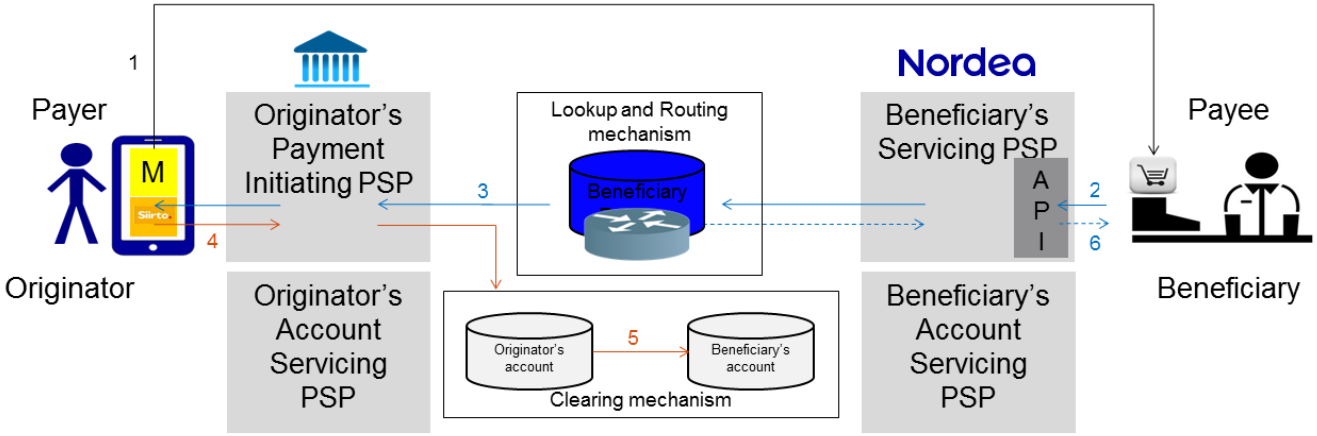
Nordea

(06.2020)

Nordea Siirto for Corporates: API specification



1. Originator shares his/her Siirto Proxy identifier to Beneficiary *, and initiates a purchase through a Merchant-app or a web-browser.
2. The Beneficiary initiates a request-for-payment, addressed to the Originator's Proxy.
3. The request-for-payment is routed to the Originator for payment.
4. The Originator initiates the payment, as defined in the request-for-payment message.
5. Funds are immediately transferred from Originator's bank account to Beneficiary's bank account
6. Beneficiary is notified of received funds

*) eg. typing in to webshop,, etc.

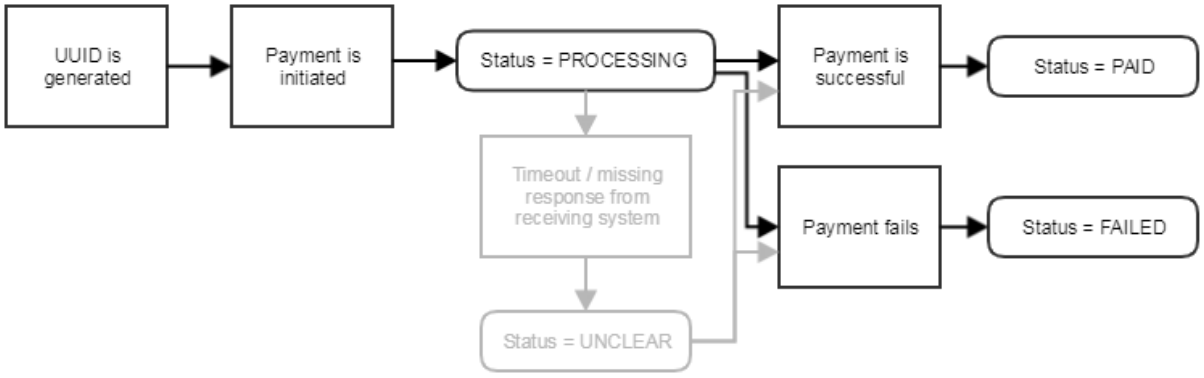
Use cases

The API covers the following main use cases:

1. Check if a specific proxy (phonenumber) is registered in Siirto
2. Sending a payment request to a customer using Siirto
3. Receiving a notification of a completed Siirto-payment (or of a declined Siirto-payment request)
4. Query the status of a payment request
5. Cancelling of a pending Siirto-payment request
6. Sending a Siirto-payment to the customer:
   1st step: Lookup of the receiver based on the Proxy
   2nd step: Payment details, initiation, and confirmation of the payment
7. Refunding a received Siirto-payment to the original payer
   1st step: Lookup of the receiver based on the previous archiving reference ID
   2nd step: see above
8. Realtime payments to a account (realtine IBAN-payments)
   Payments to banks / accounts not supporting realtime payments (fallback IBAN-payments)
9. Query payment status of a Siirto- or IBAN-payment
10. Age or identifier check when initiating a Siirto-payment, sending a Siirto-payment request, or initiating a IBAN-payment to a Nordea account.

Payment states

Each payment initiation shall be assigned a unique UUID (see chapter 8.1). Throughout the payment processing the state of the initiated payment can be determined by using this UUID as the query identifier (see chapter 9). A re-attempt to initiate a payment with a previously used UUID will be prevented as a duplicate. There is an interim state *processing,* and the final states are either *paid* or *failed*.



The exception state *unclear* will only be set if the status couldn't be reliably determined. The *unclear* state may later be updated to *paid* of *failed* - if further information becomes available.

API Authentication

Authentication is implemented with Oauth2 JWT tokens. Corporate requests authentication token with credentials provided to them by Nordea. Corporate adds the acquired token to HTTP Authorization header. Authorization header example value: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Payment notification authentication

Corporate payment notifications are signed by Nordea services using HMAC-SHA1 algorithm. Shared secret provided by Nordea is used for calculation of the checksum. The checksum value is added to the HTTP Authorization header by Nordea Siirto for Corporates services. The request date is added to HTTP Date header in RFC 1123 format by Nordea Siirto for Corporates services.

Field definitions

Data	Definition
Time stamps	ISO 8601 including time zone information - eg 2016-11-11T08:39:05.123Z
Mobile phone number	International format - eg +358501234567
SIIRTOID	SiirtoID format: BusinessID in international format + optional suffix (total length is max 30 chars). Examples: FI12345678, FI123456780001, FI12345678KAUPPA
Proxy Type	The proxy type is either PHONE or SIIRTOID.
Reference number	Payment reference in Domestic Finnish or international RF-format (Structured Creditor Reference ISO 11649). Examples: 12345678901234567894, RF091234567890123456789 4
Payment message	Maximum 140 characters. Supported characters ISO-646-FI / SFS 4017 (7-bit ASCII + ÅÄÖ) charset UTF-8.
Description	Maximum 2000 characters, supported charset UTF-8.
Amount	Amount without decimal sign.
Currency	Three letter currency code (ISO 4217) - eg EUR
Business ID	Business identifier in international format (country code + identifier without space and hyphenation). eg FI12345678
Industrial classification code	5 digit Industrial classification code. According to Tilastokeskus Toimialaluokitus TOL 2008 (NACE Rev. 2. 2008). eg 47912 "Retail sale via mail order houses or via Internet: Vaatteiden postimyynti ja verkkokauppa"
Country Code	ISO 3166-1 alpha-2 country code. eg FI

Siirto API endpoints

Acquire an authentication token

`POST /auth`

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-Type	M	Content-Type of the request body: application/x-www-form-urlencoded

Request parameters will be sent in request body using application/x-www-form-urlencoded Content-Type.

Parameter name	Mandatory	Description
grant_type	M	Grant type for the authorization request. Currently supported value is 'password'
username	M	The username provided for the merchant
password	M	The password provided for the merchant
client_id	M	Same as the username

Note. The url addresses in the examples are testenvironment addresses.

Example request

```
curl -X POST --header 'Accept: application/json' --header 'Content-Type:
application/x-www-form-urlencoded' --data
"grant_type=password&username=ME000000012345678&password=8a3191ff-3b6e-46d6-
b2a2-4e3c81986c8d&client_id=ME000000012345678"
https://merchant.trescomas.express/auth
```

Response will be returned as JSON object.

Parameter name	Mandatory	Description
access_token	M	Access token
expires_in	M	Validity time of the access token. Expiration time in seconds.

Example JSON response

```
{
  "access_token": "eyJhbGciOiJSUzI1NiJ9......HM8jIhCyKHET2rB0cqCl1Knw",
  "expires_in": 899
}
```

1 Proxy-check: Check if a specific proxy (phonenumber) is registered in Siirto

Prior initiating a Siirto payment or sending a Siirto payment request, it is possible to check if a specific PHONE proxy is registered in Siirto. If the proxy can not be found in Siirto, then it is not possible to initiate a Siirto-payment or to send a Siirto-payment requests to that specific proxy / user.

```
GET /lookup/proxy-status/PHONE/{proxyid}
```

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-type	M	The MIME type of the body of the request. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Request parameters will be sent as HTTP path parameters.

Parameter name	Description
proxyId	Proxy of the payment recipient (ie. Phone number)

Proxy status Response

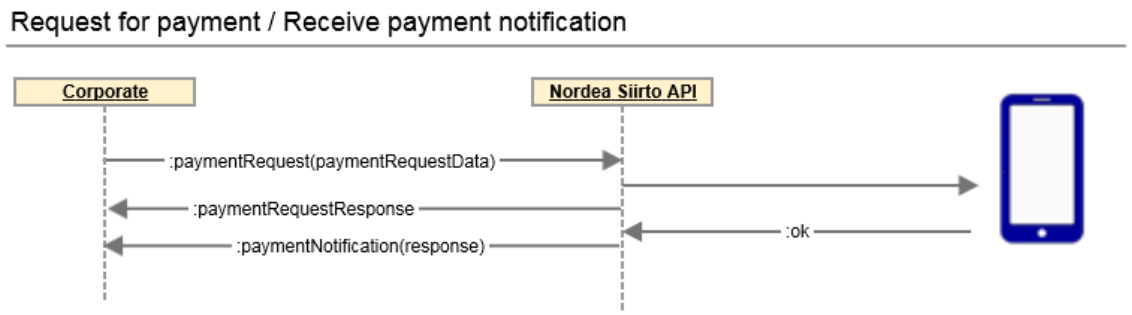
Response will be returned as JSON object.

Parameter name	Mandatory	Description
userRegistered	M	true / false

Example JSON response

```
{ "userRegistered": true }
```

2 Sending a Siirto-payment request



The payment request is delivered to the payer for confirming the payment. The payer can not edit the details in the payment request.

The payment request is assigned a unique identifier. The corporate can prior sending the payment request perform a lookup for the *paymentRequestId* through the uuid lookup endpoint (/lookup/uuid, for details see chapter 8.1). If the *paymentRequestId* is not included in the payment request, the service will generate one.

`POST /payment-request/create`

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-type	M	The MIME type of the body of the request. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Request parameters will be sent as JSON object in the HTTP message body.

Parameter name	Mandatory	Description
paymentRequestId	O	The uuid identifier of the payment request. The parameter is optional, but if the parameter is provided, then the uuid must be fetched with the /lookup/uuid (for details see section 8.1). If the parameter is not provided, then the service will generate a uuid for the payment request.
proxyId	M	Proxy of the recipient of the payment-request, the payer.
proxyType	M	Type of the proxy.
amount	M	The amount of the payment, without decimal sign.
currency	M	The currency of the payment. Currently supported 'EUR'

expires	O	Timestamp when the payment request expires, unless the payment has been completed or rejected. The payment request can be set to expire in anytime between current time and maximum expire time. The maximum expire time is 38 days from current time. If expires-parameter is not provided, then the payment request will be set to be valid for the entire default expire time. The default expire time is one month from the current time. Note. If the payment request expires according to the timestamp set in the *expires* parameter, then there will be no notification.
referenceNumber	O	Structured payment reference from the Originator to the Beneficiary. Reference will be included in the bank account statement.
paymentMessage	O	Payment message from the Originator to the Beneficiary. Message will be included in the bank account statement.
description	O	Additional information about the payment, a message from the Beneficiary to the Originator. Eg. name of the product that will be purchased. This message will not be visible on the account statement.
recurringRequestId	O	Indicator of a recurring payment request, and that payment requests with the same recurring identifier belongs to a series of payment requests. Enables the receiver to indicate and process this payment request as a recurring payment request. Any numeric value, max 20 digits.
backLinkUrl	O	Return-link to the merchant's app. After successful payment, the Siirto-payment enabled app calls this link to return to the merchant's app.
originatorMinimumAge	O	Additional age check. See chapter 10.
originatorIdentifier	O	Additional identifier check. See chapter 10.

Example request

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' --header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNRTAwM...' -d '{
 "paymentRequestId": "3d838bc9-0365-4e9e-ac23-57e8842b4bd2",
 "proxyId": "+358509999991",
 "proxyType": "PHONE",
 "amount": 10,
 "currency": "EUR",
 "expires": "2017-09-15T11:24:47.525Z",
 "referenceNumber": "RF09123456789012345678 94",
 "backLinkUrl": "https://mywebshop.com",
 "paymentMessage": "Salad Spinner GLX Deluxe 9000",
 "description": "Test company payment request"
}' 'https://merchant.trescomas.express/payment-request/create'
```

Response will be returned as JSON object.

Parameter name	Mandatory	Description
status	M	The status of the created Siirto-payment request.
paymentRequestId	M	The id for the payment request.
deepLinkUrl	O	Link to the payment request in the user's Siirto-payment enabled app. By calling this link, the Corporate can call the user's Siirto-payment enabled app and guide the user directly to the payment request. (If backlink-url was provided in the payment request, then the backlink-url is appended in the responded deeplink url.)
timestamp	M	The timestamp when the request was created. (ISO 8601)

Example JSON response

```
{
  "status": "PENDING",
  "paymentRequestId": "51fcf71e-c6de-4d6d-9d58-f41c2684d95f",
  "deepLinkUrl": "https://siirto.nordea.fi/paymentrequest/51fcf71e-c6de-4d6d-
9d58-f41c2684d95f?backlink=https://mywebshop.com",
  "timestamp": "2017-03-06T12:47:21.185Z"
}
```

Response code	Description
200	The Siirto-payment request was succesfully sent.
400	Bad request parameters in payload. Additional explanation may be included, eg "Bad Request - Invalid expiry date 2017-09-15T11:24:47.525Z" If additional checks were used, see also chapter 10.
404	ProxyId was not found from registry
5xx	Failed

3 Receiving a notification of a received Siirto-payment

Corporate provides a REST service for receiving notifications of received Siirto-payments. Nordea Siirto for Corporates service execute a HTTP POST request against this endpoint when the payment is either successfully completed, or if a payment request has been declined.

Notifications will be sent for both payments initiated by customer and for payments initiated based on a payment request issued by the corporate.

If initial notification request fails, it will be re-tried two times with 30 second delay.

```
POST <corporate URL>
```

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-type	M	The MIME type of the body of the request. (JSON)
Date	M	Date in RFC 1123 format. Added by Nordea Siirto for Corporates services when creating the callback request.
Authorization	M	Request signature

Request signature

Request signature

```
Authorization = Base64( HMAC-SHA1( YourSecretAccessKeyID, ASCII-Encoding-Of(
StringToSign ) ) );

StringToSign = HTTP-Verb + "\n" +
 Content + "\n" +
 Content-Type + "\n" +
 Date + "\n" +
 merchant-url;
```

The authorisation signature is using standard Base64 encoding (no character modification, and padding character is included, RFC 4648)

See example implementation at the end of the document.

Callback payload will be sent as JSON object in the HTTP message body.

The order of properties in the payload is listed below. Properties with null values are omitted completely.

Field name	Mandatory	Description
status	M	Status indicator: • PAID - Notification of received payment • DECLINED - is an indication that the payment request was declined by the receiver / the payer. • ERROR - is an indication that the delivery of the payment request was unsuccessful. Note. If the payment request expires according to the timestamp set in the *expires* parameter, then there will be no notification.
timestamp	M	Timestamp when the action was completed.
paymentRequestId	M / -	The uuid identifier that was created for the payment request. Mandatory if payment is based on a payment request.
archiveReference	O (M if status = PAID)	The archiving reference of the successful payment. This reference will be visible on the account statement.
payer	O (M)	Proxy id of the payer.
payerName	O (M)	Name of the payer.
payee	O (M)	Proxy id of the payee.
amount	O (M)	The amount of the payment, without decimal sign.
currency	O (M)	The currency of the payment.
referenceNumber	O	Structured payment reference from the Originator to the Beneficiary. Reference will be included in the bank account statement.
paymentMessage	O	Payment message from the Originator to the Beneficiary. Message will be included in the bank account statement.
description	O	Additional information about the payment, a message from the Originator to the Beneficiary. This message will not be visible on the account statement.
fallbackPayment	O (M if true)	false or *fallbackPayment* parameter not provided = realtime payment solutions was used. true = fallback / alternative payment solution was used for processing the payment.
ultimateOrigRefName	O	Name of the Ultimate Originator.

Example JSON callback content


```
{
 "status": "PAID",
 "timestamp": "2016-11-11T08:40:02.223Z",
 "paymentRequestId": "3d838bc9-0365-4e9e-ac23-57e8842b4bd2",
 "archiveReference": "16111409P00000073B",
 "payer": "+358401234567",
 "payerName": "Paavo T. Kataja",
 "payee": "+358505555555",
 "amount": 1200,
 "currency": "EUR",
 "referenceNumber": "123456789012345678 94",
 "paymentMessage": "Salad Spinner GLX Deluxe 9000"
}
```

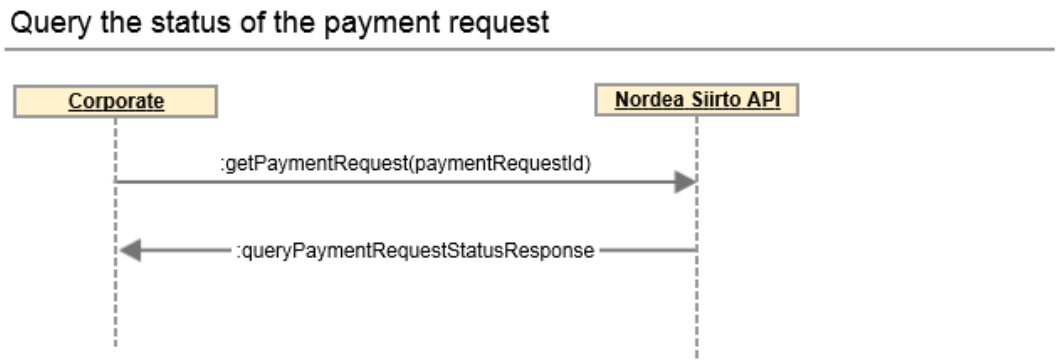

Corporate shall respond with HTTP response code. If the response from corporate is missing, then here will be two re-trials to send the notification to the corporate.

Response code	Description
200	OK
anything else	FAILED

JSON response (body)


```
{ "acknowledged": true }
```

4 Query the status of a Siirto-payment request




```
GET /payment-request/status/{paymentRequestId}
```

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-type	M	The MIME type of the body of the request. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Request parameter will be sent as HTTP path variable.

Parameter name	Mandatory	Description
paymentRequestId	M	The id of the payment request

Response will be returned as JSON object.

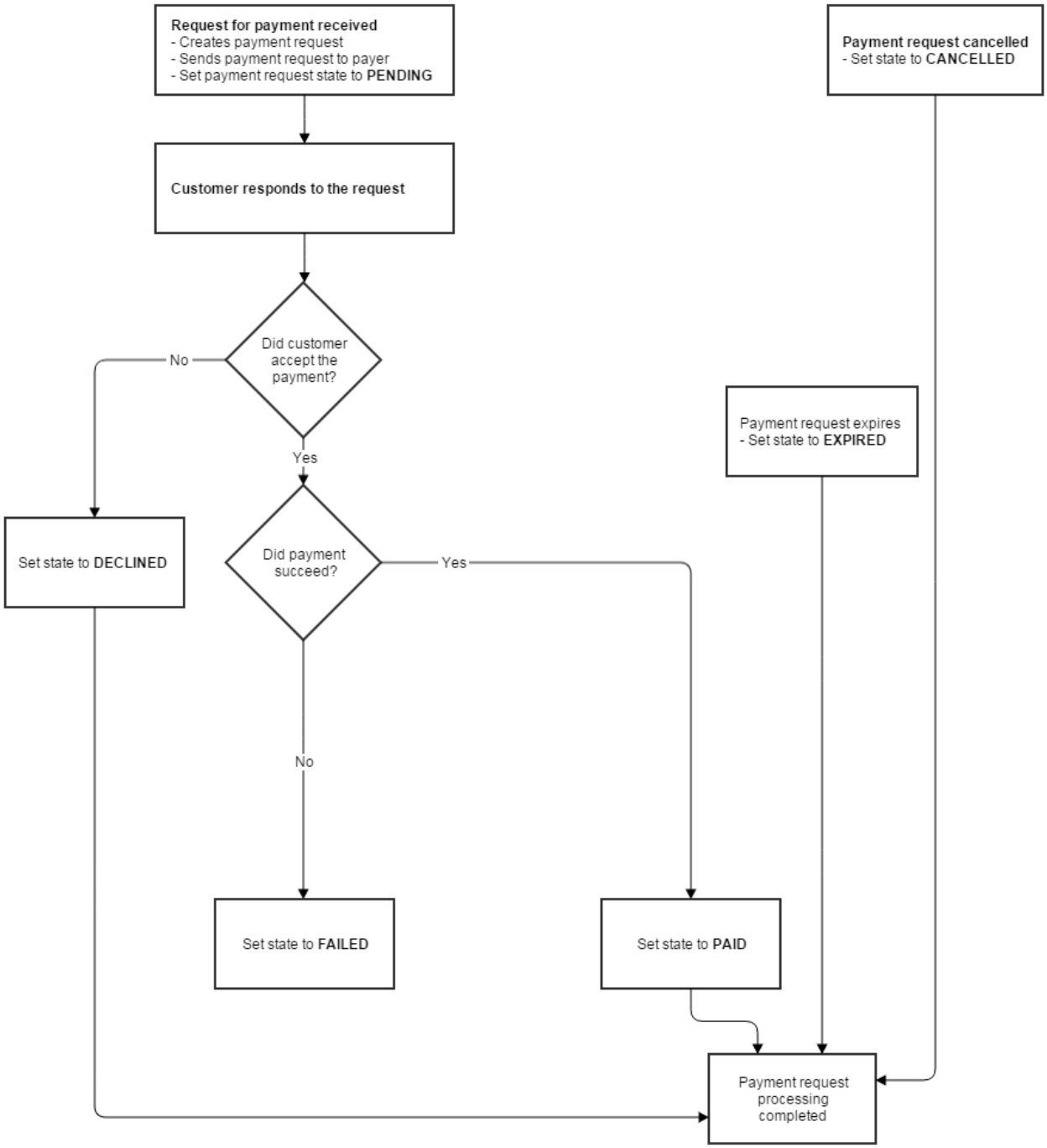
Parameter name	Mandatory	Description
status	M	Status indicator: • PAID: The payment has been completed. • PENDING: The payment request has been delivered, but the user has not yet taken any action. • FAILED: Payment request has failed. Failure is permanent. • ERROR: An error occurred. Try again later. • DECLINED: The payment request has been declined by the payer. • CANCELLED: The requestor has cancelled the payment request. • EXPIRED: The payment request has expired, user did not take any action within the expiration time. • UNCLEAR: The payment request has been sent, but there was no response received (eg timeout). • DEFERRED: tbd - for future use
timestamp	M	The timestamp (ISO 8601) when the payment request was created or if the status is PAID the timestamp the payment was completed.
paymentRequestId	M (if success)	The id of the payment request.
archiveReference	M (if status=PAID)	The archiving reference of the completed payment.

Examples of JSON responses

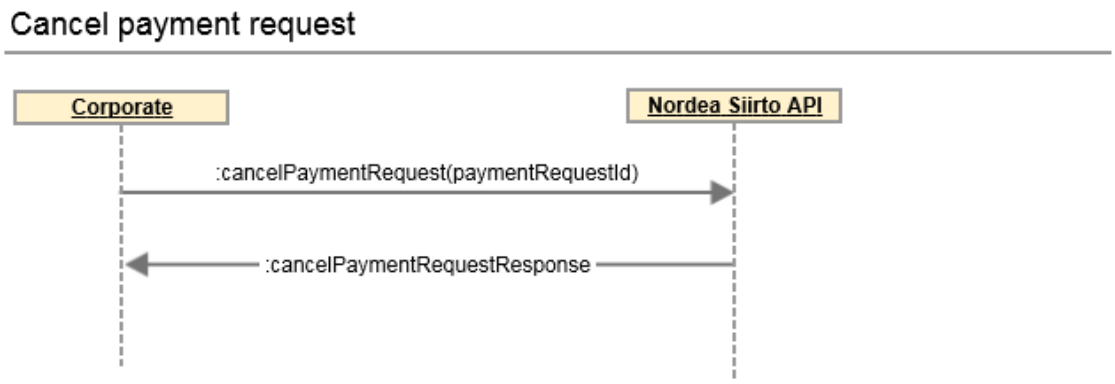
```
{
  "status": "PENDING",
  "timestamp": "2016-11-11T08:39:06.321Z",
  "paymentRequestId": "3d838bc9-0365-4e9e-ac23-57e8842b4bd2"
}

{
  "status": "PAID",
  "timestamp": "2017-10-18T15:33:21.321Z",
  "paymentRequestId": "df3b82bb-9744-496c-8d03-20892e1cd6c4",
  "archiveReference": "18102588MMMI0002"
}
```

Siirto-payment request states



5 Cancelling a pending Siirto-payment request




```
DELETE /payment-request/cancel/{paymentRequestId}
```

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Parameters

Parameter name	Mandatory	Description
paymentRequestId	M	The id that was created for the payment request.

Response will be returned as JSON object.

If the payment request was successfully cancelled, the http response code is 200.

If the payment request could not be cancelled, a http error code with response data will be provided.

Parameter name	Mandatory	Description
status	M	Status indicator: • CANCELLED: Payment request was successfully cancelled. • FAILED: Cancelling the payment request failed. Failure is permanent. • ERROR: An error occurred. Try again later. • DECLINED: The payment request has already been declined by the payer. • PAID: The payment was already completed, and the request can't anymore be cancelled. • EXPIRED: The payment request was already expired and can't be cancelled
timestamp	M (if success)	The timestamp when the request was cancelled. (ISO 8601)
paymentRequestId	M	The uuid identifier of the payment request.

Example JSON response

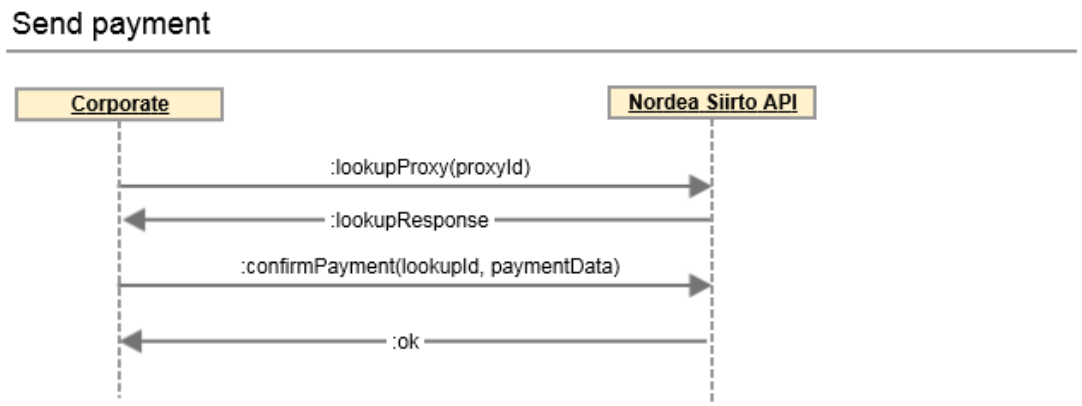
```
{
 "status": "CANCELLED",
 "timestamp": "2016-11-11T08:39:06.432Z",
 "paymentRequestId": "3d838bc9-0365-4e9e-ac23-57e8842b4bd2"
}

{
 "status": "PAID",
 "paymentRequestId": "3d838bc9-0365-4e9e-ac23-57e8842b4bd2"
}
```

Response code	Description
200	Request was successfully cancelled.
400	Bad request parameters in payload.
404	Request was not found or not pending. Check details from *status*-parameter in body.
5xx	Failed

6 Sending a Siirto-payment

The Siirto-payment procedure is two phased: first a Siirto proxy lookup is done, and in the second phase the payment is confirmed.



Prepare for a Siirto-payment by first performing a proxy lookup

Proxy lookup

Lookup PHONE / SIIRTOID proxies:

```
GET /lookup/proxy/{proxyType}/{proxyid}
```

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-type	M	The MIME type of the body of the request. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

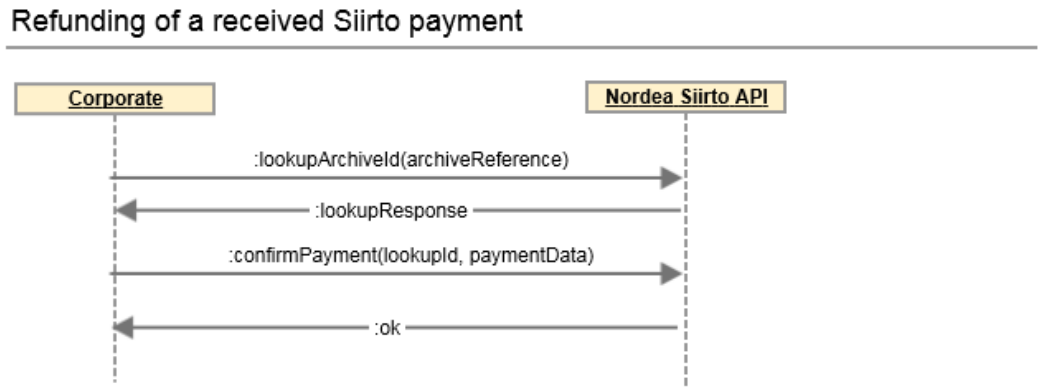
Request parameters will be sent as HTTP path parameters.

Parameter name	Description
proxyId	Proxy of the payment recipient (ie. Phone number)
proxyType	Type of the proxy (eg PHONE)

Response will be returned as JSON object in format of proxy lookup / archive id lookup response (see chapter 7.1).

Confirm the payment by calling POST /payment/confirm (see chapter 7.2)

7 Look up a received Siirto-payment for refunding (Archive reference lookup)




```
GET /lookup/archive-reference/{archiveReference}
```

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-type	M	The MIME type of the body of the request. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Request parameter will be sent as HTTP path variable.

Parameter name	Mandatory	Description
archiveReference	M	The archiving reference of the payment. Possible to search for up to one year old payments.

Response will be returned as JSON object in format of proxy lookup / archive id lookup response (see chapter 7.1), same as for proxy-lookup.

Complete the refund by calling POST /payment/confirm (see chapter 7.2)

The cumulative amount of the refund(s) may be less or equal to the original payment.

7.1 Proxy Lookup / ArchiveID Lookup Response

Response will be returned as JSON object.

Parameter name	Mandatory	Description
lookupId	M	Lookup id for the payment. Valid for confirming the payment until expired, as set in the *expires* parameter in this message.
proxyId	M	Proxy of the payment recipient (ie. Phone number)
name	M	Name of the payee.
timestamp	M	The timestamp when the request was created. (ISO 8601)
expires	M	The timestamp when the lookup id expires. (ISO 8601)
lookupType	M	Type of the lookup (PROXY_LOOKUP or ARCHIVE_REFERENCE_LOOKUP)

Example JSON response

```
{
  "lookupId": "e745eab4-130e-4e1f-b7d8-b12bd37fdf91",
  "proxyId": "+35840123456",
  "name": "Company Oy",
  "timestamp": "2017-04-27T12:54:52.471Z",
  "expires": "2017-04-27T12:59:52.471Z",
  "lookupType": "PROXY_LOOKUP"
}
```

Response code	Description
200	Lookup id successfully obtained.
404	Proxy id or archive reference not found
400	Invalid parameters in the request payload

7.2 Confirm paying a Siirto-payment

`POST /payment/confirm`

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...
Content-type	M	The MIME type of the body of the request. (JSON)

Request parameters will be sent as JSON object in the HTTP message body.

Parameter name	Mandatory	Description
lookupId	M	lookup id of the payment. (In case of performing a retrial payment for a previously unsuccessful payment attempt, then the same *lookupId* shall be used in the retrial as was used in the original payment attempt.)
amount	M	The amount of the payment, without decimal sign.
currency	M	The currency of the payment.
referenceNumber	O	Structured payment reference from the Originator to the Beneficiary. Reference will be included in the bank account statement.
paymentMessage	O	Payment message from the Originator to the Beneficiary. Message will be included in the bank account statement.
description	O	Additional information about the payment, a message from the Originator to the Beneficiary. This message will not be visible on the account statement.
beneficiaryMinimumAge	O	Additional age check. See chapter 10.
beneficiaryIdentifier	O	Additional identifier check. See chapter 10.

Example JSON payload

```
{
 "lookupId": "6633118d-b69f-4e72-a613-85b98886175f",
 "amount": 1200,
 "currency": "EUR",
 "referenceNumber": "12345678901234567894",
 "paymentMessage": "Salad Spinner GLX Deluxe 9000"
}
```

7.3 Response of a completed Siirto- or IBAN-payment

Response will be returned as JSON object.

Parameter name	Mandatory	Description
status	M	Status indicator: • PAID - Notification of completed payment.
archiveReference	O	The archiving reference of the successful payment. This reference will be visible on the account statement.
fallbackPayment	O (M if true)	Indicator for payment solution used: false or *fallbackPayment* parameter not provided = realtime payment solutions was used. true = fallback / alternative payment solution was used for processing the payment.
timestamp	M	The timestamp when the request was created. (ISO 8601)

Example JSON response

```
{
 "status": "PAID",
 "archiveReference": "16111409P00000073B",
 "timestamp": "2016-11-11T08:55:06.434Z"
}
```

Response code	Description
200	Instruction was successful. See the payment status from the response body *status*-parameter.
202	Payment is queued for payment. Applicable only for IBAN-payments with *fallbackPayment*=auto. Check the final payment status with the payment status query (see chapter 9)
400	Bad Request - lookupId was not found, has expired, or has already been used. / Bad request parameters in payload / Invalid payment reference number / Bad Request - insufficient_funds / Bad Request - originator_bank_declines / Bad Request - lookup id already used / Bad Request - other. If additional checks were used, see also chapter 10.
403	Payment status unclear, check payment status or investigate manually.
404	Beneficiary bank account can't be reached or it does not exist.
5xx	Failed / Temporary failure in payment. Check payment status.

These are the final states, there is no automatic re-trial to perform the payment.

8 IBAN-payment

8.1 Prepare for a IBAN-payment by first performing a UUID lookup

`GET /lookup/uuid`

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Response will be returned as JSON object.

Parameter name	Mandatory	Description
lookupId	M	LookupID for the payment. (universally unique identifier (UUID) RFC 4122)
expires	M	The timestamp when the lookup id expires. (ISO 8601)

8.2 Sending a payment using IBAN account number

`POST /payment/pay`

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Content-type	M	The MIME type of the body of the request. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Request parameters will be sent as JSON object in the HTTP message body.

Parameter name	Mandatory	Description
lookupId	M	Universally unique identifier (UUID) of the payment. The *lookupId* is generated through the GET /lookup/uuid -operation. (In case of performing a retrial payment for a previously unsuccessful payment attempt, then the same *lookupId* shall be used in the retrial as was used in the original payment attempt. Payment retrials apply primarily for payments with 5xx error.)
amount	M	The amount of the payment, without decimal sign.
currency	M	The currency of the payment.
beneAccountNumber	M	IBAN number of the Beneficiary account.
fallbackPayment	O	Use of fallback / alternative payment solutions for processing the payment: **false** (or *fallbackPayment* parameter is not provided) = only realtime payment solutions will be used. (If the realtime payment is unsuccessful, then there will be no retrial through alternative payment solutions after response has been issued.) **auto** = primarily realtime payment solution will be used for payment, with fallback to alternative payment solution. A 200 (paid) or 202 (queued) response will be issued after the initial realtime payment attempt. If the initial realtime payment attempt was unsuccessful, then there will be realtime payment retrials for 15 minutes and as a final payment attempt a fallback to alternative payment solution. If the initial realtime payment attempt was unsuccessful, then the interim status in the response is processing and the final payment status must later be queried with the payment status-query (see chapter 9). The used payment solution (realtime or fallback) will be indicated with the *fallbackPayment* parameter in the payment status-query response. **true** = fallback / alternative payment solution will be used for processing the payment. (In case of an unsuccessful payment, no retrial will be attempted after response has been issued.) Parameter type is string.

beneLastName	M (person)	Last name of the Beneficiary. This parameter is mandatory if the Beneficiary is a person.
beneFirstNames	M (person)	First names of the Beneficiary in the array. This parameter is mandatory if the Beneficiary is a person.
beneCompanyName	M (company)	Name of the Beneficiary company. This parameter is mandatory if the Beneficiary is a company.
referenceNumber	O	Structured payment reference from the Originator to the Beneficiary. Reference will be included in the bank account statement.
paymentMessage	O	Payment message from the Originator to the Beneficiary. Message will be included in the bank account statement.
beneficiaryMinimumAge	O	Additional age check. See chapter 10.
beneficiaryIdentifier	O	Additional identifier check. See chapter 10.

Message and names fields support characters defined in ISO-646-FI / SFS 4017 (7-bit ASCII + ÅÄÖ)

Example JSON payloads

```
{
  "lookupId": "01db84ee-15f8-4b82-9f77-7f6007b2ad77",
  "amount": 1000,
  "beneAccountNumber": "FI3815723500045661",
  "beneCompanyName": "Integration company Oy",
  "currency": "EUR",
  "paymentMessage": "IBAN payment",
  "referenceNumber": "RF111232"
}

{
  "lookupId": "d289afdf-29e6-4452-86d8-e61e14213d25",
  "amount": 1000,
  "beneAccountNumber": "FI3815723500045661",
  "beneLastName": "Kankkunen",
  "beneFirstNames": ["Timo","Pekka"],
  "currency": "EUR",
  "paymentMessage": "IBAN payment",
  "referenceNumber": "RF111232"
}
```

Response will be returned as JSON object. The content is same as for POST /payment/confirm. See chapter 7.3.

9 Query the payment status of a sent Siirto- or IBAN-payment

`GET /payment/payment-status/{lookupId}`

HTTP header fields	Mandatory	Description
Accept	M	Content-Types that are acceptable for the response. (JSON)
Authorization	M	The authorization token for the request. Example: Bearer eyJhbGciOiJSUzI1NiJ9 ...

Request parameters will be sent as HTTP path parameters.

Parameter name	Description
lookupId	lookupId used in Siirto-payment or lookupId in IBAN-payment payload

Response will be returned as Json object:

Parameter name	Mandatory	Description
status	M	Possible values: PAID - payment is completed successfully PROCESSING - payment is not completed yet FAILED - payment has failed UNCLEAR - payment status is unclear, check the bank account
archiveReference	O	Archive reference id for the created payment, if payment status is PAID.
fallbackPayment	O (M if true)	false or *fallbackPayment* parameter not provided = realtime payment solutions was used. true = fallback / alternative payment solution was used for processing the payment.
paymentTime	O	Time of payment (UTC), if payment status is PAID.

Response code	Description
200	Query was successful. See the payment status from the response body *status*-parameter.
404	Lookup id not found Unknown lookup identifier was used in the query. The query was attempted for a lookup identifier that do not exist in the system.

10 Additional checks

When sending a Siirto-payment request, initiating a Siirto-payment, or initiating a IBAN-payment to a Nordea account, then the age or identifier of the Beneficiary can be checked:

originatorMinimumAge or beneficiaryMinimumAge	O	Additional check criteria for delivering the payment, minimum age of person: 18 The age is checked. The payment (or -request) is delivered to the person only if the check criteria is met. Parameter value: 18
originatorIdentifier or beneficiaryIdentifier	O	Additional check criteria for delivering the payment: BusinessID or SSN identifier. The BusinessID or SSN identifier is checked. The payment (or -request) is delivered only if the check criteria is met. Parameter value: Business ID in international format or SSN incl century divider.

Note: For IBAN-payments the check is available only for payments to beneficiary accounts that are resident in Nordea (IBAN is beginning with FInn1... or FInn2...)

Response code	**Description**
400	Age check was unsuccessful (age is below criteria or it was not possible to perform the check) / Business ID or SSN identifier does not match.

Testenvironment

A Nordea Siirto API testenvironment is available, and it is accessible through:
https://merchant.trescomas.express port 443

Prerequisites for accessing the testenvironment:

- A corporate login-id and a test access-key issued by Nordea.
- A https address at the corporate to where Nordea will post the notifications. (Sender IP addresses 52.30.73.12)
- (Nordea do not need the corporates IP addresses)

Testing of Siirto-payments

A predefined action can be initiated by sending a payment request to a specific proxy. All user can receive payments.

+358509999991 "Essi Payxlviietelä"	The payment request will be paid. There will be a notification of a received payment. User is an adult (age is 18 or above). SSN is 100868-996K
+358509999992 "Vertti Paylverraton"	The payment request will be declined. There will be a notification of a declined payment request. User is an adult (age is 18 or above). SSN is 100868-993F
+358509999993 "Saara Mattila"	The payment request will not be answered, the payment request will be pending until it expires. No notification will be sent when the payment request expires. User is an adult (age is 18 or above). SSN is 030380-1007
+358509999994	The payment request will not be answered, the payment request will be pending until it expires. No notification will be sent when the payment request expires. User is a company. BusinessID is FI18040238
+358509999995	The payment request will be paid. There will be a notification of a received payment. User is underaged (age is between 15-17). SSN is 100802A999M
+358509999996	The payment request will be paid. There will be a notification of a received payment. User is underaged (age is below 15). SSN is 100805A981U

Testing of IBAN payments

Beneficiary IBAN is FI38 1572 3500 0456 61

Testuser for accessing Siirto-API testenvironment

Corporate can access the Siirto-API testenvironment for development purposes. This testuser supports:

- Acquisition of Siirto Payments
- Sending Siirto refund payments
- Sending Siirto payments
- Sending IBAN payments

Note: This testuser does not send any callback notifications of received Siirto-payments.

Siirto API username: ME35912345671
Siirto API password: ee582575-7941-4d5a-b9b9-7a5101c2e2bc

Production environment

A Nordea Siirto for Corporates API production environment is accessible through: https://merchant.mobilewalletservices.nordea.com port 443 (193.234.184.20, 193.234.184.36)

Prerequisites for accessing the production environment:

- A corporate login-id and a production access-key issued by Nordea.
- A https address at the corporate to where Nordea will post the notifications. Sender IP addresses: 158.233.246.130, 158.233.246.131, 158.233.247.130, 158.233.247.131 (158.233.246.128/27 and 158.233.247.128/27)
- For redundancy support: Identical notification may be sent also to a additional https address at the corporate.
- (Nordea do not need the corporates IP addresses)

Reference implementation: Receiving payment notification

During validation of the received notification, shall the whole body be handled as a string. As new parameters may be introduced in the future.

Authorization token from authorization request header should be validated.

Receiving payment notification

```
@RequestMapping(value = "receivePaymentNotification", method = RequestMethod.POST)
@ApiOperation(value = "Receive a payment notification and notify web shop", response = String.class)
public Single<String> receivePaymentNotification(
        @RequestBody String notification, // <-- MAKE SURE TO TAKE BODY AS STRING
        @RequestHeader(value = HttpHeaders.DATE, required = false) String requestDate,
        @RequestHeader(value = HttpHeaders.CONTENT_TYPE, required = false) String contentType,
        @RequestHeader(value = HttpHeaders.AUTHORIZATION, required = false) String token) throws
Exception {

    // First validate token
    boolean valid = service.validateToken(
            RequestMethod.POST,
            contentType,
            callbackUrl,
            notification,
            OffsetDateTime.parse(requestDate, DateTimeFormatter.RFC_1123_DATE_TIME),
            "<CallbackSecret>", // <-- REPLACE THIS WITH THE SECRET WHICH IS KNOWN TO YOU.
            token);

    // Then do something....
    if (valid) {
        PaymentNotification n = mapper.readValue(notification, PaymentNotification.class);
    }

    return just("Ok");
}
```

Reference implementation: Callback signature

Signature implementation

```
import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.time.OffsetDateTime;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.bind.annotation.RequestMethod;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import lombok.extern.slf4j.Slf4j;
import rx.exceptions.Exceptions;
@Slf4j
@Service
public class RestSignService {
    public static final DateTimeFormatter HTTP_DTF =
DateTimeFormatter.RFC_1123_DATE_TIME.withZone(ZoneId.of("UTC"));
    public static final DateTimeFormatter ISO_DTF =
DateTimeFormatter.ISO_OFFSET_DATE_TIME.withZone(ZoneId.of("UTC"));

    private static final String ASCII = "ASCII";
    private static final String HMAC = "HmacSHA1";

    @Autowired
    protected ObjectMapper objectMapper;

    public boolean validateToken(RequestMethod method,
            String contentType,
```

```
            String uri,
            Object body,
            OffsetDateTime timestamp,
            String secret,
            String tokenToValidate) {
        log.debug("Validating token: {}, {}, {}, {}  | Original date: {}", method, uri, body,
HTTP_DTF.format(timestamp), timestamp);

        byte[] bytes = getAuthorizationTokenBytes(method, contentType, uri, body, timestamp,
secret);
        String expected = Base64.encodeBase64String(bytes);

        log.debug("Token to validate: {} | expected: {}", tokenToValidate, expected);

        return MessageDigest.isEqual(bytes, Base64.decodeBase64(tokenToValidate));
    }

    public String getAuthorizationToken(RequestMethod method,
            String contentType,
            String uri,
            Object body,
            OffsetDateTime timestamp,
            String secret) {
        return Base64.encodeBase64String(getAuthorizationTokenBytes(method, contentType, uri, body,
timestamp, secret));
    }

    private byte[] getAuthorizationTokenBytes(
            RequestMethod method,
            String contentType,
            String uri,
            Object body,
            OffsetDateTime timestamp,
            String secret) {

        try {
            String content = null;
            if(body != null) {
                content = objectMapper.writeValueAsString(body);
            }

            StringBuilder b = new StringBuilder();
            b.append(method.name());
            b.append('\n');
            b.append(content);
            b.append('\n');
            b.append(contentType);
            b.append('\n');
            b.append(HTTP_DTF.format(timestamp));
            b.append('\n');
            b.append(uri);

            log.debug("Generating token for signature: '{}' \n\n secret: '{}' Original date: '{}'",
b.toString(), secret, timestamp);

            byte[] secretBytes = secret.getBytes();
            log.debug("Secret Bytes: {}", Base64.encodeBase64String(secretBytes));
            SecretKeySpec key = new SecretKeySpec(secretBytes, HMAC);
            Mac mac = Mac.getInstance(HMAC);
            mac.init(key);
            return mac.doFinal(b.toString().getBytes(ASCII));
        } catch(
                JsonProcessingException |
                InvalidKeyException |
                UnsupportedEncodingException |
                NoSuchAlgorithmException e) {
            throw Exceptions.propagate(e);
        }
    }
}
```