

# Recuro

# API reference

Version 1.7 – 26-09-2010

## 1 Revision Log

Version	Date	Description	Author
1.0	17-03-2009	Initial version	OB
1.1	23-05-2009	Optional return- and callback URL arguments Added editsubscription API call	OB
1.2	09-06-2009	Changed subscription report	OB
1.3	24-06-2009	New subscription states	OB
1.4	23-09-2009	Next payment date return parameter for subscription status and cancel option to pause subscription.	OB
1.5	13-12-2009	Added reference in return parameters to addextracharge and discount References to extra charges and discounts in payment callback Subscription status with reference to pending extra charges and discounts	OB
1.6	13-07-2010	Added extra charges and discounts description in callback. Extra charges cancel API call Discount cancel API call	OB
1.7	26-07-2010	Added startdate and customer details to subscription status API call.	OB

## Table of Contents

1	Revision Log .....	1
2	Introduction .....	3
2.1	HTTP POST.....	3
2.2	Notation.....	3
3	Subscription .....	4
3.1	Create Subscription.....	4
3.2	Create types.....	5
3.2.1	Redirect create type.....	5
3.2.2	Link create type.....	6
3.2.3	Email create type.....	6
3.3	Create model.....	6
3.4	Failure model.....	7
3.5	Subscription status.....	9
3.6	Close subscription.....	10
3.7	Delete subscription.....	11
3.8	Renew subscription.....	11
3.9	Edit subscription.....	11
3.10	Pause subscription.....	12
3.11	Resume subscription.....	13
3.12	On demand.....	13
3.13	Extra charges.....	14
3.14	Cancel extra charge.....	14
3.15	Discount.....	14
3.16	Cancel discount.....	15
4	Rules .....	16
4.1	Create rule.....	16
4.2	Rule types.....	17
4.3	Edit rule.....	17
4.4	Erase rule.....	18
5	Callback .....	19
5.1	Card information.....	19
5.2	Payment approved.....	19
5.3	Payment failed.....	20
5.4	Update failed.....	20
6	Reports .....	21
6.1	Subscription report.....	21
6.2	Payment report.....	21
6.2.1	CSV format.....	22
6.2.2	XML format.....	22
7	Templates .....	23
7.1	Mail templates.....	23
7.2	HTML templates.....	23
8	Test .....	24
8.1	Creation and approved.....	24
8.2	Failure.....	24

## 2 Introduction

In this reference the API for Reкуро is described. The Reкуро API allows a merchant application to interact with Reкуро. To manage a Reкуро account manually the Merchant Administration can be used by logging in at [www.reкуро.dk](http://www.reкуро.dk).

Reкуро operates with three main notions, *subscriptions*, *rules* and *payments*.

**Subscription** – A subscription represents a periodic recurrent payment. For example a subscription to newspaper or the like.

**Rule** – Every subscription has a rule. A rule can be seen as a product and defines price, frequency, number of payments, etc. A rule could for example be 100,00 DKK on the first day of every month.

**Payment** – The rule of a subscription will result in a number of payments for the subscription. A payment is scheduled when the rule of a subscription is due. For example on the first day of the month.

### 2.1 HTTP POST

The integration with Reкуро is done mainly through HTTP POST requests. In most cases server-to-server requests, where the response is in content type `text/plain` and on the form.

```
key1=value1&key2=value2&...&keyN=valueN
```

The values are URL encoded. A response always contain as minimum the following

```
result=xx
```

where *xx* is a numeric error code. Zero (0) means no error. All other values will indicate an error and the response will also contain a textual error message in the parameter `errortext`. Example:

```
result=15&errortext=Unknow+merchant
```

### 2.2 Notation

To describe the format of the parameters for the API calls, the following notation will be used.

<code>Nxx</code>	Numeric with a maximum of <i>xx</i> digits.
<code>ANxx</code>	Alphanumeric with a maximum of <i>xx</i> characters.
<code>[parameter]</code>	Optional parameter. All other will be mandatory.

## 3 Subscription

Handling subscriptions represent the most API calls. In this section API calls are described.

### 3.1 Create Subscription

Creating a subscription is the most basic operation, but also the most complex, as creating a subscription requires attaching card information which must be done in a certified environment.

Creating a subscription consists of two steps, creating the subscription, and attaching card information. These two steps can either be done in immediate continuation of each other, or separated as two individual steps. Below the HTTP POST request for creating a subscription is described. The rest of the section will describe individual parameters and their effect.

URL: <https://www.recuro.dk/api/http/createsubscription>

Example: <https://www.recuro.dk/api/createsubscription.html>

Parameter		Description
merchant	N10	Identification of the merchant. Handed out by Recuro when the merchant is created.
rule	N10	The rule to attach to this subscription. Rules are managed in the merchant administration or through API calls described later.
subno	AN50	The merchants unique identification of the subscription. An error is reported if a subscription already exists with the same subscription number.
[test]	AN	Any value will indicate test subscription. For test, only test cards can be used, and no actual money transfers will be performed. Se section Test for further information on how to test.
[type]	AN	The method to use for attaching card informations to the subscription. Three methods are available: LINK, REDIRECT or EMAIL. Default is LINK. The three methods will be described below.
[startdate]	AN	An optional parameter that indicates that the subscription should be active from the given date. Format: dd-mm-yyyy
[enddate]	AN	An optional parameter that indicates that the subscription should only be active until this date. Format: dd-mm-yyyy
[create_model]	AN	The create model indicates in what state a subscription should be born. The following values can be used: INSTANT_ACTIVE and TICKET_PEND. Default is TICKET_PEND. Create model will be described below.
[fail_model]	AN	The fail model gives the action to be taken when a payment for a subscription fails. The following values can be used: REPORT_CANCEL, REPORT and AUTO_UPDATE. Default is REPORT. Fail model will be described below.
[grace_time]	N10	For create model INSTANT_ACTIVE, grace time gives the time period in minutes where it is accepted that payments cannot be approved because of missing card information.
[lang]	AN2	Language for the page where card information is entered. The format is two letter ISO-3166. As default danish is used.
[returnurl]	AN255	After the flow of entering card information there will be returned to this URL. See below for parameters.
[callbackurl]	AN255	After the flow of entering card information a HTTP POST will be performed to this URL with the result. See below for parameters. For subsequent events for the

		subscription this callback url will also be used. See more in the section Callback.
email	AN50	The customers email. Email is mandatory as the email is used for sending receipts to the customer.
[firstname]	AN50	The customer first name
[lastname]	AN50	The customer last name
[address]	AN50	The customer address
[address2]	AN50	Additional address information for the customer.
[city]	AN50	The customers city
[postalcode]	AN10	The customers postal code
[phone]	AN20	The customers phone
[birthdate]	AN	The birthdate of the customer. Format: dd-mm-yyyy.
[gender]	AN1	Customer gender. Male (M) or female (F).
[comment]	AN	A comment about the customer.

The result of the call depends on the used create type. The three types will be described in the next subsection. The request to return- and callback url looks the same, with the difference that a GET request will be performed to return url and a POST request will be performed to callback url. The following parameters will be used.

Parameter	Description
subno	The merchants unique identification of the subscription.
merchant	The identification of merchant. This is used for systems that maintains subscriptions for multiple merchants.
action	Will in this case be fixed as <b>cardinfo</b> .
[cancel]	If this parameter has a value, the operation of attaching card information to the subscription has been cancelled by the cardholder.
result	Error code. 0 indicates success.
[errortext]	If an error has occurred, an error text will be given.

## 3.2 Create types

The type parameter determines how the flow of attaching card information to the subscription is performed. Entering card information requires a PCI certified environment, which is the reason that card information cannot just be sent to Recuro. Recuro uses a PCI certified Payment Service Provider to handle the card information.

### 3.2.1 Redirect create type

Illustration 1 shows the redirect flow when using type *REDIRECT*. In redirect a flow is started on merchant web page or indirectly through a merchant application. The customer is guided through the pages 1 to 4 and ends either at a receipt page (4), or a merchant return url (5), if this has been supplied. Below the different pages will be described.

1. A merchant web page with a form having createsubscription as action, and with the parameters just described. Se for example <https://www.recuro.dk/api/createsubscription.html>
2. A page showing terms and an Accept and Decline button. The content and style of the page is controlled by the merchant in the merchant administration. See section Templating for a description of how to control page content.
3. A PCI certified PSP page with input box for card number, expire date and CVC.

4. A page showing a receipt.
5. If return url is given, the customer will return to this page.

If the flow is cancelled on page 2 or page 3, the customer will be returned to receipt or return url with the cancel parameter set.

www.recuro.dk  
Terms page

PSP  
Card information

www.recuro.dk  
Receipt page

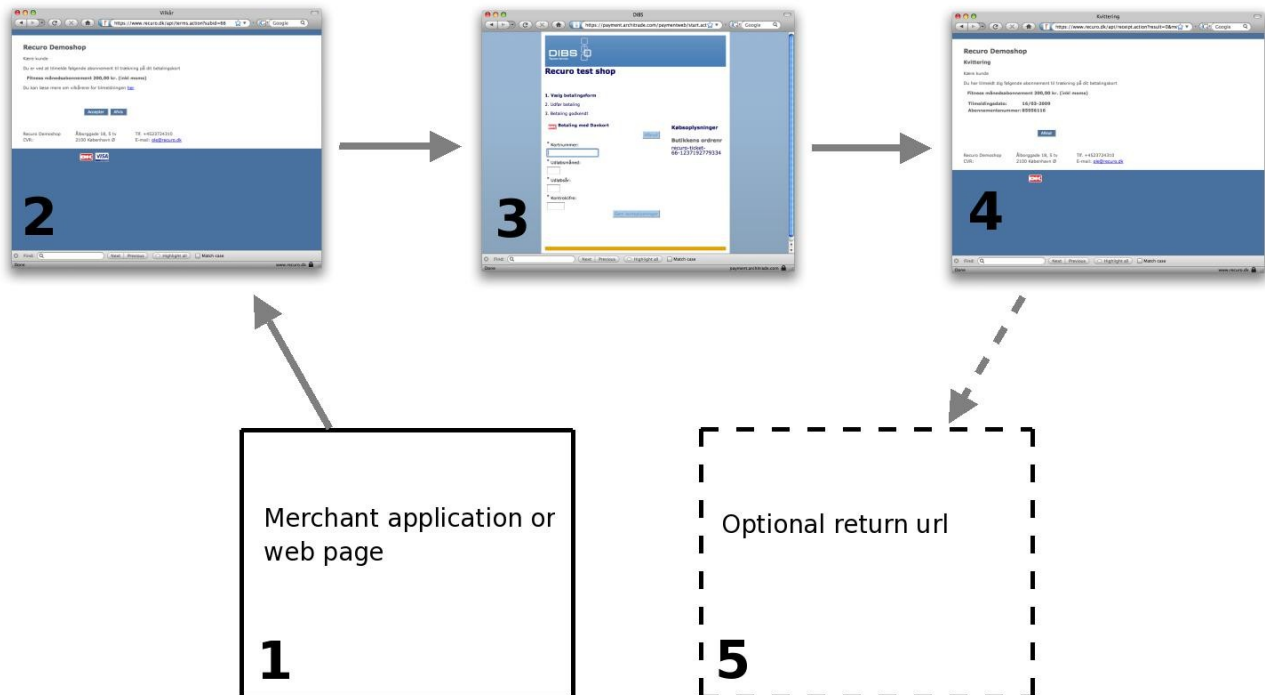


Illustration 1: Redirect flow

### 3.2.2 Link create type

With type *LINK*, a server-to-server call is made to `createsubscription`. The result is not a redirect, but a text reply giving a link to the terms page (2). Notice that the link is URL encoded. Below an example.

```
result=0&link=https%3A%2F%2Fwww.recuro.dk%2Fapi%2Fhttp%2Faddcardinfo%3Fsubid%3D16
```

The returned link can be used when the customer is ready for entering card information.

### 3.2.3 Email create type

The last type is *EMAIL*. In type email, a mail is sent to the customer with the link. The content of the mail is controlled in the merchant administration. See the Templating section for details on how to set email, terms page and receipt page content.

When the customer receives the email and uses the link, the customer will enter the flow on the terms page (2).

## 3.3 Create model

A subscription can through its lifetime be in six main states: *TICKET\_PENDING*, *ACTIVE\_PENDING*, *ACTIVE*,

*ACTIVE\_MISSING*, *TICKET\_FAILED*, and *CLOSED*. How a subscription change state, and what happens at state changes, depends on the create and failure model given at creation.

At creation the the subscription can use one of two models, *INSTANT\_ACTIVE* and *TICKET\_PEND*, defined by the parameter *create\_model*. When a subscription is created as instant active, it means that the subscription is active right away, and therefore payments for the subscription will be scheduled, even if there is no card information. When a payment is tried completed, it will fail if no card information is attached to the subscription. When the missing card information should be seen as an error, can be controlled with the *grace\_time* parameter. This parameter defines how long, given in minutes, that it should be tolerated that payments cannot be completed.

In create model *TICKET\_PEND*, which is the default, the subscription is not active until card information has been attached to subscription. Illustration 2 shows a state diagram for the two create models.

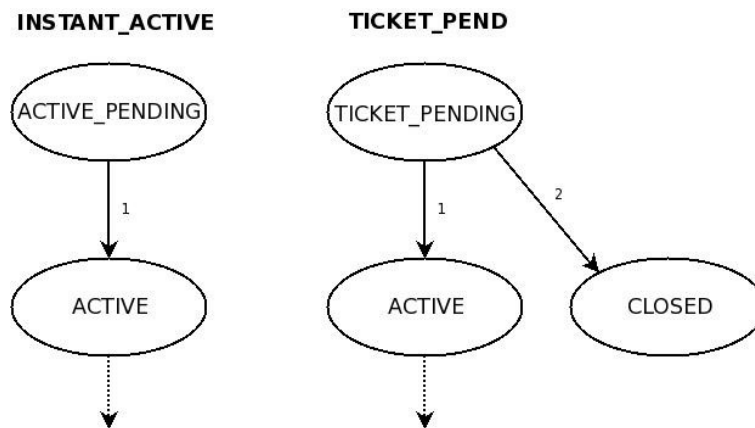


Illustration 2: Create models

The tables below describe the state changes for *INSTANT\_ACTIVE* and *TICKET\_PEND*, respectively.

State chg.	Reason	Effect
1	Card information is attached to the subscription. The state changes from an active state indicating no card information to an active state indicating card information.	Callback is performed and return url will be reached if defined at creation.

State chg.	Reason	Effect
1	Card information is attached to the subscription. The state is changed to active, and payments can now be scheduled.	Callback is performed and return url will be reached if defined at creation.
2	The subscription is actively closed.	-

### 3.4 Failure model

In most cases a subscription will not encounter payment errors through its lifetime. Payment errors is due to expired card, blocked card, or the like.

If payments fail for a subscription, a fail procedure takes effect. The fail procedure depends on the parameter *fail\_model* given at creation. There are three different fail models.

- *REPORT\_CANCEL* – On payment error the subscription is closed and the error is reported with callback to the merchant.
- *REPORT* – On payment error the error is reported to the merchant but the subscription stays active. It is up to the merchant to try to add new card information, for example by calling renew subscription described later.
- *AUTO\_UPDATE* – On payment error, a process is started where Recuro sends a number of renewal request emails to the customer. The number and interval is defined on the rule of the subscription. Only if the customer does not add new card information is the error reported to the merchant.

Illustration 3 shows state diagram for the three fail models.

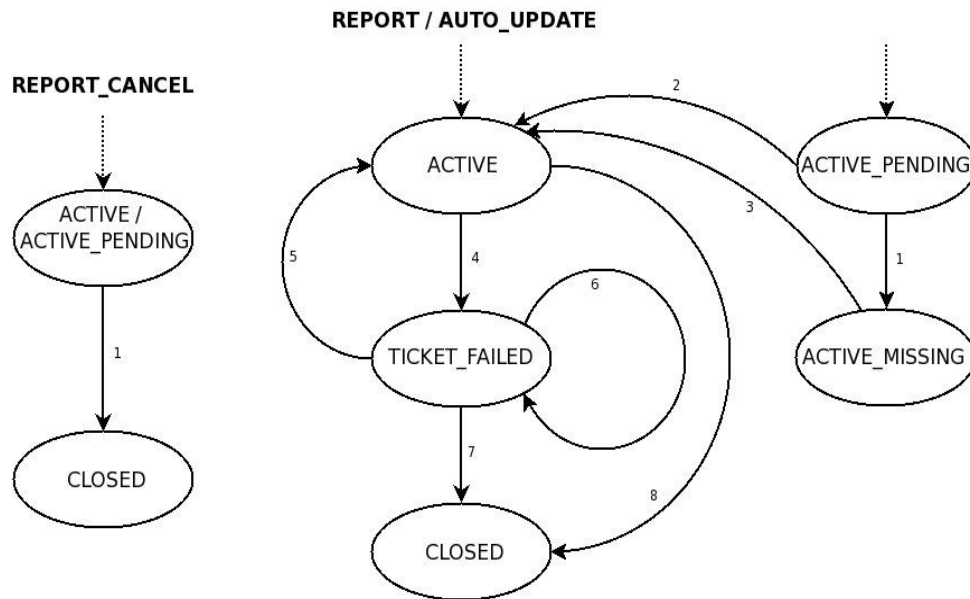


Illustration 3: Fail models

*REPORT\_CANCEL* has the following state changes.

State chg.	Reason	Effect
1	Payment fails, either because subscription has no card information, or invalid card, and grace period, if defined, has expired.	<ul style="list-style-type: none"> <li>• Subscription is closed.</li> <li>• Payment is declined.</li> <li>• The merchant is notified with payment failed callback.</li> </ul>

*REPORT* has the following state changes.

State chg.	Reason	Effect
1	Payment fails because of no card information, and grace period, if defined, has expired.	<ul style="list-style-type: none"> <li>• State is changed to <i>ACTIVE_MISSING</i> which indicates that the subscription is active, but payments cannot be completed due to missing card information.</li> <li>• The merchant is notified with payment failed callback.</li> </ul>
2	Card information is attached	<ul style="list-style-type: none"> <li>• The state is changed to active</li> <li>• The merchant is notified with callback</li> </ul>

3	Card information is attached	<ul style="list-style-type: none"> <li>The state is changed to active</li> <li>The merchant is notified with callback</li> </ul>
4	Payment fails	<ul style="list-style-type: none"> <li>The state is changed to <i>TICKET_FAILED</i> which indicates that the subscription is active but no without valid card information.</li> <li>The merchant is notified with callback</li> </ul>
5	New card information is attached, or existing card information is reactivated, and payment is approved. Card information can be reactivated by setting retry interval in the administration.	<ul style="list-style-type: none"> <li>The state is changed back to active</li> <li>The merchant is notified with callback if new card information is added.</li> </ul>
6	Payment fails	<ul style="list-style-type: none"> <li>State remains the same</li> </ul>
7	The subscription is closed actively or because auto update procedure has failed, and the merchant has the option to close in this event.	<ul style="list-style-type: none"> <li>Subscription is closed</li> <li>Payments are declined</li> </ul>
8	The subscription is closed actively	<ul style="list-style-type: none"> <li>Subscription is closed</li> <li>Payments are declined</li> </ul>

*AUTO\_UPDATE* is totally analogue to *REPORT*, with the exception that in *AUTO\_UPDATE* a process is started to try to get the customer to update the card information. In the administration a setting will control if the subscription is closed when auto update fails.

### 3.5 Subscription status

The state of a subscription can be requested with the following call.

URL: <https://www.recuro.dk/api/http/subscriptionstatus>

Example: <https://www.recuro.dk/api/subscriptionstatus.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.

The response has the following parameters.

Parameter	Description
result	Success if 0.
subno	The merchants unique identification of the subscription.
merchant	Identification of the merchant.
status	<i>TICKET_PENDING</i>   <i>ACTIVE_PENDING</i>   <i>ACTIVE</i>   <i>ACTIVE_MISSING</i>   <i>TICKET_FAILED</i>   <i>CLOSED</i>   <i>PAUSED</i>
created	Date of creation. Format: dd-MM-yyyy-HH:mm:ss
activated	Date of activation. Format: dd-MM-yyyy-HH:mm:ss. Activation is the

	time when state is changed to active, only returned if subscription is activated.
startdate	Startdate in format dd-MM. If not defined in creation, or subscription is activated, it will be same date as activation.
next_paydate	Date for next payment, if exists. Format: dd-MM-yyyy.
closed	Date of closing. Format: dd-MM-yyyy-HH:mm:ss.
enddate	Date of expiration Format: dd-MM-yyyy.
customer_email	Customer email.
customer_firstname	Customer first name, if defined.
customer_lastname	Customer last name, if defined.
customer_address	Customer address name, if defined.
customer_address2	Customer secondary address, if defined.
customer_postalcode	Customer postal code, if defined.
customer_city	Customer city, if defined.
customer_country	Customer country, if defined.
customer_phone	Customer phone, if defined.
customer_birthdate	Customer birth date in format dd-MM-yyyy, if defined.
customer_gender	Customer gender MALE   FEMALE, if defined.
customer_comment	Customer comment.
captured_count	The number of captured/completed payments.
captured_amount	The amount sum of captured payments.
pending_count	The number of pending payments.
pending_amount	The amount sum of pending payments.
declined_count	The number of declined payments.
declined_amount	The amount sum of declined payments.
currency	The currency of payments. Format: <a href="#">ISO-4217</a>
pending_extra_charges_count	Number of pending extra charges. See extra charges.
pending_extra_charges_amount	Amount sum of pending extra charges.
pending_discount_count	Number of pending discounts. See discount.
pending_discount_amount	Amount sum of pending discount.
pending_extra_charges	List of references to pending extra charges.
pending_extra_charges_amounts	List of amounts for pending extra charges.
pending_discounts	List of references to pending discounts.
pending_discounts_amounts	List of amounts for pending discounts.
rule	The rule attached to this subscription.

### 3.6 Close subscription

A subscription can be closed with the following call.

URL: <https://www.recuro.dk/api/http/closesubscription>

Example: <https://www.recuro.dk/api/closesubscription.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.

The response consist of `result` and `errortext` on error.

Pending payments will be tried captured. If they fail, they will be declined. If there is pending extra charges, these will scheduled as a single new payment.

### 3.7 Delete subscription

Subscriptions in state *TICKET\_PENDING*, *ACTIVE\_PENDING* and *ACTIVE\_MISSING* can be deleted to allow reuse of subscription numbers. Notice that subscriptions in other states cannot be deleted.

URL: <https://www.recuro.dk/api/http/deletesubscription>

Example: <https://www.recuro.dk/api/deletesubscription.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.

The response consist of `result` and `errortext` on error.

### 3.8 Renew subscription

At any point in time a subscription in state *ACTIVE*, *TICKET\_FAILED*, *ACTIVE\_PENDING*, *ACTIVE\_MISSING*, *CLOSED* or *PAUSED* can be renewed. To renew a subscription means that additional card information is added to a subscription. A subscription can carry multiple card informations, and will always use the newest valid card. Renew subscription can for example be called when a payment failure has been reported, if *AUTO\_UPDATE* fail model is not used.

Renew subscription is completely analogue to create subscription. The `type` parameter is the same as for create subscription.

If renew subscription is used on a closed subscription, the subscription will be reopened and a new activation date set.

URL: <https://www.recuro.dk/api/http/renewsubscription>

Example: <http://www.recuro.dk/api/renewsubscription.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.

[type]	AN	Same as for create subscription.
[returnurl]	AN255	An optional new returnurl. If not given the one from creation will be used.
[callbackurl]	AN255	An optional new callbackurl. If not given the one from creation will be used.

The result of the call is the same as for create subscription.

### 3.9 Edit subscription

A subscription can be edited through the following call.

URL: <https://www.recuro.dk/api/http/editsubscription>

Example: <https://www.recuro.dk/api/editsubscription.html>

Parameter		Description
merchant	N10	Identification of the merchant. Handed out by Recuro when the merchant is created.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
subno	AN50	The merchants unique identification of the subscription.
[rule]	N10	The rule to attach to this subscription. Rules are managed in the merchant administration or through API calls described later.
[startdate]	AN	An optional parameter that indicates that the subscription should be active from the given date. Format: dd-mm-yyyy
[enddate]	AN	An optional parameter that indicates that the subscription should only be active until this date. Format: dd-mm-yyyy
[grace_time]	N10	For create model INSTANT_ACTIVE, grace time gives the time period in minutes where it is accepted that payments cannot be approved because of missing card information.
[lang]	AN2	Language for the page where card information is entered. The format is two letter ISO-3166. As default danish is used.
[returnurl]	AN255	After the flow of entering card information there will be returned to this URL. See below for parameters.
[callbackurl]	AN255	After the flow of entering card information a HTTP POST will be performed to this URL with the result. See below for parameters. For subsequent events for the subscription this callback url will also be used. See more in the section Callback.
[email]	AN50	The customers email. Email is mandatory as the email is used for sending receipts to the customer.
[firstname]	AN50	The customer first name
[lastname]	AN50	The customer last name
[address]	AN50	The customer address
[address2]	AN50	Additional address information for the customer.
[city]	AN50	The customers city
[postalcode]	AN10	The customers postal code
[phone]	AN20	The customers phone
[birthdate]	AN	The birthdate of the customer. Format: dd-mm-yyyy.

[gender]	AN1	Customer gender. Male (M) or female (F).
[comment]	AN	A comment about the customer.

The response consist of `result` and `errortext` on error.

### 3.10 Pause subscription

A subscription can be paused for a known or unknown duration. Pausing a subscription will not adjust the amount of payments, but will only stop payment scheduling while the subscription is on pause. Therefore the duration of a paused is recommended to be a multiple of the interval between payments. For example one month. For some rule types a shorter pause makes sense. See the description of rule types for further details.

URL: <https://www.recuro.dk/api/http/pausesubscription>

Example: <http://www.recuro.dk/api/pausesubscription.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
[from]	AN	When to pause from. Default now. This date is inclusive in the pause, so payments are not scheduled on this day, except if a payment has already been scheduled. Format: dd-mm-yyyy
[to]	AN	When to pause to. Default eternity. This date is not included in the pause, so payments will be scheduled again on this date. Format: dd-mm-yyyy
[duration]	N	A multiple of the subscription rule duration. For example 2 months.
[cancel]	AN	If any value given for this parameter, all other parameters will be ignored, and pause period in future cancelled.

The response consist of `result` and `errortext` on error.

### 3.11 Resume subscription

A paused subscription can instantly be resumed by a call to resume subscription.

URL: <https://www.recuro.dk/api/http/resumesubscription>

Example: <http://www.recuro.dk/api/resumesubscription.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
[from]	AN	When to resume from. Default now. Format: dd-mm-yyyy

The response consist of `result` and `errortext` on error.

### 3.12 On demand

A manual payment for a subscription can be forced with an on demand call.

URL: <https://www.recuro.dk/api/http/ondemand>

Example: <http://www.recuro.dk/api/ondemand.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
[duedate]	AN	The payment should be scheduled on this date. Format: dd-mm-yyyy
[amount]	N10	Amount for payment. The default is the amount defined for the subscription rule.
[currency]	AN3	Currency for the payment. The default is the currency defined for the subscription rule. Format: <a href="#">ISO-4217</a>

The response consist of `result` and `errortext` on error. On success an addition `payment` parameter will be returned. The value is a reference to the newly created payment.

### 3.13 Extra charges

It is possible to add extra charges to an active subscription. The extra charges will be added to the next scheduled payment, or will result in a single payment if the subscription is closed.

URL: <https://www.recuro.dk/api/http/addextracharge>

Example: <http://www.recuro.dk/api/addextracharge.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
amount	N10	Amount for extra charge
ordertext	AN	A description of the extra charge. Will be on the payment receipt sent to customer. May not contain character  .

The response consist of `result` and unique reference `extra_charge`, or `errortext` on error.

### 3.14 Cancel extra charge

It is possible to cancel an extra charge as long as it is pending.

URL: <https://www.recuro.dk/api/http/cancelextracharge>

Example: <http://www.recuro.dk/api/canelextracharge.html>

Parameter	Description
-----------	-------------

merchant	N10	Identification of the merchant.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
extra_charge	N	Reference to extra charge.

### 3.15 Discount

As it possible to add extra charge, it is also possible to add a discount to a subscription. The discount will be drawn from the amount of the next scheduled payment. If the discount is larger than rule amount, the remaining discount will be forwarded. If there is still remaining discount when a subscription stops, the discount will be lost.

URL: <https://www.recuro.dk/api/http/discount>

Example: <http://www.recuro.dk/api/discount.html>

Parameter		Description
merchant	N10	Identification of the merchant.
subno	AN50	The merchants unique identification of the subscription.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
amount	N10	Discount amount
description	AN	A description of the discount. Will be on the payment receipt sent to customer. May not contain character  .

The response consist of `result` and unique reference `discount`, or `errortext` on error.

### 3.16 Cancel discount

It is possible to cancel a discount as long as it is pending.

URL: <https://www.recuro.dk/api/http/canceldiscount>

Example: <http://www.recuro.dk/api/canceldiscount.html>

Parameter		Description
merchant	N10	Identification of the merchant.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
discount	N	Reference to discount.

## 4 Rules

Rules can be managed in merchant administration as shown in Illustration 4 shown below.

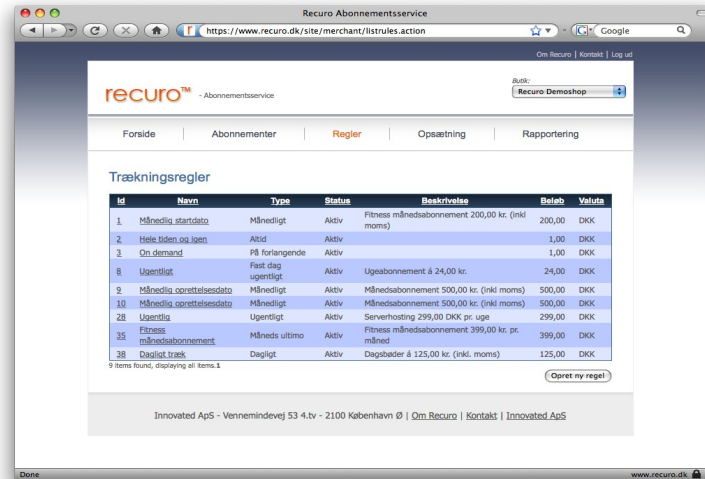


Illustration 4: Rules management in merchant administration

Rules can also be managed through a number of API calls described in this section.

### 4.1 Create rule

A new rule can be created with the following call.

URL: <https://www.recuro.dk/api/http/createrule>

Example: <https://www.recuro.dk/api/createrule.html>

Parameter		Description
merchant	N10	Identification of the merchant.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
ruletype	AN20	The type of rule: <i>ONDEMAND</i> , <i>WEEKLY</i> , <i>MONTHLY</i> , <i>MONTHLY_CREATED</i> , <i>WEEKLY_CREATED</i> , <i>MONTH_ULTIMO</i> , <i>MONTH_PRIMO</i> , <i>QUARTERLY_ULTIMO</i> , <i>QUARTERLY_PRIMO</i> and <i>DAILY</i> .  The rule types <i>MONTHLY_CREATED</i> and <i>WEEKLY_CREATED</i> will schedule payments on the day of month of activation and day of week of activation, respectively.
name	AN50	A descriptive name for the rule.
description	AN	A description of the rule. This will be shown in the terms when a customer attaches card information to a subscription.

amount	N10	Amount for payments.
currency	AN3	Currency for payments. Format: <a href="#">ISO-4217</a>
[vat]	N10	Vat in whole percent. Default is 25.
[day]	N3	For rule types with a variable day of month or week, this parameter will indicate this day. The week days monday ... sunday, are numbered 1 ... 7. Month days can be 1 ... 28.
[runtime]	N10	Indicates the number of days that a subscription should be active.
[maxcount]	N10	Indicated the number of payments for a subscription. For example 12 and the rule type is monthly, then the subscription will be active for one year.
[autoupdate_max_count]	N10	If auto update fail model is used, this parameter defines the number of renewal requests to send.
[autoupdate_interval]	N10	If auto update fail model is used, this parameter defines the interval between renewal requests.
[terms]	AN	This parameter can be used to add special terms to this rule. The terms can be shown on the terms page. See templating section.

## 4.2 Rule types

A description of the possible rule types.

Rule type	Description
<i>ONDEMAND</i>	No payments will automatically be scheduled. Payments can only be created with the on demand API call.
<i>WEEKLY</i>	A payment is scheduled each week. The <code>day</code> parameter will determine the day of week.
<i>MONTHLY</i>	A payment is scheduled each month. The <code>day</code> parameter will determine the day of month.
<i>WEEKLY_CREATED</i>	A payment is scheduled each week on the day where the subscription was activated. If a subscription is paused for a period that is not a multiple of one week, the day of week will move.
<i>MONTHLY_CREATED</i>	A payment is scheduled each month on the day where the subscription was activated. If a subscription is paused for a period that is not a multiple of one month, the day of month will move. If activation day does not exist for a given month, for example the 30 <sup>th</sup> of february, the last day of month will be used.
<i>MONTH_ULTIMO</i>	A payment is scheduled on the last day of month.
<i>MONTH_PRIMO</i>	A payment is scheduled on the first day of month.
<i>QUARTERLY_ULTIMO</i>	A payment is scheduled on the last day of a quarter.
<i>QUARTERLY_PRIMO</i>	A payment is scheduled on the first day of a quarter.
<i>DAILY</i>	A payment is scheduled daily.

## 4.3 Edit rule

A rule can be edited with the following call.

URL: <https://www.recuro.dk/api/http/editrule>

---

Example: <https://www.recuro.dk/api/editrule.html>

Edit rule takes the same parameters as create rule, except for the extra parameter `rule`, which refer to the rule, and that all data parameters are optional. Only defined parameters will be updated.

#### 4.4 Erase rule

A new rule can be created with the following call.

URL: <https://www.recuro.dk/api/http/eraserule>

Example: <https://www.recuro.dk/api/eraserule.html>

Parameter		Description
rule	N10	Identification of the rule.
merchant	N10	Identification of the merchant.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.

The response consist of `result` and `errortext` on error.

## 5 Callback

The `callback` parameter used when creating a subscription will be used for subsequent notification of events for the subscription. Additional callback url can be defined in the merchant administration. This callback url will be the same for all subscriptions.

Callbacks are HTTP POST requests. The calls will retry until a 200 OK response is received, or one day passes.

### 5.1 Card information

When card information is tried added to a subscription, the following callback will be made.

Parameter	Description
subno	The merchants unique identification of the subscription.
merchant	The identification of merchant.
action	Will in this case be fixed as <b>cardinfo</b> .
[cancel]	If this parameter has a value, the operation of attaching card information to the subscription has been cancelled by the cardholder.
result	Error code 0 indicates success.
[errortext]	If an error has occurred, an error text will be given.

### 5.2 Payment approved

When a payment has been approved the following callback will be made.

Parameter	Description
subno	The merchants unique identification of the subscription.
merchant	The identification of merchant.
action	Will in this case be fixed as <b>payment_captured</b> .
captured	Date of capture. Format: dd-MM-yyyy-HH:mm:ss.
payment	Unique identification of the payment.
amount	Amount of captured payment
currency	Currency of captured payment
rule	Possible rule responsible for scheduling the payment. If not given the payment is an ondemand payment.
extra_charges	List of references to extra charges added to this payment. The list is a comma separated string, e.g. <code>extra_charges=234,455,566</code> .
extra_charges_amounts	List of amounts for extra charges added to this payment. This list is comma separated and in the same order as the references.
extra_charges_descriptions	List of descriptions for extra charges added to this payment. This list is separated with character   and in the same order as the references.
discounts	List of references to discounts added to this payment. The list is a comma separated string, e.g. <code>discounts=234,455,566</code> .
discounts_amounts	List of amounts for discounts added to this payment. This list is comma

	separated and in the same order as the references.
discounts_descriptions	List of descriptions for discounts added to this payment. This list is separated with character   and in the same order as the references.

### 5.3 Payment failed

When a payment fails the following callback will be made.

Parameter	Description
subno	The merchants unique identification of the subscription.
merchant	The identification of merchant.
action	Will in this case be fixed as <b>payment_failed</b> .
payment	Unique identification of the payment.
amount	Amount of failed payment
currency	Currency of failed payment
rule	Possible rule responsible for scheduling the payment. If not given the payment is an ondemand payment.
extra_charges	List of references to extra charges added to this payment. The list is a comma separated string, e.g. extra_charges=234,455,566.
extra_charges_amounts	List of amounts for extra charges added to this payment. This list is comma separated and in the same order as the references.
discounts	List of references to discounts added to this payment. The list is a comma separated string, e.g. discounts=234,455,566.
discounts_amounts	List of amounts for discounts added to this payment. This list is comma separated and in the same order as the references.

### 5.4 Update failed

When renewal requests to customer have not resulted in new valid card information, the following call will be made.

Parameter	Description
subno	The merchants unique identification of the subscription.
merchant	The identification of merchant.
action	Will in this case be fixed as <b>update_failed</b> .
pending_count	Number of pending payments
amount $n$	Amount of the $n$ 'th pending payment
currency $n$	Currency of the $n$ 'th pending payment

## 6 Reports

Callback is one way to keep the merchant system in sync with Recuro. Another way is through the request of reports, giving changes to subscriptions and payments.

### 6.1 Subscription report

A subscription report contains information on subscriptions matching given search criteria.

URL: <https://www.recuro.dk/api/http/subreport>

Example: <https://www.recuro.dk/api/subreport.html>

Parameter		Description
merchant	N10	Identification of the merchant.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.
[subno]	AN50	Optional subscription number. If given the report will only be for this subscription.
[status]	AN	The following states can be used: TICKET_PENDING, ACTIVE, ACTIVE_PENDING, ACTIVE_MISSING, TICKET_FAILED, PAUSED and CLOSED.
[created_from]	AN	Subscriptions created after this date.
[created_to]	AN	Subscriptions created before this date.
[rule]	N	Subscriptions with this rule.
[email]	AN	Subscriptions with this customer email. Parts of email can be used.
[name]	AN	Subscriptions with this customer name. Parts of name can be used.
[address]	AN	Subscriptions with this customer address. Parts of address can be used.
[phone]	AN	Subscriptions with this customer phone number. Parts of phone number can be used.
[card]	AN	Subscriptions with card attached. A search can be done on the first six and last four digits. More criteria can be separated by space. Example "501910 0000".

### 6.2 Payment report

A payment report contains information about payments, created, pending, captured or failed for a given period.

URL: <https://www.recuro.dk/api/http/payreport>

Example: <https://www.recuro.dk/api/payreport.html>

Parameter		Description
merchant	N10	Identification of the merchant.
username	AN100	API username. Controlled in merchant administration.
password	AN100	API password.

[subno]	AN50	Optional subscription number. If given the report will only be for this subscription.
[startdate]	AN	Payments created/captured/failed on or after this date. Format: dd-mm-yyyy. Default today.
[enddate]	AN	Payments created/captured/failed on or before this date. Format: dd-mm-yyyy. Default today.
[status]	AN	The state of payments. The following values can be used: <i>CREATED</i> , <i>PENDING</i> , <i>CAPTURED</i> and <i>FAILED</i> . Default is <i>CAPTURED</i> .
[type]	AN	The output format. <i>CSV</i> and <i>XML</i> can be used. <i>CSV</i> is default.

### 6.2.1 CSV format

Example of CVS format.

```
subno,sub_status,pay_id,pay_amount,pay_currency,pay_captured,pay_created
sub005,ACTIVE,400779,CAPTURE_APPROVED,1000,DKK,21-01-2009-01:31,21-01-2009-00:00
sub005,ACTIVE,400780,CAPTURE_APPROVED,2000,DKK,21-01-2009-01:31,21-01-2009-00:00
sub006,ACTIVE,400781,CAPTURE_APPROVED,2000,DKK,21-01-2009-01:53,21-01-2009-00:00
sub006,ACTIVE,400782,PENDING,2000,DKK,,21-01-2009-00:00
```

```
subno,amount,currency,count
sub005,3000,DKK,2
sub006,4000,DKK,2
```

```
amount,currency,count
7000,DKK,4
```

The report has three parts separated by a blank line. The first part contains all payments. The second part contains totals for each subscription. The third part is totals for all subscriptions.

### 6.2.2 XML format

The XML version has the same structure as the CSV version. Below the same example in XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<payreport>
  <payments>
    <payment pay_amount="1000" pay_captured="21-01-2009-01:31" pay_created="21-01-2009-00:00"
pay_currency="DKK" pay_id="400779" pay_status="CAPTURE_APPROVED" sub_status="ACTIVE" subno="sub005"/
  >
    <payment pay_amount="2000" pay_captured="21-01-2009-01:31" pay_created="21-01-2009-00:00"
pay_currency="DKK" pay_id="400780" pay_status="CAPTURE_APPROVED" sub_status="ACTIVE" subno="sub005"/
  >
    <payment pay_amount="2000" pay_captured="21-01-2009-01:53" pay_created="21-01-2009-00:00"
pay_currency="DKK" pay_id="400781" pay_status="CAPTURE_APPROVED" sub_status="ACTIVE" subno="sub006"/
  >
    <payment pay_amount="2000" pay_captured="" pay_created="21-01-2009-00:00" pay_currency="DKK"
pay_id="400782" pay_status="PENDING" sub_status="ACTIVE" subno="sub006"/>
  </payments>
  <subtotal>
    <subscription amount="3000" count="2" currency="DKK" subno="sub005"/>
    <subscription amount="4000" count="2" currency="DKK" subno="sub006"/>
  </subtotal>
  <total amount="7000" count="4" currency="DKK"/>
</payreport>
```

## 7 Templates

The content of mails sent from Recuro on behalf of the merchant, and content of web pages shown, can be controlled through a number of templates where a tags can be used to insert dynamic content. The templates can be controlled in the merchant administration.

### 7.1 Mail templates

The following mail templates exists.

- Email to customer with request to add card information to new subscription.
- Email to customer with request to renew card information.
- Email receipt sent to customer when card information has been added to subscription.
- Email receipt on approved payment.

### 7.2 HTML templates

The following HTML templates exists.

- Terms page
- Receipt page
- Receipt page when adding card information fails or is cancelled

## 8 Test

Testing an integration is mainly concerned with the synchronization between merchant application and Recuro. Below a short description is given on how to test subscription creation and callbacks.

### 8.1 Creation and approved

Creating a subscription can be done with a test account. The following test card can be used:

5019 1000 0000 0000 exp: 06/24 cvc: 684

When the subscription has been created, payments will be scheduled. To make payments schedule right away use *WEEKLY\_CREATED* or *MONTHLY\_CREATED* rule type. For these types a payment will be scheduled within a few minutes.

This allows testing creating a subscription, activation callback and payment approved callback.

### 8.2 Failure

To test failure a button has been added in the merchant application when showing test subscriptions. This button will invalidate the attached card information.



The screenshot shows a table titled "Kortoplysninger" with the following data:

Id	Ticket	Oprettet	Udløbsdato	Aktiv	Fejlet	Fejlkode
45	197686792	16/03-2009 09:40		true		

Below the table, it says "One item found. 1". To the right of the table is a button labeled "Inaktiver kortoplysninger".

*Illustration 5: Inactivate card information button*

Invalidating card information before the payment is tried approved will make the payment fail. This allows testing failed payment callback.

Testing auto update takes a little longer time. Use low number of renewal request emails, and an interval of one day.