soap** call initiated by the merchant to CASHU server exactly to this URL:
(<https://sandbox.cashu.com/secure/payment.wsdl>) for the Testing Area (Sandbox), and this URL (<https://secure.cashu.com/payment.wsdl>) for the Live Area.

```
<?php

ini_set("soap.wsdl_cache_enabled", "0");
ini_set('customer_agent', 'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:20.0)
Gecko/20100101 Firefox/20.0');
//The above line will add the User-Agent to the header of your request,
and the soap library 'SoapClient' will add the Host header name
automatically.

$merchant_id = 'test';
$encryption_key = 'encryption key_value';
$amount = $_POST["amount"]; // This value will be posted from the above
HTML page
$currency = 'usd';
$display_text = 'Description about the item that will appear to the
customer';
$language = 'en';
$session_id = '123ct2';
$txt1 = "item";
$testmode = 0;

// The below Parameters are not required, especially in the default
Merchant Checkout Page:
$txt2 = '';
$txt3 = '';
$txt4 = '';
$txt5 = '';
$service_name = '';

// If using Enhanced Encryption:
$token = md5(strtolower($merchant_id) . ':' . $amount . ':' .
strtolower($currency) . ':' . strtolower($session_id) . ':' .
$encryption_key);
// If using Default Encryption:
$token = md5(strtolower($merchant_id) . ':' . $amount . ':' .
strtolower($currency) . ':' . $encryption_key);

$client = new
SoapClient("https://sandbox.cashu.com/secure/payment.wsdl",
array('trace' => true)); // Change the URL to
"https://secure.cashu.com/payment.wsdl" for Live Area.
$request = $client-
>DoPaymentRequest($merchant_id,$token,$display_text,$currency,$amount,$l
anguage,$session_id,$txt1,$txt2,$txt3,$txt4,$txt5,$testmode,$service_nam
e);

//The Parameters must be in this order: merchant_id, token,
display_text, currency, amount, language, session_id, txt1, txt2, txt3,
txt4, txt5, test mode, service name
```

- In the second step, the interaction is based on secure **HTTP POST** to the payment page exactly to this URL (<https://sandbox.cashu.com/cgi-bin/payment/pcashu.cgi>) on the Testing Area (Sandbox), and to this URL (<https://www.cashu.com/cgi-bin/payment/pcashu.cgi>) on the Live Area. It requires one parameter that is returned from CASHU server as a result of the **soap** call in the first step.

```
//Get transaction code from the response
$tmp = strstr($request, '=');
$Transaction_Code = substr($tmp, 1);
echo '<html>';
if ($Transaction_Code != '') {

    //Change the URL in the blow HTML form to "https://www.cashu.com/cgi-bin/payment/pcashu.cgi" for live area
    echo '
<form action="https://sandbox.cashu.com/cgi-bin/payment/pcashu.cgi"
method="post">
<input type="hidden" name="Transaction_Code"
value="'. $Transaction_Code. '">
<input type="submit" name="but" value="Pay with CASHU!">
</form>';
}
else
{
    echo $request;
}
echo '</html>';

?>
```

The below table shows the errors description that might return to you, in case your SOAP request contains wrong data:

Error Message	Description
INSECURE_REQUEST	Request was not sent through HTTPS.
SYSTEM_NOT_AVAILABLE	CASHU servers are not responding.
INVALID_PARAMETER	One or more of the required parameters is null or has invalid character.
INACTIVE_MERCHANT	The Merchant Account is inactive.
TOKEN_CHECK_FAILURE	The submitted value for the token parameter is incorrect.
GENERAL_SYSTEM_ERROR	Error happened while processing the transaction

Handling CASHU Parameters on Return URL:

- Return URL Notifications is the default type of CASHU notifications.
- CASHU uses the HTTP POST to submit all the following data to the merchant's Return URL:

Parameter Name	Description
amount	The same amount sent by the merchant.
currency	The same currency sent by the merchant.
language	The interface language sent by the merchant.
txt1	The item description sent by the merchant.
txt2	The txt2 parameter sent by the merchant.

txt3	The txt3 parameter sent by the merchant.
txt4	The txt4 parameter sent by the merchant.
txt5	The txt5 parameter sent by the merchant.
token	The MD5 String sent by the merchant.
trn_id	A unique reference number to the transaction created by CASHU.
trnDate	The transaction's date and time in GMT.
session_id	The merchant's unique reference to the transaction (<i>if it was sent by the merchant</i>). This parameter will appear in the exported Statement of Account under column named "Merchant Transaction Reference".
verificationString	<p>It is a hex-encoded SHA1 HASH (40 HEX characters) of a concatenation of the following parameter values separated by ":" Merchant ID, CASHU Transaction ID and the Encryption Key. Please note that the Merchant ID should be in lower case.</p> <p>The example below shows how to create the hash. The used values are fictitious.</p> <p>First, values should be converted to lower case: Parameter: merchant_id / value: test Parameter: trn_id / value: 4566 Then their values are concatenated: test:4566</p> <p>The Encryption Key is appended to the end of the string: test:4566:encryptionkey</p> <p>The SHA1 HASH function is called on the resulting string: SHA1 HASH ("test: 4566: encryptionkey")</p> <p>The result of the function is the hash to be included in the response.</p>
netAmount	The net amount credited to the merchant.
test_mode	In most cases, the value will be '0'.
servicesName	The name of the service (<i><u>merchantCheckout</u> or <u>merchantSubCheckout</u></i>) where the purchase transaction has been processed.

**NOTE!**

In addition to the above table, CASHU also returns the values of the parameters from (txt2 to txt5). CASHU offers these parameters to allow the merchant to send any values they want, and CASHU sends them back to the Return URL and Notification URL along with the successful transaction.

As for the Default Checkout, servicesName parameter will be an empty string. However, for any sub Checkout, the servicesName must be passed as configured in the Merchant Account under Service Setup.

- Below is a sample code in PHP language showing how you can get the posted parameters, and how to verify the transaction before releasing any goods or services:

```
<?php

// First get the parameters values:

$merchant_id = 'test';
$encryptionKey = 'encryption_key_value';
$amount = $_POST["amount"];
$currency = $_POST["currency"];
$language = $_POST["language"];
$txt1 = $_POST["txt1"];
$token = $_POST["token"];
$trn_id = $_POST["trn id"];
$session_id = $_POST["session id"];
$verificationString = $_POST["verificationString"];
$netAmount = $_POST["netAmount"];

// And the below parameters, if used:
// $txt2 = $_POST["txt2"];
// $txt3 = $_POST["txt3"];
// $txt4 = $_POST["txt4"];
// $txt5 = $_POST["txt5"];

$originalAmount = "Original_amount"; // Pass the original amount for this
order from your system, not the one sent by CASHU.
$originalCurrency = "Original_currency"; // Pass the original currency for
this order from your system, not the one sent by CASHU.

// Then you need to calculate the verification string, and make sure that it
matches the verificationString parameter value as below:
$calculatedVerificationString = sha1(strtolower($merchant_id) . ':' . $trn_id
. ':' . $encryptionKey);
if($calculatedVerificationString!=$verificationString)
{
    // If they did not match, do not release the service or the product, as
this request is not from CASHU's side.
}
else
{
    // If they matched, then calculate the token, and make sure that it matches
the token parameter value as below to verify the amount:

    // For Enhanced Encryption:
    $calculatedtoken = md5(strtolower($merchant_id) . ':' . $originalAmount .
':' . strtolower($originalCurrency) . ':' . strtolower($session_id) . ':' .
$encryptionKey);
    // For Default encryption:
    //$calculatedtoken = md5(strtolower($merchant_id) . ':' . $originalAmount
. ':' . strtolower($originalCurrency) . ':' . $encryptionKey);
```



```
if($calculatedtoken!=$token)
{
    // If they did not match, do not release the service.
    // In such case, please refund the transaction.
}
else
{
    // If they matched, then release the service.
}
}

?>
```

**NOTE!**

Make sure to use the original amount and the original currency in the Token verification step, to make sure that no one manipulated the response.

Do not use the amount and currency values that sent to you by CASHU in the verification process.

Handling CASHU Parameters on Notification URL (or) on SDK Notification URL:

- CASHU offers an extra way to make sure that the merchant receives the notification, even if the customer has closed the internet browser before reaching the merchant's Return URL. It is called the Notification URL (*XML Notification*).
- In case the merchant uses CASHU SDK on their mobile app, CASHU will also make sure that the merchant receives the notification, even if the customer has closed the app. It is called the SDK Notification URL (*XML Notification*).
- CASHU always recommends to use this feature for the following reasons:
 - CASHU keeps sending the notification (*up to six times on the Live Area*) with time interval of 2 hours, until a confirmation is received from the merchant.
 - CASHU sends the XML notification as a value of a parameter called "sRequest" using the POST method.
- The followings are short descriptions of the fields in the XML notification:

Parameter Name	Description
responseCode	The expected value is "OK".
trnDate	The transaction's date and time in GMT.
cashU_trnID	A unique reference number to the transaction created by CASHU.

cashUToken	<p>The MD5 string consists of the following values: Merchant ID CASHU transaction Encryption key The Merchant ID and the Encryption Key should be in lower case. The example below shows how to create the hash. The used values are fictitious. First, values should be converted to lower case: Parameter: merchant_id / value: test Parameter: cashU_trnID / value: 401231 Parameter: Encryption Key / value: encryptionkey Then their values are concatenated: test:401231:encryptionkey</p> <p>The MD5 function is called on the resulting string: MD5("test:401231:encryptionkey") The result of the function is the hash to be included in the response.</p>
amount	<p>A number identifying the value of the payment transaction. It must be presented in maximum 2 decimal points.</p>
currency	<p>The currency in which the service/item is sold on the merchant's website. The accepted currencies are: USD (<i>United States Dollar</i>) AED (<i>Emirati Dirham</i>) EUR (<i>Euro</i>) JOD (<i>Jordanian Dinar</i>) EGP (<i>Egyptian Pound</i>) SAR (<i>Saudi Riyal</i>) DZD (<i>Algerian Dinar</i>) LBP (<i>Lebanese Pound</i>) MAD (<i>Moroccan Dirham</i>) QAR (<i>Qatari Riyal</i>) TRY (<i>Turkish Lira</i>) KWD (<i>Kuwaiti Dinar</i>) BHD (<i>Bahraini Dinar</i>) OMR (<i>Omani Riyal</i>) GBP (<i>British Pound Sterling</i>)</p>
language	<p>It identifies the interface language of the payment page. Supported values are: "en" for the English interface and "ar" for the Arabic interface.</p>
display_text	<p>English or Arabic description of the transaction. This is displayed to the customer on the payment page. Maximum length allowed is 250 characters. Allowed characters are [A-Za-z_\. \- @\^"0-9] as well as Arabic characters. This parameter will appear in the Statement of Account under a column named "Item/Service".</p>

txt1	Maximum length allowed is 150 characters. Allowed characters are [A-Za-z_\.0-9].
txt2	The txt2 parameter sent by the merchant.
txt3	The txt3 parameter sent by the merchant.
txt4	The txt4 parameter sent by the merchant.
txt5	The txt5 parameter sent by the merchant.
test_mode	In most cases, the value will be '0'.
session_id	A unique reference to the transaction generated by the merchant. Maximum length allowed is 100 characters from [A-Za-z_\.0-9]. This parameter will appear in the exported Statement of Account under a column named "Merchant Transaction Reference".
servicesName	Name of the service or domain entered by the merchant that has multiple checkout pages. This parameter cannot be empty if the merchant uses multiple checkout pages. Maximum length allowed is 50 characters from [A-Za-z_\.0-9].

**NOTE!**

In addition to the above table, CASHU also returns the values of the parameters from (txt2 to txt5). CASHU offers these parameters to allow the merchant to send any values he wants, and CASHU sends them back to the Return URL along with the successful transaction.

As for the Default CheckoutservicesName parameter will be empty string.

- The following example shows the **XML structure** returned by CASHU (*values are fictitious*):

```
<cashUTransaction>
  <merchant_id>test</merchant_id>
  <token>66a31cd699d8d9cb454df1f6cec30c2c</token >
  <display_text>Baseball Hat</display_text>
  <currency>AED</currency >
  <amount>100</amount>
  <language>en</language>
  <session_id>asdasd-234-asdasd</session_id>
  <txt1>item27</txt1>
  <txt2>123</txt2>
  <txt3></txt3>
  <txt4></txt4>
  <txt5></txt5>
  <serviceName></serviceName>
  <responseCode>OK</responseCode>
  <trnDate>2008-01-01 09:20:01</trnDate>
  <cashU_trnID>401231</cashU_trnID>
  <cashUToken>234c36b77ffac7905165bb72d582342</cashUToken>
  <netAmount>97.00</netAmount>
</cashUTransaction>
```

- The following are short descriptions of the fields that CASHU expect from the merchant's XML response:

Parameter Name	Description
merchant_id	The Merchant ID registered in CASHU.
cashU_trnID	The unique reference number of the transaction created by CASHU.
cashUToken	The MD5 string sent by CASHU to the merchant for the notification.
responseCode	The expected value is "OK" .
responseDate	The timestamp of the response should include the date and time in GMT. The accepted format is: "yyyy-mm-ddhh:mm:ss" .

- The response should be sent via secure HTTP POST to this URL on the Testing Area (*Sandbox*):

<https://sandbox.cashu.com/cgi-bin/notification/MerchantFeedBack.cgi>

And to this URL on the Live Area:

<https://www.cashu.com/cgi-bin/notification/MerchantFeedBack.cgi>

- CASHU expects to receive the response in the following format:

```
"sRequest=<cashUTransaction><merchant_id>test<merchant_id><cashU_trnID>401231</c
ashU_trnID><cashUToken>234c36b77ffac7905165bb72d582342</cashUToken><responseCode
>OK</responseCode><responseDate>2008-01-01
09:21:01</responseDate></cashUTransaction>" ;
```

- The below example is for 'XML Notification sent by CASHU to the merchant' and 'the merchant's response on that notification' using **PHP** language:

```
<?php

// First get the parameters values:
$encryptionKey = 'encryption_key_value';
$sRequest = $ POST["sRequest"];
$successTransaction = new SimpleXMLElement($sRequest);
$merchant_id = $successTransaction->merchant_id;
$amount = $successTransaction->amount;
$currency = $successTransaction->currency;
$language = $successTransaction->language;
$txt1 = $successTransaction->txt1;
$token = $successTransaction->token;
$cashU_trnID = $successTransaction->cashU_trnID;
$session_id = $successTransaction->session_id;
$cashUToken = $successTransaction->cashUToken;
$display_text = $successTransaction->display_text;
$responseCode = $successTransaction->responseCode;
$serviceName = $successTransaction->serviceName;
$trnDate = $successTransaction->trnDate;

// And the below parameters if used:
// $txt2 = $successTransaction->txt2;
// $txt3 = $successTransaction->txt3;
// $txt4 = $successTransaction->txt4;
// $txt5 = $successTransaction->txt5;

$originaAmount = "Original_amount"; // Pass the original amount for this
order from your system.
$originaCurrency = "Original_currency"; // Pass the original currency for
this order from your system.

// Then, you need to calculate the cashUToken and make sure that it
matchcashUToken parameter value as below:
$calculatedCashuToken = MD5(strtolower($merchant_id) . ':' . $cashU_trnID .
':' . strtolower($encryptionKey));
if($calculatedCashuToken!=$cashUToken)
{
    // If they didn't match, then don't release the service or the product,
    as this request is not from CASHU's side:
}
else
{
    // If they matched, then calculate the token and make sure that it
    matches the token parameter value as below:

    // For Enhanced Encryption:
    $calculatedtoken = md5(strtolower($merchant_id) . ':' . $originaAmount
    . ':' . strtolower($originaCurrency) . ':' . strtolower($session_id) .
    ':' . $encryptionKey);
    // For Default Encryption:
    //$calculatedtoken = md5(strtolower($merchant_id) . ':' . $originaAmount
    . ':' . strtolower($originaCurrency) . ':' . $encryptionKey);

    if($calculatedtoken!=$token)
    {
        // If they didn't match, then don't release the service.
    }
    else
    {
        // If they matched, then release the service.
    }
}

// The below part is the merchant response on the xml notification:

if($responseCode = 'OK')
```

```

{
    $sRequest = "sRequest=<cashUTransaction><merchant id>" . $merchant id .
    "</merchant_id><cashU_trnID>" . $cashU_trnID .
    "</cashU_trnID><cashUToken>" . $cashUToken .
    "</cashUToken><responseCode>" . $responseCode .
    "</responseCode><responseDate>" . date("Y-m-d H:i:s") .
    "</responseDate></cashUTransaction>";
    $ch = curl_init();

    // Change the below URL to - https://www.cashu.com/cgi-
    bin/notification/merchantFeedBack.cgi for Live Area:
    $customeragent = "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:20.0)
    Gecko/20100101 Firefox/20.0";
    curl_setopt($ch, CURLOPT_USERAGENT, $customeragent);
    curl_setopt($ch, CURLOPT_URL, 'https://sandbox.cashu.com/cgi-
    bin/notification/merchantFeedBack.cgi');
    curl_setopt($ch, CURLOPT_VERBOSE, 0);
    curl_setopt($ch, CURLOPT_HEADER, true);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'POST');
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_TIMEOUT, 0);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $sRequest);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Connection: close'));
    $result = curl_exec($ch);
    curl_close($ch);
}
?>

```

- CASHU does not send any notification to the merchant in the case of a failed transaction, but displays the error message to the customer in the Payment Page.



NOTE!

Make sure to use Original Amount and Original Currency in the Token verification step to make sure that no one manipulated the response.

Do not use the amount and currency values that sent to you by CASHU in the verification process.

Handling CASHU Parameters on Sorry URL:

- CASHU offers a way to be able to redirect the customer to the merchant's website in case of transaction failure; it is called **Sorry URL**.
- **Sorry URL** is a mandatory service that allows the merchant to redirect the customer to a predefined page, if the payment transaction failed for any reason (*insufficient funds, restricted country, KYC compliance restriction, or incorrect authentication details*).

- The following is a list of parameters returned to the **Sorry URL** by CASHU:

Parameter Name	Description
errorCode	Details are available in the following table (<i>Error Codes</i>).
txt1	The item's description sent by the merchant.
session_id	The merchant's unique reference to the transaction (<i>if it was sent by the merchant</i>). This parameter will appear in the exported Statement of Account under column named "Merchant Transaction Reference".

- The following describes the error codes returned to the **Sorry URL** and the error messages related to them:

Error Code	Description
2	Inactive Merchant ID.
4	Inactive Payment Account.
6	Insufficient funds.
7	Incorrect Payment Account details.
8	Invalid account.
15	The password of the Payment Account has expired.
17	The transaction has not been completed.
20	The merchant has limited his sales to some countries; and the purchase attempt is coming from a country that is not listed in the merchant's profile.
21	The transaction value is more than the limit. This limitation is applied to Payment Accounts that do not comply with KYC rules.
22	The merchant has limited his sales to only KYC-compliant Payment Accounts; and the purchase attempt is coming from a Payment Account that is NOT KYC-compliant.
23	The transaction has been cancelled by the customer. If the customer clicks on the "Cancel" button.
24	The Payment Account has been locked.
27	The customer is already subscribed to standing order.
32	User profile is incomplete, and the customer needs to upload their identification document inside their Payment Account in order to process this transaction.

33	User profile is incomplete, and the customer needs to upload their identification document inside their Payment Account in order to process this transaction.
34	The Payment Account exceeded the allowed spending limit, due to the incompleteness of customer profile. In order to process this transaction, the customer needs to upload their identification document.

- Below is an example on how to get the posted parameters to the Sorry URL using **PHP** language:

```
<?php
$errorCode = $_POST["errorCode"];
$txt1 = $_POST["txt1"];
$session_id = $_POST["session_id"];
?>
```

**NOTE!**

Once you complete building your **Integration form**, you are able to **handle CASHU Parameters on all your predefined URLs**, and then simply move to the next step, which is **Test Your Integration Form**.

5.4 Performing a Successful Payment Transaction on the Testing Area (Sandbox)

Now, you need to make at least one successful payment transaction on the Testing Area (Sandbox) to be eligible to request a **Live Merchant Account** on CASHU. To do so, CASHU provides you with **Testing Payment Account Credentials**. You can find them under “**CASHU Prepaid Credentials**” on the left column of your Sandbox merchant account.

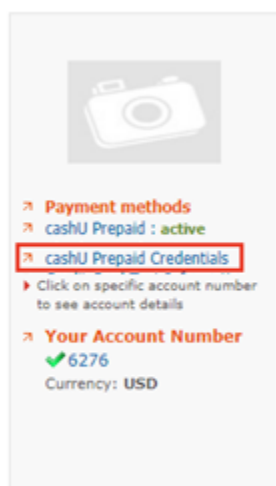
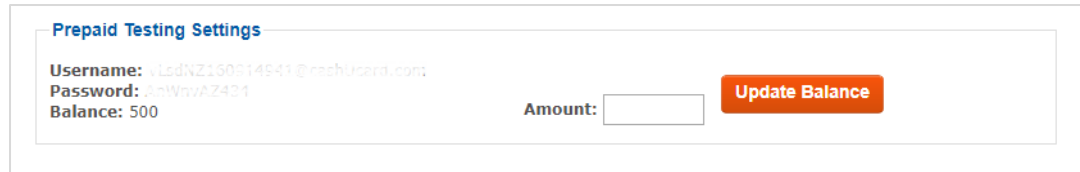


Figure 9: CASHU Prepaid Credentials

After clicking it, the below figure will appear to you:

A screenshot of a web form titled "Prepaid Testing Settings". The form contains three lines of text: "Username: vLsdNZ160814941@cashu.com", "Password: JnWwvAZ421", and "Balance: 500". To the right of the password field is an "Amount:" label followed by a text input box. To the right of the input box is an orange button labeled "Update Balance".

Prepaid Testing Settings

Username: vLsdNZ160814941@cashu.com
Password: JnWwvAZ421
Balance: 500

Amount:

Update Balance

Figure 10: Prepaid Testing Settings

Use the **Username** and **Password** that show in that page to perform your testing transaction(s).

You can update the testing balance by filling the amount on the same page, and clicking "**Update Balance**".

After CASHU verifies the account authentication details, sufficient balance, restricted countries and maximum amount, and upon successful authentication, CASHU will redirect the customer to the **Return URL** that is defined by the merchant.



NOTE!

*If you received an **insufficient balance** error while performing a testing transaction, you can update your testing balance by filling the **Amount** text box with the needed amount and clicking the **Update Balance** button.*



NOTE!

Up to this step, always log in to your Sandbox Merchant Account using the testing area login URL: <https://sandbox.cashu.com/Merchants/en/login>

After you go to the Live Area, which is explained in the next step, this URL will change.

5.5 Going Live

After you are done with at least one successful test payment transaction and you are sure that everything went well, you can now leave the Testing Area (*Sandbox*) and move to the Live Area.

To do so, click on the **“Go to Live”** button that appears at the bottom of the left-side column inside your Merchant Account in the Testing Area (*Sandbox*).

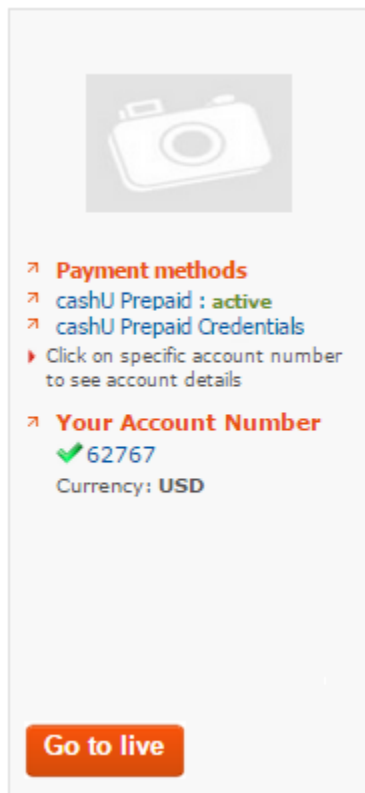


Figure 11: "Go to Live" Button

**NOTE!**

After you move to the Live Area, always log in to your Merchant Account using the Live Area URL: <https://www.cashu.com/Merchants/en/login>

Never use the Testing Area (Sandbox) login URL again.

5.6 Provide the required “Know Your Customer” (KYC) documents and Settlement Account details.

After you click on the “Go to Live” button, you will be redirected to the KYC (*Know Your Customer*) page, where you should enter your information and provide CASHU with the needed identification documents.

The screenshot shows the 'Basic Information' form with the following fields and options:

- Security Question: * (Dropdown menu with 'Your city of birth' selected)
- Security Answer: * (Text input field)
- First Name: * (Text input field)
- Middle Name: (Text input field)
- Last Name: * (Text input field)
- Date of Birth: * (Dropdown menus for Month, Day, and Year)
- Nationality: * (Dropdown menu with '-Select-' selected)
- National ID Number: * (Text input field)
- Address Line 1: * (Text input field, with subtext '(Building No, Street, Area)')
- Address Line 2: (Text input field)
- City: * (Text input field)
- Country: * (Dropdown menu with '-Select-' selected)
- Merchant Website Address(es): * (Text input field with a '+ Add another website' link)
- KYC Document: * (File upload area with 'Choose File' button and 'No file chosen' text)

Below the KYC Document field, there is a yellow box with the following text:

Allowed File(s)(doc, docx, pdf, jpeg, jpg)
Please upload scanned copy of your ID taking into consideration that the document has to be:
1. updated/renewed;
2. clear, colored and captured by a scanner not a mobile camera.

At the bottom of the form, there are two buttons: 'Next' and 'Clear'.

Figure 12: Basic Information

Fill the required **Basic Information** as shown above, and then click on "Next".

**NOTE!**

Please note that you will need to upload your Trade License as a KYC Document before clicking on "Next" and proceeding to the following page.

* This form is required for the Know Your Customer (KYC) validation in compliance with the Financial Services Authority (FSA) requirements. Please complete it to the best of your knowledge before you shift to the Live Mode.

Settlement Account Details

Country: *	<input type="text" value="Jordan"/>
City: *	<input type="text"/>
Bank Account Holder Name: *	<input type="text"/>
Bank Account Number:	<input type="text"/>
IBAN: *	<input type="text"/>
Bank Name: *	<input type="text"/>
Bank Branch: *	<input type="text"/>
SWIFT / BIC-code: *	<input type="text"/>
	<small>Refer to your bank in case you don't know it.</small>
Sort Code <small>(For UK and Ireland only):</small>	<input type="text"/>
	<small>Refer to your bank in case you don't know it.</small>
Routing Number <small>(for US only):</small>	<input type="text"/>
	<small>Refer to your bank in case you don't know it.</small>
Special Instructions:	<div><input type="text"/></div>

Figure 13: Settlement Account Details

After that, fill the **Settlement Account Details** as shown in the above figure, and then click on **"Next"**.

* This form is required for the **Know Your Customer (KYC)** validation in compliance with the Financial Services Authority (FSA) requirements. Please complete it to the best of your knowledge before you shift to the Live Mode.

Contact Information

Account Manager / Business Owner:

Full Name: *

Country Code: *

Phone Number: *

Email: *

Customer Support:

Full Name: *

Country Code: *

Phone Number: *

To be displayed to customers/buyers who need to contact the Merchant's support.

Email: *

* Please note that before you request transferring your earnings from your Merchant account to your bank account, you will be asked to upload a scanned copy of your ID/Passport.

[Register Now](#) [Back](#) [Clear](#)

Figure 14: Contact Information Details

Finally, fill the **Contact Information** details, taking into consideration the below major points, then click on "**Register Now**":

- Full names for both the **Account Manager/Business Owner** and the **Customer Support** should be added.
- Email addresses for both the **Account Manager/Business Owner** and the **Customer Support** cannot be duplicated.

Please let us know once finished to follow up internally on the approval process for your Live Account.

Once your account is approved, an email message will be sent to your email inbox containing the Live Account Credentials.

Please go back and redo all the previous steps on the [Live Area](#).

**NOTE!**

You will not be able to use same values of integration URLs and Encryption Key that were used in Testing Area (Sandbox); you need to use different values.

In case you want to use the same values, you should go back to your Sandbox account and replace the existing configuration to some dummy values.

YOU CAN NOW START ACCEPTING PAYMENTS FROM CASHU USERS!

Appendix 1 – URLs

The below table lists all URLs used in the Integration Form for both areas (*Sandbox & Live*):

URL Name	Testing Area (<i>Sandbox</i>)	Live Area
CASHU Payment Page	https://sandbox.cashu.com/cgi-bin/payment/pcashu.cgi	https://www.cashu.com/cgi-bin/payment/pcashu.cgi
SOAP WSDL	https://sandbox.cashu.com/secure/payment.wsdl	https://secure.cashu.com/payment.wsdl
Merchant Response URL on (Notification URL)	https://sandbox.cashu.com/cgi-bin/notification/merchantFeedBack.cgi	https://www.cashu.com/cgi-bin/notification/merchantFeedBack.cgi

Appendix 2 – (Important) Security Measures

1. **Enhanced Encryption:**

It is recommended that you always use the Enhanced Encryption instead of the Default Encryption, since it allows you to include the **session_id** in the **Token** calculation.

2. **Notification URL:**

You should deliver the service depending on the notification that you will receive on your Notification URL if the transaction was done on the web, instead of delivering it on the Return URL, or on your SDK Notification URL if the transaction was done through the mobile SDK.

3. **Using and Recalculating CASHU Verification Parameters:**

- verificationString / to make sure that CASHU is the source of the “call back”
- cashutoken / to make sure that CASHU is the source of the “offline notification”
- Token / recalculating this parameter is based on the original amount, to make sure that the transaction details were never been manipulated.

4. **IP Restriction or Domain Restriction :**

We also recommend using Domain Restriction on your Notification URLs, to make sure that no one will send requests to your URLs except CASHU.

Helpful Tips:

- No one should know your Security Key “Encryption Keyword”.
- No one should know your merchant area login information.
- No one should know the Return URL nor the Notification URL.
- Each transaction must be linked with an Order ID / Session ID, where each Order ID / Session ID must be unique.
- Make sure you are using a valid SSL certificate on your server.