Key Client/CartaSì Payment for Magento2

This is the documentation of the Key Client/CartaSì Payment module for Magento2 which integrates the platform with **Nexi** (formerly QuiPago) payment gateway.

Installation

The installation procedure could be "wizard" or "manual". If the module has been purchased on the Magento Marketplace, both installation procedures are available. Instead, if the module has been purchased on the Webgriffe Store, only the "manual" procedure is available.

Wizard installation (only Magento Marketplace)

If you purchesed the module on Magento Marketplace, you have two 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 puchased 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.

- **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-quipago
```

In case of errors such as "*unable to find webgriffe/module-quipago*", 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
```

Manual installation (Magento Marketplace and Webgriffe Store)

Once you downloaded the module's ZIP file from Magento Marketplace or Webgriffe Store, follow these steps:

- Create an *artifacts* folder inside your Magento's root folder.

```
$ mkdir artifacts
```

- Deny public access to this folder by properly configuring your webserver (for example, if you use Apache, put an *.htaccess* file with the content *Deny from all*).

```
$ echo "Deny from all" > artifacts/.htaccess
```

- Copy the module's ZIP file inside the *artifacts* folder (do not extract the contents of the zip file, just move the whole archive there)
- Add the *artifacts* folder as Composer repository

```
$ composer config repo.artifacts "artifact" "artifacts/"
```

- Add the module to your Magento's dependencies using Composer

```
$ composer require webgriffe/module-quipago
```

If you receive an error saying that the module can't be found, open your composer.json file, locate the *repositories* section and edit the file to make sure that *artifacts* is the first item in that section. Save the file and try again with the *composer require webgriffe/module-quipago* command.

- Launch the Magento's *setup:upgrade* script

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

- Clear the cache:

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

- Rrecompile the dependency injection data:

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

Configuration

General configuration is located in *Stores -> Configuration -> Sales -> Payment Methods -> Nexi/QuiPago Payment Gateway*. Here are the following settings available:

- **Title**: allows to specify the name of the payment method which will be shown to the customer during the checkout
- **Enabled**: allows to enable/disable the payment method
- **Test Mode Enabled**: allows to enable the test mode. This mode is useful to test an end-to-end payment before putting it into production environment. With test mode enable the payment module will interface with Nexi test environment; in this enviroment all transactions are fake and there is no actual funds transfers. When the test mode is enabled will be used the following parameters:

  - **Payment Page URL (test)**: it's the URL of Nexi's payment page in test environment. Unless unexpected changes in the specifications this value shuold be set to *https://int-ecommerce.nexi.it/ecomm/ecomm/DispatcherServlet*
  - **Merchant Alias (test)**: is the merchant's identification code by Nexi to use in the test environment. Read the *Test Mode* section below for more detailed information.
  - **MAC Calculation Secret Key (test)**: it's the secret key for MAC calculation to be used in test environment. Read the *Test Mode* section below for more detailed information.

  Instead, when the test mode is **not** enabled the module will interface with the production environment of Nexi where funds are really transfered. In this case will be used the following parameters:

  - **Payment Page URL (production)**: it's the URL of Nexi's payment page in production environment. Unless unexpected changes in the specifications this value shuold be set to *https://ecommerce.nexi.it/ecomm/ecomm/DispatcherServlet*
  - **Merchant Alias (production)**: is the merchant's identification code by Nexi to use in the production environment. This value is comunicated by Nexi to the merchant upon contract activation.
  - **MAC Calculation Secret Key (production)**: it's the secret key for MAC calculation to be used in test environment. This value is comunicated by Nexi to the merchant upon contract activation.

- **Payment Page Language**: allows to choose in which language the payment page should be shown
- **Payment Action**: allows to choose if payments done with this payment methods should be considered as *only authorized* or *authorized and captured*. This option should be set accordingly with authorization/capture mode of Nexi gateway chosen during contract activation.
- **Encryption Method (for MAC code)**: allows to choose which encryption algorithm should be used for MAC calculation. With new contract activations it should always use the *SHA1* method but some old contracts require the *MD5* method. If in doubt refer to Nexi technical support.
- **Cancel Order on Failure**: allows to enable automatic order cancel in case of failed payment.

- **Debug**: allows to enable the debug mode which logs more informations duiring module operation. It's useful to enable this mode in case of technical issues to better understand what is going wrong.
- **Payment From Applicable Countries** e **Payment From Specific Countries**: allows to specify from which countries this payment method is allowed
- **Minimum Order Total** e **Maximum Order Total**: allows to specify a minimun and/or maximum total for which this payment method is allowed
- **Sort Order**: allows to set a sort order number used to determine in which order the payment method will show in the checkout page

MyBank payments

This extension supports payments placed through MyBank. In order to enable this feature, it's necessary to go to Nexi's backoffice and enable MyBank in the list of available payment methods. If MyBank is enabled, when the customer confirms the order he/she will be redirected to a page hosted by Nexi where it will be possibile to choose which payment method to use among the ones that are available ("regular" credit card, MyBank and, perhaps, others). Once a method has been chosen, the customer will then proceed with the actual payment according to what method he/she had chosen. In Magento the order paid through MyBank will be updated with the payment outcome just like orders paid by credit card (see the "Usage and payment result handling" section). **Currently it's not possible to choose the specific payment method to use (credit card, MyBank or others) from the Magento checkout. The choice happens only after the customer has been redirected to Nexi's page.** Currently Nexi's test environment does not support the simulation of MyBank payments, therefore it's not possible to use test mode to try out MyBank payments. The only way is to place a real payment with the module set to production mode.

Test Mode

Test mode is used to test an end-to-end payment before putting it into production environment. With test mode enabled the payment module will interface with Nexi's test environment; in this enviroment all transactions are fake and there is no actual funds transfers. In order to do this, you have to register a new test account here https://ecommerce.nexi.it/area-test . After you log in, you land on a page that contains a lot of useful information:

- Under the section **Simple Payment** you find the *Alias* and the *MAC key* that you have to insert in our module configuration test fields in Magento (*Alias* and *MAC Calculation Secret Key* respectively);
- In the same section (**Simple Payment**) are specified the data with which you can access the Back Office to check for your transactions and get some reports.
- Under the section **Test cards** you find the cards data with which you can simulate all types of payment results.

Usage and payment result handling

When a customer completes a checkout using the Nexi payment method, it will be redirected on an external payment page. In that moment Magento sets the order state to *Pending Payment*. Then, after that the customer completes the payment filling credit card data, Nexi sends a server-to-server message to communicate the transaction result. The payment module receives this notification and handles, automatically, the order payment. If a customer never completes the payment process, then the order will remain in *Pending Payment* state, as the module will never receive any notification by Nexi and no mechanism is provided to automatically close these orders. **The Cancel Order on Failure setting has no effect on these orders**. In case of an unauthorized transaction (for example because the customer used an invalid credit card), the order is either canceled or it remains in the *Pending Payment* state depending on the *Cancel Order on Failure* setting. Instead, in case of an authorized transaction, the order is set, automatically, to the *Processing* state. In this case, the payment module performs further operations which depends on the *Payment Action* set:

- In case of **Authorize Only**, a *Transaction* is created with type *Authorization* and it is tied to the order. This tells Magento that the amount is authorized. But for Magento the order is not paid yet and in fact there is no *Invoice* tied to the order.
- Instead, in case of **Authorize and Capture**, a *Transaction* with type *Capture* is created and tied to the order. In addition, the *Invoice* is automatically created. In this way, for Magento the order is completely paid.

Internationalization

The Nexi payment module is developed with a native user interface in english language. The module also comes with the italian translation (*it_IT*). It is possibile to translate the interface in other languages using the Magento's native translation system. For further information refers to Magento's official documentation.

Allowed currencies

The Nexi/QuiPago/XPay/CartaSì payment gateway accepts only payment with the **Euro** currency. Accordingly, the module will try to convert the amount of the order payment in Euro, if the base currency is not Euro, by using the currency rates of Magento (you can find the relative section in the admin under Stores->Currency Rates). For this reason, when the base currency is not Euro, there must be a currency rate from the base currency to Euro, otherwise the module will generate an error and not allow the payment. For more information regarding currency rates referes to Magento2 documentation:
http://docs.magento.com/m2/ce/user_guide/stores/currency-overview.html