

# Triveneto payment method for Magento 2

This is the documentation for the Triveneto module for Magento 2, which integrates Magento 2 with the aforementioned payment gateway.

## 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 following steps:

- Create the following directory structure: `app/code/Webgriffe/Triveneto`;
- Unzip the file downloaded from the Magento Marketplace;
- Copy the contents of the zip file in `app/code/Webgriffe/Triveneto`;
- Install the library that the module depends on using composer:

```
$ composer require webgriffe/lib-triveneto
```

- Run Magento setup upgrade script:



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

- o Clear the cache:

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

- o Recompile the dependency injection data:

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

- **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-triveneto
```

In case of errors such as “*unable to find webgriffe/module-triveneto*”, 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

If you purchased the module from Webgriffe Store, follow these instructions for manual installation:

- Unzip the file downloaded from the store;
- Copy the files of the extracted folder to the project root directory;
- Install the required library using composer:

```
$ composer require webgriffe/lib-triveneto
```

- 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 *Stores -> Configuration -> Sales -> Payment Methods -> Triveneto Payment Gateway*. 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 Triveneto. 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:
  - **Triveneto TranPortalID (test):** This identifies the test merchant in Triveneto's systems. This data is provided by Triveneto when the contract is signed.
  - **Triveneto Password (test):** A password provided by Triveneto and used to check that test transactions are authentic.
  - **Init URL (test)** URL that the module sends the initial request to in order to start the payment procedure when test mode is enabled. This value should always be *https://ipg-test4.constriv.com/IPGWeb/servlet/PaymentInitHTTPServlet*

On the other hand, when test mode is not enabled, the module will use Triveneto's production environment where transactions are real. In production mode the following values will be used:

- **Triveneto TranPortalID:** This identifies the merchant in Triveneto's systems. This data is provided by Triveneto when the contract is signed.
  - **Triveneto Password:** A password provided by Triveneto and used to check that transactions are authentic.
  - **Init URL:** URL that the module sends the initial request to in order to start the payment procedure when test mode is disabled. This value should always be *https://ipg.constriv.com/IPGWeb/servlet/PaymentInitHTTPServlet*
- **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.
  - **Action:** This field allows one to indicate which action to execute when a customer places an order:
    - **Authorize:** 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 Triveneto backoffice (currently the module does not implement the fund capture when one marks an invoice as "paid" in Magento's backoffice).
    - **Capture:** this value immediately executes the payment transaction, immediately transferring the funds from the customer's credit card to the merchant's account.

- **Secret key:** To provide greater security, the module uses this field (automatically generated when the module is installed) to “sign” the data that is sent to Triveneto and check that what comes back from the gateway matches what was sent. Normally you should not need to modify this field. In case a database is imported that comes from a different Magento installation where there were some orders paid with Triveneto, then it will be necessary to input in this field the value that was used by the installation that generated the orders in the database that is being imported. This way the module will be able to correctly verify the outcome of payments that were started in the old Magento installation.
- **Payment From Applicable Countries** and **Payment From Specific Countries:** these are used to specify whether the Triveneto 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 Triveneto payment method must be available only for orders having a total amount smaller and/or greater 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 flow before going live, as the module will use Triveneto’s test environment. In this environment all transactions are fake and there is no transfer of real money. The payment flow includes a server to server call by Triveneto to the merchant’s server in order to communicate the transaction outcome. This means that in order to use the Triveneto module (both in test and production mode), the server that the module is installed on must be reachable through a public URL without any kind of authentication. As a minimum the `triveneto/notify/index` URL must be publicly accessible.

It is possible to use these credit cards to make the needed tests in the test environment. Please note that some of these cards return a negative outcome, which may be useful to check the system behavior in case of errors. Expiration dates can be set to any future date. Similarly, the first name, last name, verification code and email address can be set to any value in the test environment:

- 4539990000000012 -> **positive outcome**
- 4539990000000020 -> **transaction not approved**
- 4999000055550000 -> **invalid card number**

In the test environment it’s possible to choose whether to pay by credit card, MyBank or MasterPass. At the time of writing this manual, it seems like only the “simple” credit card payment method works in Triveneto’s test environment. In the production environment, on the other hand, all of these models should be operational.

## Usage and payment outcome handling

When a customer completes the checkout of an order and chooses the Triveneto payment method, he/she is redirected to the Triveneto payment page. At this time Magento sets the order state as *Pending Payment*. After the customer has completed the payment by entering the credit card data, the Triveneto server contacts the notification URL that the module provides in order to communicate the outcome of the transaction. If the transaction is successful, then the order is marked as paid, it passes in the *Processing* state and the customer is redirected to the checkout/onepage/success URL. In case of errors, on the other hand, the order remains in the previous state and the customer is redirected to the checkout/onepage/failure URL.

In case of a positive outcome, additional operations are performed depending on what *Action* is set in the module configuration:

- In case of **Capture** then a *Transaction* is created for the order, and this transaction is of *Capture* type. Also an *Invoice* is automatically generated. This way, for Magento, the order is completely paid.
- In case of **Authorize** on the other hand, a *Transaction* of type *Authorization* is created for the order, indicating that the payment amount was authorized. For Magento, however, the order is not yet paid and indeed there is no *Invoice* for the order.

If, on the other hand, the payment outcome reports an error, then the order remains in whatever state it was before the payment attempt. Both in case of a positive and negative outcome, the data returned by Triveneto is saved in a comment to the Magento order, so that it's easily possible to check the cause of any error.

## Internationalization

The Triveneto payment 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.