Recurring Payments

This document outlines how to perform Recurring Payments.

Published: 1 August 2017

4.1 (c)

Table of Contents

1 Introduction

This document explains how to process recurring payments on your site (also known as continuous authority, periodic billing or repeat payments).

> ⚠ **Your merchant ID must be capable of processing recurring or continuous authority payments.** If in doubt, we strongly recommend contacting your bank for clarification.

This document assumes a basic understanding of how to process authorisations.
We recommend reading our full **XML Specification** before proceeding.

This document contains XML examples throughout, but you may prefer to download XML examples separately. These can be found within the Web Services example files which can be downloaded from our website: http://webapp.securetrading.net/examples/WEBSERVICES.zip

Advantages of Recurring Payments

Process repeat payments for customers without needing to store their card details on your own system.

A recurring payment is only processed when a request has been submitted by your system, allowing for greater control.

Alternative solution: Subscriptions

Our subscriptions solution allows you to submit a single request and we will process recurring payments automatically at regular intervals. Learn more >>>

1.1 Process overview

1	Customer enters their card details on your secure website.
2	Your system processes a **parent request** including either of the following request types: • AUTH • ACCOUNTCHECK *Refer to section 2 for further information.*
3	At a later time, your system processes a **child request** that inherits the previously-stored payment, billing and delivery details and seeks authorisation for a new payment. *Refer to section 3 for further information.*
4	Your system can submit **child requests** at various intervals to process new payments. The customer will only be debited after a new request is submitted.

2 Parent

This section describes the initial XML Request to be submitted. You can inherit the payment details submitted in this request in subsequent recurring payments.

2.1 Parent request overview

The parent request can either be “AUTH” or “ACCOUNTCHECK”. You will need to read the two flows below and choose the request type that best suits your business requirements:

“AUTH”: Processes the first payment immediately

1. Customer enters their card details and agrees to schedule a subscription, with the first payment being taken immediately.

2. Your system submits an AUTH request. All payment, billing and delivery details submitted in the request are stored securely in our records for future use.

3. We will contact the acquiring bank to perform AVS and security code checks, before seeking authorisation for the payment.

   Funds are reserved immediately on the customer’s bank account.

4. Your system receives an AUTH response, including the outcome of the request. The XML Response returned contains a unique transaction reference.

“ACCOUNTCHECK”: Does not process the first payment immediately (Only performs checks on the card and billing address)

⚠ **Only supported by certain acquiring banks.**
Please contact your bank before developing a solution that includes ACCOUNTCHECKs.

1. Customer enters their card details and agrees to schedule a subscription, with the first payment being taken at a later time.

2. Your system submits an ACCOUNTCHECK request. All payment, billing and delivery details submitted in the request are stored securely in our records for future use.

3. We will contact the acquiring bank to perform AVS and security code checks.

   Funds are not reserved and the customer is not charged.

4. Your system receives an ACCOUNTCHECK response with the results of the checks performed. The XML Response returned contains a unique transaction reference.

2.2 Parent request example

Insert your preferred request type “AUTH” or “ACCOUNTCHECK” in the **request type** field. Fields of interest for processing recurring payments are highlighted in **bold**.

```
<requestblock version="3.67">
  <alias>test_site12345</alias>
  <request type="AUTH">
    <billing>
      <postcode>TE45 6ST</postcode>
      <premise>789</premise>
      <payment type="VISA">
        <expirydate>10/2031</expirydate>
        <pan>4111110000000211</pan>
        <securitycode>123</securitycode>
      </payment>
      <amount currencycode="GBP">1234</amount>
      <subscription type="RECURRING">
        <number>1</number>
      </subscription>
    </billing>
    <operation>
      <sitereference>test_site12345</sitereference>
      <accounttypedescription>ECOM</accounttypedescription>
    </operation>
  </request>
</requestblock>
```

2.3 Parent request field specification

Both AUTH and ACCOUNTCHECK parent requests have a similar XML structure, as described in the XML Specification, but are subject to different requirements when employed as part of a recurring payments solution. These differences are outlined in the table below.

> We recommend that the customer’s billing address and security code details are submitted in the parent request, as these will be used in the AVS and security code checks (see the **AVS & Security Code document** for further information).

Tag	Type	Length	Required	Comment
operation			Y	
accounttype description	an	20	Y	The correct account type value for the parent request is dependent on your acquiring bank. Please contact your bank for further information. This field must be one of the following values: “ECOM” - e-commerce “MOTO” - Mail or Telephone Order
parent transaction reference	an	25	N	You can submit the transaction reference of a previously submitted ACCOUNTCHECK in this field, in order to inherit previously-submitted payment details.
billing				
amount currencycode=””	an	3	Y	The transaction currency. This must remain the same in all future child requests processed that inherit from this parent request.
amount	n	13	Y	The amount associated with this payment in base units, with no commas or decimal points. *e.g. £10.99 would be submitted as “1099” but ¥246 would be submitted as “246”*. This will be inherited in all child requests that inherit from this parent request, unless manually overridden.
subscription type=””	an	11	Y	This is the type of subscription: “RECURRING” is for when the customer is making a recurring payment for a new product/service each time. “INSTALLMENT” is for when a customer is purchasing a single item over several installments (only supported by certain acquirers; contact Support for further info).
number	n	1	Y	For the first request for a given customer, this must be submitted with value “1” (indicating the 1st payment). **For each subsequent payment, the number submitted must be incremented by 1.** For example the second transaction is “2”, the third is “3”, etc.

2.4 Parent response

The response returned has the same XML structure as a standard AUTH or ACCOUNTCHECK Response, as documented in the **XML Specification**. This includes the outcome of **AVS and security code** checks, which can be found in the **<security>** tag.

3 Child

This section describes the subsequent recurring payment. This section assumes you have already processed a parent request as described in section 2.

Before processing a child request, you must ensure that the parent request has completed successfully. Investigate any issues raised and if assistance is required contact our support team.

About declines: Mastercard has mandated that in cases where a recurring payment has been declined by the card issuer, your system **must not** retry the request more than once per day for the next 31 days. After this period has passed, your system must not send further requests for the affected customer.

3.1 Child request overview

The child XML Request type must be "AUTH":

1. Your system requests that a new payment is processed by submitting an AUTH request that includes the transaction reference of the parent.
2. We contact the acquiring bank to seek authorisation for the payment, using the billing and delivery details inherited from the parent request.

   (Your system does not need to re-submit these details)
3. Your system receives an AUTH response, including the results of the request.

3.2 Child request examples

The request type submitted for the child request must be "AUTH" (submitted in the **request type** field). In the following examples, fields of interest for processing recurring payments are highlighted in **bold**.

3.2.1 Process a new payment inheriting all details from the parent

```
<?xml version="1.0" encoding="utf-8"?>
<requestblock version="3.67">
<alias>test_site12345</alias>
 <request type="AUTH">
  <operation>
    <sitereference>test_site12345</sitereference>
    <accounttypedescription>RECUR</accounttypedescription>
    <parenttransactionreference>12-3-4567</parenttransactionreference>
  </operation>
  <billing>
    <amount currencycode="GBP">1234</amount>
    <subscription type="RECURRING">
      <number>2</number>
    </subscription>
  </billing>
 </request>
</requestblock>
```

The example above assumes that the previous request was the first to be processed, therefore the **subscription/number** field is assigned value "2". Subsequent child requests are processed with the same structure, incrementing this number each time (i.e. next payment in sequence would be "3", then "4" and so on).

3.2.2 Process a new payment with updated details

The following examples are for child requests that inherit all payment, billing and delivery details from a parent, except some fields have been updated with new values. (Updated fields are highlighted in **bold**). *See section 6.1 for a list of fields that can be updated in this way.*

3.2.2.1 One-off change

In this child request example. the billing address has been overridden with updated values:

```
<?xml version="1.0" encoding="utf-8"?>
<requestblock version="3.67">
<alias>test_site12345</alias>
 <request type="AUTH">
  <operation>
    <sitereference>test_site12345</sitereference>
    <accounttypedescription>RECUR</accounttypedescription>
    <parenttransactionreference>12-3-4567</parenttransactionreference>
  </operation>
  <billing>
    <premise>789</premise>
    <street>Test Street</street>
    <town>Bangor</town>
    <county>Gwynedd</county>
    <postcode>TE45 6ST</postcode>
    <country>GB</country>
    <subscription type="RECURRING">
      <number>2</number>
    </subscription>
  </billing>
 </request>
</requestblock>
```

Provided all future child requests continue to include the same reference to the original parent, they will continue to inherit all details from the original parent, unless overridden in the manner outlined above.

3.2.2.2 Carrying changes forward in future recurring payments

In this child request example. the card expiry date has been overridden with updated values:

```
<?xml version="1.0" encoding="utf-8"?>
<requestblock version="3.67">
<alias>test_site12345</alias>
 <request type="AUTH">
  <operation>
    <sitereference>test_site12345</sitereference>
    <accounttypedescription>RECUR</accounttypedescription>
    <parenttransactionreference>12-3-4567</parenttransactionreference>
  </operation>
  <billing>
      <payment>
        <expirydate>10/2031</expirydate>
      </payment>
    <subscription type="RECURRING">
      <number>2</number>
    </subscription>
  </billing>
 </request>
</requestblock>
```

To retain new details in future child requests, they must inherit from the new child instead, essentially making it the new parent. **You must always ensure that the subscription/number continues to be incremented in an unbroken sequence.**

We strongly recommend against updating the card number, as this may go against the terms agreed with your bank. If the customer needs to update their card number, it is best practice to start an entirely new sequence of recurring payments by submitting a new parent (section 2), ensuring the **subscription/number** has been reset to “1”. If you are unsure, please contact your acquiring bank or our Support Team (section 5.1).

3.3 Child request field specification

The AUTH child request has the same XML structure as the standard AUTH request described in the **XML Specification**, but is subject to different requirements, which are outlined below.

You will need to include the parent transaction reference within the **operation** tag.
We use this reference to select the customer's details from our records.

Tag	Type	Length	Required	Comment
operation			Y	
accounttype description	an	20	Y	**This value must be "RECUR".**
parent transaction reference	an	25	Y	The transaction reference returned in the XML Response of the initial request.
sitereference	an	25	Y	Your site reference.
billing			Y	
amount currencycode= ""	an	3	N	If submitted, **this must be the same currency used in the parent request.**
amount	n	13	N	The amount associated with this payment in base units, with no commas or decimal points. *e.g. £10.99 would be submitted as "1099" but ¥246 would be submitted as "246".* If you don't submit this field, the amount processed will be inherited from the preceding parent request.
subscription type=""	an	11	Y	This is the type of subscription: "RECURRING" is for when the customer is making a recurring payment for a new product/service each time. "INSTALLMENT" is for when a customer is purchasing a single item over several installments (only supported by certain banks; contact Support for further info).
number	n	6	Y	This is used to identify a payment's position within a sequence of recurring transactions. **For each subsequent payment, the number submitted should be incremented by 1 (without gaps).** e.g. 2nd transaction is "2", 3rd is "3", then "4" etc. (You should only increment this number if the previous recurring payment request was successful) We do not impose limits on the number of payments made against a card.
payment				
security code	n	4	N	To gain the additional advantages of security code checks, the security code value may optionally be submitted in recurring payment requests. However, you must prompt the customer for the security code prior to submitting the child request. **You must never store the security code on your system.**

We allow any valid **number** to be submitted in each request, but strongly recommend following an incrementing sequence. Your acquiring bank may mandate this (contact your acquiring bank for clarification). You must ensure the correct **number** is submitted.

3.4 Child response

The child response has the same XML structure as a typical AUTH response described in the **XML Specification**, but is subject to different specifications, which are outlined below.

3.4.1 Child response example

```
<responseblock version="3.67">
  <requestreference>X1234abcd</requestreference>
  <response type="AUTH">
    <merchant>
      <merchantnumber>12345678</merchantnumber>
      <merchantcountryiso2a>GB</merchantcountryiso2a>
      <operatorname>test_site12345</operatorname>
    </merchant>
    <transactionreference>1-2-3</transactionreference>
    <timestamp>2015-09-23 07:56:02</timestamp>
    <acquirerresponsecode>00</acquirerresponsecode>
    <operation>
      <splitfinalnumber>1</splitfinalnumber>
      <parenttransactionreference>1-5-1</parenttransactionreference>
      <accounttypedescription>RECUR</accounttypedescription>
    </operation>
    <settlement>
      <settleduedate>2015-09-23</settleduedate>
      <settlestatus>0</settlestatus>
    </settlement>
    <billing>
      <amount currencycode="GBP">1234</amount>
      <payment type="VISA">
        <pan>411111######0211</pan>
        <issuercountry>GB</issuercountry>
      </payment>
      <dcc enabled="0"/>
    </billing>
    <authcode>231</authcode>
    <live>1</live>
    <error>
      <message>Ok</message>
      <code>0</code>
    </error>
    <security>
      <postcode>2</postcode>
      <securitycode>0</securitycode>
      <address>2</address>
    </security>
    <acquireradvicecode>0</acquireradvicecode>
  </response>
  <secrand>qweRtY1Yu2Iop</secrand>
</responseblock>
```

3.4.2 Child response field specification

Tag		Type	Length	Required	Comment
operation				Y	
	accounttype description	an	20	Y	This value is “RECUR”.
acquirer advicecode		n	1	C	A numeric value returned if supported by your acquiring bank, indicating if further payments can be processed. For a full mapping, see section 4.2.

⚠ Mastercard has mandated that in cases where a recurring payment has been declined by the card issuer, you **must not** retry more than once per day for the next 31 days. After this period has passed, your system must not send further requests for the affected customer.

4 Additional Notes

Below are some additional notes to consider when processing recurring payments.

4.1 Payment Pages

If processing your initial parent request through **Payment Pages**, please ensure your POST to Secure Trading includes the following:

subscriptionnumber=1
subscriptiontype=**`RECURRING`** (or **`INSTALLMENT`**)

4.2 Acquirer advice code mapping

The acquirer advice code is a numeric value returned if supported by your acquiring bank, indicating if further payments can be processed. The codes are mapped as follows:

Code	Description	Action
0	N/A	No action required
1	"New account information available" The bank has reason to believe that the customer's card details are out-of-date (e.g. the expiry date has passed).	Please check with the customer that their card details are still correct. This field is advisory and we will allow you to re-process even if the card details are not updated, however the acquiring bank may not accept the payment. You may find it useful to contact your bank in such cases for clarification.
2	"Cannot approve at this time"	Try again later. If you are continuing to have difficulties, please contact your acquiring bank.
4	"Do not try again"	Do not process further recurring transactions.
8	"Payment blocked by card scheme"	

For information on how to test for different acquirer advice code values, please refer to the **Testing document**.

4.3 3-D Secure Liability Shift

By processing the first AUTH with 3-D Secure, the subsequent recurring payments may also be covered by the benefits of the schemes, depending on your acquirer. Please contact your bank for further information on recurring payments and 3-D Secure.

For further info on processing 3-D Secure payments, please refer to the **3-D Secure document**.

4.4 Testing

After Secure Trading has provided you with a test site reference, you can process transactions and check that your system handles the responses correctly. You can process successful transactions by submitting the following details:

Payment type	Test PAN
Visa	4111111111111111
Mastercard	5100000000000511

Field name	Value
Billing premise	789
Billing postcode	TE45 6ST
Card security code	123

Please ensure you have tested your integration with all payment methods that will be made available to your customers. The **Testing document** contains a full list of payment credentials that can be used when testing.

5 Further Information and Support

This section provides useful information with regards to documentation and support for your Secure Trading solution.

5.1 Secure Trading Support

If you have any questions regarding integration or maintenance of the system, please contact our support team using one of the following methods.

Method	Details
Telephone	+44 (0) 1248 672 050
Fax	+44 (0) 1248 672 099
Email	support@securetrading.com
Website	http://www.securetrading.com/support/support.html

5.2 Secure Trading Sales

If you do not have an account with Secure Trading, please contact our Sales team and they will inform you of the benefits of a Secure Trading account.

Method	Details
Telephone	0800 028 9151
Telephone (Int'l)	+44 (0) 1248 672 070
Fax	+44 (0) 1248 672 079
Email	sales@securetrading.com
Website	http://www.securetrading.com

5.3 Frequently Asked Questions

Please visit the FAQ section on our website (http://www.securetrading.com/support/faq).

6 Appendix

6.1 Updatable fields

When submitting a child request with updated details (as described in section 3.2.2), the following fields can be overridden:

billing/premise
billing/street
billing/town
billing/county
billing/postcode
billing/country
billing/email
billing/telephone type=””
billing/name/prefix
billing/name/first
billing/name/middle
billing/name/last
billing/name/suffix

customer/premise
customer/street
customer/town
customer/county
customer/postcode
customer/country
customer/email
customer/telephone type=””
customer/name/prefix
customer/name/first
customer/name/middle
customer/name/last
customer/name/suffix
customer/ip
customer/forwardedip

operation/authmethod
merchant/orderreference
merchant/chargedescription
merchant/operatorname

For a specification on these fields, please refer to the **XML Specification**.