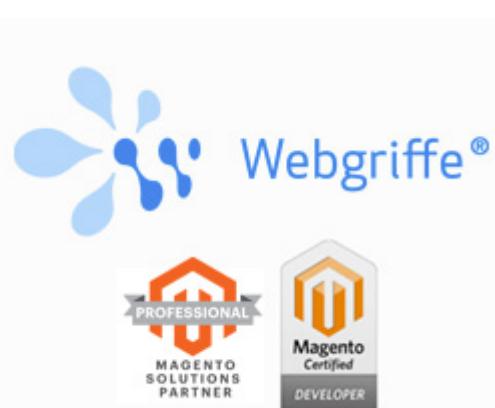


Unicredit PagOnline Imprese payment method for Magento2

This is the documentation for the Unicredit PagOnline Imprese module for Magento 2, which integrates Magento 2 with the aforementioned payment gateway. In addition, the module supports payment via MyBank.



The module provides two different payment methods: one for payment by credit card and one for payment via MyBank. The two payment methods are configured and managed independently. For details regarding the configuration take a look at the chapter **Configuration** further down.

Installation

The installation procedure is different if you purchased the module on Magento Marketplace or on the Webgriffe Store.

Module purchased on Magento Marketplace

If you purchased the module on Magento Marketplace, you have three possibilities:

- **Installation via Magento Component Manager**

If not already done, enter your Magento Marketplace credentials under **System -> Web Setup Wizard -> System Configuration** and follow the instructions.

When you're done you can access into the **Component Manager** at **System > Web Setup Wizard > Component Manager**, you should see the newly purchased extension under the **New Purchases** section, choose then **Install** to install the extension. More information are available at <http://devdocs.magento.com/guides/v2.1/comp-mgr/module-man/compman-main-pg.html>.

- **Manual Installation**

Once the zip file has been downloaded from the Magento marketplace, you can proceed with the same installation procedure as the one used for purchases made on the Webgriffe Store. Please refer to the corresponding section below in this document.

- **Command line installation with Composer**

First of all ensure that the `repo.magento.com` repository has been added to the `composer.json` file and the associated authentication keys have been configured. More information on how to get the authentication keys and how to set them are available at <http://devdocs.magento.com/guides/v2.0/install-gde/prereq/connect-auth.html> and at <https://getcomposer.org/doc/articles/http-basic-authentication.md>.

So the `auth.json` file should be like this:

```
{  
    "http-basic": {  
        "repo.magento.com": {  
            "username": "<MAGENTO_MARKETPLACE_PUBLIC_KEY>",  
            "password": "<MAGENTO_MARKETPLACE_PRIVATE_KEY>"  
        }  
    }  
}
```

Now is it possible to execute the following command

```
composer require webgriffe/module-unicredit-imprese
```

In case of errors such as “*unable to find webgriffe/module-unicredit-imprese*”, please make sure that the authentication keys are correct, that the file is in the correct folder and that, more generally, the file is being used by Composer.

And then run Magento setup upgrade script:

```
$ php bin/magento setup:upgrade
```

Clear the cache:

```
$ php bin/magento cache:clean
```

And recompile the dependency injection data:

```
$ php bin/magento setup:di:compile
```

Module Purchased on Webgriffe Store

Follow these instructions for manual installation using Composer artifacts:

- In the main directory of the store create a folder called *artifacts*, if it doesn't already exist;
- Paste the downloaded archive in this folder;
- Run the command:

```
$ composer config repositories.artifacts artifact artifacts/
```

- Install the module using composer:

```
$ composer require webgriffe/module-unicredit-imprese
```

- Run Magento setup upgrade script:

```
$ php bin/magento setup:upgrade
```

- Clear the cache:

```
$ php bin/magento cache:clean
```

- Recompile the dependency injection data:

```
$ php bin/magento setup:di:compile
```

Configuration

The module configuration can be found in the following admin page: *Stores -> Configuration -> Sales -> Payment Methods*. Here you can find two groups of settings:

- **Unicredit Imprese Payment gateway**, which groups the settings for the credit card payment
- **Unicredit Imprese MyBank Payment gateway**, which groups the settings for the MyBank

Since the configurations shown in the two groups are the same, below you can find only those related to the first group, the ones of the other are similar. These are the available configuration options:

- **Title:** allows one to indicate the name of the payment method that will be shown during the checkout process
- **Enabled:** allows one to enable or disable the payment method.
- **Test Mode Enabled:** enables and disables test mode. This mode is useful to test the payment flow before going live with the module, as when this mode is enabled the module will use the test environment provided by Unicredit. In this environment every transaction is fake and there is no real money transfer involved. When test mode is enabled, the following fields will be used:
 - **Payment Page URL (test):** this is the URL of the test environment of Unicredit PagOnline Imprese. Unless the gateway changes its specs, this field should be set to [https://testeps.netswgroup.it/UNI\(CG_SERVICES/services/PaymentInitGatewayPort?wsdl](https://testeps.netswgroup.it/UNI(CG_SERVICES/services/PaymentInitGatewayPort?wsdl)
 - **Merchant Terminal id (test):** this is the merchant's identifier for the test transaction. Unless the gateway changes its specs, this value should be set to *UNI_ECOM* for credit card payment method, while it should be set to *UNI_MYBK* for MyBank
 - **Merchant Signature Key (test):** this is the secret key used to digitally sign the messages that are sent between Magento and the gateway, thus ensuring that the communication is secure. Unless the gateway changes its specs, this value should be set to *UNI_TESTKEY*

On the other hand, when test mode is disabled the module will use Unicredit's production environment, where transactions involve real money. The production mode can be used only if you have activated a real contract with Unicredit and you have a real **Merchant Terminal Id** and **Merchant Signature Key**. In production mode the following fields will be available:

- **Payment Page URL (production):** this is the URL of the gateway's production environment. Unless the gateway changes its specs, this value should be set to [https://pagamenti.unicredit.it/UNI\(CG_SERVICES/services/PaymentInitGatewayPort?wsdl](https://pagamenti.unicredit.it/UNI(CG_SERVICES/services/PaymentInitGatewayPort?wsdl)
 - **Merchant Terminal Id (production):** this is the code that identifies the merchant. This value is provided by Unicredit when the contract is activated. Please pay attention when inserting this value: respect uppercase and lowercase letters (if present), and pay special attention not to inadvertently add spaces before or after the value
 - **Merchant Signature Key (production):** this is the secret key used to digitally sign every message sent between Magento and the gateway, thus ensuring that the communication is secure. This value is provided by Unicredit. Pay attention to insert this value exactly as indicated by Unicredit, keeping the distinction between uppercase and lowercase letters (if present) and paying especial attention not to inadvertently add spaces before or after the value. This code should be similar to this sample code: *4g4i3d5c0b1d8e3j0j0a4g4i3d5j0j0a*. In case of problems, if the format of the code is significantly different from this example (wildly different length, presence of non-alphanumeric characters etc.) please contact Unicredit to make sure that the code is correct. An error of even a single character will make it impossible to verify the authenticity of the sent and received information, thus preventing the module from working.
- **Payment Currency:** This is the currency that the payment amount will be converted to and that will be shown to the customer after he/she reaches the Unicredit payment page. Currently only Euro and US Dollar are supported.
 - **Payment Page Language:** This is the language that the Unicredit payment page will be shown in. The only values currently supported by Unicredit are Italian and English
 - **Payment Transaction Type:** This field allows one to indicate which action to execute when a customer places an order:
 - **Authorize and Capture:** this value immediately executes the payment transaction, immediately transferring the funds from the customer's credit card to the merchant's account.
 - **Authorize Only:** this action only "authorizes" the fund transfer, but does not actually execute the transfer. In essence the funds are locked on the customer's card, but not transferred. This transfer must be done manually at a later time from the Unicredit backoffice (currently the module does not implement the fund capture when one marks an invoice as "paid" in Magento's backoffice).
 - **Pending Payment expiration days:** In case the customer does not return to the merchant's store after completing the payment, or if it's not possible to verify right away the outcome of the transaction, this field indicates how long the module should keep on trying to verify the payment. For example, if a customer places an order but Unicredit's systems have some problems and cannot provide the authorization confirmation at once, then the module will continue to poll Unicredit once every hour to check the outcome of this transaction up to

the number of days specified in this field. If the number of days is reached without a response being received (be it positive or negative), then the order will remain stuck in the “pending payment” state.

- **Cancel order on payment failure:** If the customer chooses to cancel the payment by clicking on the “cancel” button in the Unicredit payment page or if the Unicredit gateway reports an error to Magento, then if this option is enabled the order will be canceled. In the same situations, if this option is not enabled, then the order will remain in the “pending payment” state. Please note that this options **HAS NO EFFECT FOR ABANDONED PAYMENTS**, that is those payments where the customer clicks the browser’s “back” button or simply closes the browser window. In these instances the order will remain in the “pending payment” state, regardless of the value of this configuration option. This payment module does not react to abandoned payments, so if it becomes necessary to handle these abandoned payments (for example by canceling the matching orders), then it is suggested to use a third party module that provides this functionality to cancel orders that have remained in “pending payment” for some time or to develop a custom logic to do the same.
- **Debug:** this enables debug mode, used to store additional information during the module’s operation. It’s especially useful to enable this mode in case of technical problems to better understand what isn’t working.
- **Payment From Applicable Countries and Payment From Specific Countries:** this is used to specify whether the Unicredit PagOnline Imprese payment method will be available for every country in the world or only from selected countries
- **Minimum Order Total and Maximum Order Total:** used to indicate whether the Unicredit PagOnline Imprese payment method must be available only for orders having a total amount smaller and/or grater than some specified values.
- **Sort Order:** This sets a number that will be used to sort the payment methods in the checkout. This is an ascending order sort, so that a payment method with a lower value in this field will be shown above another method with a higher value.

Test Mode

Test mode is useful to check the payment workflow before deploying the module in the production environment, as the module will then use Unicredit’s test environment. In this environment all transactions are fake and there’s no real transfer of funds. Unlike other payment gateways, Unicredit MagOnline Imprese does not use a server to server request to complete a transaction. This allows one to easily test the module even on a computer that is not reachable from the outside and that does not have a public URL associated with it.

It is possible to test the **credit card** payment method by using these credit cards and make all the test transactions that are needed. Please note that, by design, some of these cards return a negative transaction outcome, which may be useful to verify the system behavior in case of errors. The expiration dates can be set to any future date. Even first and last name can be set to any value:

- **CartaSi** 4539990000000038 CVV2: any (3 digits) -> **positive outcome**
- **Amex** 37520000000003 CVV2: any (4 digits) -> **positive outcome**

- Visa 4020070300280212 CVV2: 301 -> **positive outcome**
- Visa 4172340000123456 CVV2: 784 -> **negative outcome**
- Mastercard 5204740000001002 CVV2: 100 -> **negative outcome**
- Mastercard 5457210002001040 CVV2: 586 -> **positive outcome**
- Maestro 6799990100000000043 CVV2: 004 -> **positive outcome**

Test di MyBank

At the time of writing this documentation the Unicredit test environment for Mybank payments is not accessible outside of the Unicredit internal network. This means that if the module is configured to use Mybank's test mode, it will be possible to get only to the bank selection page. If one tries to go on, no matter what one does, then an error will be received. The only way to complete a test payment with Mybank is to contact the Unicredit's tech support (e-merchant.ucbanca@unicredit.eu) and ask them to manually complete the payment for a test order you created. To do so, the tech support staff will need the payment URL that the customer would be redirected to in order to complete the payment; however the difficulty is that these URLs can be used only once, therefore it is imperative to retrieve such URLs without first opening them in a browser. To do so it's necessary to temporarily modify the module's code to suppress the automatic redirect to the payment gateway: open the file `rc/Controller/Redirect/Index.php` and find the line starting with `return $this->_redirect()`. Then comment that line (add two forward slashes before the `return` keyword so that the line becomes `//return $this->_redirect()`). Then recompile the dependency injection data and clear the cache (see the section about the module installation to get the commands to perform these tasks). The purpose of this operation is to prevent the module from automatically redirecting the customer to the payment page. Also ensure that the debug flag is enabled for Mybank and that Mybank is in test mode. Now it is possible to place a test order. Proceed as usual and click on the "place order" button. Normally you'd see the payment page at this point, but with the modification you just made you should get a white error page. This is normal. At this point we can retrieve the URL that the customer would have been redirected to: go to the `/var/log` directory inside Magento's root directory and open the `debug.log` file. Among the last lines you should find one like so `[unicreditimpresemybank]: Unicreditimprese - Initialize response has returned redirect url ...`. The URL in place of the ellipsis is the one you need and you can send it to the Unicredit's tech support. Make sure you do not open that URL with a browser. Once this URL has been sent you can restore the original content of the `src/Controller/Redirect/Index.php` file, then recompile the dependency injection data and clear the cache.

Usage and payment outcome handling

When a customer completes the checkout and places an order using the Unicredit PagOnline Imprese payment method, he/she will be redirected to the gateway's payment page. At this time Magento sets the order state to *Pending Payment*. Once the customer completes the payment by entering his credit card data, the customer will be redirected to an URL that is provided by this module (`unicreditpagonlineimprese/payment/notify`, provided that the customer does not close the browser before this redirect takes place). When that page is

reached, the module retrieves the last order from the customer's session and performs a request to Unicredit to try and verify the outcome of the transaction. In case Unicredit indicates an error, then the customer is redirected to the error page (checkout/onepage/failure), while in every other case (included the one when the payment is still being processed by Unicredit) the customer is sent to checkout/onepage/success.

If it is not possible to check the state of a payment when the customer comes back to the merchant's website, then the module will keep on trying to verify the state of the payment. This series of attempts will continue until a positive or negative outcome is received, or until the configured time limit is reached. Once the module manages to determine the outcome of a transaction some operations are performed. In case the transaction succeeds, the order automatically goes to the *Processing* state. Then, depending on the value set in the *Payment Transaction Type* configuration field, further operations are executed:

- In case of **Authorize and Capture**, a *Capture* type *Transaction* object is created on the order, and an *Invoice* is automatically generated. This way, for Magento, the order is completely paid.
- In case of **Authorize Only**, on the other hand, an *Authorization* type *Transaction* object is created on the order, which signals Magento that the amount has been authorized. From Magento's point of view, however, the order has not been paid yet, and indeed there is no *Invoice* for this order.

If, on the other hand, the payment verification reports a failure, then the order remains in the state it was before, and a comment is added to the comment history of the order. This comment contains some information about the error that occurred.

Internationalization

The Unicredit PagOnline Imprese module is developed with a native user interface in English (*en_US*). The module also includes the Italian language translation (*it_IT*). It is possible to translate the interface in new languages using the Magento 2 translation system. For further information about this system please refer to the official Magento 2 documentation.