

PayPlug

The payment solution that increases your sales

PAYPLUG EXTENSION FOR MAGENTO V1

Table of Contents

| | |
|---|----|
| 1. INTRODUCTION..... | 3 |
| 2. CONFIGURATION | 4 |
| 2.1. CONNECT | 4 |
| 2.2. SETTINGS..... | 5 |
| 2.3. PAYMENT PAGE | 6 |
| 2.4. ONE-CLICK PAYMENT | 7 |
| 2.5. INVOICE GENERATION | 7 |
| 2.6. CUSTOM ORDER STATUSES | 7 |
| 3. PAYMENT PAGE..... | 7 |
| 3.1. REDIRECT | 8 |
| 3.2. EMBEDDED | 9 |
| 3.3. ONE-CLICK PAYMENTS | 11 |
| 3.4. SAVED CARDS MANAGEMENT | 12 |
| 4. MAGENTO BACK OFFICE | 12 |
| 4.1. ORDER - PAYMENT INFORMATION | 12 |
| 4.2. ORDER – REFRESH PAYMENT STATUS | 13 |
| 4.3. ORDER - REFUNDS | 13 |
| 5. TECHNICAL DOCUMENTATION..... | 14 |
| 5.1. PLUGIN INSTALLATION | 14 |
| 5.2. TESTING THE PLUGIN..... | 14 |
| 5.3. PLUGIN WORKFLOW..... | 15 |
| 5.4. PLUGIN COMPATIBILITY..... | 17 |
| 5.5. PLUGIN UNINSTALLATION | 22 |

1. INTRODUCTION

The PayPlug extension for Magento is designed to help merchants get paid by their customers.

You can register, configure the extension and begin accepting card payments immediately. The extension is designed to simplify the administration of payments while integrating essential features that allow you to develop your business.

PayPlug payment solution complies with the VISA and MasterCard requirements and conforms to the very latest standards and technologies within the field of security.

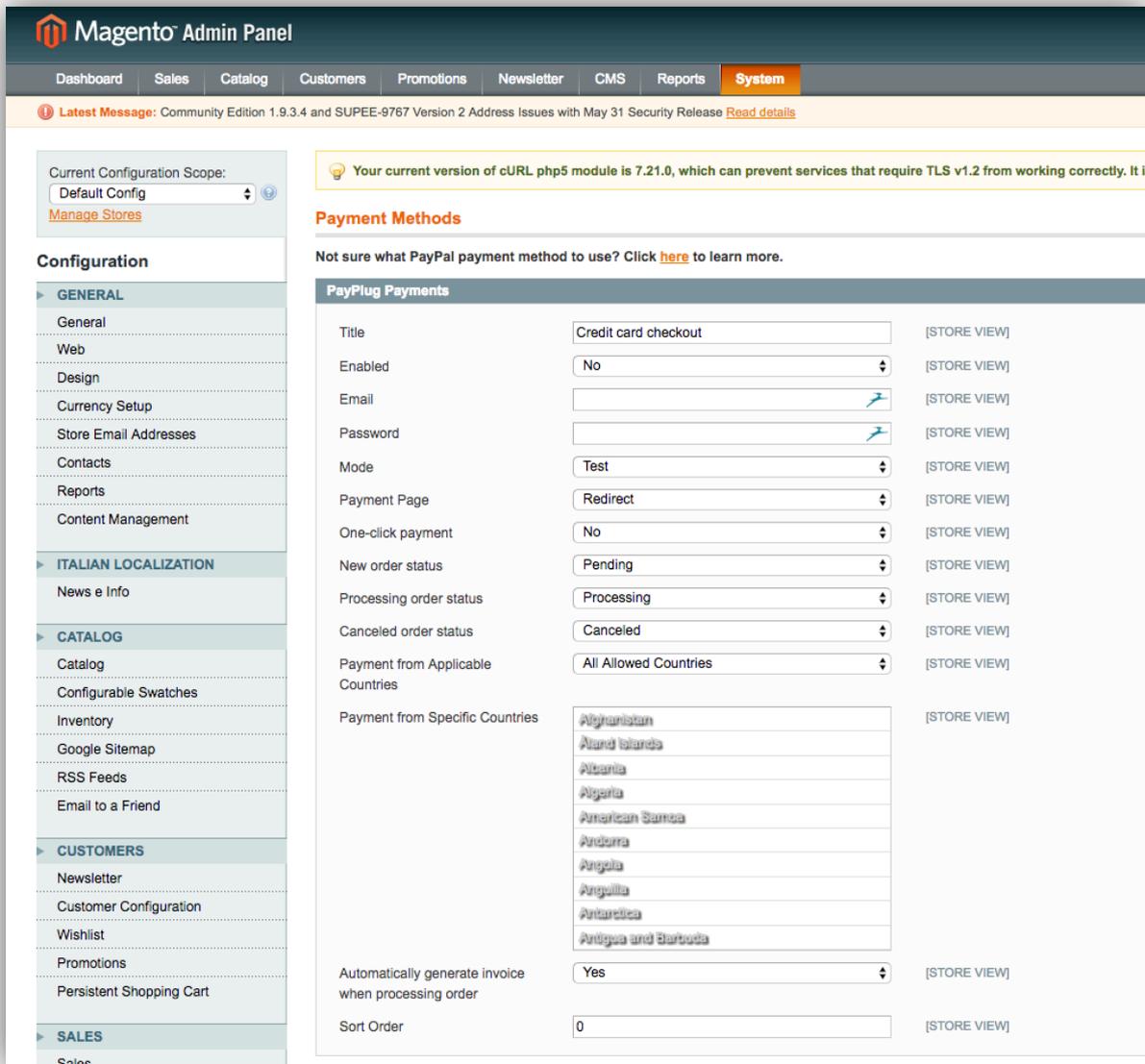
All transactions are carried out on payment forms secured by HTTPS.

Card numbers and the sensitive data are fully encrypted and protected by the TSL protocol. All PayPlug servers are 100% secure and compliant with the PCI-DSS standard, thus relieving merchants of any responsibility for hosting the banking data of their customers.

2. CONFIGURATION

Before accepting payments with PayPlug the you will need to configure the extension. This can be done from the Magento back-office.

You can access the configuration panel in System > Configuration > Sales > Payment Methods > PayPlug Payments.



2.1. CONNECT

| | | |
|----------|----------------------|--------------|
| Email | <input type="text"/> | [STORE VIEW] |
| Password | <input type="text"/> | [STORE VIEW] |

In order to be able to accept payments with PayPlug, you will need to connect your PayPlug account with Magento.

For this, a valid PayPlug account username and password need to be provided.

| | | |
|-----------------|--|--------------|
| PayPlug Account | test@payplug.com ▲ Disconnect | [STORE VIEW] |
|-----------------|--|--------------|

2.2. SETTINGS

Once the PayPlug account is connected, you will be able to configure how the extension will be presented to your customers.

| PayPlug Payments | | |
|--|--|--------------|
| Title | <input type="text" value="Credit card checkout"/> | [STORE VIEW] |
| Enabled | <input type="button" value="No"/> | [STORE VIEW] |
| Mode | <input type="button" value="Test"/> | [STORE VIEW] |
| Payment Page | <input type="button" value="Redirect"/> | [STORE VIEW] |
| One-click payment | <input type="button" value="No"/> | [STORE VIEW] |
| PayPlug Account | test@payplug.com ▲ Disconnect | [STORE VIEW] |
| New order status | <input type="button" value="Pending"/> | [STORE VIEW] |
| Processing order status | <input type="button" value="Processing"/> | [STORE VIEW] |
| Canceled order status | <input type="button" value="Canceled"/> | [STORE VIEW] |
| Payment from Applicable Countries | <input type="button" value="All Allowed Countries"/> | [STORE VIEW] |
| Payment from Specific Countries | <input type="text" value="Afghanistan"/> <input type="text" value="Åland Islands"/> <input type="text" value="Albania"/> <input type="text" value="Algeria"/> <input type="text" value="American Samoa"/> <input type="text" value="Andorra"/> <input type="text" value="Angola"/> <input type="text" value="Anguilla"/> <input type="text" value="Antarctica"/> <input type="text" value="Antigua and Barbuda"/> | [STORE VIEW] |
| Automatically generate invoice when processing order | <input type="button" value="Yes"/> | [STORE VIEW] |
| Sort Order | <input type="text" value="0"/> | [STORE VIEW] |

2.2.1. GENERAL SETTINGS

Title

Label that will be displayed to customers during checkout process

Enabled

Set if PayPlug method can be displayed during checkout process

Please note that the method can only be displayed once the PayPlug account has been connected to Magento.

Please note that PayPlug payment method works exclusively with the Euro currency. You'll need to make sure that Euro is part of the allowed currencies (System > Configuration > Currency Setup > Currency Options > Allowed Currencies).

Countries

If set to "All Allowed Countries", it will use Magento default configuration (System > Configuration > General > Countries Options > Allow Countries).

You can also select specific countries for which PayPlug method will be displayed (based on billing address' country).

Sort Order

Set the order in which the payment methods will be displayed during checkout process (lowest value will be displayed first)

2.2.2.MODE

You can choose between LIVE and TEST modes.

When in TEST mode, the customers will see a test payment page where test payments can be performed. The test payments will not generate real transactions and the card used to perform test payments will not be debited. In the same way, you will not receive any money.

This mode is designed to give you a look and feel of how the extension is used and presented to your customers.

When in LIVE mode the customers will be able to perform real transactions and you will be able to transfer the resulted amount to your bank account (from the PayPlug portal).

In order to be able to perform LIVE transactions, you will need to activate your PayPlug account from the PayPlug portal.

When switching from TEST to LIVE mode, you will be asked to fill in your PayPlug account password again.

2.3. PAYMENT PAGE

You can select how PayPlug is displayed to customers once they chose to pay with PayPlug.

Redirect

The redirect option will redirect your customers to a PayPlug payment page where they can complete the transaction. Once the payment form is validated, customers will be redirected to the order confirmation page.

You can customize the payment page with your logo, your colors and also by adding a background image.

Embedded

The embedded option will open the payment form as an overlay on your website and will display the payment form, having your website in the background. This way, the customers are not redirected away to complete the transaction. Once the payment form is validated, the overlay will close itself and customers will be redirected to the order confirmation page.

2.4. ONE-CLICK PAYMENT

When activated, the option will allow your customers to save their card info for future purchases. This way, the next time they will order something they will only select the saved card without needing to enter the card details.

Please note that in LIVE mode, this option can only be activated by Premium merchants. In TEST mode, it can be activated without restriction.

2.5. INVOICE GENERATION

When activated, once the payment is validated, the order will be set to Processing state and an invoice will automatically be created.

Otherwise, once the payment is validated, a comment will be added to the order to inform you of the payment capture and order state will not be modified.

2.6. CUSTOM ORDER STATUSES

These three parameters allow you to choose which status will be used when PayPlug plugin changes the order state (“Pending Payment” when PayPlug transaction has been created; “Processing” when payment is validated; “Canceled” if payment is rejected or canceled).

By default, Magento default statuses are selected. Please check section Order Flow for more details.

3. PAYMENT PAGE

In order to pay with PayPlug, the customers will need to fill their credit card information in the payment form and confirm the payment.

The way the payment form is displayed is based on the settings configured and can be either using a redirect to a PayPlug secured payment page or using an embedded secured form displayed as an overlay on your website.

3.1. REDIRECT

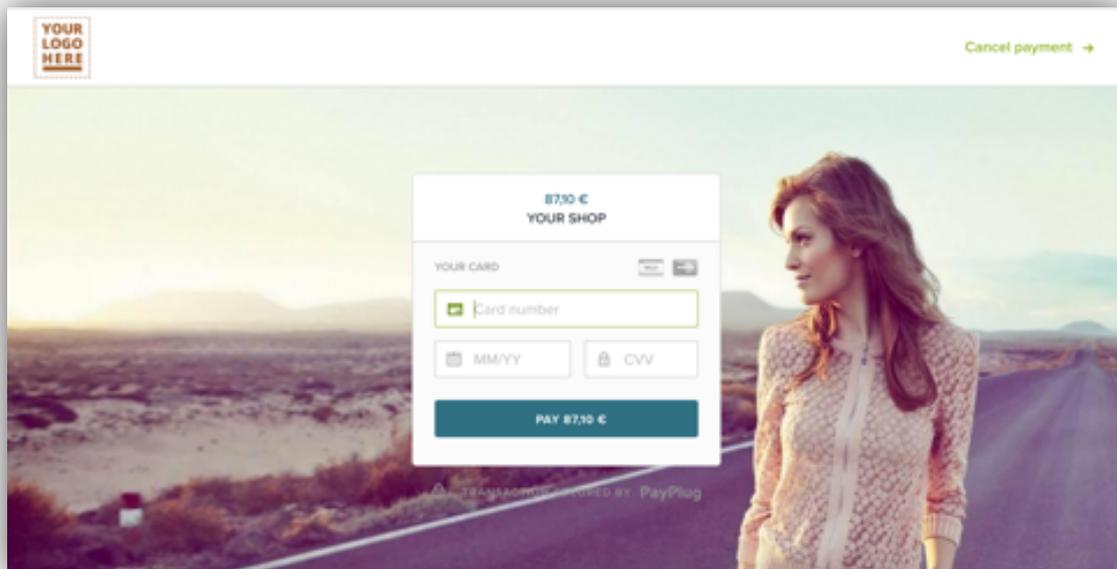
The redirect option will redirect the customers to a PayPlug payment page where they can complete the transaction.

The error messages (if any) are also displayed inside the page.

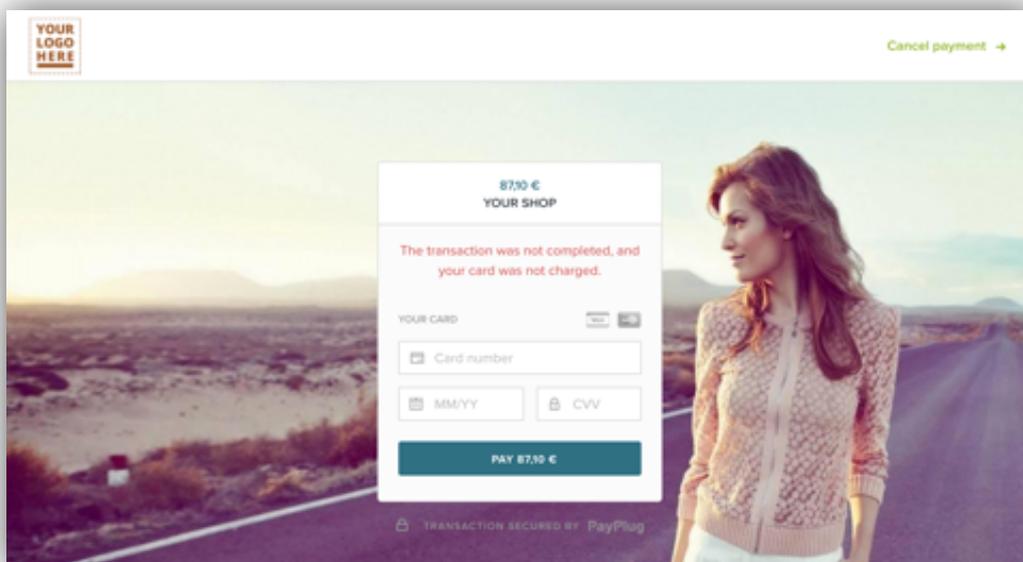
The customers can also cancel the payment in which case they are redirected to the cart page.

Once the transaction is completed, the customers will be redirected to the order confirmation page and the order status is updated.

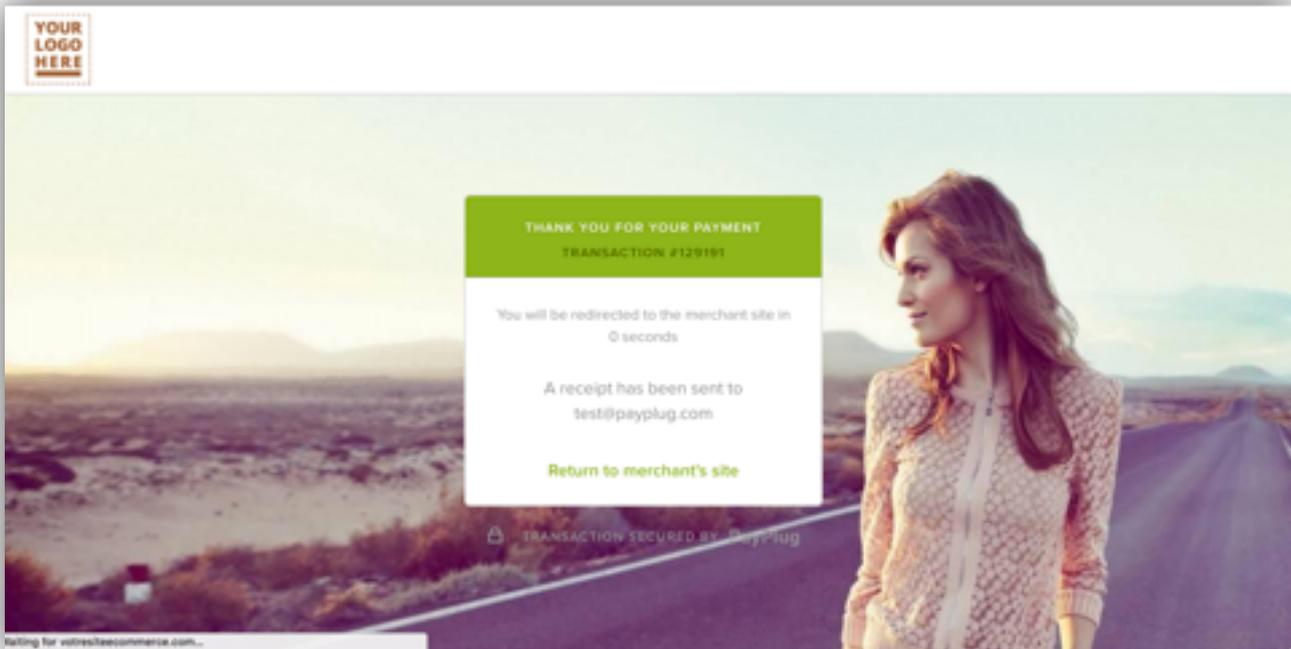
The REDIRECT payment page



Payment errors in the REDIRECT payment page



Successful payment in the REDIRECT payment page



3.2. EMBEDDED

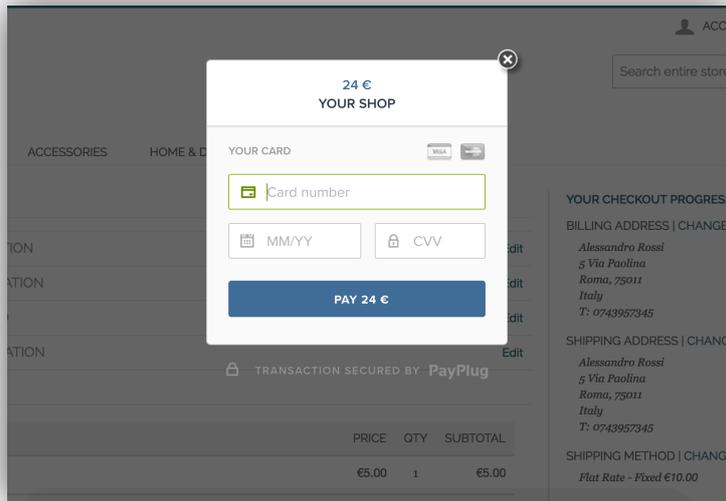
The embedded option will open as an overlay on your website and will display the payment form with your website in the background. This way the customers are not redirected away to complete the transaction.

The error messages are also displayed inside the overlay.

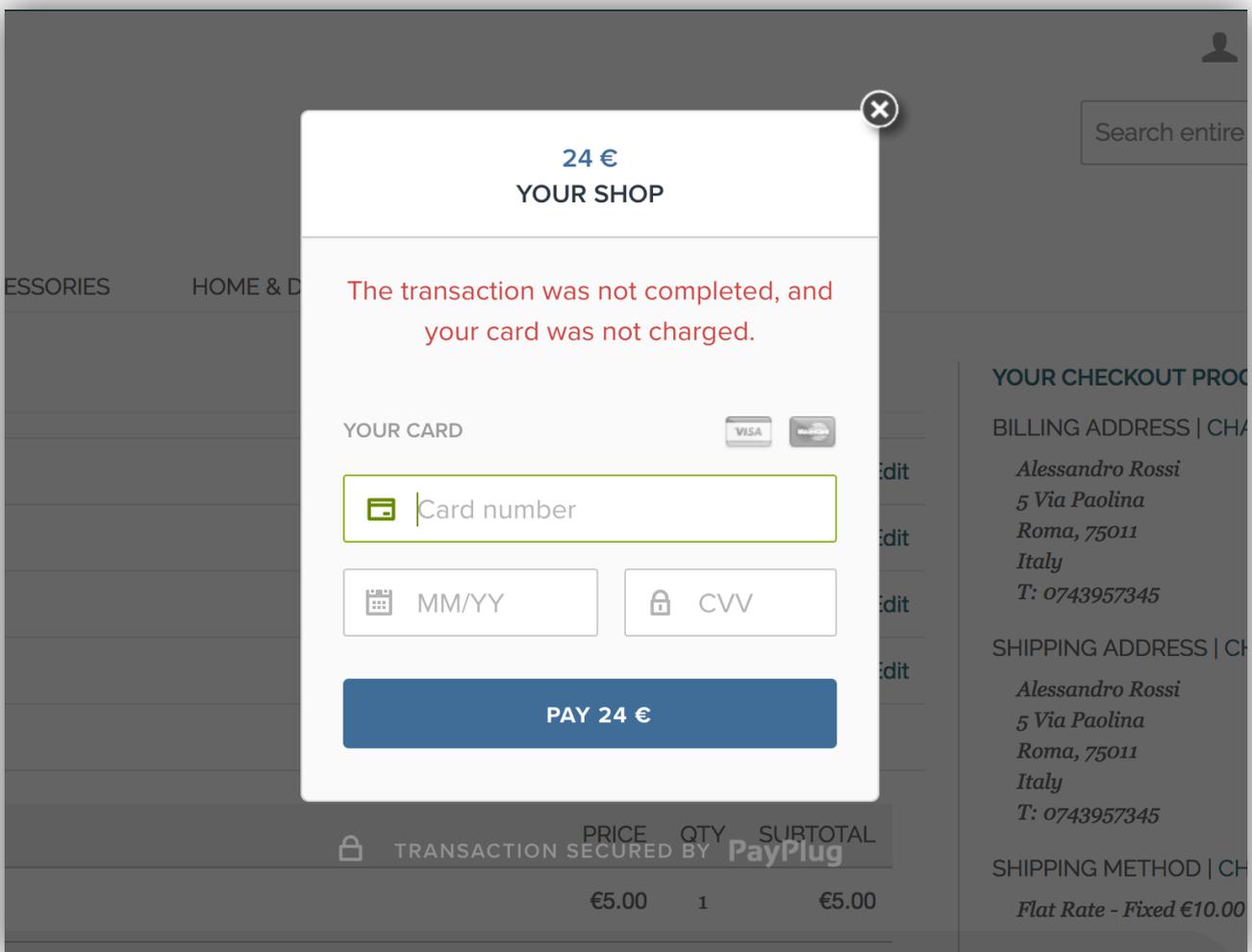
The customer can also cancel the payment in which case the overlay is closed and the customer is then redirected to the cart page.

Once the payment is successful, the customers will be redirected to the order confirmation.

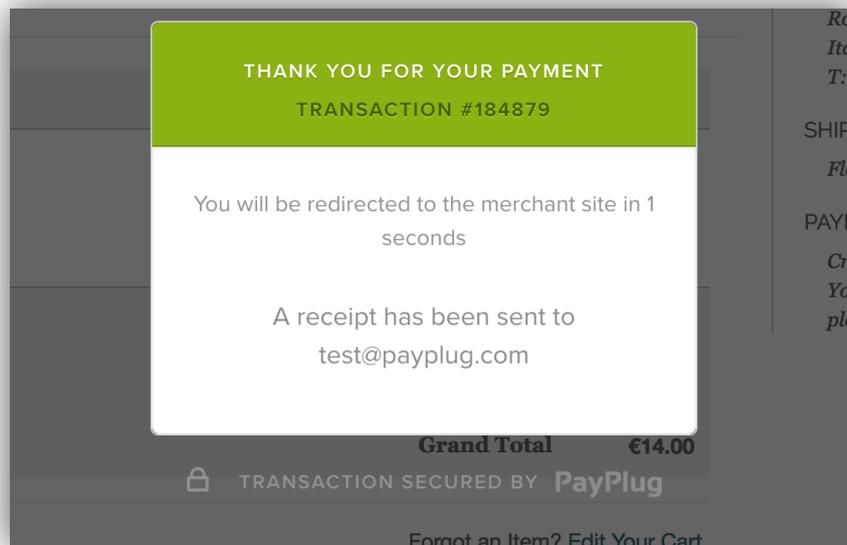
The EMBEDDED payment page



Payment errors in the EMBEDDED payment page

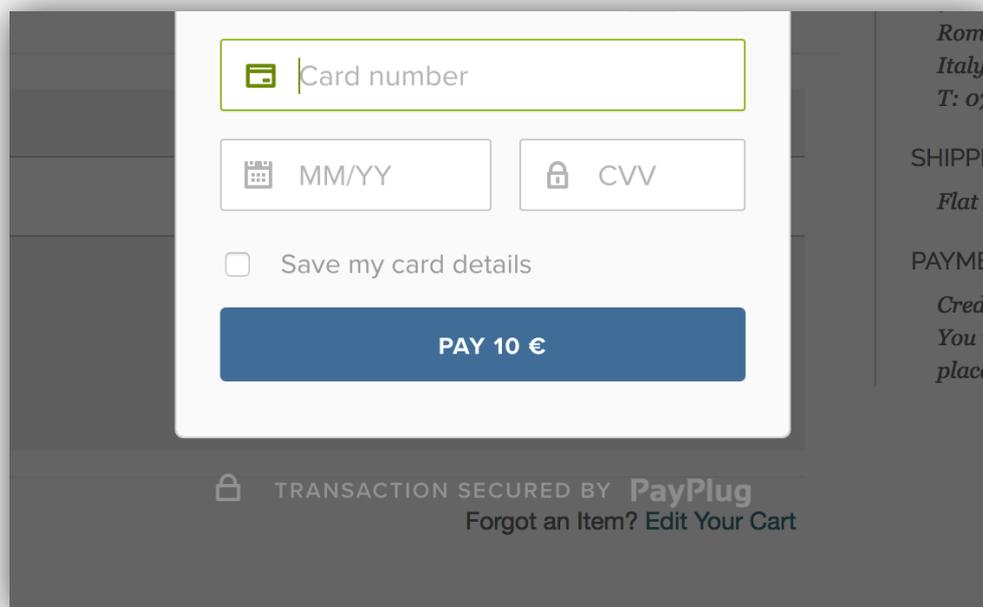


Successful payment in the EMBEDDED payment page



3.3. ONE-CLICK PAYMENTS

If one-click payments option is enabled, the customers will have the option to save their card information for later purchases.



Next time when they shop on your website, when the payment methods are displayed, they will see the saved cards displayed and they will be able to select any of them. If they choose “Pay with another card”, then the selected “payment page” behavior will be used (Redirect or Embedded).

The payment validation or the error messages will be displayed inside the “Credit card checkout” section.

If the payment is successful, the customers will be redirected to the order confirmation page.

CHECKOUT

| | | |
|---|----------------------|----------------------|
| 1 | BILLING INFORMATION | Edit |
| 2 | SHIPPING INFORMATION | Edit |
| 3 | SHIPPING METHOD | Edit |
| 4 | PAYMENT INFORMATION | |

  Credit card checkout

 Your card **** * 4242 (01/99)

Pay with another card

Check / Money order

CONTINUE
[« Back](#)

| | | |
|---|--------------|--|
| 5 | ORDER REVIEW | |
|---|--------------|--|

3.4. SAVED CARDS MANAGEMENT

Your customers will be able to manage their saved cards from My Account > My Saved Cards section. From there they can see the cards previously saved and they can also delete them.

MY ACCOUNT

- ACCOUNT DASHBOARD
- ACCOUNT INFORMATION
- ADDRESS BOOK
- MY ORDERS
- BILLING AGREEMENTS
- RECURRING PROFILES
- MY PRODUCT REVIEWS
- MY WISHLIST
- MY APPLICATIONS
- NEWSLETTER SUBSCRIPTIONS
- MY DOWNLOADABLE PRODUCTS
- MY SAVED CARDS

MY SAVED CARDS

| CARD # | BRAND | CARD MASK | EXPIRATION DATE | DELETE |
|--------|-------|-----------|-----------------|------------------|
| 1 | Visa | 4242 | 10/19 | DELETE THIS CARD |

[« Back](#)

4. MAGENTO BACK OFFICE

4.1. ORDER - PAYMENT INFORMATION

The PayPlug payment ID will be available in the order details page along with various information regarding the payment (status, amount, card mask, date at which the payment was processed, card expiration date, information regarding 3D-Secure)

| Payment Information | |
|---------------------|----------------------------|
| Payplug Payment ID | pay_5cmrwhoGnDB6X3pEyJdgYX |
| Status | Paid |
| Amount | €266.74 |
| Paid at | 06/03/2018 16:31 |
| Credit card | Visa (US) |
| Card mask | **** * 4242 |
| 3-D Secure | No |
| Expiration Date | 01/99 |
| Mode | Test |

Order was placed using EUR

Please note that if a saved card has been used, there will be no details on the selected card in this section.

4.2. ORDER – REFRESH PAYMENT STATUS

Between payment form validation and order confirmation page, a redirection is made to an internal PayPlug page (transparent for the customer).

It allows the plugin to check payment status and process order accordingly.

If the customer closes his page after payment form validation (or if the internal redirection is not possible for some reason), the order will remain in “Pending Payment” state, but the payment will be accepted on PayPlug side.

Order # 145000256 | Mar 6, 2018 9:31:42 AM

Back Edit Send Email Credit Memo Hold Ship Reorder Update Payment

In this case, a button “Update Payment” is available on the order view.

Clicking on it will launch a request to Payplug, to get current payment status.

If validated, order will be processed automatically (if Auto generate invoice is set to Yes).

If payment has been refused or canceled, order will be canceled automatically.

4.3. ORDER - REFUNDS

Online refunds are allowed as long as an invoice has been created (automatically by the module, or manually in the Back-Office) :

- From the order’s invoice page, click on “Credit Memo”.
- Fill in Credit Memo information (product quantities, refund adjustment, ...).
- Click on “Refund”

This will launch a refund request to PayPlug.

If request is accepted, the Credit Memo will be created.

If not, error message(s) will be displayed on the Credit Memo creation page.

5. TECHNICAL DOCUMENTATION

5.1. PLUGIN INSTALLATION

1. Download the plugin package in its latest version
2. Copy the package in your Magento root directory.
3. Uncompress archive
4. Flush the Magento cache

5.2. TESTING THE PLUGIN

You can create an account on our website: <https://portal.payplug.com/signup>

Connect your PayPlug account to Magento (see Configuration section).

Some specific card numbers will allow to successfully create test payments: <http://support.payplug.com/customer/en/portal/articles/1701656-comment-tester-le-service-quest-ce-que-le-mode-test->

Once logged in, you can see all your test payments in your PayPlug account: <https://portal.payplug.com/#/payments>

The payment ID is the same one that is displayed on order view in Magento Back-Office, so you can easily match Magento orders with PayPlug payments.

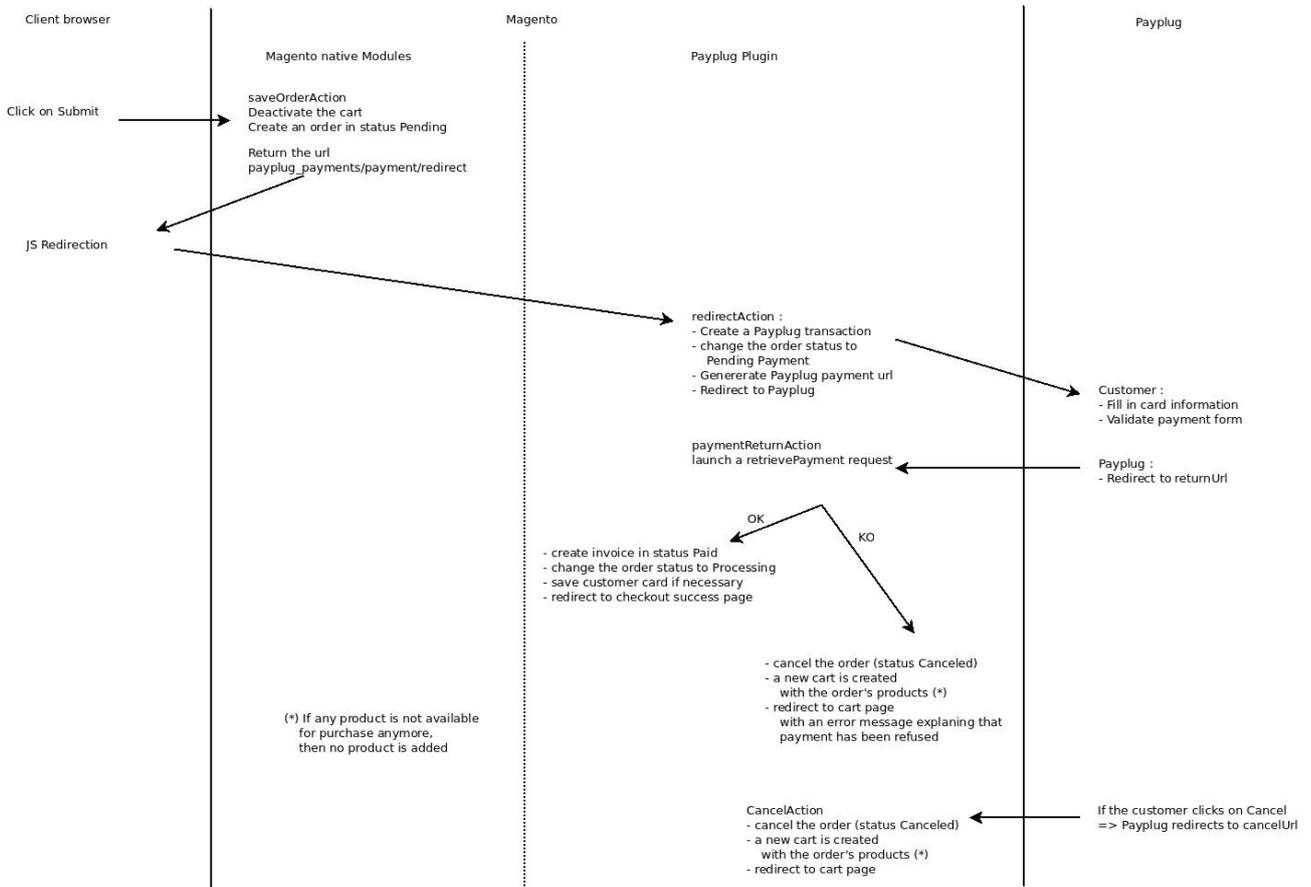
The screenshot shows the PayPlug 'Paielements' dashboard. At the top, there is a toggle for 'TEST' (active) and 'LIVE'. A message states: 'Vous êtes en mode TEST. Tous les paiements, produits et virements sont des simulations et ne généreront pas de transactions réelles. En savoir plus.' Below this, there is a yellow banner for 'Activation du compte' with a button 'ACTIVER MON COMPTE'. The main area contains a search bar, date filters (07-12-2017 and 07-03-2018), a 'Statut' dropdown, and a 'Réinitialiser' button. An 'Exporter' button is also present. The payment list table is as follows:

| MONTANT | CLIENT | DATE | STATUT | ID |
|----------|----------------------|------------------|------------|-----------------------------|
| 430,00 € | XXXXXXXXXXXXXXXXXXXX | 07/03/2018 12:38 | Échoué | pay_Zvt8JaUXrxZDD7nP8DIV |
| 266,74 € | XXXXXXXXXXXXXXXXXXXX | 06/03/2018 17:31 | Payé | pay_5cmrwhoGnDBX3pEyJdgYX |
| 266,74 € | XXXXXXXXXXXXXXXXXXXX | 06/03/2018 17:31 | Payé | pay_5llkeaAhESoxs33NkZmrMU |
| 266,74 € | XXXXXXXXXXXXXXXXXXXX | 06/03/2018 17:30 | Payé | pay_4nG4gdDIFUn9cMlqubNAOf |
| 115,00 € | XXXXXXXXXXXXXXXXXXXX | 06/03/2018 14:38 | Payé | pay_4GkXUMAeufcqQMmnz2sqyVa |
| 115,00 € | XXXXXXXXXXXXXXXXXXXX | 02/03/2018 17:16 | Payé | pay_6pNGg3Fml4Usa7460qGVD0 |
| 115,00 € | XXXXXXXXXXXXXXXXXXXX | 02/03/2018 17:10 | Échoué | pay_6N70juv5jAVHljkC90WVPT |
| 250,00 € | XXXXXXXXXXXXXXXXXXXX | 02/03/2018 17:09 | Payé | pay_Azeox5IWA6PrCNqy45yc3 |
| 215,00 € | XXXXXXXXXXXXXXXXXXXX | 02/03/2018 17:07 | En attente | pay_1SpCBpwwdFKHmXwKsuZ83BC |
| 145,00 € | XXXXXXXXXXXXXXXXXXXX | 02/03/2018 16:48 | Échoué | pay_2NYesLlGYV1AjvHiPpkzNW |

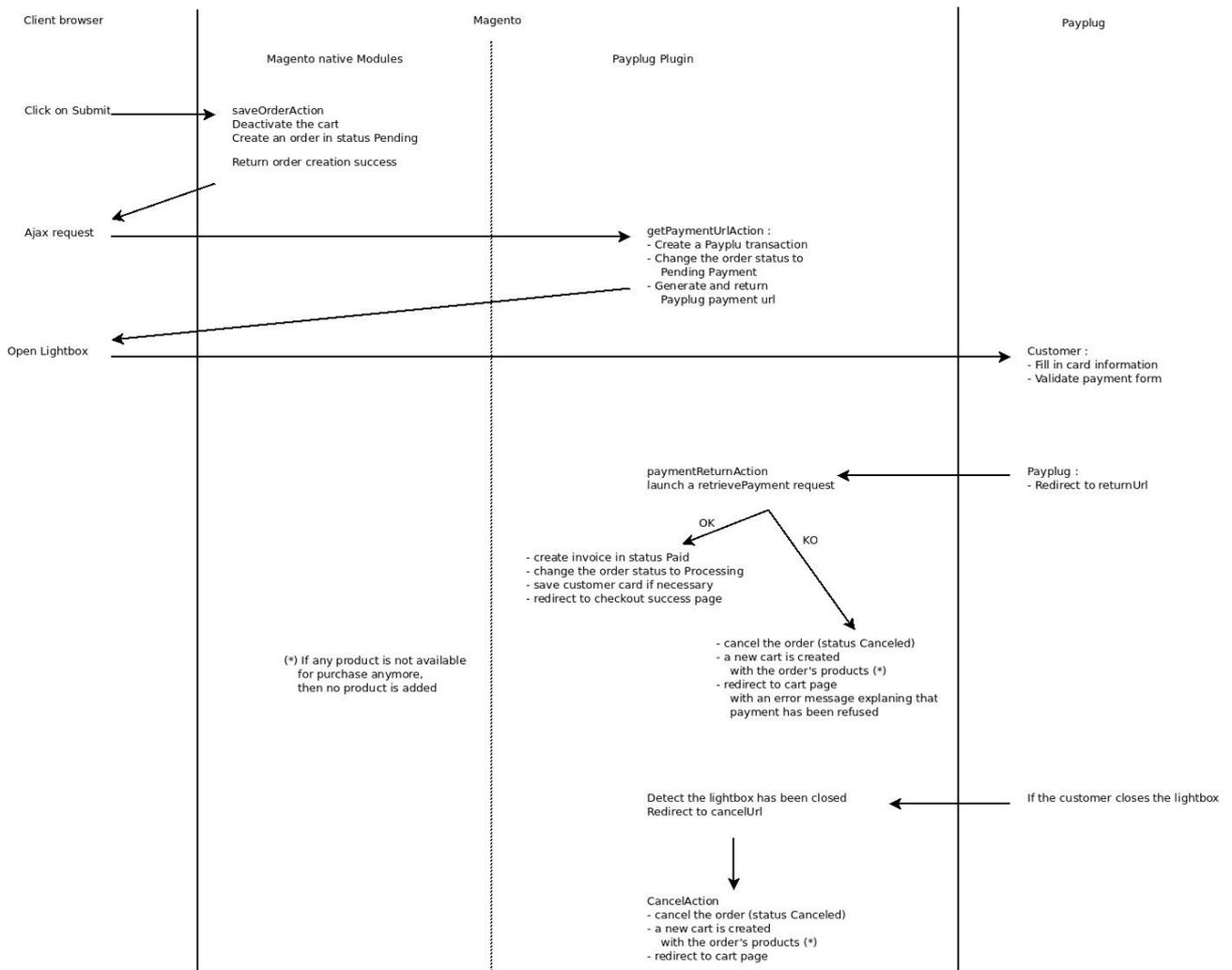
At the bottom right, it shows '1:10 SUR 221 PAIEMENTS' and navigation arrows.

5.3. PLUGIN WORKFLOW

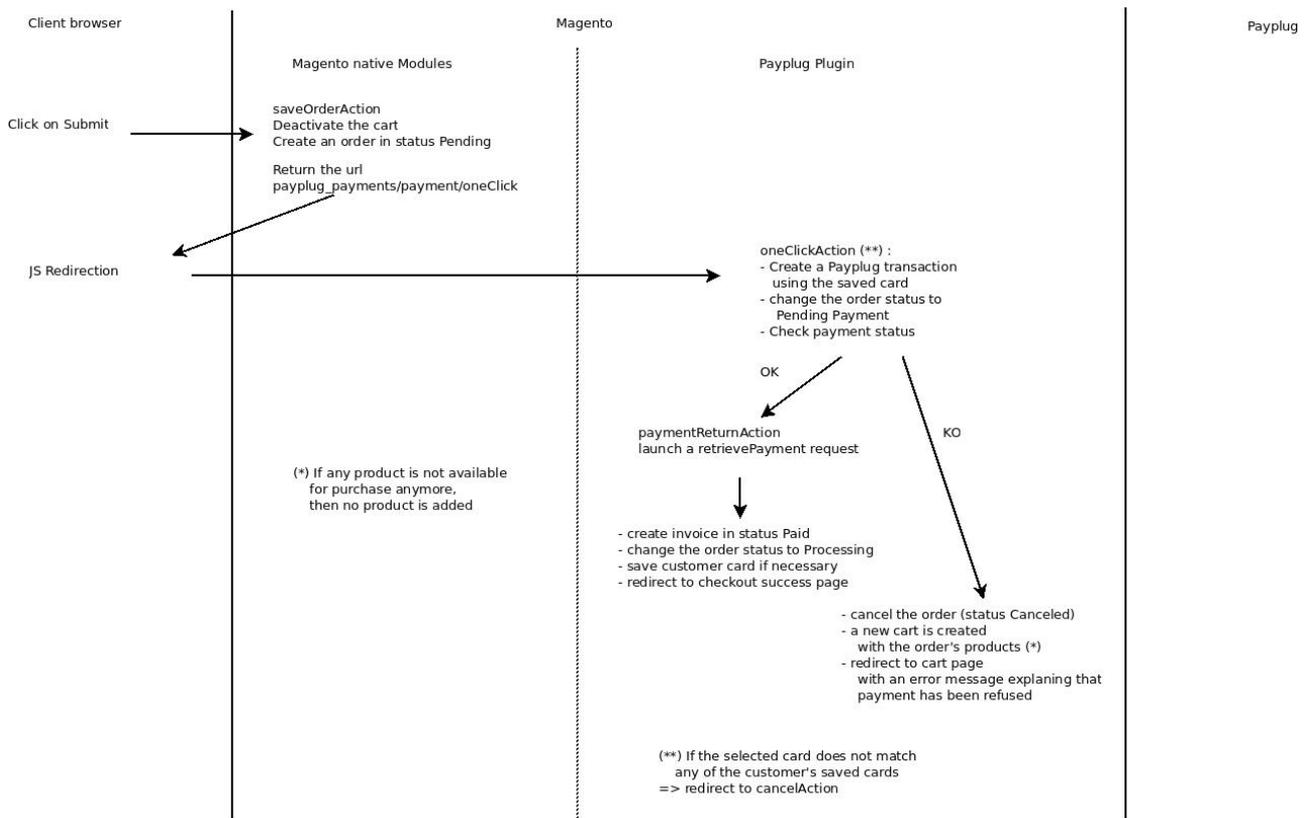
5.3.1. Redirect payment page



5.3.2. Embedded payment page



5.3.3. One-Click payment



5.3.4. Payment status update

Three methods are available to update payments:

- paymentReturnAction : automatically called by Payplug after payment form submission
- Instant Payment Notification (IPN) : triggered by Payplug after payment form submission. Please note that to benefit from the IPN, your server will have to be reachable by Payplug. As a result, IPN calls will fail on most development environments.
- button Update Payment on order view in Magento Back-Office : will launch a request to Payplug to get the current payment status

If the payment is canceled by the customer or refused by PayPlug, the order will be canceled.

If the payment is validated, the order status will be set to Processing and an invoice will be created (if Auto generate invoice is set to Yes in the configuration).

5.4. PLUGIN COMPATIBILITY

When the customer clicks on the checkout final Submit button, if all data is validated, an order will be created by Magento.

If the selected payment requires a redirection, the customer will then be redirected to the relevant page.

However, in case you need some specific JavaScript to be executed (our case for the Embedded payment page), it is not possible out of the box and you will need to tweak a bit the original behavior.

All our JavaScript customizations are placed in file `payplug_payments/checkout/onepage/review_js.phtml` and will be detailed in section “Other plugin compatibility”

5.4.1. Native checkout compatibility

Our plugin is compatible with Magento OnePage checkout.

To implement Embedded functionality, native JS `Review.nextStep` function has been overridden:

```
if (typeof Review != 'undefined') {
    var parentMethod = Review.prototype.nextStep;
    Review.prototype.nextStep = function (transport) {
        if (typeof payplugPaymentType == 'undefined') {
            parentMethod(transport);
            return;
        }
        if (transport) {
            var response = transport.responseJSON || transport.responseText.evalJSON(true) || {};

            if (response.redirect) {
                parentMethod(transport);
                return;
            }
            if (!response.success) {
                parentMethod(transport);
                return;
            }
        }

        if (payplugPaymentType == 'embedded') {
            processPayplugLightboxPaymentUrl();
        }
    }
}
```

As you can see, in most cases, the original function is called (the full function has not been rewritten):

- if our custom variable `payplugPaymentType` is not defined
- if the selected payment method requires a redirection to a specific page
- if an error occurred while saving the order

5.4.2. AW Onestepcheckout compatibility

Our plugin is compatible with AW Onestepcheckout plugin.

To implement Embedded functionality, the native JS `AWOnestepcheckoutForm.onComplete` function has been overridden:

```

if (typeof AW0nestepcheckoutForm != 'undefined') {
  var parentMethod = AW0nestepcheckoutForm.prototype.onComplete;
  AW0nestepcheckoutForm.prototype.onComplete = function(transport) {
    checkPaymentMethod();
    if (currentMethod != 'payplug' || !Lightbox) {
      parentMethod(transport);
      return;
    }
    if (transport && transport.responseText) {
      try{
        response = eval('(' + transport.responseText + ')');
      } catch (e) {
        response = {};
      }
      if (response.redirect) {
        parentMethod(transport);
        return;
      }
      if (!response.success) {
        parentMethod(transport);
        return;
      }
      processPayplugLightboxPaymentUrl();
    }
  }
}

```

We use the function `checkPaymentMethod()` to verify what was the selected payment method.

In case the customer chose PayPlug, with Embedded mode, we call the function `processPayplugLightboxPaymentUrl()` which will trigger the Lightbox opening.

5.4.3. Idev OneStepCheckout compatibility

Our plugin is compatible with Idev OneStepCheckout plugin.

The original behavior of this plugin is a regular POST of the whole checkout page.

After processing the data and creating the order, the plugin checks if the selected payment method requires a specific redirection. If not, the plugin will automatically redirect the customer to the checkout success page.

To prevent redirecting to checkout success page, we had to override the original behavior for the Embedded functionality:

```

if ($('#onestepcheckout-form')) {
  $('#onestepcheckout-form').submit = function () {
    checkPaymentMethod();
    if (currentMethod != 'payplug' || !Lightbox) {
      HTMLFormElement.prototype.submit.apply(this, arguments);
      return;
    }

    new Ajax.Request(this.action, {
      parameters: this.serialize(),
      method: 'post',
      asynchronous: true,
      onSuccess: function (transport) {
        processCheckoutFormResponse(transport);
      },
      onFailure: function (transport) {
        processCheckoutFormResponse(transport);
      }
    });
  };
}

function processCheckoutFormResponse(transport) {
  if (transport.transport && transport.transport.responseText) {
    var respUrl = transport.transport.responseText;
    if (respUrl.search("forceLightbox") != -1) {
      processPayplugLightboxPaymentUrl();
    } else if (respUrl.search("onestepcheckout") != -1) {
      location.reload();
    } else {
      cancelPayplugPayment();
    }
  } else {
    location.reload();
  }
}

```

If selected payment method is PayPlug, with Lightbox mode, then the checkout form will be posted via an Ajax request.

We also had to adapt the method that provides the payment method redirect url `PayPlug_Payments_Model_PaymentMethod::getOrderPlaceRedirectUrl`, which should be empty in case of the Lightbox.

For Idev, we changed it, to return a given value that we could catch in our JavaScript function.

```
// app/code/community/PayPlug/Payments/Model/PaymentMethod.php

/**
 * To provide compatibility with Idev_OneStepCheckout
 * return fake url to be caught by JS and open lightbox
 *
 * @return bool
 */
protected function _forceLightbox()
{
    return Mage::helper( 'core' )->isModuleEnabled( moduleName: 'Idev_OneStepCheckout' ) &&
        Mage::getStoreConfig( path: 'onestepcheckout/general/rewrite_checkout_links' );
}

/**
 * Get URL for "Place Order" button according to configuration
 *
 * @return String
 */
public function getOrderPlaceRedirectUrl()
{
    $customerCardId = Mage::helper( 'checkout' )->getQuote()->getPayment()->getData( 'customer_card_id' );
    $quoteId = Mage::helper( 'checkout' )->getQuote()->getData( 'entity_id' );
    if ( !empty( $customerCardId ) && $this->isOneClick() ) {
        return Mage::getUrl(
            route: 'payplug_payments/payment/oneClick/',
            array( '_secure' => true, 'customer_card_id' => $customerCardId, 'quote_id' => $quoteId )
        );
    } elseif ( $this->isEmbedded() ) {
        $url = '';
        if ( $this->_forceLightbox() ) {
            $url = 'forceLightbox';
        }
        return $url;
    }

    return Mage::getUrl( route: 'payplug_payments/payment/redirect', array( '_secure' => true ) );
}
```

5.4.4. Other plugins compatibility

If your module follows the native behavior and does not bring any modification to the Review step, then no code update should be required to be compatible with PayPlug plugin.

Otherwise, you will have to add some code modifications.

Redirect or One-Click payments do not require any code modifications.

If you want to use the Embebbed payment, here are our recommendations:

Layout update

```
<your_custom_handle>
<reference name="head">
<block type="payplug_payments/checkout_onepage_formjs" name="payplug_payments.checkout.onepage.formjs" />
</reference>
<reference name="before_body_end">
<block type="core/template" name="payplug_payments.checkout.review.js" template="payplug_payments/checkout/onepage/review_js.phtml" />
</reference>
</your_custom_handle>
```

Two blocks have to be included on your checkout page:

- formjs : load the PayPlug JS library
- review_js : specific functions to handle the lightbox mode

Add custom JS function

In order to open the lightbox, you will have to execute the function `processPayplugLightboxPaymentUrl()`.

Find out in your module what is the last JavaScript function that will trigger order creation and then redirect to the relevant page.

Override this function.

To check if you are in the Embedded case, you have two options, depending on the module layout:

- as for native Onepage checkout, there are several steps in your checkout. You can include Lightbox block

```
<checkout_onepage_review>
  <reference name="checkout_onepage_review.info.items.after">
    <block type="payplug_payments/checkout_onepage_review_lightbox" name="payplug_payments.checkout_onepage_review.lightbox" />
  </reference>
</checkout_onepage_review>
```

The block will declare a custom variable and you can then check the value in your own JavaScript.

```
function customFunction() {
  if (typeof payplugPaymentType == 'undefined') {
    // Call parent function
    return;
  }
  if (payplugPaymentType == 'embedded') {
    processPayplugLightboxPaymentUrl();
  }
}
```

- as for AW Onestepcheckout or Idev OneStepCheckout, the checkout is in a single page. You can check what was the selected payment method with the function `checkPaymentMethod()` :

```
function customFunction() {
  checkPaymentMethod();
  if (currentMethod != 'payplug' || !lightbox) {
    // Call parent function
    return;
  }
  processPayplugLightboxPaymentUrl();
}
```

If you do not wish to use our template `review_js.phtml` but use your own custom one, here are the JS functions that you will need:

`getPayplugPaymentUrl()` : ajax call that will trigger the initialization of a PayPlug transaction for the order and that will return the lightbox url. If an error occurred, the customer will be redirected to the relevant page.

```
function getPayplugPaymentUrl()
{
    var paymentUrlController = '<?php echo Mage::getUrl( route: 'payplug_payments/payment/getpaymenturl', array('_secure'=>true)); ?>';
    var url = false;
    new Ajax.Request(paymentUrlController, {
        parameters: {},
        method: 'get',
        asynchronous: false,
        onSuccess: function (transport) {
            var response = transport.responseText.evalJSON();
            if (response.error === true) {
                alert(response.message);
                location.href = encodeURI(response.url);
            } else {
                url = response.url;
            }
        }
    });
    return url;
}
```

processPayplugLightboxPaymentUrl() : open the lightbox with the corresponding url.

```
function processPayplugLightboxPaymentUrl()
{
    var paymentUrl = getPayplugPaymentUrl();
    if (paymentUrl !== false) {
        Payplug.showPayment(paymentUrl);
    } else {
        cancelPayplugPayment();
    }
}
```

cancelPayplugPayment : will redirect to cancelAction, which will : cancel the order, rebuild the customer's cart and redirect him to the cart page.

```
function cancelPayplugPayment()
{
    location.href = '<?php echo Mage::getUrl( route: 'payplug_payments/payment/cancel', array('_secure' => true)); ?>';
}
```

To detect when the lightbox is closed (in order to call cancelPayplugPayment())

```
if (typeof Payplug != "undefined") {
    Payplug._closeIframe = function (callback) {
        var node = document.getElementById("payplug-spinner");
        if (node) {
            node.style.display = "none";
            node.parentNode.removeChild(node);
        }
        node = document.getElementById("wrapper-payplug-iframe");
        if (node) {
            this._fadeOut(node, function () {
                if (callback) {
                    callback();
                }
            });
        }
        // Hard Remove iframe
        node.parentNode.removeChild(node);
        node = document.getElementById("iframe-payplug-close");
        if (node && node.parentNode) {
            node.parentNode.removeChild(node);
        }
        cancelPayplugPayment();
    }
}
```

5.5. PLUGIN UNINSTALLATION

As any other payment methods, it not advised to uninstall payment plugins.

Once orders are created using a given payment method, Magento will need the payment method module code to display order information and will crash when not found.

As a result, if you ever wish to stop using our plugin, we recommend that you just set the Enabled parameter to No, so that PayPlug method is not displayed anymore in the checkout process.