

Verifone eCommerce Saved card and one-click payments quick guide

This is a supplementary documentation for the eCommerce interface documentations, meant to help with the integration process. More specific details and additional functions can be found in the interface documentation proper.

General notes:

Production environment uses three different payment servers.

Webstore/app needs to be able to check if the server is usable prior to sending the customer to the hosted pages-environment. This is done by sending a GET request to the hosted payment page (ex. <https://epayment1.point.fi/pw/payment>), and the server is usable if the reply has 200 OK-http header. With any other response, the server cannot be used, and status polling should be done for the other server instead. The webstore/app must also alternate between which server to send the status request to at first between transactions.

Server interface doesn't need to be polled prior to sending the message, and causes unneeded delays to transactions. Send the message directly to the server interface, and based on the reply either proceed as normal, or resend the request to the second server.

All example requests have had their signature fields stripped out to save space. Refer to the interface reference guide on how to calculate signature fields (s-t-256-256_signature-one and/or s-t-256-256_signature-two)

1. Saving the card with a 0,01€ authorization

Request:

s-f-32-32_payment-token=20B27724BC7C488A2C4E148958991B62

locale-f-2-5_payment-locale=fi_FI

t-f-14-19_payment-timestamp=2015-11-25 12:17:19

s-f-1-36_merchant-agreement-code=demo-merchant-agreement

s-f-1-36_order-number=ORD1448453837

t-f-14-19_order-timestamp=2015-11-25 12:17:19

i-f-1-3_order-currency-code=978

l-f-1-20_order-gross-amount=1

l-f-1-20_order-net-amount=1

l-f-1-20_order-vat-amount=0

i-t-1-1_save-payment-method=2

s-f-1-30_buyer-first-name=John

s-f-1-30_buyer-last-name=Doe

s-f-1-100_buyer-email-address=john.doe@verifone.com

s-t-1-255_buyer-external-id=johndoe

s-t-1-30_style-code=custom-template

i-t-1-1_skip-confirmation-page=1

s-f-5-256_success-url=<http://verifone.local/store/success/>

s-f-5-256_rejected-url= <http://verifone.local/store/rejected/>

s-f-5-256_cancel-url= <http://verifone.local/store/cancelled/>

s-f-5-256_expired-url= <http://verifone.local/store/expired/>

s-f-5-256_error-url= <http://verifone.local/store/error/>

s-f-1-30_software=ePayment test

s-f-1-10_software-version=1.0

i-f-1-11_interface-version=5

When using i-t-1-1_save-payment_method with value of 2, the transaction isn't captured but only authorized, and the authorization is automatically reversed. The amount to be authorized isn't shown to the customer, and 'Pay'-button label changes to 'Save card'-label on the payment page (note that if 3D Secure is performed, the selected amount may be shown in the 3D Secure UI). First name, last name and email address must be sent to the interface, and in addition to these, either phone number or an external-id (or both) must be sent. External-id would be a identifier decided on by the webstore/app (userID, user account database index etc).

Other fields of note are s-t-1-30_style-code, which can be used for a custom payment page style to match it better with the app or webstore. I-t-1-1_skip-confirmation-page can be used to skip past the Verifone payment confirmation page to streamline the process (for card saving purposes, this isn't required; with save only the confirmation page is skipped automatically).

The amount specified in the request will be the amount that is used to make the authorization. After successful authorization, an automatic reversal is issued (when saving card only, for save & pay, payment naturally is processed). As authorization is passed onto the customer's card, the sum should be kept small, in 0,01-1,00€ range. Zero sum authorization (account check) isn't allowed, however unless separately agreed to.

Response when successful will be:

i-f-1-11_interface-version=5

i-f-1-3_order-currency-code=978

I-f-1-20_order-gross-amount=1

I-f-1-20_transaction-number=5209011712

s-f-1-10_software-version=1.46.1.55

s-f-1-30_payment-method-code=visa

s-f-1-36_order-number=ORD1448453837

s-t-1-26_filing-code=151125018341

s-t-1-6_card-expected-validity=112020

t-f-14-19_order-timestamp=2015-11-25 12:17:20

Things that should be noted:

Card expiry date, MMYYYY in s-t-1-6_card_expected_validity (returned only when using interface version 5+), expired cards will be automatically removed from the payment system when they expire.

Transaction number, I-f-1-20_transaction-numberPayment method, s-f-1-30_payment-method-code

+signature verifications. Note that the return message might have fields s-t-1-40_shop-receipt__phase, s-t-1-40_shop-order__phase or s-t-1-40_submit in it, and they should be excluded from signature verification.

2. Listing saved payment methods (server interface, list-saved-payment-methods operation)

When querying for saved cards, the **exact** same customer details must be used, as were used when the card was. For example, if email address is different from what it was while saving, the saved payment method will not be returned. However, if using buyer-external-id, only it needs to match.

Request:

```
I-f-1-20_request-id=1448455863
t-f-14-19_request-timestamp=2015-11-25 12:51:03
s-f-1-36_merchant-agreement-code=demo-merchant-agreement
s-f-1-30_software=ePayment test
s-f-1-10_software-version=1.0
i-f-1-11_interface-version=5
s-f-1-30_operation=list-saved-payment-methods
s-f-1-30_buyer-first-name=John
s-f-1-30_buyer-last-name=doe
s-f-1-100_buyer-email-address=john.doe@verifone.com
s-t-1-255_buyer-external-id=johndoe
```

Response:

```
i-f-1-11_interface-version=5
I-f-1-20_request-id=1448455863
I-f-1-20_response-id=1448455863
I-t-1-20_payment-method-id-0=3846487
s-f-1-10_software-version=1.46.1.55
s-f-1-30_operation=list-saved-payment-methods
s-t-1-30_payment-method-code-0=visa
s-t-1-30_payment-method-title-0=492987*****4825
s-t-1-6_card-expected-validity-0=112020
t-f-14-19_response-timestamp=2015-11-25 12:56:39
```

Saved card ID is returned as I-t-1-20_payment-method-id-X=? which then can be used to make payments.

Again, attention should be paid to the card expiry date, if it wasn't noted down in step 1 already. Payment system will not notify the customer that the card is about to expire, and the webstore/app should take care of this. Reply may contain multiple cards if more than one was saved, the index in the end of the field will be incremented accordingly (payment-method-id-1, payment-method-id-2 etc.)

3. Removing a saved payment method (server interface; delete-saved-payment-method)

For deleting a saved card, only the card ID in form of saved-payment-method-id is required.

Request:

I-f-1-20_request-id=1448457874
t-f-14-19_request-timestamp=2015-11-25 13:24:34
s-f-1-36_merchant-agreement-code=demo-merchant-agreement
s-f-1-30_software=ePayment test
s-f-1-10_software-version=1.0
i-f-1-11_interface-version=5
s-f-1-30_operation=remove-saved-payment-method
I-t-1-20_saved-payment-method-id=3846487

Response:

i-f-1-11_interface-version=5
I-f-1-20_request-id=1448457874
I-f-1-20_response-id=1448457874
I-t-1-10_removed-count=1
s-f-1-10_software-version=1.46.1.55
s-f-1-30_operation=remove-saved-payment-method
t-f-14-19_response-timestamp=2015-11-25 13:25:22

4. Making a payment (server interface; process-payment)

Once again, payment is done using the customer details that **exactly match** the details used in card saving and the card ID number for the card which payment will be used for the transaction as gathered in step 2. (or as earlier, only buyer-external-id needs to match if used)

Request:

I-f-1-20_request-id=1448456697
t-f-14-19_request-timestamp=2015-11-25 13:04:57
s-f-1-36_merchant-agreement-code=demo-merchant-agreement
s-f-1-30_software=ePayment test
s-f-1-10_software-version=1.0
i-f-1-11_interface-version=5
s-f-1-30_operation=process-payment
locale-f-2-5_payment-locale=fi_FI

t-f-14-19_payment-timestamp=2015-11-25 13:14:57
t-f-14-19_order-timestamp=2015-11-25 13:14:57
s-f-1-36_order-number=ORD1448456700
i-f-1-3_order-currency-code=978
l-f-1-20_order-gross-amount=12000
s-f-1-30_buyer-first-name=John
s-f-1-30_buyer-last-name=Doe
[s-f-1-100_buyer-email-address=john.doe@verifone.com](#)
s-t-1-255_buyer-external-id=johndoe
l-t-1-20_saved-payment-method-id=3846487

Response:

i-f-1-11_interface-version=5
i-f-1-3_order-currency-code=978
l-f-1-20_order-gross-amount=12000
l-f-1-20_request-id=1448456697
l-f-1-20_response-id=1448456697
l-f-1-20_transaction-number=5209021440
s-f-1-10_software-version=1.46.1.55
s-f-1-30_operation=process-payment
s-f-1-30_payment-method-code=visa
s-f-1-36_order-number= ORD1448456700
s-t-1-26_filing-code=151125018361
t-f-14-19_order-timestamp=2015-11-25 13:14:57
t-f-14-19_response-timestamp=2015-11-25 13:17:19

Special care should be had when checking the result for payment. Make sure filing-code, order-gross-amount and payment-method-code are present.

5. Testing and test cases

It is recommended to use Verifone test environment for all testing before entering production.

Test environment can be accessed at:

<https://epayment.test.point.fi/pw/payment> for hosted pages

<https://epayment.test.point.fi/pw/serverinterface> for server interface

For custom payment page, the template should be uploaded via TCS Client which can be accessed at

<https://tcs.test.point.fi/tcs-client/index.html>

For general testing, demo-merchant-agreement can be used as s-f-1-36_merchant-agreement-code in requests, the keys are included in the integration package. For custom template testing, a specific test account needs to be created, for further details please contact Verifone.

Test cards:

Test environment doesn't use actual acquirer test environments by default, and all payments are directed to a mock authorization server, which will accept all properly formatted payment card numbers, which can be generated via <http://www.getcreditcardnumbers.com/> for example.

There is a special case for testing authorization errors, if the payment sum is in the range of 90,01€ - 99,99€, an error is returned when paying.

Test cases

What:	How:	Expected result:
Sanity checks for customer inputs	Try long user details (name, last name, email address etc), special characters in details etc	Relevant fields are truncated to proper sizes for the interface, special characters are encoded properly and there is no issues with signature generation.
Customer attempts payment with an expired card	Use incorrect l-t-1-20_saved-payment-method-id. Verifone system deletes all expired cards, so using a bogus payment-method-id simulates this error.	App/webstore returns a proper error to the customer before or after issuing process-payment. It is also a good idea to notify customer before card expires in-app, expiration date is supplied to the app/store in s-t-1-6_card-expected-validity when using list-saved-payment-methods
Card payment is rejected	Use l-f-1-20_order-gross-amount=9116 or another amount in range of 90,01-99,99€	App/webstore should handle rejected payments in a graceful manner.