



**Payment page design change
eCommerceConnect Gateway**

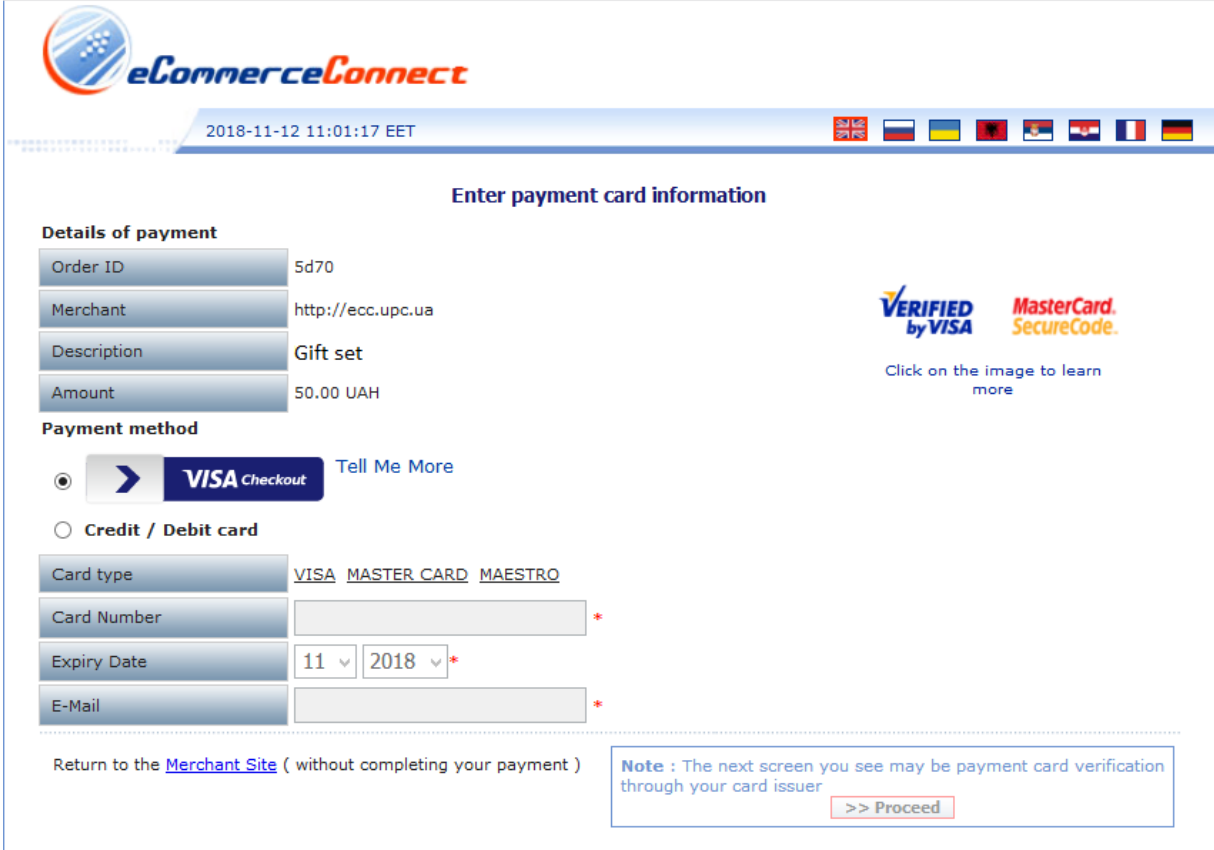
Table of contents

1. Introduction
2. Creation payment page layout
3. Localization
4. Step-by-step algorithm for creation of own payment page
5. Page design after payment completion
6. Sample page

1. Introduction

The instruction describes eCommerceConnect Gateway payment page as well as the step-by-step procedure for its design change. The document describes structure of the payment page, mandatory elements as well as the content available for design change.

The Ukrainian Processing Center (hereinafter UPC) has the protected payment page of a standard pattern that is active for all service users.



eCommerceConnect

2018-11-12 11:01:17 EET

Enter payment card information

Details of payment

Order ID	5d70
Merchant	http://ecc.upc.ua
Description	Gift set
Amount	50.00 UAH

Payment method

VISA Checkout [Tell Me More](#)

Credit / Debit card

Card type: [VISA](#) [MASTER CARD](#) [MAESTRO](#)

Card Number: *

Expiry Date: 11 / 2018 *

E-Mail: *

Return to the [Merchant Site](#) (without completing your payment)

Note : The next screen you see may be payment card verification through your card issuer

[>> Proceed](#)

VERIFIED by VISA **MasterCard. SecureCode.**

Click on the image to learn more

Each merchant can develop his own payment page stylized for his Internet website, to achieve deeper integration of payment gateway. Customized payment page allows to transfer individual design of website to the payment page, which will permit users to perceive the payment page as a part of merchant's website.

2. Creation of own payment page

During implementation of the design and page layout, it is necessary to consider some limitations and rules for correct service operation:

1. Fields with the customer data shall be with approved names, the same as on standard payment page


Card type	VISA MASTER CARD MAESTRO
Card Number	<input type="text"/> *
Expiry Date	<input type="text" value="05"/> <input type="text" value="2018"/> *
E-Mail	<input type="text"/> *

name="CardNumber"

name="ExpMonth"

name="ExpYear"

name="Email" (optionally)

Card verification code	<input type="text"/>	
------------------------	----------------------	---

name="Cvc"

2. Data entered by the customer shall be validated – card shall have a valid number, expiry date shall exceed or be equal to the current month, security code shall consist of 3 digits, e-mail line shall correspond to common-used format.

3. Payment form shall contain information about 3DSecure (Verified By Visa & MasterCard Secure Code). Explanation as to CVV input is also mandatory.

4. Fields shall be transferred via HTTP POST to a new address

<https://ecg.test.upc.ua/go/pay>

5. Html layout with the field to enter card details shall have hidden mandatory field `<input type="hidden" name="action" value="process" />`

6. Design of the payment page shall be adoptive and reflected correctly on all possible devices.

7. The page shall comply with CSP (Content Security Policy) requirements. Inline scripts, styles are not allowed.

The page design has no other restrictions. The merchant is able to develop a unique design without using any patterns or case frames. To activate own payment page, it's necessary to

send html layout and necessary accompanying files, such as css-tables, images, js scripts to e-mail **ec@upc.ua**.

The following special standard addresses are used for references to image files and scripts in the payment page layout:

1. Image

```

```

2. Scripts

```
<script type="text/javascript" src="js/${payment.templateName}/script.js"></script>
```

2. CSS-table

```
<link rel="stylesheet" type="text/css" href="css/${payment.templateName}/style.css">
```

8. Prohibited to use:

1. Inline scripts (js code in the body of the document) - should be connected via file
2. File names should not contain dots and comas (dot before file extension is allowed). Symbols "_" and "-" are allowed to use in names.
3. Path to the file in the page markup should not contain attachments, except for the mentioned above: "img", "css", "js"

4 Allowed file sizes:

- a. Js- upto 500 kb
- b. Css- upto 200 kb
- c. Img – upto 75 kb

3. Localization of the payment page

Procedure of the payment page localization may be selected by the merchant at his own discretion and implemented using any standard or specialized libraries. All necessary files for localization shall be transferred to e-mail address of the electronic commerce department to be loaded to the payment gateway (**ec@upc.ua**).

Except for standard fields for entering card details, the gateway gives list of additional information fields, which may be used for transaction details by payment – transaction amount, payment description, name of point of sale, order number.

These data are accessed via key name:

1. `${amount} ${currency}` – amount of transaction

2. \${payment.orderId} – order number
3. \${name} – terminal name
4. \${payment.description} – payment description

There is also a possibility to use a standard localization procedure for the merchant.

Locale shift mechanism - HTTP GET shall be called for gateway address with the parameters
action=refresh&locale={locale name}

List of available locales: en, ru, uk, sq, rs, hr, fr, de.

<https://ecg.test.upc.ua/go/pay?action=refresh&locale={ locale name }>

List of localized messages:

\${msg[lang].TITLE_PAYMENT} – Enter data of payment card
 \${msg[lang].PAYMENT_DETAILS} - Payment data
 \${msg[lang].PAYMENT_ORDER} - order ID
 \${msg[lang].PAYMENT_MERCHANT} - Merchant
 \${msg[lang].PAYMENT_DESCRIPTION} - Description
 \${msg[lang].TRAN_AMOUNT} - Amount
 \${msg[lang].PAYMENT_SYSTEM_GET_MORE} Click on the image to learn more

 \${msg[lang].TRAN_CODE} - **Transaction code**
 \${msg[lang].TRAN_APPROVAL_CODE} - **Approval code**
 \${msg[lang].TRAN_AMOUNT} - **Amount**
 \${msg[lang].PAYMENT_TOTAL_AMOUNT} - **Amount**
 \${msg[lang].PAYMENT_RESULT} - **Result of payment**
 \${msg[lang].PAYMENT_DETAILS} - **Details of payment**
 \${msg[lang].PAYMENT_ORDER} - **Order ID**
 \${msg[lang].PAYMENT_MERCHANT} - **Merchant**
 \${msg[lang].PAYMENT_DESCRIPTION} - **Description**
 \${msg[lang].ENTER_CARD(BUNDLE_PROJECT)} - **Enter payment card information**
 \${msg[lang].ENTER_CARD_METHOD} - **Payment method**
 \${msg[lang].CARD_CHOOSE} - **Credit / Debit card**
 \${msg[lang].CARD_DETAILS} - **Card data**
 \${msg[lang].CARD_TYPE} - **Card type**
 \${msg[lang].CARD_NUMBER} - **Card Number**
 \${msg[lang].EXPIRE_DATE} - **Expiry Date**
 \${msg[lang].RETURN_TO_MERCHANT} - **Return to the [Merchant Site]({0}) (without completing your payment)**
 \${msg[lang].ENTER_CARD_SUBMIT_NOTE} - **Note** : The next screen you see may be **payment card verification through your card issuer**
 \${msg[lang].BUTTON_CONTINUE} - **Proceed**
 \${msg[lang].PAYMENT_SYSTEM_GET_MORE} - **Click on the image to learn more**
 \${msg[lang].ENTER_CVC_TEXT} - **Please enter the CVV2/CVC2 code to perform the payment**
 \${msg[lang].ENTER_CVC} - **Card verification code**
 \${msg[lang].JAVASCRIPT_ERROR_PAN_EMPTY} - **Card number must be defined**
 \${msg[lang].JAVASCRIPT_ERROR_PAN_FORMAT} - **There is a wrong symbol in card number**
 \${msg[lang].JAVASCRIPT_ERROR_PAN} - **Incorrect card number**
 \${msg[lang].JAVASCRIPT_ERROR_EXPIRE_DATE} - **Incorrect expiration date**
 \${msg[lang].JAVASCRIPT_ERROR_EMAIL_EMPTY} - **E-mail must be defined**
 \${msg[lang].JAVASCRIPT_ERROR_EMAIL_FORMAT} - **Wrong E-mail**

`\${msg[lang].JAVASCRIPT_ERROR_CVC_EMPTY}` - **The CVC value cannot be empty**
`\${msg[lang].JAVASCRIPT_ERROR_CVC_FORMAT}` - **Wrong CVC/CVV value**
`\${msg[lang].TITLE_PAYMENT}` - **Enter payment card information**
`\${msg[lang].TITLE_FAILURE}` - **The request completed with failure**
`\${msg[lang].TITLE_SUCCESS}` - **Transaction is approved**

4. Step-by-step algorithm for creation of own payment page

1. Create html layout of the payment page. There are no restrictions; you have an opportunity to create any design. Add tables of styles, necessary scripts and images to the document layout in the above-mentioned format.

```
<!DOCTYPE html>
<html><head>
<meta http-equiv="content-type" content="text/html; charset=windows-1251">
  <title>Credit Card</title>

  <!-- Styles -->
  <link href="css/${payment.templateName}/css.css" rel="stylesheet" type="text/css">
  <link rel="stylesheet" href="css/${payment.templateName}/style.css">
  <meta name="robots" content="noindex, follow">
</head>
```

HTML form with fields to enter the card details. Names to fields are given according to the above-mentioned documentation.

```
<form class="credit-card" action="https://ecg.test.upc.ua/go/pay" method="POST" >
<input type="hidden" name="action" value="process" />
  <div class="form-header">
    <h4 class="title">Credit card detail</h4>
  </div>
  <div class="form-body">
    <!-- Card Number -->
    <input class="card-number" placeholder="Card Number" type="text" name="CardNumber">
    <!-- Date Field -->
    <div class="date-field">
      <div class="month">
        <select name="ExpMonth">
          <option value="01" selected="selected">January</option>
          <option value="02">February</option>
          <option value="03">March</option>
          <option value="04">April</option>
          <option value="05">May</option>
          <option value="06">June</option>
          <option value="07">July</option>
          <option value="08">August</option>
          <option value="09">September</option>
          <option value="10">October</option>
          <option value="11">November</option>
          <option value="12">December</option>
        </select>
      </div>
      <div class="year">
        <select name="ExpYear">
          <option value="2018" selected="selected" >2018</option>
          <option value="2019">2019</option>
          <option value="2020">2020</option>
          <option value="2021">2021</option>
          <option value="2022">2022</option>
          <option value="2023">2023</option>
          <option value="2024">2024</option>
        </select>
      </div>
    </div>
    <!-- Card Verification Field -->
    <div class="card-verification">
      <div class="cvv-input">
        <input placeholder="CVV" type="text" name="Cvc">
      </div>
      <div class="cvv-details">
        <p>3 or 4 digits usually found <br> on the signature strip</p>
      </div>
    </div>
    <!-- Buttons -->
    <button type="submit" class="proceed-btn"><a href="#">Proceed</a></button>
  </div>
</form>
```

2. Add html layout of the page, css files, necessary images and scripts to archive and send to ec@upc.ua.
3. Upon confirmation from the department employees of loading files to the gateway, you can try to carry out transaction and check a new payment form.



```

<form action="https://ecg.test.upc.ua/go/pay" method="POST">
  <input name="Version" type="hidden" value="1" />
  <input name="MerchantID" type="hidden" value="          " />
  <input name="TerminalID" type="hidden" value="          " />
  <input name="TotalAmount" type="hidden" value="1000" />
  <input name="Currency" type="hidden" value="980" />
  <input name="locale" type="hidden" value="ru" />
  <INPUT TYPE="HIDDEN" NAME="PurchaseTime" VALUE="120820161430"/>
  <input id="order" name="OrderID" type="hidden" value="121212121"/>
  <input name="PurchaseDesc" type="hidden" value="custom" />
  <input name="Signature" type="hidden" value="" />
<button id="submit">go</button>

```

4. Updated page

Credit card detail

 [Tell Me More](#)

Credit / Debit card

3 or 4 digits usually found on the signature strip

Proceed

5. Page design after payment completion

In personal account of the merchant, terminal settings have addresses, where customer is redirected, when payment is completed.

Terminal data	
Merchant ID	1753945
Terminal ID	E7881945
Merchant	polka.com
Cutt-Off time	-AUTO ▾ : -AUTO ▾ (hh:mm)
Number of attempts to enter card	1 ▾
Success URL (SUCCESS_URL)	https://www.google.com.ua/
Failure URL (FAILURE_URL)	
Notify URL (NOTIFY_URL)	
Revers transaction on unsuccessful notification (NOTIFY_URL)	No ▾
E-Mail	Oleksiy.RYABCHENKO@upc.ua
To accept cards put into Stop-list by other merchants	Yes ▾
<input type="button" value="Update"/>	<input type="button" value="Reset"/>

If addresses SUCCESS_URL i FAILURE_URL are filled in, then after payment the customer is redirected to the shop page, indicated in the settings. If fields are empty, then after the transaction the customer receives a standard page of the payment results.

2018-11-12 12:07:40 EET

Result of payment

Transaction code	000 : Transaction is approved
Approval code	394656
Amount	10.00UAH



E-mail: Oleksiy.RYABCHENKO@upc.ua

In order to change the page of payment results for more stylized without interventions to the web-site (SUCCESS, FAILURE pages), it's necessary to send success.html and failure.html for successful and unsuccessful pages correspondingly similar to the payment form. All accompanying files to the web-site shall be sent in archived form.

To reflect payment results, it's possible to obtain such data from the objects on the page:

1. $\${amount}\ \${currency}$ – amount of payment

2. $\{name\}$ – name of the merchant
3. $\{payment.orderId\}$ – order number
4. $\{payment.description\}$ – payment description
5. $\{payment.tranCode\}$ – transaction code
6. $\{payment.approvalCode\}$ – transaction code
7. $\{payment.transTime?string["dd.MM.yyyy, HH:mm"]\}$ – date and time of transaction
8. $\{pan\}$ – masked card number.

SUCCESS, FAILURE pages shall comply with all above mentioned rules similarly to the payment page. For unsuccessful page it's necessary to implement the mechanism of the payment page opening for repeated attempt.

Sample of a successful page:



6. Sample of individual payment page



Amount	Order ID
1000 UAH	7705
Merchant	Description
http://www.polka.com	custom

VISA

Card Number

MM YY Expiry Date

CVV*

Email

CONTINUE

Verified by
VISA

[Click here for detail](#)

[← Back to the order without paying](#)

*Enter CVV2/CVC2 code on the back of your card