

Network Partner API v5

Table of Contents

1. Overview	1
1.1. General Note	1
1.2. Version information	2
1.3. Contact information	2
1.4. URI scheme	2
1.5. Tags	2
1.6. Documentation structure	3
2. Introduction	4
2.1. Scope	5
2.2. System interfaces	5
2.3. Primary network partners	6
2.4. Billing and payment process overview	6
3. General documentation	8
3.1. Design principles	8
3.1.1. Definition language	8
3.1.2. Payload	8
3.1.3. API versioning	8
3.2. Security	9
3.2.1. HTTPS	9
3.2.2. Authentication	9
3.3. Conventions	10
3.3.1. Models	10
3.3.2. Id-handling	10
3.3.3. Multilingual support	10
3.3.4. Assets	11
3.3.5. HTTP headers	11
3.3.6. Strings must conform to XML1.0 Character Subset	12
3.3.7. HTTP status codes	12
3.3.8. Error handling	12
3.3.9. Event handling	13
3.3.10. Search operations	14
3.3.11. Guidelines for polling the events	15
3.3.12. Maintenance windows	16
4. Use cases	17
4.1. Biller management	17
4.2. Business case management	17
4.2.1. eBill format	19
4.2.2. eBill format versioning	22

4.3. Subscriptions and subscription cancellations	23
4.3.1. Bill recipient driven subscription	23
4.3.2. Biller driven subscription	27
4.3.3. Subscription cancellation	29
4.4. Donation inquiries	30
4.5. eBill Direct Debit	32
4.5.1. Enabling biller for eBill Direct Debit	32
4.5.2. Submit an eBill Direct Debit proposal	33
4.5.3. Submit an eBill Direct Debit proposal (opt-in)	33
4.5.4. Submit a bill with eBill Direct Debit payment	34
5. Resources	36
5.1. Billers	36
5.1.1. Create a biller	36
5.1.2. Search billers	38
5.1.3. Get biller by id	39
5.1.4. Update biller	40
5.1.5. Get asset data by id	41
5.1.6. Add/update asset	42
5.1.7. Delete asset	44
5.1.8. Get the custom subscription form of a biller	45
5.1.9. Create or update the custom subscription form of a biller	46
5.1.10. Delete the subscription form of a biller	47
5.1.11. Search for multiple bill recipients for this biller	48
5.1.12. Create business case in PDF/A-3b-format	49
5.1.13. Get business case	51
5.2. Biller Driven Subscription	53
5.2.1. Initiate bill recipient subscription	53
5.2.2. Confirm bill recipient subscription initiation	54
5.2.3. Initiate a bill recipient subscription and return the URL to the eBill infrastructure	55
5.3. Events	56
5.3.1. Find events for business cases which changed status	57
5.3.2. Find events for instalments which changed status	58
5.3.3. Find events for bill recipient subscriptions which changed status	59
5.3.4. Find events for bill recipients email address changes	60
5.4. Sectors	62
5.4.1. Find Sectors	62
5.5. Bill-recipients	63
5.5.1. Get bill recipient by id	63
5.6. EBill Direct Debit	64
5.6.1. Propose eBill Direct Debit to the bill recipient	64
5.7. Healthcheck	65

5.7.1. Health check using GET method	65
5.7.2. Health check using PUT method	66
5.7.3. Health check using POST method	67
5.7.4. Health check using DELETE method	68
6. Definitions	70
6.1. BillerDetail	70
6.2. Biller	71
6.3. LocalizedBillerData	73
6.4. BillerAddress	75
6.5. RecipientAddress	76
6.6. BillerAccount	76
6.7. BillRecipientsForBillerSearchRequest	77
6.8. BillRecipientsForBillerSearchResponse	77
6.9. BillRecipientsForBillerSearchResponseItem	78
6.10. BillRecipientIdentification	79
6.11. BillRecipientById	80
6.12. BillRecipientSubscription	80
6.13. BillRecipientURLSubscription	80
6.14. BillRecipient	80
6.15. Bill	81
6.16. InstalmentBill	82
6.17. CreditNote	83
6.18. Reminder	84
6.19. Advice	85
6.20. DonationInquiry	86
6.21. BusinessCase	87
6.22. ReferencedBill	88
6.23. Sector	89
6.24. LocalizedSectorData	89
6.25. Event	89
6.26. BusinessCaseStatusChangedEvent	90
6.27. BillRecipientEmailAddressChangedEvent	91
6.28. InstalmentStatusChangedEvent	92
6.29. BillRecipientSubscriptionStatusChangedEvent	92
6.30. BillRecipientSubscriptionForm	94
6.31. OptionalAmountWithCurrency	94
6.32. ApprovalAmountWithCurrency	94
6.33. AmountValue	95
6.34. EbillDebitCurrenyCode	95
6.35. CurrencyCode	95
6.36. AccountNumber	95

6.37. Problem	96
6.38. HealthCheckRequest	97
6.39. HealthCheckResponse	97
6.40. DefaultLanguage	98
6.41. BillRecipientSubscriptionStatus	98
6.42. BillerSearchFilter	98
6.43. BillerSearchResult	99
6.44. SubscriptionInitiationEmailAddress	99
6.45. SubscriptionInitiationToken	99
6.46. SubscriptionInitiationActivationCode	99
6.47. MaintenanceWindow	99
6.48. BillerSubscriptionForm	99
6.49. BillerSubscriptionFormField	100
6.50. LocalizedBillerSubscriptionFormField	102
6.51. BillerSubscriptionFormChoice	102
6.52. LocalizedBillerSubscriptionFormChoice	103
6.53. LocalizedBillerSubscriptionFormInfoText	103
6.54. AnomalyDetectionConfig	104
6.55. AllowedEbillDebitSubmission	105
6.56. EbillDebitProposal	105
6.57. ChargebackMode	105
6.58. Certification	105
7. Problem Descriptions Overview	107
7.1. Basic Validations	107
7.2. Shared Validations	108
7.3. Bill Recipient Validations	109
7.4. Biller Validations	110
7.5. Asset Validations	111
7.6. Custom Subscription Form Validations	111
7.7. Custom Subscription Form Data Validations	112
7.8. Business Case Validations	112
7.9. Business Case Payment Validations	114
7.10. Business Case Instalment Validations	115
7.11. Business Case Donation Inquiry Validations	116
7.12. Business Case eBill Direct Debit Validations	117
7.13. PDF Validations	117
7.14. Subscription Validations	118
7.15. eBill Direct Debit Validations	118
7.16. Technical Errors	119
Appendix A: Roadmap API versions	120
Appendix B: Changelog	121

B.1. Changes between v4 and v5	121
B.1.1. Overview	121
B.1.2. Changes of the eBill XSD	121
B.1.3. Changes of the Swagger definitions	121
B.1.4. Changes of the Documentation	122
B.1.5. Changes of the Swagger operations	122
B.2. Changes between 5.0.5 and 5.0.10	122
B.2.1. Changes of the Documentation	122
B.2.2. Changes of the eBill XSD	122
B.2.3. Changes of the Swagger operations	122
B.2.4. Changes of the Swagger definitions	122

Chapter 1. Overview

The application programming interface for network partners (Network Partner API) allows communication with the eBill infrastructure on the "biller side". It is therefore the central entry point for billers.

The eBill service comprises of electronic bills, reminders, credit notes, advices and donation inquiries which are summarized under the term "business case". Business cases are delivered from network partners to the SIX eBill infrastructure and can be received online by bill recipients. The eBill infrastructure is the runtime system at SIX. Core functionality is the management of system participants and the processing of business cases. It consists of software and hardware needed to provide the entire service.

Documentation of the Network Partner API comprises of three parts:

Handbook for network partners

Business-level description targeting product and IT management.

Network Partner API documentation

Technical description targeting IT architects and developers.

OpenAPI Specification

Technical specification of the interface for developers and code generators.

1.1. General Note

SIX reserves the right to amend this documentation as required within the scope of the applicable contractual conditions.

All rights are reserved with respect to this documentation, including with regard to photocopying and storage on electronic media as well as translation into foreign languages.

This documentation has been compiled with the greatest care, but may nevertheless contain errors or inaccuracies. SIX cannot assume any legal responsibility or any liability for erroneous information or its consequences.

If you notice any errors in this documentation or have any suggestions for improvements, we would be grateful to receive your feedback.

Contact

SIX

eBill & Direct Debit Support

Pfingstweidstrasse 110

8005 Zürich

Switzerland

Phone: 0041 58 399 4800

1.2. Version information

Version : 5.0.10

1.3. Contact information

Contact : SIX eBill & Direct Debit Support

Contact Email : support.billing-payments@six-group.com

1.4. URI scheme

Host : api.six-group.com

BasePath : /api/pns/networkpartner/v5

Schemes : HTTPS

1.5. Tags

- billers : **Biller Management**

All operations that are associated with a biller are located within the biller resource, including the creation of business cases.

The operations are designed to be self contained, because of this, the data objects can be rather large. However this allows for complete validation and avoids chains of calls that depend on each other.

- biller driven subscription : **Biller driven subscription**

Further information about subscription processes can be found in [Section 4.3.2, “Biller driven subscription”](#).

- events : **Notification Events**

The event resource allows the network partner to retrieve all changes addressed to it. There is one specific operation for every type of event (for example: Subscription status changes or business case status changes), where the network partner can pull new changes from.

More details can be found in [Section 3.3.11, “Guidelines for polling the events”](#).

- sectors : **Industry Sector**

Industry sectors are valid system wide. Each biller will reference one or several industry sectors.

- bill-recipients : **Bill Recipients**

This resource can be used to verify the existence of a specific billRecipientId in the eBill infrastructure.

- eBill Direct Debit : **eBill Direct Debit**

Operations that are associated with eBill Direct Debit.

More details can be found in [Section 4.5, “eBill Direct Debit”](#).

- healthcheck : **System Healthcheck**

This allows to check the basic state of the system (can it be reached, does authentication and authorization work, does it respond).

As some infrastructures block certain HTTP methods by default, the healthcheck allows to test if

all required methods (GET, POST, PUT, DELETE) work across all layers.

1.6. Documentation structure

This documentation consists of the following sections:

Overview

Overview of the document.

Introduction

High-level information about the eBill service with roles, interfaces, business introduction.

General Documentation

Basic design principles, general concepts applied during API and model design like multilingual support, event and error handling.

Use cases

Describes the interactions between network partners and eBill in order to process business cases and manage system participants.

Resources

Describes the endpoints of the API. Generated from the Swagger definition and enriched with additional information.

Security

Contains security considerations and explains the authentication and access mechanism.

Definitions

Request and Response definitions. Generated from the Swagger definition.

Problem Descriptions

Lists and describes all error responses.

Chapter 2. Introduction

The main goal of the Network Partner API is to offer network partners an easy to use interface to deliver electronic invoices in the name of their customers (billers) to the eBill infrastructure. Electronic invoices delivered to this channel target online consumers on the financial institution / online banking side.

System participants of the eBill service are:

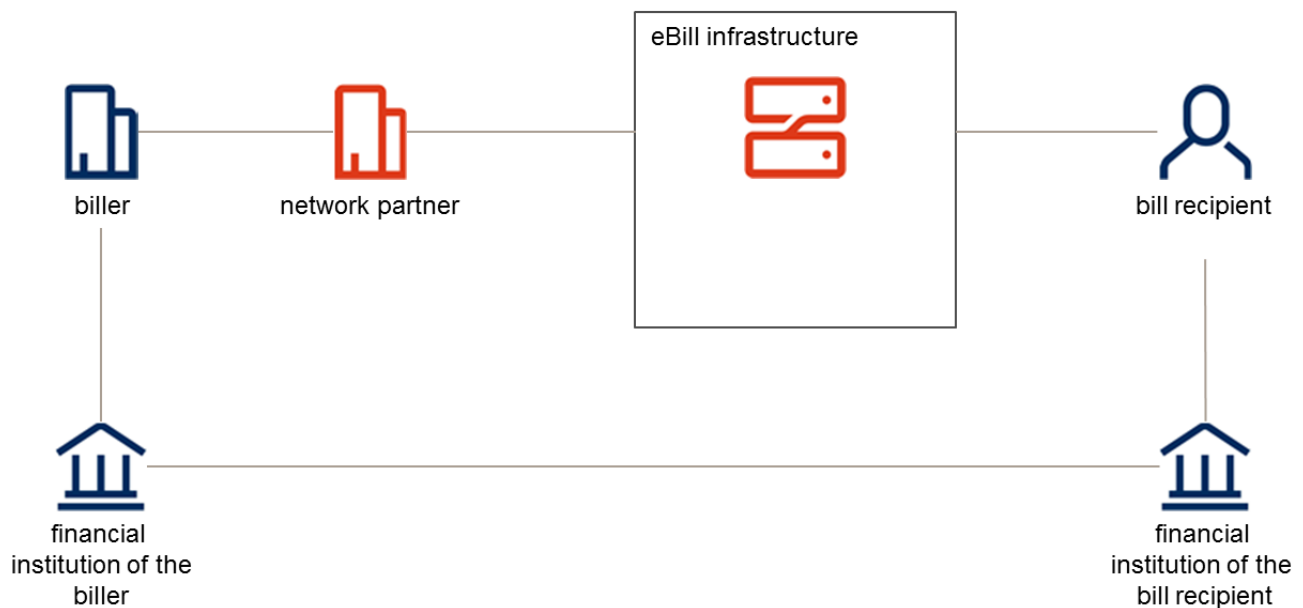


Figure 1. Participants of the eBill service

- **Biller** (creditor) is a system participant that uses the eBill service to send electronic invoices to bill recipients.
- **Network partners** offer the eBill service to billers. A system participant that delivers electronic invoices to the eBill infrastructure.
- **eBill infrastructure** offers the eBill service of SIX to network partners and financial institutions. The eBill infrastructure manages master data, receives and processes business cases, creates payment instructions and sends them to financial institutions. Furthermore it offers an Events endpoint which consumers can poll to get information on changes that occurred asynchronously.
- **Financial institution of the bill recipient** offers the eBill service to its customers (bill recipients). It provides banking services including the processing of payment instructions generated by the eBill infrastructure.
- **Bill recipients** get access to the electronic invoices while using the online banking functionality of their financial institution.
- **Financial institution of a biller** receive and book the payment on behalf of the biller. There is no direct relation to the eBill infrastructure.

2.1. Scope

This documentation focuses on the eBill service, which offers electronic invoices that can be viewed and processed by bill recipients using the eBill portal.

The documentation is specifically targeting network partners and does not include details for billers and financial institutions.

Other services of SIX, namely direct debit service ("LSV"), "E-Rechnung EDI" and "E-Rechnung Workflow" are not in scope.

2.2. System interfaces

System participants use different interfaces to communicate with the eBill infrastructure. The most important interfaces are described below:

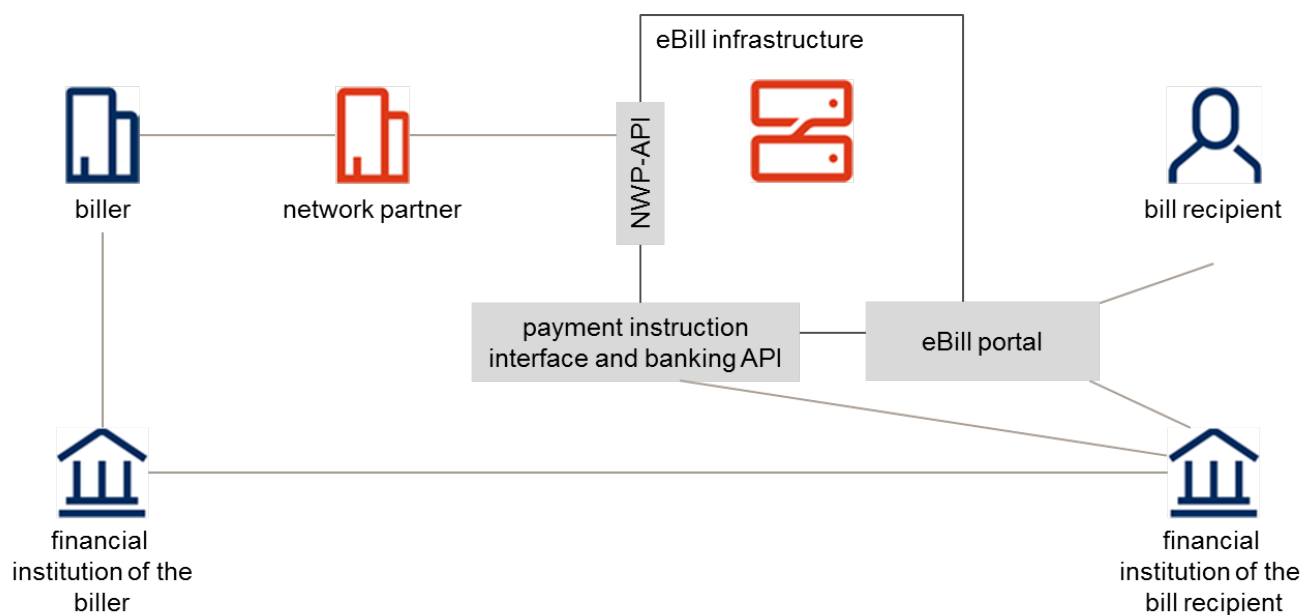


Figure 2. Interfaces of the eBill infrastructure

- **Network Partner API** allows network partners and - indirectly - the billers to interact with the eBill infrastructure.
- **Bank API** is a webservice interface for financial institutions on the bill recipient side. It e.g. allows for single sign on with the eBill infrastructure.
- **Payment instruction interface** is an asynchronous communication channel to send payment instruction messages (pain.001) to financial institutions and receive status report messages (pain.002) from financial institutions.
- **eBill portal** is a central web application that can be used by all participating financial institutions. It allows bill recipients to use the eBill functionality in the web. Access to the eBill portal is always initiated from an online banking session.

NOTE

Network partners can offer additional services to their customers, e.g. a web portal for billers. This kind of additional functionality is not part of the service offering of SIX and therefore not depicted above.

2.3. Primary network partners

The eBill infrastructure does not restrict billers to work with a single network partner. Specifically, it is possible for a biller to deliver business cases through several network partners.

However, to allow secure data management there are some restrictions and it is necessary that each biller assigns one network partner as primary network partner.

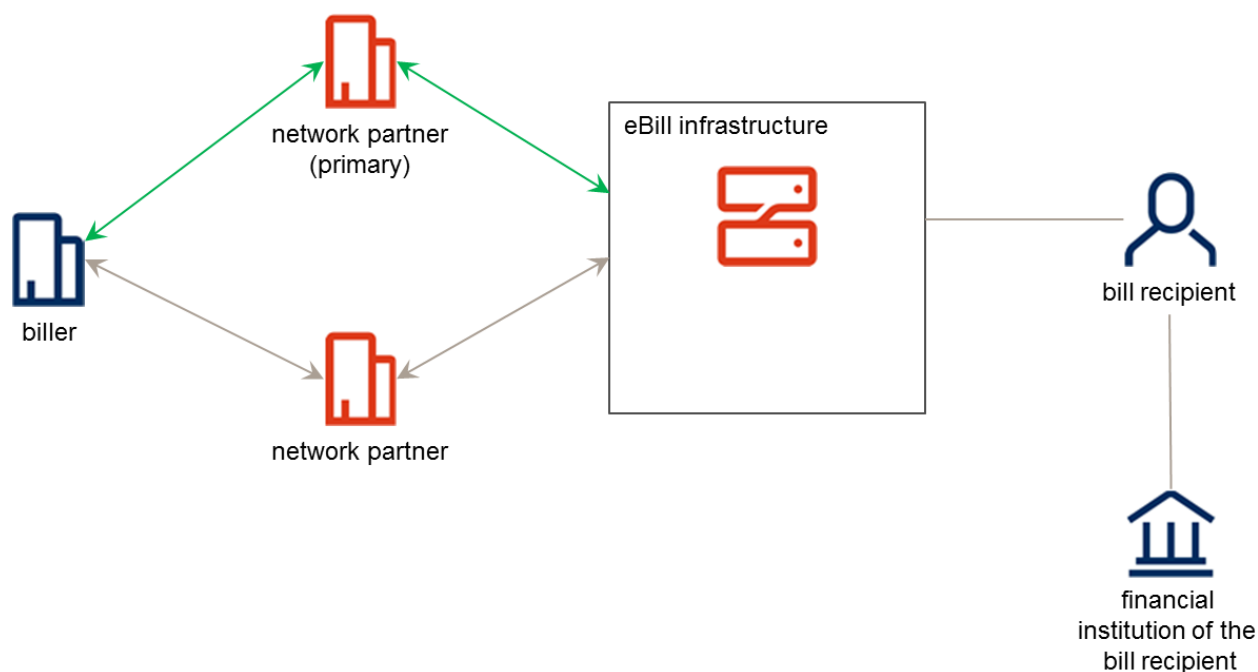


Figure 3. Network partner and primary network partner

The following functionality of the Network Partner API must be executed by the primary network partner (green above):

- Creation of new billers in the eBill infrastructure. The creation of a biller will assign the registering network partner as primary network partner.
- Management of biller master data.
- Subscriptions and cancellation of subscriptions of bill recipients with billers.

After a biller was created by his primary network partner, the eBill infrastructure allows to deliver business cases using other network partners, too (grey above). The necessary contractual agreements and technical setup have to be completed between biller and network partner.

2.4. Billing and payment process overview

An entire billing and payment process using the eBill infrastructure is shown in the following overview:

NOTE

Roles within a colored area can be represented by the same party. Example: Often, the bill recipient will be the same party as the one finally paying the bill with his account.

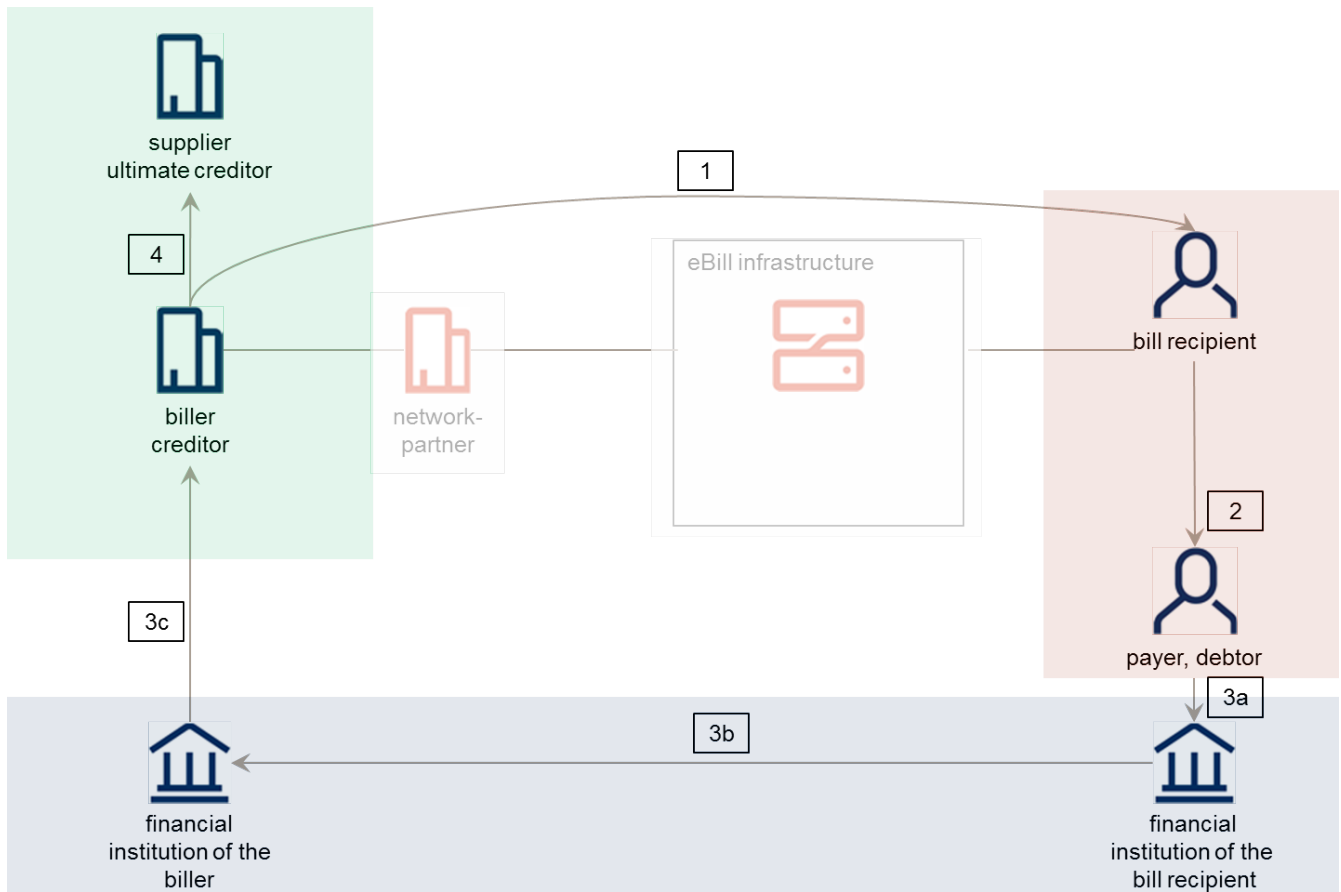


Figure 4. Billing and Payment Process

1. The biller sends an electronic invoice to a bill recipient using the services of his network partner and the eBill infrastructure. If the supplier or ultimate creditor is a different party than the biller, this has to be handled outside of the eBill infrastructure.
2. The bill recipient approves the electronic invoice and a payment instruction is sent by the eBill infrastructure to his or her financial institution. If the bill recipient is not the owner of the account being debited, a debtor party may be involved. However, these structures are implemented by the financial institution and are not known to the eBill infrastructure.
3. The debtor pays the bill and at the financial institutions the following steps occur:
 - a. The financial institution of the debtor debits the account of the debtor.
 - b. The financial institution of the debtor initiates the funds transfer to the financial institution of the biller.
 - c. The financial institution of the biller credits the account of the biller and sends a credit note to the biller.
4. A potential ultimate creditor may be informed by the biller (out of scope of the eBill infrastructure).

Chapter 3. General documentation

General information about the Network Partner API.

3.1. Design principles

The Network Partner API is designed and implemented as a RESTful API.

The REST resources are usually designed to be self-contained. However, complex business-objects (for example: a biller with multiple properties like logos) may be split into different resources and sub-resources.

3.1.1. Definition language

The Network Partner API is available as an OpenAPI Specification Version 2.0 (Swagger). This detailed specification of the API is in itself also the documentation of the API. Furthermore consumers of the API have the possibility to generate the client-side code from the specification.

The specification is provided in a separate file:

FileName	Description
networkpartner-api-v5-swagger.yaml	OpenAPI Specification of the Network Partner API. The OpenAPI Specification is best viewed in an editor such as https://editor.swagger.io/

3.1.2. Payload

The payload of the Network Partner API is defined in a format independent way in the Network Partner API specification.

The implementation of the Network Partner API expects and produces JSON-Payloads.

3.1.3. API versioning

API Versioning (Version number in Swagger and XML-Schema):

1.2.1

| | |

| | +--- Defines the Patch Version, is incremented in case of a Bugfix in the API or a change of the documentation

| +----- Defines the Minor Version, is incremented in case of a Non-Breaking-Change in the API

+----- Defines the Major-Version, is incremented in case of a Breaking-Change in the API

Versioning in Namespace and in URLs:

1

|

+----- Defines the Major-Version and is incremented in case of a Breaking-Change in the API

The major version of the API is defined in the basepath URL.

Example: "https://api.six-group.com/api/pns/networkpartner/v5"

The following API changes are defined as backward compatible, meaning not a Breaking-Change, and will not lead to a new major version of the API specification:

- adding a new resource
- new, optional headers
- new, optional query parameters
- new, optional properties in a request
- new properties in a response
- mandatory property, query parameter or header becomes optional
- changing the order of response properties
- adding new problem types

Minor and patch level versions can be redeployed in the backend at any time.

API consumers have to deal with these backward compatible changes

The following API changes are examples of breaking-changes:

- existing functionality is removed
- mandatory data structure changes
- renaming of a property, query parameter or header
- optional property, query parameter or header becomes mandatory

Major version changes are always done with a transitional period, in this period both versions of the API are accessible.

Requests are always answered with the same major version as the requests were given (via basepath URL).

3.2. Security

3.2.1. HTTPS

The REST-API is exposed on an HTTPS endpoint supporting TLS 1.2 **only**.

3.2.2. Authentication

The Network Partner API authentication works via mTLS (mutual TLS).

The network partner must provide a valid certificate meeting the requirements documented at the following location:

<https://billing.six-group.com/private/en/home/certificates.html>

Furthermore, the network partner must provide the x-networkpartner-id header that unambiguously identifies the current caller.

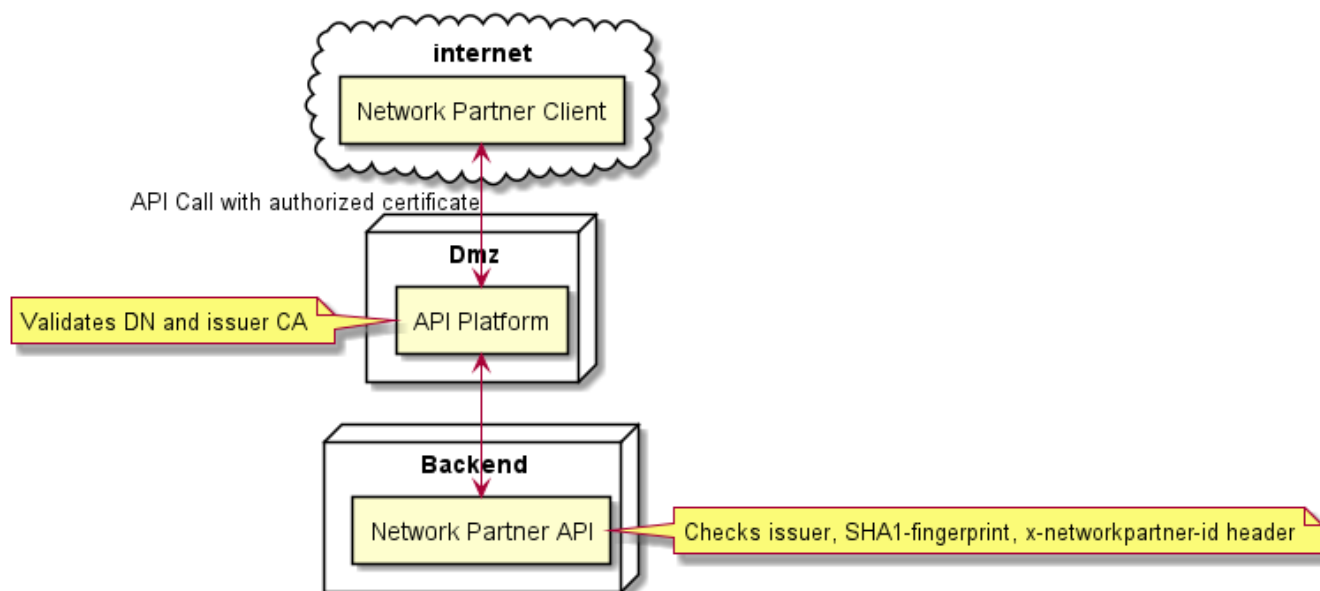


Figure 5. mTLS Flow Overview

References:

For details on headers that the eBill infrastructure expects see [Section 3.3.5, “HTTP headers”](#).

3.3. Conventions

3.3.1. Models

In [Chapter 6, Definitions](#), all models that define the resources of the Network Partner API are listed.

3.3.2. Id-handling

Identifications ("id") within the Network Partner API are always generated by the Network Partner API. This means that when the consumer of a resource creates a new object, such as billers or business cases, the "id" field in these models are optional.

When the creation of these new objects is successful, the endpoint will respond with a unique "id" that is generated by the Network Partner API.

This "id" can then be used as a unique shared identifier for this particular object and will be mandatory in any update or deletion actions.

3.3.3. Multilingual support

To support a biller that operates in multiple language regions, the API offers the concept of localizedData. Within localizedData, the biller has to specify its default language as well as provide at least the localizedData belonging to the selected default language.

There is direct support for multiple languages and the setting of a default language in the following resources:

- [Section 5.1.1, “Create a biller”](#)
- [Section 5.1.4, “Update biller”](#)
- [Section 5.1.9, “Create or update the custom subscription form of a biller”](#)

The localizedData models for biller, and biller subscription forms can be found here: [Section 6.3](#),

[“LocalizedBillerData”](#), [Section 6.50](#), [“LocalizedBillerSubscriptionFormField”](#), [Section 6.53](#), [“LocalizedBillerSubscriptionFormInfoText”](#)

3.3.4. Assets

The Network Partner API supports the creation and management of binary assets such as images. Management of the assets is split into two separate steps: First an entity (e.g. a biller) is created using a POST call. This returns asset identification (**assetId**). In a second step the binary data for the **assetId** can be uploaded.

These calls are separated to avoid embedding binary data in Base64 encoded fields or MIME/multipart uploads.

Biller example: When creating a biller via [Section 5.1.1, “Create a biller”](#), an empty asset is created for each logo in the [Section 6.3, “LocalizedBillerData”](#) entry. When the biller is created successfully, the operation will respond with a unique **assetId** for each one of these assets.

This returned **assetId** can then be used in combination with the [Section 5.1.6, “Add/update asset”](#) operation to upload the logo for the biller.

When changing a biller via [Section 5.1.4, “Update biller”](#), without changes to the logo, an existing **assetId** can be reused.

3.3.5. HTTP headers

The network partner must provide the following HTTP header fields with each request:

HTTP-Header-Field	Description	Example
x-networkpartner-id	This field contains the network partner identity of the caller A detailed description of the associated authentication is described in Section 3.2.2, “Authentication”	NWID00000012345

x-correlation-id	<p>This field contains an ID, which will unambiguously identify this request to the API, this field is mandatory.</p> <p>When the field is not included in the request the API will respond with a Code 400 (Bad Request). The uniqueness of this value is validated for each network partner, the api will keep a buffer of the last 1000 correlation ids per network partner and validate the given value against this set.</p> <p>The ID is used in to analyze problem cases and enables the eBill infrastructure to uniquely identify requests between a network partner and the API.</p> <p>For responses, the ID is taken from the request. This allows queries and answers to be linked. We recommend to use a RFC 4122 Version 4 UUID as correlation id.</p>	d36d37e7-bfad-...
------------------	--	-------------------

3.3.6. Strings must conform to XML1.0 Character Subset

Strings used in modifying API calls must conform to the defined XML1.0 character subset throughout the API.

This restriction is imposed to avoid downstream issues when communicating with systems that are limited to this character set.

Here an example for the regex pattern used to achieve this restriction:

```
'[\u0009\u000A\u000D\u0020-\u007E\u0085\u00A0-\uD7FF\uE000-\uFDCF\uFDF0-\uFFFF]*'
```

This includes all the non-control characters in the Basic Latin block as well as some control characters.

3.3.7. HTTP status codes

Each HTTP request of the client is answered to with a HTTP status code by the Network Partner API. The status code is an indication for the client whether or not the request was successful.

In the event of an error, the response body contains additional information about how the error can be resolved, see [Section 3.3.8, “Error handling”](#).

3.3.8. Error handling

If a request was not successful, the according HTTP status code provides basic information (see [Section 3.3.7, “HTTP status codes”](#)).

Further details are in the response body, see <https://tools.ietf.org/html/rfc7807>.

Example of an error response:

```

HTTP/1.1 400 Bad Request ①
Content-Type: application/problem+json
Content-Language: en
{
  "type": "/problems/REQUEST_BODY_VALIDATION_FAILED", ②
  "title": "Payload has missing or invalid values", ③
  "status": 400, ①
  "detail": "The submitted request contains invalid or missing data which can not be
processed.", ④
  "instance": "/api/pns/networkpartner/v5/billers/errors/NWID0090000001/ef46fa53-
6377-40d3-9f35-39a4a507792e" ⑤
  "technicalReason": "some field validations failed" ⑥
  "fieldErrors": [ ⑦
    {
      "fieldName": "fieldNameWithValidationError",
      "message": "size must be between 1 and 1000",
      "rejectedValue": ""
    }
  ]
}

```

- ① The HTTP status provides basic information.
- ② A problem URI specifies the type of problem that occurred. See [Chapter 7, Problem Descriptions Overview](#) for details.
- ③ Human readable title of the problem.
- ④ Details of the problem that occurred.
- ⑤ A specific reference of this occurrence. A combination of request URL, network partner id and correlation id.
- ⑥ A more detailed technical reason of the problem, if available.
- ⑦ An object containing validation errors on field level, if available. The object contains the field name, a message and optionally a rejected value.

3.3.9. Event handling

All events are accessible on the specific operations of the resource [Section 5.3, “Events”](#).

The eBill infrastructure will generate events specifically for each consumer (e.g. network partner).

Each operation offers the following parameters to control which and how many events are returned in a single response:

Parameter	Description
lastEventId	when provided, the operation will only respond with events that occurred after the provided lastEventId

limit	the maximum number of events the endpoint will respond with in a single call
-------	--

3.3.10. Search operations

Search operations in the form of POST requests, as for example [Section 5.1.2, “Search billers”](#), use a common set of arguments and return identically structured response. In addition, they follow a common pagination behaviour. This chapter describes the commonalities.

The search query arguments allow the retrieval of result lists in consecutive requests. The search query may contain the arguments explained below. The order of arguments is not relevant.

Argument	Description
limit	Restricts the result set to the specified number of items. Less or none may be returned. The parameter may be omitted, in which case the default is applied.
offset	The distance from the first element in the resulting list to the first item to be returned. The very first item complying to the search arguments has an offset of 0. If omitted, 0 is assumed.

Examples:

URL query string	Explanation
/billers/search	Defaults apply and biller 1 to 100 are returned
/billers/search?limit=100	Equivalent to above
/billers/search?offset=0&limit=100	Equivalent to above
/billers/search?offset=0&limit=0	Returns an empty response
/billers/search?limit=500&offset=1000	Returns biller 1001 to 1500
/billers/search?offset=2000	Returns biller 2001 to 2100

Result lists always follow the same ordering.

The request body is a mandatory json type. It contains a sequence of filter arguments, wrapped in a type of name "filter". Filter arguments may be empty or omitted, the structure "filter" is mandatory but may be empty.

Search without arguments:

```
{
  "filter": {}
}
```

Search with some arguments:

```
{
  "filter": {
    "name": "SIX",
    "iban": "CH100023000A109822346"
  }
}
```

Search operations always return a response object. The response might however be empty, or contain less items then requested.

It contains a total count which indicates the number of items complying to the filter arguments, regardless of limit or offset. This may be used for pagination.

Finally, the response contains an array of items in the "items" structure. The array can be empty, in which case the total count is 0.

No items found:

```
{
  "totalCount": 0,
  "items": []
}
```

Two items found:

```
{
  "totalCount": 2,
  "items": [
    {
      ...
    },
    {
      ...
    }
  ]
}
```

3.3.11. Guidelines for polling the events

Depending on the network partner, different types of events may be of interest. We recommend the network partner to consume those events of interest on a regular basis.

Generally it is recommended to poll each events endpoint once every couple of minutes, whereas it's important to consume all events until the endpoint does not return any more events.

Please avoid querying events unnecessary often.

In order to keep our services performant, events won't be available after 60 days. If you are unable to find your Event-Id, please query without Event-Id to get the oldest events available.

3.3.12. Maintenance windows

The healthcheck endpoints [Section 5.7, “Healthcheck”](#) provide information about the next planned maintenance windows. For each maintenance window the start, and the end time of maintenance is returned.

As soon as the maintenance is completed, the maintenance window will be removed from the list, and the eBill infrastructure is available again.

If the maintenance can't be completed in the planned window, it won't be updated. It will remain until the maintenance is done. In general this should not happen and therefore there should not be a maintenance window in the past.

Chapter 4. Use cases

4.1. Biller management

Billers are managed through the [Section 5.1, “Billers”](#) endpoint which offers a number of different operations:

- The operation [Section 5.1.2, “Search billers”](#) is the public listing of all billers. This means it exposes all public biller data to all network partners, not only the biller’s primary network partner.
- The operation [Section 5.1.3, “Get biller by id”](#) is used to retrieve a single specific biller by its Id. It returns the complete biller information, but can only be called by the primary network partner of this biller.
- The operation [Section 5.1.1, “Create a biller”](#) is used to create a new biller. The network partner that created the biller is automatically assigned as primary network partner. If its status is **ACTIVE**, a duplicate check will be performed on existing **ACTIVE** billers.
- The operation [Section 5.1.4, “Update biller”](#) is used by the primary network partner of the biller to change the data of the biller. It is not possible to delete a biller, but one can set its status to **INACTIVE**. If its status is **ACTIVE**, a duplicate check will be performed on existing **ACTIVE** billers.

The network partner can specify if the biller is allowed to submit donation inquiries by the property 'isAllowedToSubmitDonationInquiries'. This permission must only be granted for verified non-profit organizations.

The network partner can add or remove certifications from his billers. Certifications serve as markers that identify billers with specific characteristics, enhance their credibility and highlights their commitment to responsibility.

Ebill infrastructure oversees the management of certifications available for billers, with the understanding that they can be altered or updated at any given time.

Certifications may be displayed to users within the eBill portal or online banking. Network partners bear the responsibility of ensuring that the certifications associated with their billers are consistently kept current and accurate.

4.2. Business case management

Business cases can be submitted through the operation [Section 5.1.12, “Create business case in PDF/A-3b-format”](#). After a successful creation, the operation answers with a [Section 6.21, “BusinessCase”](#).

The id assigned to the business case by the Network Partner API can later be used to retrieve the business case information from the [Section 5.1.13, “Get business case”](#) resource.

There are six business case types [Section 6.15, “Bill”](#), [Section 6.18, “Reminder”](#), [Section 6.17, “CreditNote”](#), [Section 6.19, “Advice”](#), [Section 6.16, “InstalmentBill”](#) and [Section 6.20, “DonationInquiry”](#).

Credit note and advice only have the status **OPEN** and **COMPLETED**, which change, as soon as the user has viewed the business case.

These business cases are furthermore excluded from the status change reports.

The following state diagram shows the lifecycle of single payments (from a bill, reminder or donation inquiry) and instalments (from an instalment bill):

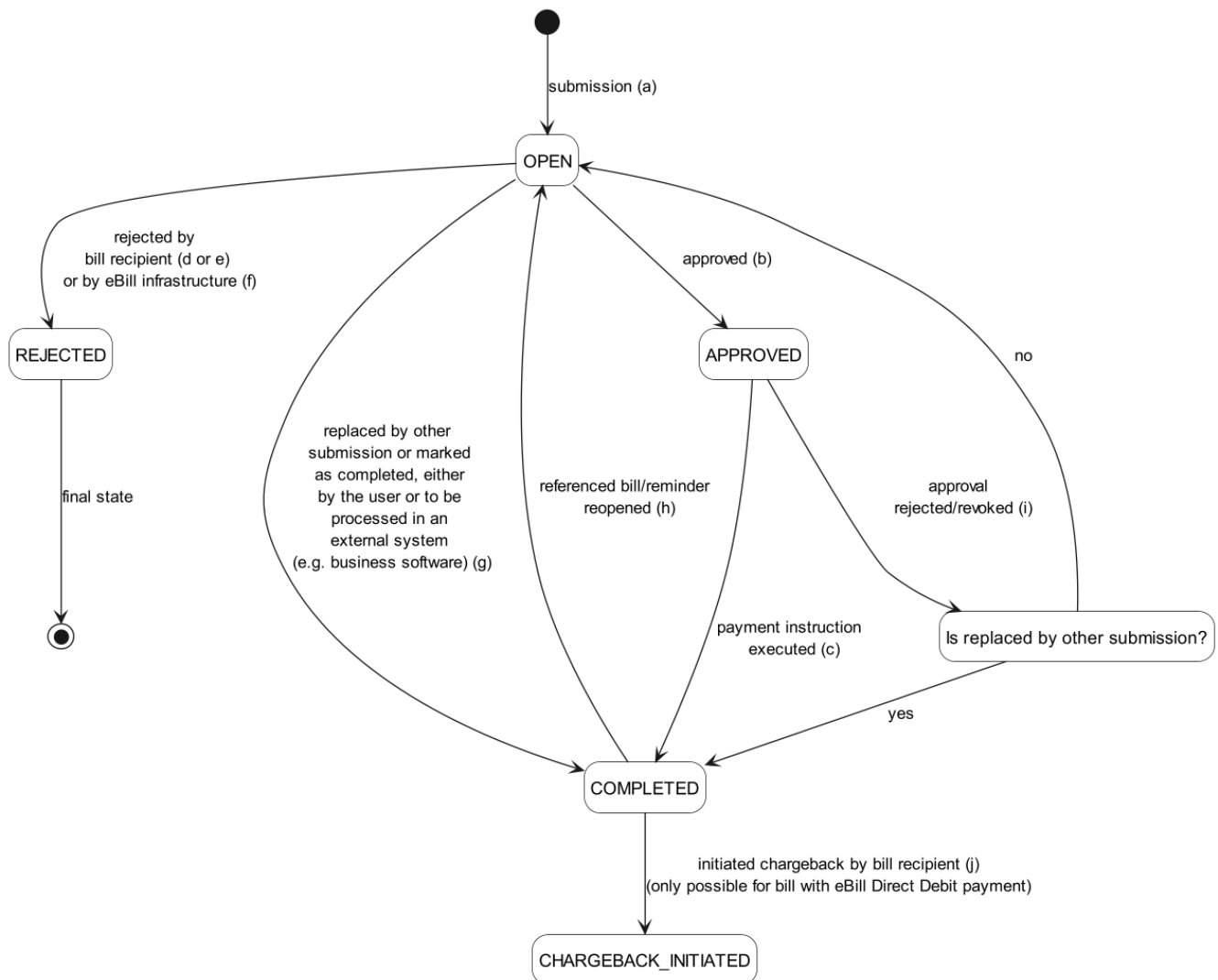


Figure 6. Lifecycle of bills, reminders, instalments and donation inquiries

Happy path

- a. A new single payment or instalment was created.
- b. The single payment or instalment was approved, either by the bill recipient or a standing approval.
- c. The payment instruction was executed. The single payment or instalment gets completed.

Direct or indirect rejection by bill recipient

- d. The bill recipient rejected the single payment or instalment. If more information is needed, the biller must contact the bill recipient.
- e. The bill recipient deregistered from eBill while the single payment or instalment was still open.

Indirect rejection by eBill infrastructure

- f. The single payment or instalment was cleaned up by the eBill infrastructure.

Completed by user or business software or replaced or referencing business cases

- g. The single payment or instalment was either processed by an external business software and the status is therefore changed to completed or the user has changed the status to completed by himself.

Please note that a donation inquiry can not be replaced or referenced by another business case and vice versa. The following applies only for single payments that are not donation inquiries and instalments.

.. In case of a replacement of a bill or an instalment bill by a new submission, the referenced bill or all instalments of the referenced instalment bill changed to completed.

If the new submission was a reminder, it did not replace the bill or the instalment bill, but only references it and both business cases stay open. If either the reminder, the bill or any instalment of the instalment bill got approved, the referenced business case will be set to completed.

Referencing is possible in eBill-SIX_v5.xml with the element `referencedBill`, where the reference number of the referenced bill has to be provided.

.. As in (i), a bill or an instalment bill was referenced by a reminder. The approved business case of the pair got reopened, which reopens the referenced business case as well.

Revocation of an approval

- i. The single payment or any instalment of an instalment bill got revoked/rejected in the online banking. As a consequence, its status changed back to open (reopened). But, if in the meantime (between approval and approval-rejection/-revocation), it was replaced by another submission (bill, reminder or instalment bill), its status will change to completed. Please note that a donation inquiry can not be replaced by another submission.

There are multiple reasons for reopening:

- The approval was revoked in the online banking by the bill recipient.
- The payment instruction was not valid.
- The payment instruction could not be executed.

Chargeback by bill recipient

- j. For a bill with eBill Direct Debit payment which supports chargeback, the bill recipient has the option to chargeback the debited amount within a period of 60 days after being notified about the payment.

Each status change will be reported by a specific event as described in [Section 5.3, “Events”](#). Excluded from the reports are status changes of the business cases [Section 6.17, “CreditNote”](#) and [Section 6.19, “Advice”](#).

It's distinguished between business case status change events (for bills, reminders and donation inquiries) and instalment status changed events (for instalment bills only).

Further information on the format of the PDF needed for the [Section 5.1.12, “Create business case in PDF/A-3b-format”](#) can be found in [Section 4.2.1, “eBill format”](#).

4.2.1. eBill format

The eBill format is the specification of the PDF and the attachment included in the PDF. The PDF with the included attachment is the required payload of the [Section 5.1.12, “Create business case in](#)

Business considerations

The business requirements that are considered by this standard are summarized as follows:

- It should be possible to easily convert existing electronic billing standards to the eBill format.
- VAT details and instalments information are supported on the business case level in the format.
- Addresses are required and structured in compliance with new regulations
- Relevant information concerning bill recipient ERP systems are supported under the term "workflow".
 1. This workflow information is modeled as optional values in the eBill-SIX_v5.xml format and supports information pertaining to VAT, delivery of goods/services, and reference identifiers see [Section 4.2.1.5, “Specification of the structured data”](#).
 2. Only the structure of this data is validated by the Network Partner API.
- The information contained in the QR-bill must be sufficient for the creation of a simple electronic invoice. A minimal set of base attributes will be enriched by the network partner (for example, bill recipient identification, biller identification and business case type).

Basic structure

To facilitate the transfer of business cases from a network partner to the eBill infrastructure a standard-format has been defined.

The eBill infrastructure only handles this standard-format and does not convert or support any other formats.

Transformations from other message formats to eBill format can be offered by the network partner as a service.

The eBill format consists of a PDF-container with an XML attachment and embedded signatures, see the figure below.

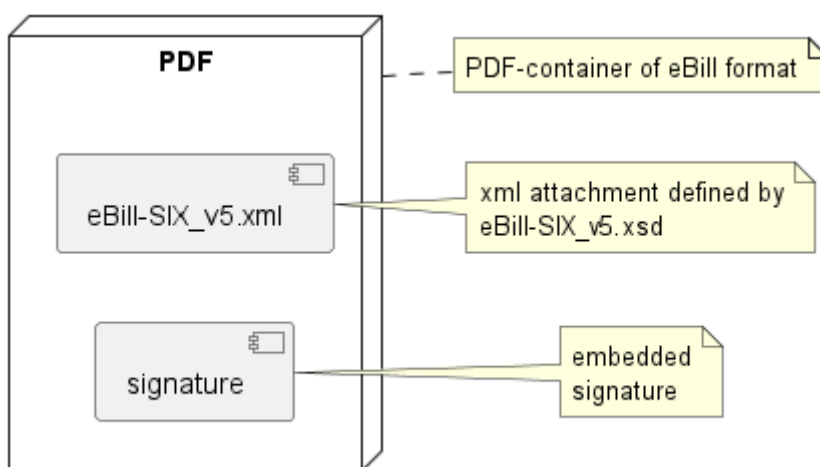


Figure 7. A representation of the PDF-container

Table 1. Glossary:

Name	Description
------	-------------

eBill format	The complete PDF (PDF incl. XML and signatures)
eBill-SIX_v5.xml	The Structured business case information in XML included as attachment in the PDF
eBill-SIX_v5.xsd	The XSD-Schema that describes the structure of the XML attachment (eBill-SIX_v5.xml)

Instalments

The structured data offers support for instalment bills. A biller can offer to pay his bill by various instalment options. Each of the instalment options can contain multiple instalments:

XML structure	Description («text given by biller and visible to bill recipient»)
paymentsByInstalments	parent element of the instalment options
paymentByInstalments	instalment option
isTotalAmountHidden=false	configuration of the instalment option
isTotalPaymentOption=true	
instalments	parent element of the instalments
instalment	an instalment «Total payment at once with 2% discount»
paymentByInstalments	instalment option
isTotalAmountHidden=true	configuration of the instalment option
isTotalPaymentOption=false	
instalments	parent element of the instalments
instalment	an instalment «First half»
instalment	an instalment «Second half»

Figure 8. Example of two instalments options

The bill recipient will pick one of the instalment options in the UI and will subsequently approve all of the instalments.

File specification and signatures

The supplied PDF needs to be a PDF/A-3b conforming document. This type of document fulfills a number of requirements "out of the box" such as the support for long term archiving, embedding data, document signing and wide tools support.

A PDF is allowed to have other embedded documents. The eBill format requires that an XML attachment is included with the name "eBill-SIX_v5.xml" that is compliant with the eBill XSD. The PDF must not contain an XML attachment with name "eBill-SIX_v1.xml" or any other version.

The document needs to be signed with a PAdES-B-B-level conforming electronic signature. The documentation of the PAdES standard can be downloaded from <https://www.etsi.org/>.

The signature must be included in the PDF document and encompasses the whole PDF document including the embedded documents and at least the eBill-SIX_v5.xml.

It is possible to provide the signing certificate in the Document Security Store (DSS).

Note: Accessibility is not required in the PDF/A-3b. It is up to the biller to ensure accessibility, if

desired.

Specification of the structured data

The specification of the eBill-SIX_v5.xml is created and maintained by SIX. The XML adheres to the following ground principles:

- Simplicity of the format (No detailed invoice positions)
- Limited user group (network partner to infrastructure)
- Possibility to convert from existing, simple formats, through the use of QR-Bills
- The format is based on the recommendations of "swissDIGIN content standard light"

A specification of the structured data (XML) including a detailed description is available.

This specification is delivered separately:

FileName	Description
eBill-SIX_v5.xsd	XSD-Schema of the structured data (eBill-SIX_v5.xml)
eBill-SIX_v5_advice.pdf eBill-SIX_v5_bill.pdf eBill-SIX_v5_creditNote.pdf eBill-SIX_v5_instalmentBill.pdf eBill-SIX_v5_reminder.pdf eBill-SIX_v5_donationInquiry.pdf	documentation of the XML schema

Character set

The eBill-SIX_v5.xml content will be parsed using the UTF-8 character set. Please note that ISO 20022 messages according to the Swiss Payment Standard only allow the Latin Character Set to be used. Therefore a conversion as documented in the pain.001 specifications will be applied for non latin characters in the relevant fields.

4.2.2. eBill format versioning

The major version of the eBill format is defined in the namespace of the XML schema (eBill-SIX_v5.xsd).

Example: xmlns:ebill="http://six-group.com/pns/networkpartner/v5/ebill/xml"

The following XML schema changes are defined as backward compatible and will not lead to a new major version of the XML schema:

- new, optional fields
- adding new error codes

Consumers of the eBill format have to deal with such changes without prior notice.

4.3. Subscriptions and subscription cancellations

The transfer of business cases from a biller to a bill recipient requires both parties to agree on this process. The resulting connection is called "biller to bill recipient relation" or for simplicity just "biller relation". Initiating a biller to bill recipient relation is called "subscription" and there are various ways to achieve this:

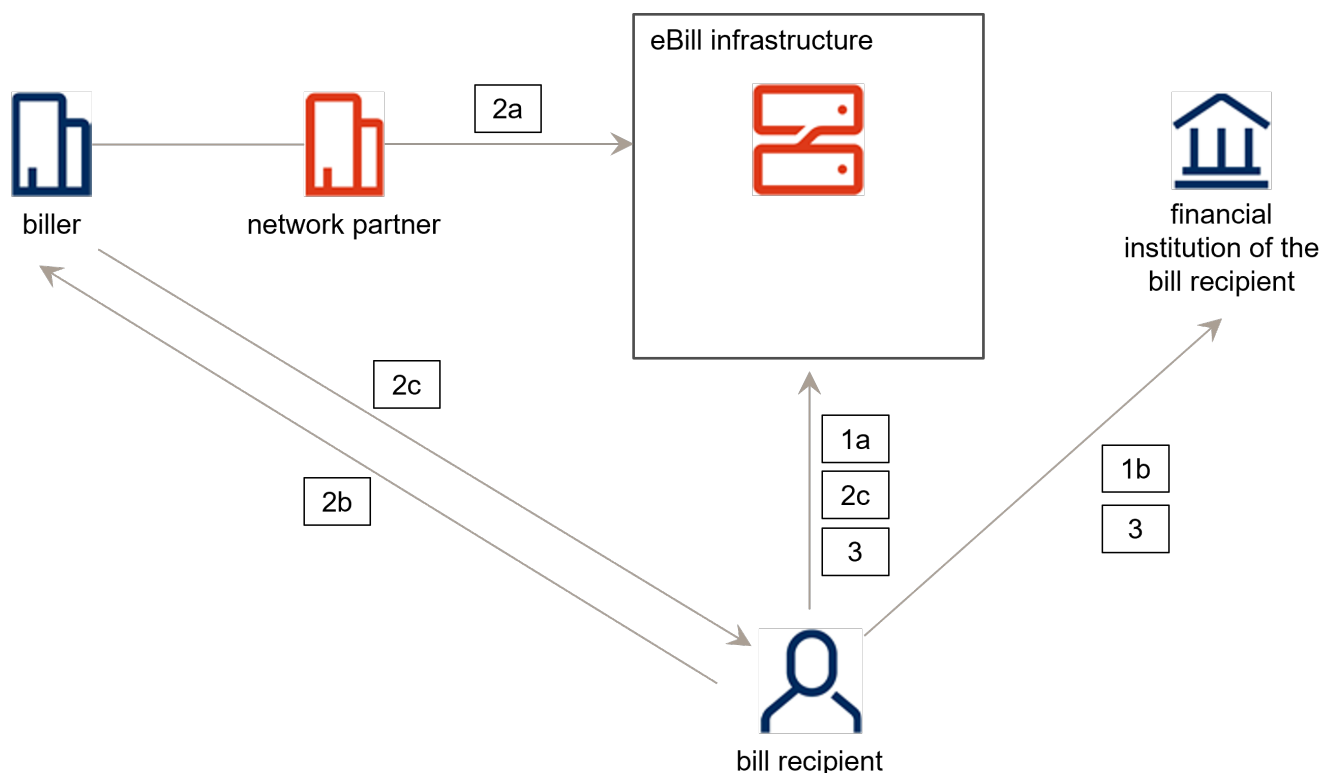


Figure 9. Overview of biller subscription options

1. Section 4.3.1, “Bill recipient driven subscription”

- a. Subscribe through the eBill portal with or without custom subscription form [Section 4.3.1.1, “Subscribe through the eBill portal”](#)
- b. Subscribe through the online banking of the financial institution [Section 4.3.1.2, “Subscribe through the online banking”](#)

2. Section 4.3.2, “Biller driven subscription”

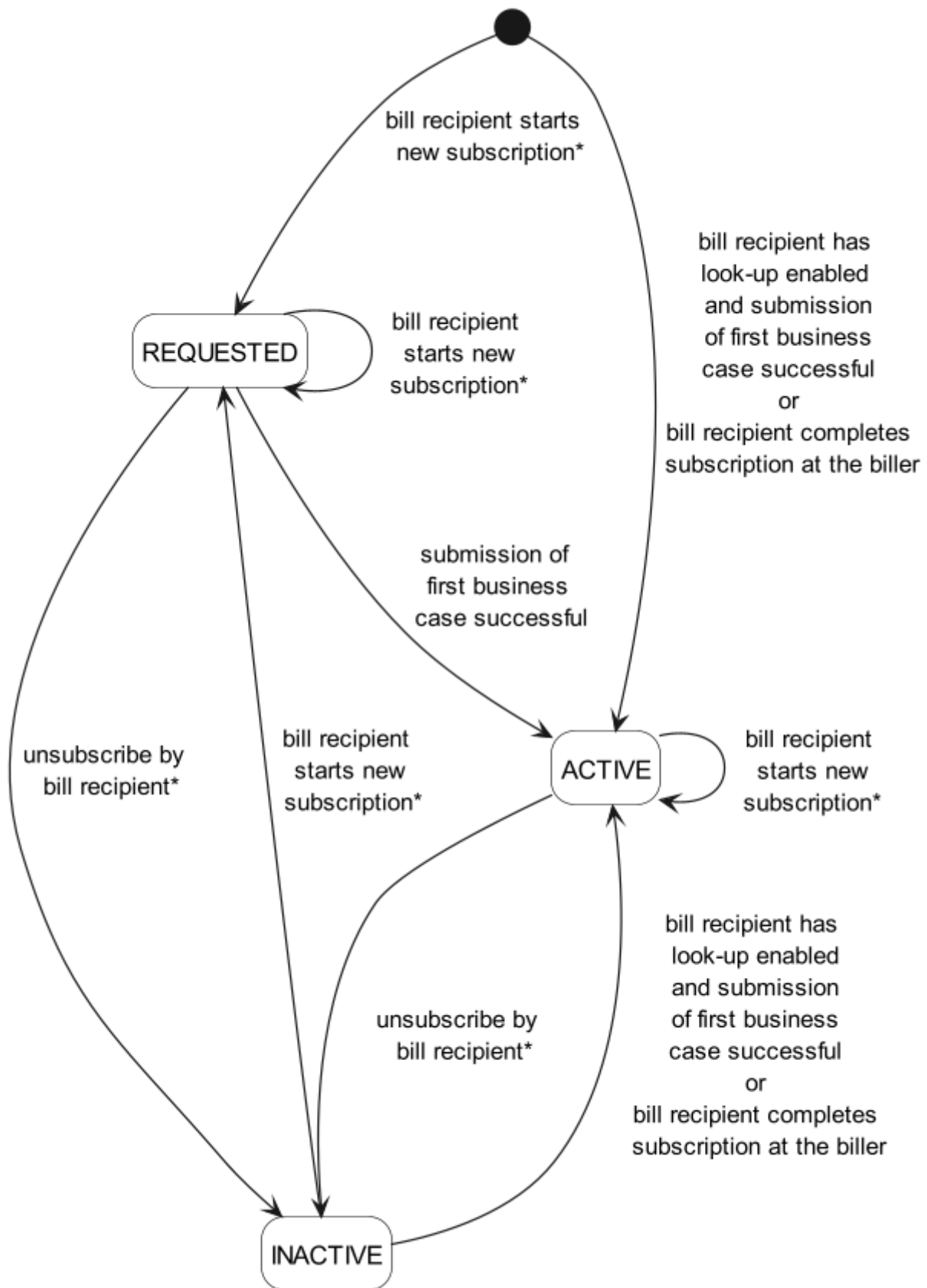
- a. Biller Look-Up [Section 4.3.2.1, “Look-Up”](#).
- b. Subscribe through the website of the Biller [Section 4.3.2.2, “Subscription at the biller”](#).
- c. Subscribe through the eBill infrastructure [Section 4.3.2.3, “Subscription at the eBill infrastructure”](#).

3. Automatically if the user accepts a proposed eBill Direct Debit proposal see [Section 4.5.2, “Submit an eBill Direct Debit proposal”](#). In this scenario the custom subscription form details will not be provided, as the biller is already able to identify the bill recipient.

4.3.1. Bill recipient driven subscription

In all variants of bill recipient driven subscriptions, the driving user is the future bill recipient. In case of eBill sharing this can also be the authorized sharing partner.

The following state diagram of the biller to bill recipient relation illustrates the status transitions and reasons for it.



status transitions marked with a * trigger an event for the network partner

Figure 10. State diagram for biller to bill recipient relations

The bill recipient starts the subscription process either in the eBill portal or in the online banking.

In ideal circumstances, the information provided by the eBill infrastructure (see [Section 6.14, “BillRecipient”](#)) is sufficient for the biller to identify and establish the specific relationship between himself and the bill recipient.

If additional information is required, a custom subscription form may be presented to the bill recipient. The form is defined by the biller within the eBill infrastructure by the primary network partner of the biller. The definition consists of input fields, explanatory information, constraints and properties for visual rendering. The form is presented to the bill recipient by the eBill portal during the subscription process, or, in a read-only mode, during a subscription at the eBill infrastructure (see [Section 4.3.2.3, “Subscription at the eBill infrastructure”](#)). The presentation is an integral part of the eBill portal and has the same characteristics regarding design, multilingualism and accessibility. For more details about the different types of biller subscription form fields, see the resource definition: [Section 5.1.9, “Create or update the custom subscription form of a biller”](#)

Custom subscription forms may be altered any time by the primary network partner of the biller. Changes take effect immediately for new subscriptions, but presently viewed forms may not reflect the changes. In order to prevent receiving deprecated form data, the biller may alter its `billRecipientSubscriptionStatus`, so that no new subscriptions are started, wait for a period of time sufficient for current subscription processes to conclude, update the form and then change back the status.

Subscribe through the eBill portal

If the bill recipient is using the eBill portal, the subscription flow is as follows:

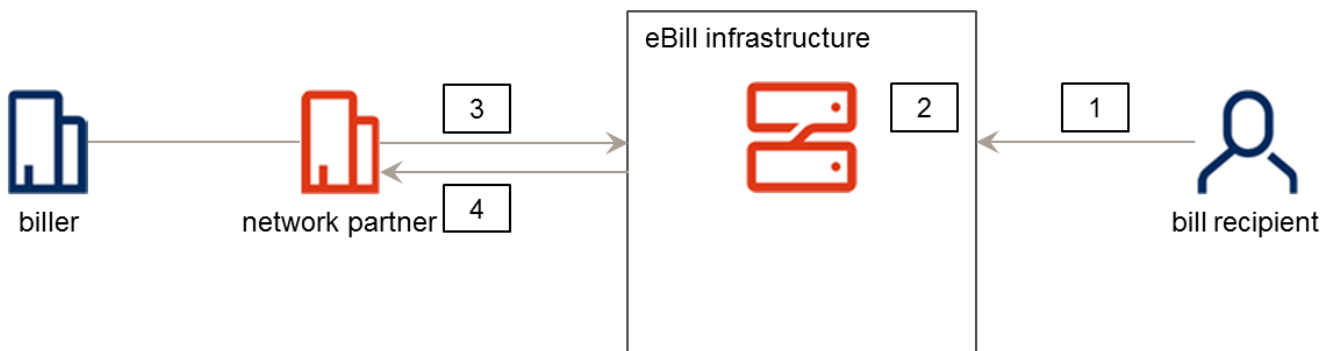


Figure 11. Subscription initiated by bill recipient

1. A bill recipient finds the biller and starts a subscription request. If the biller has defined a custom subscription form within the eBill infrastructure, the form is shown to the bill recipient.
2. The bill recipient concludes the subscription process, whereupon a biller relation in status `REQUESTED` is created and a subscription event is generated.
3. The network partner polls for new subscription events....

See the resource definition: [Section 5.3.3, “Find events for bill recipient subscriptions which changed status”](#)

4. ...and receives the subscription data as a response. The subscription data may include filled in form data.

Subscribe through the online banking

A similar process is started if the bill recipient is using the online banking of his financial institution:

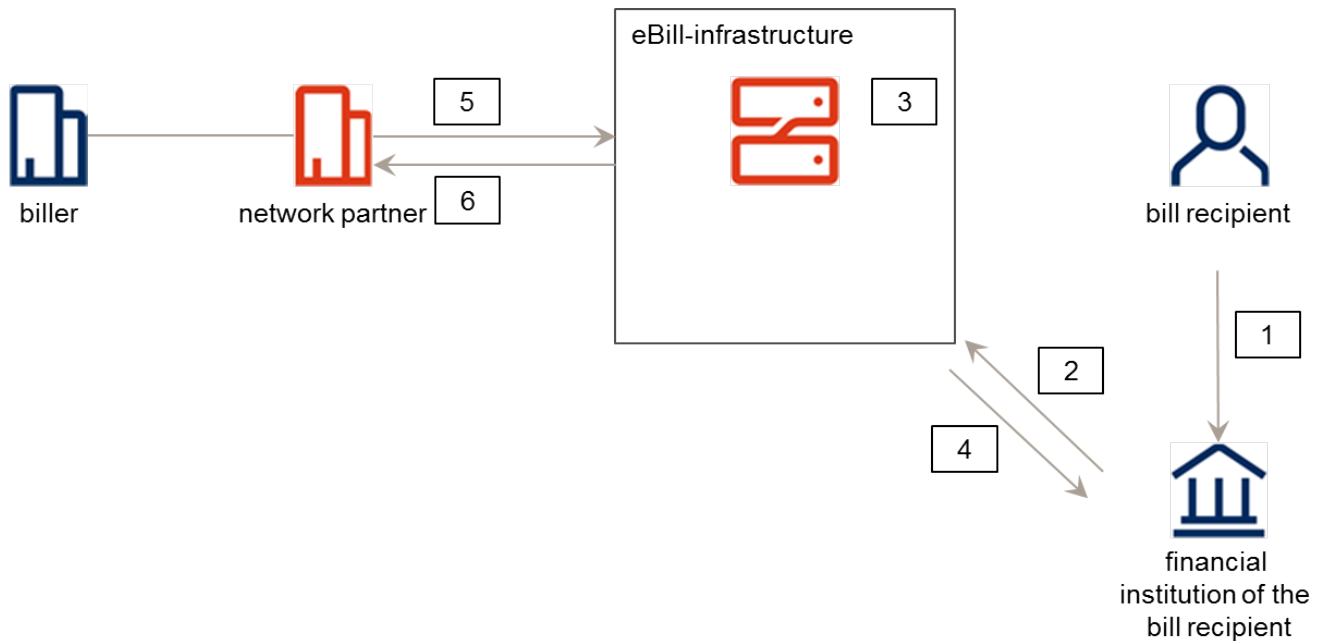


Figure 12. Subscription initiated by the online banking of the bill recipient

1. A bill recipient enters payment information in the online banking of his financial institution. The financial institution checks if the payment information (account) refers to an entry in the biller directory. If a match is found, the online banking provides the user the option to subscribe to the biller.
2. The user decides to subscribe to the biller in the online banking. The online banking calls the eBill infrastructure to subscribe.
3. The eBill infrastructure creates a bill recipient subscription in status **REQUESTED** and creates a subscription event for the network partner.
4. The response of the eBill infrastructure returns an OK code.
5. The network partner polls for new subscription events....

See the resource definition: [Section 5.3.3, “Find events for bill recipient subscriptions which changed status”](#)

6. ...and receives the subscription data as a response. The response [Section 6.12, “BillRecipientSubscription”](#) contains both **accountNumber** and **referenceStructured** which clearly identify the bill recipient to the biller.

Events for subscriptions and subscription cancellations are triggered after a bill recipient has subscribed or unsubscribed to a biller. It is also possible for the bill recipient to subscribe multiple times to the same biller, which results in multiple events that can have the same status but maybe a different email address or a different address.

4.3.2. Biller driven subscription

It is in the best interest of billers to promote biller subscriptions to their customers. There are three ways a biller can initiate such a biller subscription:

- [Section 4.3.2.1, “Look-Up”](#)
- [Section 4.3.2.2, “Subscription at the biller”](#)
- [Section 4.3.2.3, “Subscription at the eBill infrastructure”](#)

If a customer is found via Look-Up, eBills can be sent to him immediately. An additional subscription process is not necessary.

Look-Up

Billers get the possibility to search for bill recipients using the Network Partner API. With a positive match the biller can immediately submit business cases to the bill recipient.

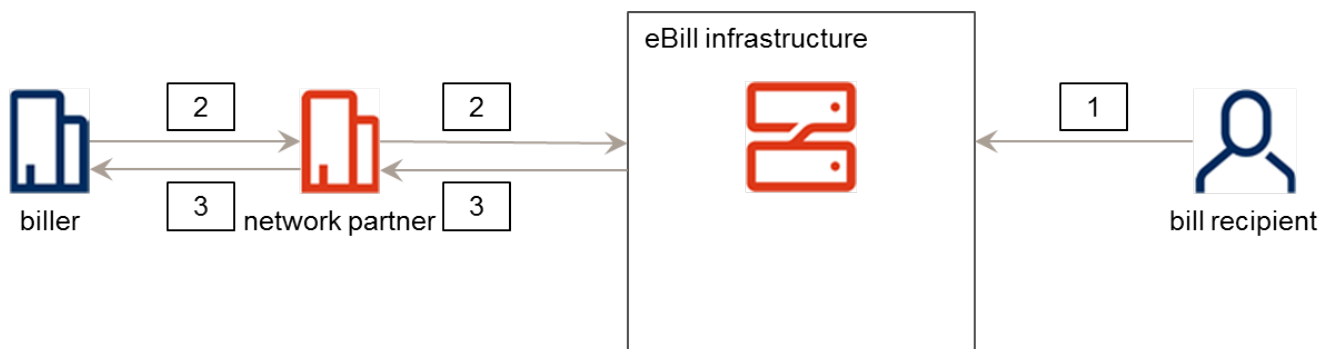


Figure 13. Subscription initiated by biller using Look-Up

1. Bill recipients can specify whether they can be found by billers by updating their Look-Up status in the user settings page. In case of non-profit organizations, bill recipients must explicitly agree to this in their user settings.
2. The biller may query the eBill infrastructure to find bill recipients and/or potential donors (e.g. using the bill recipient's email address as search criteria).

See the resource definition: [Section 5.1.11, “Search for multiple bill recipients for this biller”](#)

3. If all conditions are met, a positive response is returned and it is possible to submit a business case immediately.

Subscription at the biller

In this method, the biller relation is created directly at the biller's website. The biller can fully integrate the subscription process into its infrastructure, thus having the greatest possible flexibility and no media discontinuity. A prerequisite for this method is that the biller has identified its bill recipient (for example, by logging into the biller's customer portal).

This functionality can only be offered by the primary networkpartner.

The flow for this subscription process looks as follows:

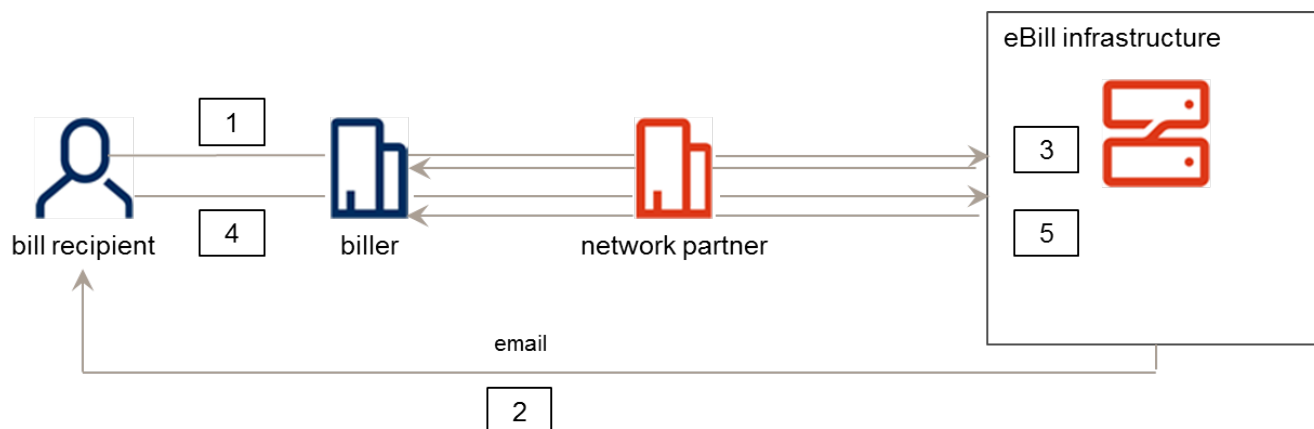


Figure 14. Subscription at the biller

1. The bill recipient chooses to use eBill as a payment method at the billers website and fills in its eBill email address, which is sent to the eBill infrastructure.

See the resource definition: [Section 5.2.1, “Initiate bill recipient subscription”](#).

2. The email is verified and an email including a verification code is sent to the user.
3. The eBill infrastructure responds with a unique token that identifies this subscription request.
4. The user fills in the verification code on the billers website, the verification code and unique token are sent to the eBill infrastructure.

See the resource definition: [Section 5.2.2, “Confirm bill recipient subscription initiation”](#).

5. If the token and code are valid, the eBill infrastructure creates a new biller relation in the status **ACTIVE**. The bill recipient data is returned to the network partner.

Furthermore, this flow could be combined with the [Section 4.3.2.1, “Look-Up”](#) flow as a first check to see if the bill recipient has Look-Up enabled. If this is the case, the biller does not need to go through the activation/subscription flow and instead can just send in eBills for this recipient.

Subscription at the eBill infrastructure

With this method, the bill recipient subscribes to a biller with a personalized link to the eBill infrastructure. The subscription process for the bill recipient takes place entirely at the eBill infrastructure. The biller can request a personalized link and share it with his customer through various channels, either as a link or QR code:

- via email
- in a letter
- on the website

The link is valid for a limited time.

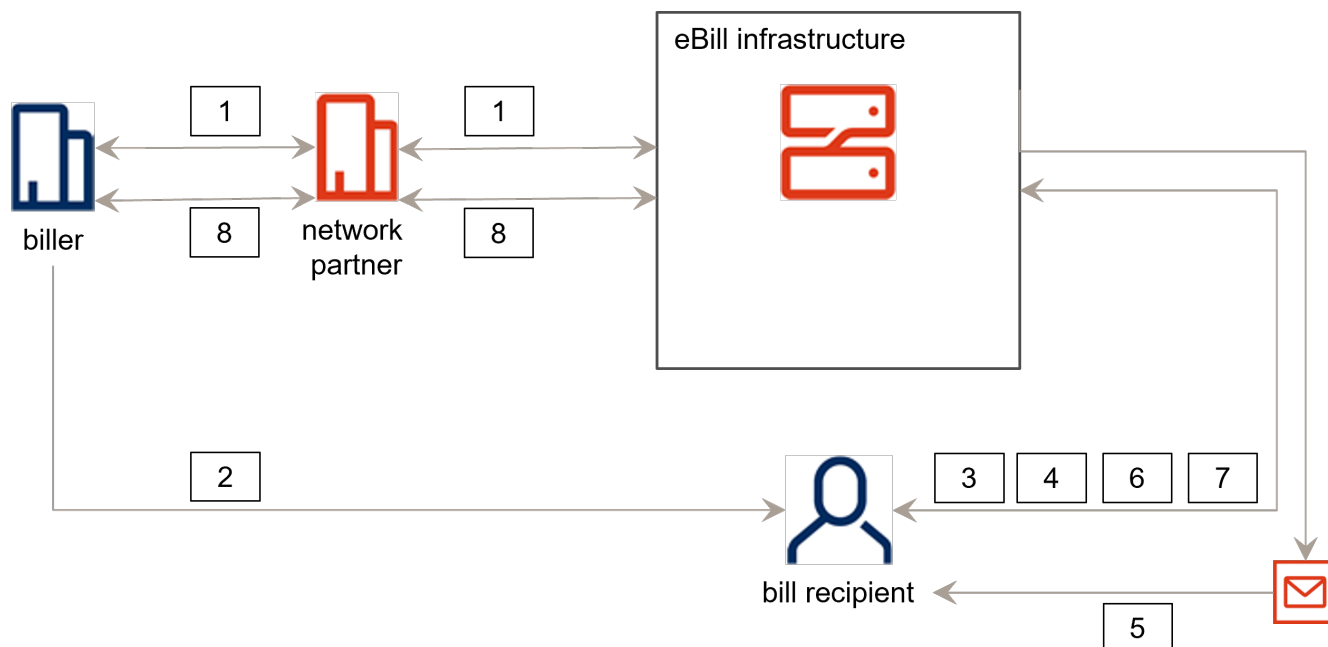


Figure 15. Subscription at the eBill infrastructure

1. The biller requests a personalized subscription URL for a specific customer. The biller may deposit one of the following customer identifying information for this subscription URL:
 - a. QR-Reference
 - b. Custom subscription form, already prefilled with customer identifying information compliant to the current definition. See [Section 5.1.9, “Create or update the custom subscription form of a biller”](#).
2. The biller presents the personalized subscription URL to its customer through any channel.
3. The customer opens his personalized subscription URL to the eBill infrastructure.
4. The customer enters the email address that he uses for eBill. The email address is validated.
5. An email with a verification code is sent to the customer.
6. The customer enters the verification code. The verification code is validated.
7. The identifying information of the customer is displayed, if any has been deposited. It cannot be changed. The customer confirms the subscription.
8. Once the subscription is confirmed, a new biller relation with the status **ACTIVE** is created and a subscription event is triggered. The biller receives the subscription event via his network partner. Customer identifying information, if given, is included.

Before offering this subscription method to a customer, it is recommended to check that the customer cannot be found via the [Section 4.3.2.1, “Look-Up”](#). However, if the customer is found via Look-Up, eBills can be sent to him immediately. A separate subscription process is not necessary.

4.3.3. Subscription cancellation

The cancellation of a subscription is always initiated by a bill recipient (a biller will simply decide not to send eBills any more - this is not represented in the eBill infrastructure).

A subscription cancellation leads to an event on the Network Partner API.

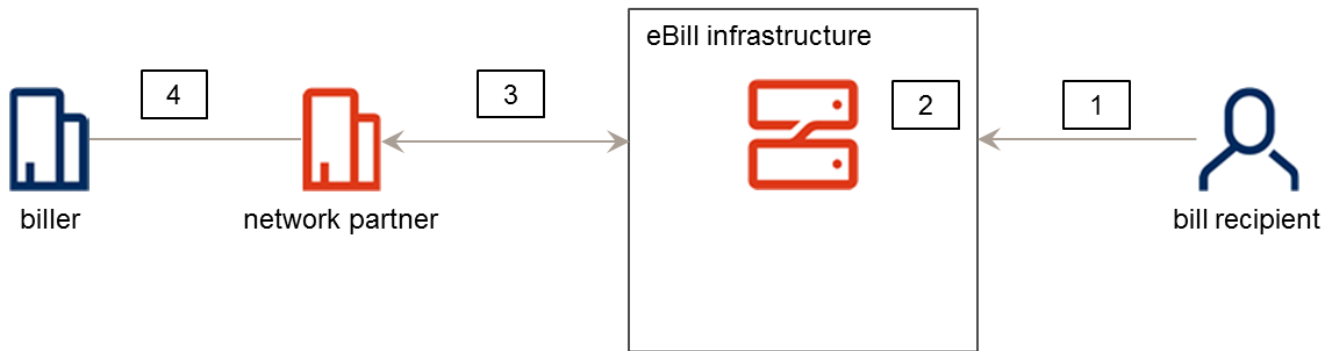


Figure 16. Subscription cancellation initiated by bill recipient

1. A bill recipient clicks "unsubscribe" in the eBill portal.
2. The eBill infrastructure sets the biller relation to **INACTIVE** and creates a subscription changed event with the necessary information.
3. The network partner polls for new subscription cancellation events...

See the resource definition: [Section 5.3.3, "Find events for bill recipient subscriptions which changed status"](#)

4. ...and receives the subscription cancellation data as a response.

4.4. Donation inquiries

Starting with v5 a new business case type 'DonationInquiry' can be submitted via the Network Partner API. This allows non-profit organizations such as clubs, associations or other charities to send donation inquiries to their members directly via eBill. Non-profit organizations are registered in the eBill infrastructure as billers which have been granted the permission to submit donation inquiries by the network partner. They operate as regular billers that can also submit regular bills like membership fees to their members. See [Section 4.1, "Biller management"](#) for more details.

DonationInquiry

Within the business case 'DonationInquiry' different donation purposes and donation amounts may be proposed (see element 'donationInquiry' in eBill-SIX_v5.xsd for more details). Here are examples of possible proposed donation purposes and donation amounts.

a. Proposed donation purposes

For a non-profit organization which focuses on preserving the nature, the proposed donation purposes could be:

1. Species protection
2. Woods
3. Oceans

b. Proposed donation amounts

1. 10 CHF
2. 50 CHF

3. 100 CHF

This offers the donor the flexibility to choose for what and how much she or he wants to donate. Both the proposed donation purposes and donation amounts are optional elements. Submitting proposed donation purposes does not imply the submission of proposed donation amounts and vice versa. If within a donation inquiry proposed donation amounts have been specified, the total and payment information amount must be omitted and vice versa. All the amount values within a donation inquiry must fulfill the requirements specified for donation inquiry amounts (see element 'donationInquiry' in eBill-SIX_v5.xsd for more details).

Donors are bill recipients that have all the options described in [Section 4.3, “Subscriptions and subscription cancellations”](#) to subscribe to non-profit organizations and receive donation inquiries via eBill.

Two possible use cases are described below:

Donation portal

1. The donor is on a donation portal and decides to make a contribution for a specific donation purpose. The donor then determines the donation amount and chooses eBill as the payment method. This initiates a biller subscription process which is equivalent to the one described in [Section 4.3.2.2, “Subscription at the biller”](#).
2. Once the subscription to the biller (in this case the non-profit organization) is complete, the network partner submits the donation inquiry into the eBill infrastructure. The donation inquiry contains the donation purpose and amount selected by the donor before. In this case, no other donation purposes and donation amounts will be proposed. By providing a QR-Reference for the donation inquiry, the non-profit organization can easily match the donation afterwards.
3. In the next step, the donation inquiry will be displayed to the bill recipient in the online banking of his/her financial institution or in the eBill portal. After approving the donation inquiry by the bill recipient (donor), the eBill infrastructure generates a network partner event enriched with the 'externalDonationPurposeId' if a proposed donation purpose has been selected by the donor.

In this way, the donor subscribes to the non-profit organization to receive also future donation inquiries and regular bills directly via eBill.

Fundraising campaigns

Fundraising campaigns are used by non-profit organizations to draw attention to their missions, particularly to programs or to initiatives for which they are currently soliciting donations. Within the donation inquiries arising during such a campaign, these missions can be represented in the proposed donation purposes like 'Species protection', 'Woods', 'Oceans', etc. Non-profit organizations may also propose different donation amounts and so give everybody the opportunity to make a small contribution. As described in [Section 4.3.2.1, “Look-Up”](#), non-profit organizations may search for potential donors using the Network Partner API, if the bill recipient has given prior consent to use its eBill identification (e.g. its email address) to them.

4.5. eBill Direct Debit

With eBill Direct Debit, the biller is provided with functionalities to setup and manage debit mandates via the eBill infrastructure and automatically collect them on the bill recipient side (by an eBill Direct Debit standing approval). The bill recipient has the option to request the chargeback of a processed eBill Direct Debit payment.

eBill Direct Debit can be used for recurring collections from classic contractual relationships to service- or consumption bills such as rent, leasing, credit card bills, electricity bills, etc., as well as for one-time claims.

Before an eBill Direct Debit payment is possible, an eBill Direct Debit standing approval has to exist and can be created as following:

1. Private eBill user
 - a. Creates an eBill Direct Debit standing approval by itself
 - b. Biller sends an eBill Direct Debit proposal, which can be accepted by the user (bill recipient)
2. Business eBill user
 - a. Biller sends an eBill Direct Debit proposal, which can be accepted by the user (bill recipient)

On existence of an eBill Direct Debit standing approval, the biller can start submitting bills with eBill Direct Debit payment, which are checked by the eBill infrastructure against the eBill Direct Debit mandate.

The eBill infrastructure processes the bill and transmits the eBill Direct Debit payment information to the bill recipient's financial institution, that debits the bill recipients's account and processes the payment in favor of the biller's account.

The bill recipient has the option to chargeback the debited amount within a period of 60 days after being notified about the payment.

4.5.1. Enabling biller for eBill Direct Debit

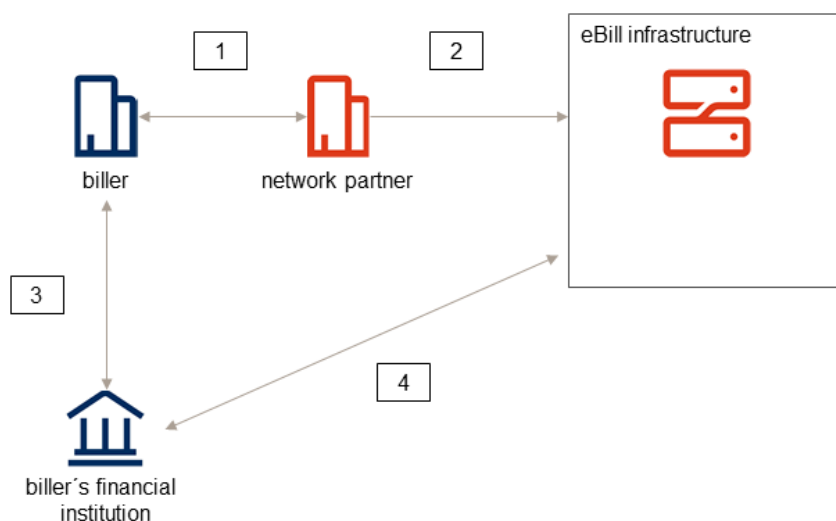


Figure 17. Enabling biller for eBill Direct Debit

Prerequisites: Network partner is enabled for eBill Direct Debit.

1. eBill Direct Debit contractual basis between biller and network partner established
2. Primary network partner creates or updates biller and sets the ebillDebitSupport to enabled (now it is possible to submit eBill Direct Debit proposals to bill recipients, see [Section 4.5.2, “Submit an eBill Direct Debit proposal”](#))
3. eBill Direct Debit contractual basis between biller and biller’s financial institution established
4. Biller’s financial institution enables biller for eBill Direct Debit in eBill infrastructure and defines an eBill Direct Debit submission limit for the biller and currency (now one or more biller accounts are enabled for eBill Direct Debit and it is possible to submit bills with eBill Direct Debit payment)

4.5.2. Submit an eBill Direct Debit proposal

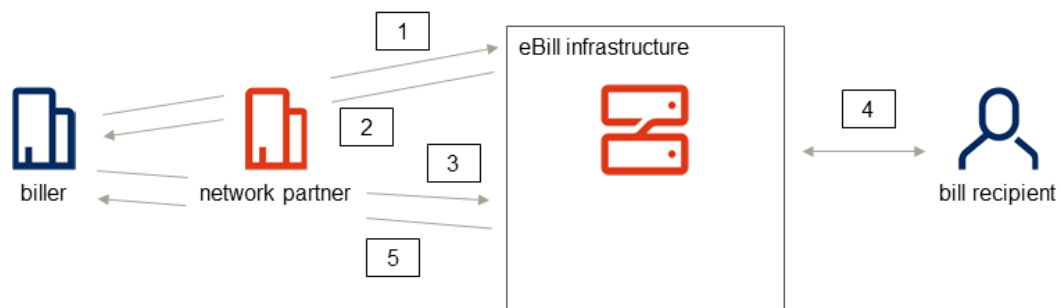


Figure 18. Submit an eBill Direct Debit proposal

Prerequisites: Network partner and biller are enabled for eBill Direct Debit.

1. Biller checks whether it is allowed to submit an eBill Direct Debit proposal by calling [Section 5.1.11, “Search for multiple bill recipients for this biller”](#)
2. In the response ebillDebitProposalStatus must be «allowed»
3. Biller submits eBill Direct Debit proposal (see [Section 5.6.1, “Propose eBill Direct Debit to the bill recipient”](#))
4. Bill recipient receives eBill Direct Debit proposal and accepts it (by accepting the proposal a new eBill Direct Debit standing approval gets created, additionally if a subscription is missing it is created on the fly [Section 4.3, “Subscriptions and subscription cancellations”](#). Submissions of bills with eBill Direct Debit payment are possible, see [Section 4.5.4, “Submit a bill with eBill Direct Debit payment”](#))
5. On acceptance the biller gets notified by an event (see allowedEbillDebitSubmissions in [Section 5.3.3, “Find events for bill recipient subscriptions which changed status”](#))

4.5.3. Submit an eBill Direct Debit proposal (opt-in)

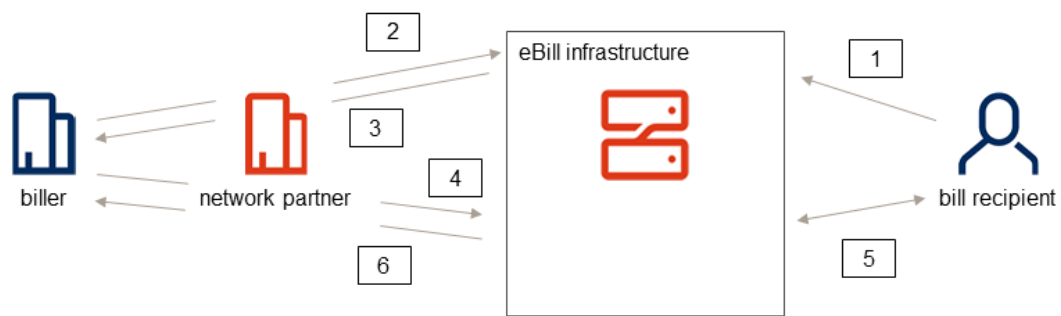


Figure 19. Submit an eBill Direct Debit proposal (opt-in)

Prerequisites: Network partner and biller are enabled for eBill Direct Debit.

1. Bill recipient enables «opt-in» for eBill Direct Debit
2. Biller checks whether it is allowed to submit a proposal to the bill recipient via Look-Up, see [Section 4.3.2.1, “Look-Up”](#)
3. In the response ebillDebitProposalStatus must be «allowed»
4. Biller submits eBill Direct Debit proposal (see [Section 5.6.1, “Propose eBill Direct Debit to the bill recipient”](#))
5. Bill recipient receives eBill Direct Debit proposal and accepts it (by accepting the proposal a new eBill Direct Debit standing approval gets created and submissions of bills with eBill Direct Debit payment are possible, see [Section 4.5.4, “Submit a bill with eBill Direct Debit payment”](#))
6. On acceptance the biller gets notified by an event (see allowedEbillDebitSubmissions in [Section 5.3.3, “Find events for bill recipient subscriptions which changed status”](#))

4.5.4. Submit a bill with eBill Direct Debit payment

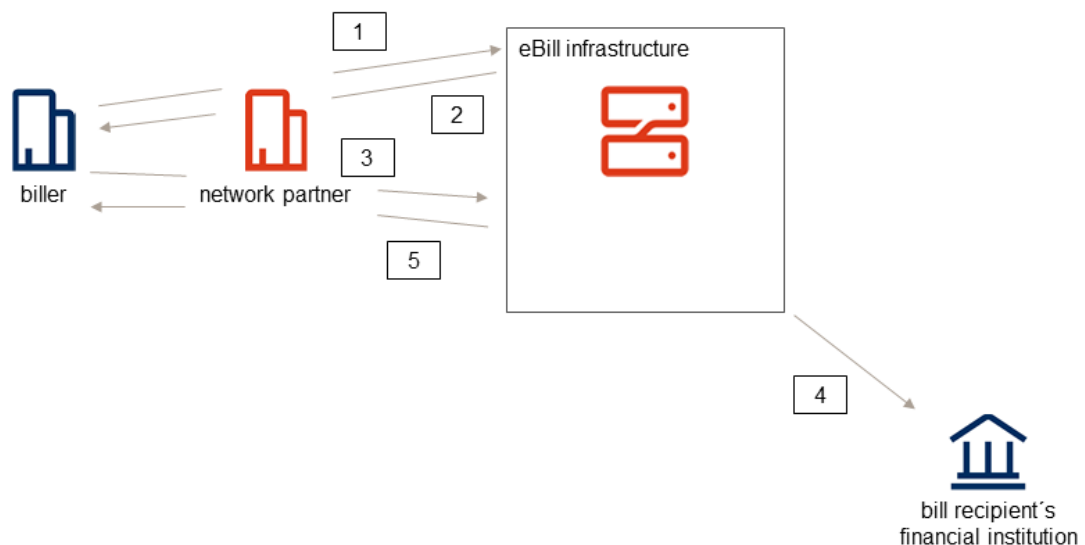


Figure 20. Submit a bill with eBill Direct Debit payment

Prerequisites: Network partner, biller and biller account are enabled for eBill Direct Debit.

1. Biller checks whether it is allowed to submit a bill with eBill Direct Debit payment by calling [Section 5.1.11, “Search for multiple bill recipients for this biller”](#)
2. In the response `allowedEbillDebitSubmissions` must be available with the correct currency chargebackMode combination (means there is an existing appropriate eBill Direct Debit standing approval)
3. Biller submits bill with eBill Direct Debit payment (see [Section 5.1.12, “Create business case in PDF/A-3b-format”](#))
4. eBill Direct Debit payment will be automatically approved and a payment instruction is delivered to the bill recipient’s financial institution
5. Biller gets notified by an event about the changed business case status, see [Section 5.3.1, “Find events for business cases which changed status”](#)

Chapter 5. Resources

5.1. Billers

Biller Management

All operations that are associated with a biller are located within the biller resource, including the creation of business cases.

The operations are designed to be self contained, because of this, the data objects can be rather large. However this allows for complete validation and avoids chains of calls that depend on each other.

5.1.1. Create a biller

POST /billers

Description

This operation attempts to create a new biller in the system.

Before creating a new biller, a series of validations are executed (these rules apply even if the biller exists in INACTIVE status, except for duplicate checks, which only apply for billers in ACTIVE status).

If all validations pass, the biller is created and the response will contain the newly created biller ID. If there is a conflict with an existing biller, its biller ID is provided in the technicalReason of the error response.

The network partner that successfully creates a biller becomes the primary network partner for this biller.

All other network partners will immediately be able to see a limited set of data of the the newly created biller.

The following diagram gives an overview of the organizational and technical steps of the biller creation:

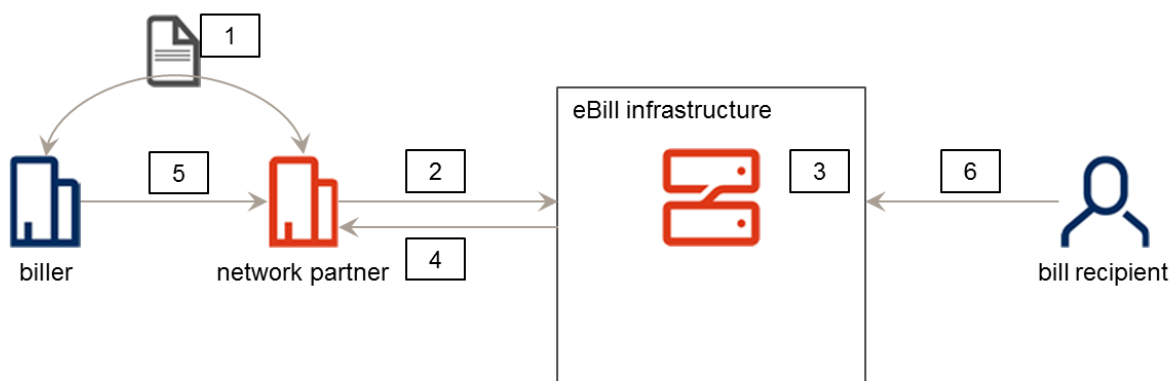


Figure 21. Create a biller

1. Biller signs a contract with the network partner to use his services.
2. The network partner creates the biller on the eBill infrastructure using the operation [Section 5.1.1, “Create a biller”](#).
3. The eBill infrastructure validates the request.
4. After a successful validation, the API returns a new biller ID.
5. The biller can start to use the eBill services via its network partner as soon as the registration is completed (e.g. searching bill recipients, creating business cases).
6. Bill recipients can search for the newly created biller and subscribe with it.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and [Section 7.4, “Biller Validations”](#) except [INVALID_ASSET_ID](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Data for the biller to be added	BillerDetail

Responses

HTTP Code	Description	Schema
201	Biller created	BillerDetail
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.1.2. Search billers

POST /billers/search

Description

A public listing of billers in the eBill infrastructure, refined by filter, limit and offset parameters. Can be used by all network partners to retrieve public data about each biller.

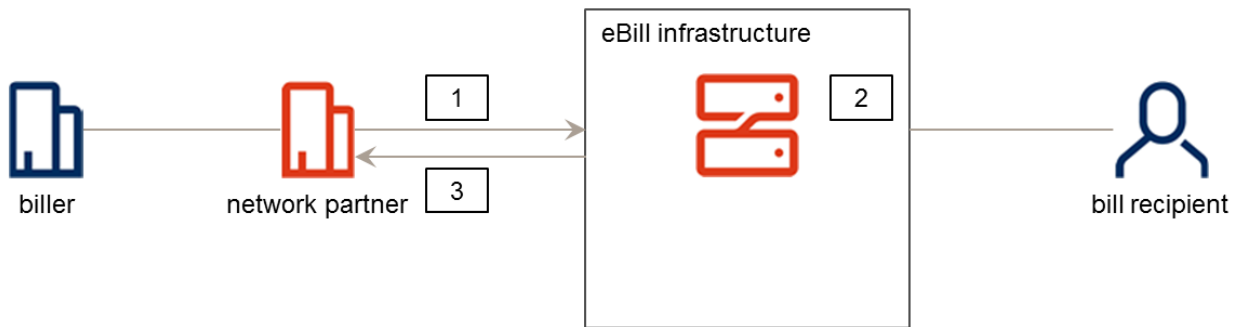


Figure 22. Search billers

1. A network partner sends the request to the eBill infrastructure
2. The eBill infrastructure finds billers (active + inactive)
3. and returns the answer to the network partner.

Validations

All validations according to [Section 7.1, “Basic Validations”](#).

Parameters

Type	Name	Description	Schema	Default
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string	
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string	
Query	limit <i>optional</i>	Maximum number of items to be returned (technical maximum is 1000, no more will be returned).	integer	100

Type	Name	Description	Schema	Default
Query	offset <i>optional</i>	Indicates the distance between the first existing item (offset=0) and the first item to be returned.	integer	0
Body	filter <i>required</i>	Filter object for search action	BillerSearchFilter	

Responses

HTTP Code	Description	Schema
200	Response object containing billers matching the filter, limit and offset parameters.	BillerSearchResult
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.1.3. Get biller by id

```
GET /billers/{billerId}
```

Description

Get all information for a specific biller. Calls are validated and only network partners are allowed to retrieve information for billers where they are assigned as primary network partner.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
200	Biller found	BillerDetail
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.1.4. Update biller

```
PUT /billers/{billerId}
```

Description

Updates a biller with the given biller details. Only the primary network partner of a biller is allowed to update biller details.

Note that if the status of the biller is set to **INACTIVE** the biller cannot submit business cases any more and it will be removed from the biller listing in the eBill infrastructure.

Validations

All validations according to [Section 7.1, “Basic Validations”](#), [Section 7.4, “Biller Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Biller, which should be updated	BillerDetail

Responses

HTTP Code	Description	Schema
200	Biller updated	BillerDetail
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.1.5. Get asset data by id

```
GET /billers/{billerId}/assets/{assetId}
```

Description

Get the binary data for a specific asset. Depending on the content type of the asset, it produces the response accordingly. Only the primary network partner of a biller is allowed to manage its assets.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Path	assetId <i>required</i>	asset id Maximal length: 14 Example: "ASID0000000001" Pattern: ASID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
200	Biller asset found	string (binary)
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `image/jpeg`
- `image/png`
- `image/gif`
- `application/pdf`

5.1.6. Add/update asset

```
PUT /billers/{billerId}/assets/{assetId}
```


Description

Update the asset's binary data or adds new binary data to the asset. Depending on the accept header the corresponding content type is stored along with the asset's binary data.

Note the different content types that are accepted: Biller logo assets have to be one of image/jpeg, image/png or image/gif.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and [Section 7.5, “Asset Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

For image assets:

[ASSET_IMAGE_INVALID](#)

[ASSET_MIME_TYPE_DOES_NOT_CORRESPOND_TO_CONTENT_TYPE](#)

[ASSET_IMAGE_EXCEEDS_MAXIMUM_SIZES](#)

[ASSET_CONTENT_TYPE_MUST_BE_IMAGE](#)

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Path	assetId <i>required</i>	asset id Maximal length: 14 Example: "ASID0000000001" Pattern: ASID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Binary of biller asset	string (binary)

Responses

HTTP Code	Description	Schema
204	Biller asset updated	No Content
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `image/jpeg`
- `image/png`
- `image/gif`

5.1.7. Delete asset

```
DELETE /billers/{billerId}/assets/{assetId}
```

Description

Delete the asset, this includes its binary and content type data. Only the primary network partner of a biller is allowed to manage its assets.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: <code>BIID[0-9]{10}</code>	string
Path	assetId <i>required</i>	asset id Maximal length: 14 Example: "ASID0000000001" Pattern: <code>ASID[0-9]{10}</code>	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: <code>NWID[0-9]{10}</code>	string

Type	Name	Description	Schema
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
204	Biller asset deleted	No Content
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

5.1.8. Get the custom subscription form of a biller

```
GET /billers/{billerId}/subscription-form
```

Description

Returns the custom subscription form of a biller.

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
200	The custom subscription form of this biller	BillerSubscription Form
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.1.9. Create or update the custom subscription form of a biller

```
PUT /billers/{billerId}/subscription-form
```

Description

Only the primary network partner of the biller may maintain the subscription form. Since there may be only one form per biller, the endpoint url not only specifies the intent but also identifies the resource. A separate post-command is therefore omitted.

Details regarding the subscription processes in general can be found in [Section 4.3, “Subscriptions and subscription cancellations”](#). A description of bill recipient subscriptions with custom subscription form is in [Section 4.3.1, “Bill recipient driven subscription”](#).

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and the problem descriptions in [Section 7.6, “Custom Subscription Form Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string

Type	Name	Description	Schema
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Biller subscription form to create or update.	BillerSubscriptionForm

Responses

HTTP Code	Description	Schema
200	Biller subscription form created or updated	BillerSubscriptionForm
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.1.10. Delete the subscription form of a biller

```
DELETE /billers/{billerId}/subscription-form
```

Description

Deletes the subscription form. Only the primary network partner of the biller may call this endpoint.

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
204	Biller subscription form deleted	No Content
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

5.1.11. Search for multiple bill recipients for this biller

```
POST /billers/{billerId}/bill-recipients/search
```

Description

For each item in the request body, the endpoint will respond if the biller is allowed to submit a business case for a bill recipient with the provided id.

In addition to the provided id, the response will always contain the billRecipientId, if a business case submission is allowed.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Parameters for the search, at least one id per entry is required	BillRecipientsForBillerSearchRequest

Responses

HTTP Code	Description	Schema
200	Search successfully executed	BillRecipientsForBillerSearchResponse
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.1.12. Create business case in PDF/A-3b-format

```
POST /billers/{billerId}/business-cases
```

Description

The creation of a business case in PDF/A-3b format works with a simple POST request where the PDF is the binary payload of the request. The PDF needs to hold an embedded attachment in XML format describing

the business case according to the eBill-SIX_v5.xsd schema definition. Note that only one PDF can be submitted at a time.

The submitted payload has to comply to the standard conformance level PDF/A-3b whose set of requirements, restrictions and rules are being validated immediately upon submission.

The response either confirms the successful creation of the business case within the eBill system or presents the respective error details for the failed submission.

Technical Hint:

Note that this POST operation is not officially supported by the specification language 'OAS 2.0' ('Swagger 2.0') as described in detail here → '<https://swagger.io/docs/specification/2-0/file-upload/>'

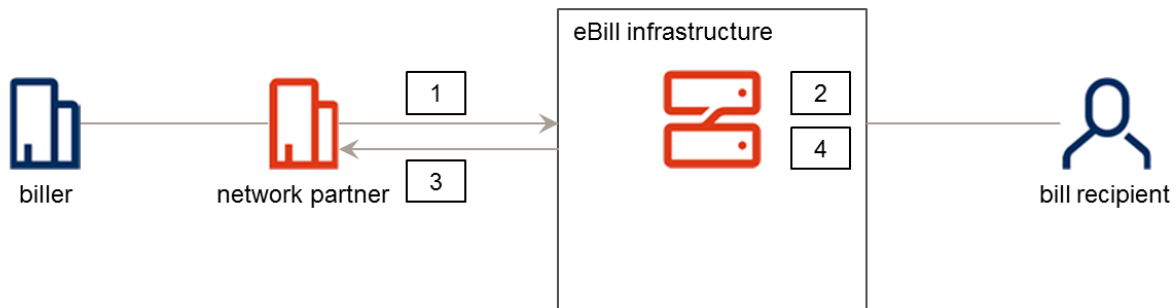


Figure 23. Create a business case

1. A network partner submits a business case on behalf of a biller.
2. Processing takes place in the eBill infrastructure.
3. The response contains the generated business case id.
4. As of this moment the business case is visible to bill recipients.

Addressing of a Business Case to a Bill Recipient

A submitted business case can be addressed to a bill recipient by the bill recipient's email address or by its billRecipientId or by its swiss enterprise identification number (UID) from the commercial register (Handelsregister). Business cases addressing a bill recipient by its email address can be successfully submitted even if the bill recipient's email address has been changed in the meantime (at least for a limited time). For further details see: [Section 5.3.4, "Find events for bill recipients email address changes"](#)

Validations

All validations according to [Section 7.1, "Basic Validations"](#), [Section 7.8, "Business Case Validations"](#), [Section 7.9, "Business Case Payment Validations"](#), [Section 7.10, "Business Case Instalment Validations"](#), [Section 7.11, "Business Case Donation Inquiry Validations"](#), [Section 7.13, "PDF Validations"](#) and some of [Section 7.2, "Shared Validations"](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Header	x-filename <i>optional</i>	Filename for the business case PDF. This is only used for analytical purposes and support. The filename is not visible for the bill recipient. Minimal length: 1 Maximal length: 99	string
Header	x-anomaly-detection <i>optional</i>	If the optional header is provided with the value 'SKIP', the anomaly detection does not prevent business case submission	string
Body	body <i>required</i>	PDF as binary data	string (binary)

Responses

HTTP Code	Description	Schema
201	Business case created	BusinessCase
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/pdf`

Produces

- `application/json`

5.1.13. Get business case

```
GET /billers/{billerId}/business-cases/{businessCaseId}
```

Description

Depending on the accept header, this operation either returns a JSON business case object or the original PDF.

The returned JSON object contains one of the business case subtypes: Bill, InstalmentBill, Reminder, CreditNote, Advice or DonationInquiry.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Path	businessCaseId <i>required</i>	business case id Maximal length: 36 Example: "BCID0FB909852BBC4D06AD8336AAE87D7FC9" Pattern: BCID[0-9A-Z]{32}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
200	Business case found	BusinessCase
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

- [application/pdf](#)

5.2. Biller Driven Subscription

Biller driven subscription

Further information about subscription processes can be found in [Section 4.3.2, “Biller driven subscription”](#).

5.2.1. Initiate bill recipient subscription

```
POST /billers/{billerId}/bill-recipient-subscription-initiations
```

Description

This endpoint initiates the [Section 4.3.2.2, “Subscription at the biller”](#) process, returns a subscription token to the caller and sends a verification code to the eBill subscribers email address. Only the primary network partner of a biller is allowed to initiate the subscription at the biller.

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	email address for bill recipient subscription initiation	SubscriptionInitiationEmailAddress

Responses

HTTP Code	Description	Schema
201	subscription initiation token which was send to the user	SubscriptionInitiationToken

HTTP Code	Description	Schema
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.2.2. Confirm bill recipient subscription initiation

```
PUT /billers/{billerId}/bill-recipient-subscription-
initiations/{subscriptionInitiationToken}
```

Description

This endpoint verifies the combination of `subscriptionInitiationToken` and activation code. If the combination is valid, it creates an active biller relation.

Only the primary network partner of a biller is allowed to confirm the subscription at the biller initiation.

More details regarding bill recipient subscriptions can be found in [Section 4.3.2.2, “Subscription at the biller”](#).

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and the "INVALID_TOKEN_AND_ACTIVATION_CODE_COMBINATION" validation from [Section 7.14, “Subscription Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Path	subscriptionInitiationToken <i>required</i>	subscription initiation token	string
Body	body <i>required</i>	activation code send to the user through	SubscriptionInitiationActivationCode

Responses

HTTP Code	Description	Schema
200	Bill recipient relation initiated	BillRecipient
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.2.3. Initiate a bill recipient subscription and return the URL to the eBill infrastructure

```
POST /billers/{billerId}/bill-recipient-subscription-initiations-url
```

Description

This endpoint initiates a bill recipient subscription. It returns an URL pointing to the subscription initiation at the eBill infrastructure, where the subscription may be concluded. Only the primary network partner of a biller is allowed to initiate a subscription.

More details regarding this type of bill recipient subscription can be found in [Section 4.3.2.3, "Subscription at the eBill infrastructure"](#).

Validations

All validations according to [Section 7.1, “Basic Validations”](#), [Section 7.4, “Biller Validations”](#) and [Section 7.7, “Custom Subscription Form Data Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	biller id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	bill recipient identifying information for the subscription initiation at the eBill infrastructure	BillRecipientURLSubscription

Responses

HTTP Code	Description	Schema
201	bill recipient personalized URL to subscribe through the eBill infrastructure	string (uri)
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.3. Events

Notification Events

The event resource allows the network partner to retrieve all changes addressed to it. There is one

specific operation for every type of event (for example: Subscription status changes or business case status changes), where the network partner can pull new changes from.
More details can be found in [Section 3.3.11, “Guidelines for polling the events”](#).

5.3.1. Find events for business cases which changed status

```
GET /events/business-case-status-changed
```

Description

Events for status changes of bills, reminders and donation inquiries.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema	Default
Query	lastEventId <i>optional</i>	Id of the last received event. If omitted, the result starts with the oldest available event. Example: "EVID82A65938766547EBBBA39BA6F7B07F24" Maximal length: 36 Pattern: "EVID[0-9A-Z]{32}"	string	
Query	limit <i>optional</i>	Maximum number of items to be returned (technical maximum is 1000, no more will be returned).	integer	100
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string	
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string	

Responses

HTTP Code	Description	Schema
200	Business case status changed events found	< BusinessCaseStatusChangedEvent > array
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- [application/json](#)

5.3.2. Find events for instalments which changed status

```
GET /events/instalment-status-changed
```

Description

Events for status changes of instalment bills only.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema	Default
Query	lastEventId <i>optional</i>	Id of the last received event. If omitted, the result starts with the oldest available event. Example: "EVID82A65938766547EBBBA39BA6F7B07F24" Maximal length: 36 Pattern: "EVID[0-9A-Z]{32}"	string	
Query	limit <i>optional</i>	Maximum number of items to be returned (technical maximum is 1000, no more will be returned).	integer	100
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string	

Type	Name	Description	Schema	Default
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string	

Responses

HTTP Code	Description	Schema
200	Instalment status changed events found	< InstalmentStatusC hangedEvent > array
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.3.3. Find events for bill recipient subscriptions which changed status

```
GET /events/bill-recipient-subscription-status-changed
```

Description

More information about the subscription process can be found in [Section 4.3, “Subscriptions and subscription cancellations”](#).

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema	Default
Query	lastEventId <i>optional</i>	Id of the last received event. If omitted, the result starts with the oldest available event. Example: <code>"EVID82A65938766547EBBBA39BA6F7B07F24"</code> Maximal length: 36 Pattern: <code>"EVID[0-9A-Z]{32}"</code>	string	

Type	Name	Description	Schema	Default
Query	limit <i>optional</i>	Maximum number of items to be returned (technical maximum is 1000, no more will be returned).	integer	100
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string	
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string	

Responses

HTTP Code	Description	Schema
200	Bill recipient subscription changed events found	< BillRecipientSubscriptionStatusChangedEvent > array
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- [application/json](#)

5.3.4. Find events for bill recipients email address changes

```
GET /events/bill-recipient-email-address-changed
```

Description

This event is triggered after a biller has submitted a business case with an outdated, so called historically available email address. It notifies about the changed email address of a bill recipient, which has been adjusted in eBill.

An email address is considered to be historically available if it was present up to 15 months prior to the submission time.

The billers are able to submit business cases with historically available email addresses of a bill recipient. However, latest 15 months after the email address changed, the billers are required to

submit the business cases with the currently valid email address of the bill recipient.

Validations

All validations according to [Section 7.1, “Basic Validations”](#) and some of [Section 7.2, “Shared Validations”](#).

Parameters

Type	Name	Description	Schema	Default
Query	lastEventId <i>optional</i>	Id of the last received event. If omitted, the result starts with the oldest available event. Example: "EVID82A65938766547EBBBA39BA6F7B07F24" Maximal length: 36 Pattern: "EVID[0-9A-Z]{32}"	string	
Query	limit <i>optional</i>	Maximum number of items to be returned (technical maximum is 1000, no more will be returned).	integer	100
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string	
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string	

Responses

HTTP Code	Description	Schema
200	Bill recipient email address changed events found	< BillRecipientEmailAddressChangedEvent > array
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.4. Sectors

Industry Sector

Industry sectors are valid system wide. Each biller will reference one or several industry sectors.

5.4.1. Find Sectors

```
GET /sectors
```

Description

Get the industry sector list of the eBill infrastructure. The sectors are more or less static and can therefore be cached.

Validations

All validations according to [Section 7.1, “Basic Validations”](#).

Parameters

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
200	Sectors found	< Sector > array
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.5. Bill-recipients

Bill Recipients

This resource can be used to verify the existence of a specific billRecipientId in the eBill infrastructure.

5.5.1. Get bill recipient by id

```
GET /bill-recipients/{billRecipientId}
```

Description

The network partner can verify if a bill recipient is part of the eBill infrastructure by its billRecipientId.

The call will return the billRecipientId if the bill recipient has been found in the eBill infrastructure. Receiving a billRecipientId does not necessary mean that the corresponding bill recipient is an active eBill user, it only means that this billRecipientId exists in the eBill infrastructure.

Validations

All validations according to [Section 7.1, “Basic Validations”](#).

Parameters

Type	Name	Description	Schema
Path	billRecipientId <i>required</i>	id of the bill recipient Maximal length: 17 Example: "41010560425610173" Pattern: ([0-9])*	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
200	Bill recipient found	BillRecipientById

HTTP Code	Description	Schema
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.6. EBill Direct Debit

eBill Direct Debit

Operations that are associated with eBill Direct Debit.

More details can be found in [Section 4.5, “eBill Direct Debit”](#).

5.6.1. Propose eBill Direct Debit to the bill recipient

```
POST /billers/{billerId}/ebill-debit-proposal
```

Parameters

Type	Name	Description	Schema
Path	billerId <i>required</i>	billers id Maximal length: 14 Example: "BIID0000123456" Pattern: BIID[0-9]{10}	string
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Definition of the eBill Direct Debit proposal	EbillDebitProposal

Responses

HTTP Code	Description	Schema
201	eBill Direct Debit proposal created	EbillDebitProposal

HTTP Code	Description	Schema
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.7. Healthcheck

System Healthcheck

This allows to check the basic state of the system (can it be reached, does authentication and authorization work, does it respond).

As some infrastructures block certain HTTP methods by default, the healthcheck allows to test if all required methods (GET, POST, PUT, DELETE) work across all layers.

5.7.1. Health check using GET method

```
GET /healthcheck
```

Description

Returns a status message of the system.

Validations

All validations according to [Section 7.1, “Basic Validations”](#).

Parameters

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
200	Healthcheck successful	HealthCheckResponse
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Produces

- `application/json`

5.7.2. Health check using PUT method

PUT /healthcheck

Description

Returns the request body. This operation will not modify the system.

Validations

All validations according to [Section 7.1, “Basic Validations”](#).

Parameters

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Any message which is expected in the response	HealthCheckRequest

Responses

HTTP Code	Description	Schema
200	Healthcheck successful	HealthCheckResponse
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.7.3. Health check using POST method

POST /healthcheck

Description

Returns the request body. This operation will not modify the system.

Validations

All validations according to [Section 7.1, “Basic Validations”](#).

Parameters

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string
Body	body <i>required</i>	Any message which is expected in the response	HealthCheckRequest

Responses

HTTP Code	Description	Schema
200	Healthcheck successful	HealthCheckResponse
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Consumes

- `application/json`

Produces

- `application/json`

5.7.4. Health check using DELETE method

DELETE /healthcheck

Description

Returns without further content. This operation will not modify the system.

Validations

All validations according to [Section 7.1, “Basic Validations”](#).

Parameters

Type	Name	Description	Schema
Header	x-networkpartner-id <i>required</i>	ID which will identify the calling network partner. Minimal length: 14 Maximal length: 14 Example: "NWID0000123456" Pattern: NWID[0-9]{10}	string
Header	x-correlation-id <i>required</i>	ID which will unambiguously identify this request to the API. Minimal length: 1 Maximal length: 36	string

Responses

HTTP Code	Description	Schema
204	Healthcheck successful	No Content
default	See section Error Handling of the Network Partner API documentation for further details about errors and error handling.	Problem

Chapter 6. Definitions

6.1. BillerDetail

Polymorphism : Composition

Name	Description	Schema
id <i>optional</i>	A unique id for this biller. Property must not be given when creating a new biller. Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
status <i>required</i>	the status of the biller	enum (ACTIVE, INACTIVE)
legalName <i>required</i>	legal name of the company which is displayed in the invoice overview, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 70 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "Verlag Neuer"	string
enterpriseIdentificationNumber <i>optional</i>	Swiss enterprise identification number (UID) without dashes, dots or extension. Note that this has to contain the swiss enterprise identification number (UID) from the commercial register (Handelsregister) which may be different from the VAT UID (Mehrwertsteuer UID). Maximal length : 12 Pattern : "CHE[0-9]{9}" Example : "CHE123456789"	string
billRecipientSubscriptionStatus <i>required</i>		BillRecipientSubscriptionStatus
localizedData <i>required</i>		localizedData
sectorIds <i>required</i>	list of assigned sector ids to the biller	< string > array
allowedToSubmitDonationInquiries <i>required</i>	States whether the biller is allowed to submit donation inquiries. Only billers that have been verified to be non-profit organisations (NPO) may be granted this permission. Default : false	boolean

Name	Description	Schema
ebillDebitSupport <i>required</i>	Defines whether the biller supports eBill Direct Debit. Only billers that support eBill Direct Debit are allowed to create eBill Direct Debit proposals and submit bills with eBill Direct Debit payment. Only network partners that support eBill Direct Debit themselves are allowed to enable eBillDebitSupport. As soon as the biller is enabled for eBill Direct Debit, it is possible to submit eBill Direct Debit proposals. For details see Section 4.5, “eBill Direct Debit” part in the documentation.	enum (ENABLED, DISABLED)
billerDirectSubscriptionSupport <i>required</i>	Defines whether the biller supports direct subscription. Default : "ENABLED"	enum (ENABLED, DISABLED)
certificationIds <i>optional</i>	List of assigned certification ids.	< string > array
accounts <i>required</i>	list of biller accounts	< BillerAccount > array
anomalyConfiguration <i>required</i>	The configuration how anomalies should be detected. If no configuration is provided, the platforms default will be applied	AnomalyDetectionConfig

localizedData

Name	Schema
defaultLanguage <i>required</i>	DefaultLanguage
ger <i>optional</i>	LocalizedBillerData
fre <i>optional</i>	LocalizedBillerData
ita <i>optional</i>	LocalizedBillerData
eng <i>optional</i>	LocalizedBillerData

6.2. Biller

Name	Description	Schema
id <i>optional</i>	A unique id for this biller. Property must not be given when creating a new biller. Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
status <i>required</i>	the status of the biller	enum (ACTIVE, INACTIVE)
legalName <i>required</i>	legal name of the company which is displayed in the invoice overview, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 70 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "Verlag Neuer"	string
enterpriseIdentificationNumber <i>optional</i>	Swiss enterprise identification number (UID) without dashes, dots or extension. Note that this has to contain the swiss enterprise identification number (UID) from the commercial register (Handelsregister) which may be different from the VAT UID (Mehrwertsteuer UID). Maximal length : 12 Pattern : "CHE[0-9]{9}" Example : "CHE123456789"	string
billRecipientSubscriptionStatus <i>required</i>		BillRecipientSubscriptionStatus
localizedData <i>required</i>		localizedData
sectorIds <i>required</i>	list of assigned sector ids to the biller	< string > array
allowedToSubmitDonationInquiries <i>required</i>	States whether the biller is allowed to submit donation inquiries. Only billers that have been verified to be non-profit organisations (NPO) may be granted this permission. Default : false	boolean

Name	Description	Schema
eBillDebitSupport <i>required</i>	Defines whether the biller supports eBill Direct Debit. Only billers that support eBill Direct Debit are allowed to create eBill Direct Debit proposals and submit bills with eBill Direct Debit payment. Only network partners that support eBill Direct Debit themselves are allowed to enable eBillDebitSupport. As soon as the biller is enabled for eBill Direct Debit, it is possible to submit eBill Direct Debit proposals. For details see Section 4.5, “eBill Direct Debit” part in the documentation.	enum (ENABLED, DISABLED)
billerDirectSubscriptionSupport <i>required</i>	Defines whether the biller supports direct subscription. Default : "ENABLED"	enum (ENABLED, DISABLED)
certificationIds <i>optional</i>	List of assigned certification ids.	< string > array

localizedData

Name	Schema
defaultLanguage <i>required</i>	DefaultLanguage
ger <i>optional</i>	LocalizedBillerData
fre <i>optional</i>	LocalizedBillerData
ita <i>optional</i>	LocalizedBillerData
eng <i>optional</i>	LocalizedBillerData

6.3. LocalizedBillerData

Name	Description	Schema
displayName <i>required</i>	display name which will be used for presentation in the biller overview, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 100 Pattern : "[\u0020-\u007E\u0085\u00A0-\uD7FF\uE000-\uFDCF\uFDF0-\uFFFD]*" Example : "Neuers Neuste Nachrichten"	string

Name	Description	Schema
emailAddress <i>optional</i>	Email address to contact the biller. It is recommended to provide a specific address for questions concerning eBill, if available. Length : 1 - 256 Example : "nnn@nnn-verlag.info"	string (email)
phone <i>optional</i>	Phone number to contact the biller. It is recommended to provide a specific phone number for questions concerning eBill, if available. Both the countryCode and the nationalNumber are as defined in International Telecommunication Union (ITU) Recommendation E.164, without any leading zeros.	phone
website <i>optional</i>	Website to contact the biller. It is recommended to provide a specific website for questions concerning eBill, if available. The pattern restricts to word characters of the US-ASCII and some special characters. Length : 1 - 1000 Pattern : "[\\w .:/@?&=%-]*" Example : "http://www.nnn-verlag.info"	string
address <i>required</i>		BillerAddress
logo <i>optional</i>	Biller logos may be provided and will be scaled down to the necessary size. Contains the reference to a logo as binary from the /biller/{billerId}/assets endpoint. For create/update: if assetId is empty, a new assetId will be returned, were the logo can be uploaded.	logo

phone

Name	Description	Schema
countryCode <i>required</i>	Maximal length : 6 Maximum value : 999999 Example : 41	integer
nationalNumber <i>required</i>	Maximal length : 18 Example : 446681800	integer (int64)

logo

Name	Description	Schema
name <i>optional</i>	<p>the name of the asset</p> <p>for create/update: will be taken as asset name if provided the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues</p> <p>Length : 1 - 256</p> <p>Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*"</p> <p>Example : "company-logo-de"</p>	string
assetId <i>optional</i>	<p>asset id</p> <p>Maximal length : 14</p> <p>Pattern : "ASID[0-9]{10}"</p> <p>Example : "ASID0000000001"</p>	string

6.4. BillerAddress

Name	Description	Schema
streetName <i>required</i>	<p>street name, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues</p> <p>Length : 1 - 70</p> <p>Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*"</p> <p>Example : "Neustadtstrasse"</p>	string
buildingNumber <i>optional</i>	<p>building number, the pattern restricts to word characters of the US-ASCII, slashes, empty spaces and hyphens</p> <p>Length : 1 - 16</p> <p>Pattern : "[\\w/ -]*"</p> <p>Example : "20a"</p>	string
postalCode <i>required</i>	<p>postal code, the pattern restricts to word characters of the US-ASCII, empty spaces and hyphens</p> <p>Length : 1 - 9</p> <p>Pattern : "[\\w -]*"</p> <p>Example : "6025"</p>	string
city <i>required</i>	<p>city name, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues</p> <p>Length : 1 - 35</p> <p>Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*"</p> <p>Example : "Neudorf"</p>	string
countryCode <i>required</i>	<p>in format ISO 3166-1 alpha 2</p> <p>Maximal length : 2</p> <p>Pattern : "[A-Z]{2}"</p> <p>Example : "CH"</p>	string

6.5. RecipientAddress

Name	Description	Schema
streetName <i>required</i>	street name, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 70 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "Neustadtstrasse"	string
postalCode <i>required</i>	postal code, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 9 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "6025"	string
city <i>required</i>	city name, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 35 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "Neudorf"	string
countryCode <i>required</i>	in format ISO 3166-1 alpha 2 Maximal length : 2 Pattern : "[A-Z]{2}" Example : "CH"	string

6.6. BillerAccount

Name	Description	Schema
accountNumber <i>required</i>		AccountNumber
iid <i>required</i>	The required institution identification of the associated financial institute. Please use leading zeroes to reach the 5 digit value. Length : 5 Pattern : "[0-9]{5}" Example : "00100"	string

Name	Description	Schema
qrIbanAccountSupplement <i>optional</i>	<p>In case the account number is of type QR-IBAN the account supplement can be used as additional information to this account number for distinguishing billers using the same QR-IBAN. The combination of QR-IBAN and account supplement must be unique. The account supplement is used during business case validation (if an account supplement is defined the QR-Reference has to start with these exact digits) and during direct subscription from online banking (if an account supplement is defined the QR-Reference has to start with these exact digits to produce a match for direct subscription).</p> <p>Length : 6 Pattern : "[0-9]{6}" Example : "345924"</p>	string
currencyCode <i>required</i>	<p>currency code in ISO-4217</p> <p>Maximal length : 3 Pattern : "[A-Z]{3}" Example : "CHF"</p>	string
eBillDebitSupport <i>optional</i> <i>read-only</i>	<p>Represents whether the associated financial institution for the biller account has enabled the account for eBill Direct Debit.</p> <p>Property must not be given when creating or updating a biller account, this is considered a 'read only' property. As soon as the biller account supports eBill Direct Debit and the biller itself supports eBill Direct Debit, the network partner can submit bills with eBill Direct Debit payment. For details see Section 4.5, "eBill Direct Debit" part in the documentation.</p>	enum (ENABLED, DISABLED)

6.7. BillRecipientsForBillerSearchRequest

Name	Schema
items <i>required</i>	< BillRecipientIdentification > array

6.8. BillRecipientsForBillerSearchResponse

Name	Schema
items <i>required</i>	< BillRecipientsForBillerSearchResponseItem > array

6.9. BillRecipientsForBillerSearchResponseItem

Polymorphism : Composition

Name	Description	Schema
emailAddress <i>optional</i>	email address of the bill recipient Length : 1 - 256 Example : "peter@muster.ch"	string (email)
billRecipientId <i>optional</i>	id of the bill recipient Length : 1 - 17 Pattern : "([0-9])*" Example : "41010560425610173"	string
enterpriseIdentificationNumber <i>optional</i>	Swiss enterprise identification number (UID) without dashes, dots or extension. Note that this has to contain the swiss enterprise identification number (UID) from the commercial register (Handelsregister) which may be different from the VAT UID (Mehrwertsteuer UID). Maximal length : 12 Pattern : "CHE[0-9]{9}" Example : "CHE123456789"	string
birthDate <i>optional</i>	birth date of the bill recipient (ISO-8601 format). Note that this field is optional and is used to narrow down the search result Example : "2015-01-01"	string (date)
postalCode <i>optional</i>	postal code of the bill recipient, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Note that this field is optional and is used to narrow down the search result Length : 1 - 9 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "6025"	string
eBillSubmissionStatus <i>required</i>	Defines if the biller can submit a business-case for the provided bill recipient identification in the request (=ALLOWED) or if the bill recipient identification is either not known to the eBill infrastructure or the submission is not allowed (=NOT_ALLOWED).	enum (ALLOWED, NOT_ALLOWED)
eBillDebitProposalStatus <i>required</i>	Defines if the biller can submit an eBill Direct Debit proposal for the provided bill recipient id in the request (=ALLOWED) or if the bill recipient is either not known to the eBill infrastructure or the submission of an eBill Direct Debit proposal is not allowed (=NOT_ALLOWED).	enum (ALLOWED, NOT_ALLOWED)

Name	Description	Schema
allowedEbillDebitSubmissions <i>optional</i>	Defines what kind of eBill Direct Debit submissions are allowed.	< AllowedEbillDebitSubmission > array

6.10. BillRecipientIdentification

One property out of emailAddress, billRecipientId or enterpriseIdentificationNumber must be set. Optionally birthDate and postalCode can be set to narrow down the results.

Name	Description	Schema
emailAddress <i>optional</i>	email address of the bill recipient Length : 1 - 256 Example : "peter@muster.ch"	string (email)
billRecipientId <i>optional</i>	id of the bill recipient Length : 1 - 17 Pattern : "([0-9])*" Example : "41010560425610173"	string
enterpriseIdentificationNumber <i>optional</i>	Swiss enterprise identification number (UID) without dashes, dots or extension. Note that this has to contain the swiss enterprise identification number (UID) from the commercial register (Handelsregister) which may be different from the VAT UID (Mehrwertsteuer UID). Maximal length : 12 Pattern : "CHE[0-9]{9}" Example : "CHE123456789"	string
birthDate <i>optional</i>	birth date of the bill recipient (ISO-8601 format). Note that this field is optional and is used to narrow down the search result Example : "2015-01-01"	string (date)
postalCode <i>optional</i>	postal code of the bill recipient, the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Note that this field is optional and is used to narrow down the search result Length : 1 - 9 Pattern : "[\u0020-\u007E\u0085\u00A0-\uD7FF\uE000-\uFDCF\uFDF0-\uFFFD]" Example : "6025"	string

6.11. BillRecipientById

Name	Description	Schema
billRecipientId <i>optional</i>	id of the bill recipient Length : 1 - 17 Pattern : "([0-9])*" Example : "41010560425610173"	string

6.12. BillRecipientSubscription

Name	Description	Schema
billerId <i>required</i>	Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
billRecipient <i>required</i>		BillRecipient
accountNumber <i>optional</i>	Account number of the biller (e.g. iban), if provided from the financial institution Maximal length : 21	string
referenceStructured <i>optional</i>	QR or creditor reference number, if provided from the financial institution. Maximal length : 27 Pattern : "([a-zA-Z0-9])*" Example : "123456123456789012345678901"	string

6.13. BillRecipientURLSubscription

Name	Description	Schema
referenceStructured <i>optional</i>	If provided, must be of type QR reference number. Maximal length : 27 Pattern : "([a-zA-Z0-9])*" Example : "123456123456789012345678901"	string
subscriptionFormData <i>optional</i>		BillRecipientSubscriptionForm

6.14. BillRecipient

Name	Description	Schema
emailAddress <i>optional</i>	email address of the bill recipient Length : 1 - 256 Example : "peter@muster.ch"	string (email)

Name	Description	Schema
billRecipientId <i>required</i>	id of the bill recipient Length : 1 - 17 Pattern : "([0-9])*" Example : "41010560425610173"	string
enterpriseIdentificationNumber <i>optional</i>	Swiss enterprise identification number (UID) without dashes, dots or extension. Note that this has to contain the swiss enterprise identification number (UID) from the commercial register (Handelsregister) which may be different from the VAT UID (Mehrwertsteuer UID). Maximal length : 12 Pattern : "CHE[0-9]{9}" Example : "CHE123456789"	string
type <i>required</i>	the type of the bill recipient	enum (PRIVATE, COMPANY)
name <i>required</i>	last name, if private bill recipient company name, if company bill recipient the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 70 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "for private bill recipient: Muster, for company name: Muster AG"	string
firstName <i>optional</i>	first name, if private bill recipient empty, if company bill recipient the pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Maximal length : 35 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "Peter"	string
correspondenceLanguage <i>required</i>	language for correspondence with this bill recipient ISO-639-2/B Length : 1 - 3 Example : "ger"	string
address <i>required</i>		RecipientAddress

6.15. Bill

A business case of type Bill

Polymorphism : Inheritance

Discriminator : type

Name	Description	Schema
id <i>required</i>	A unique id for this business case. Property must not be given when creating a new business case. Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
type <i>required</i>	the type of the business case	enum (Bill, InstalmentBill, Advice, CreditNote, Reminder, DonationInquiry)
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
referenceNumber <i>required</i>	A business case reference given by the biller. Must be unique in combination with the billerId. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Length : 1 - 120 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "2018-123456-22"	string
referencedBill <i>optional</i>		ReferencedBill
businessCaseDate <i>required</i>	The business case date (ISO-8601 format), can not be more than 90 days in the past on the date it was created. Cannot be in the future. Example : "2017-12-22"	string (date)
status <i>optional</i>	the status of the business case	enum (OPEN, APPROVED, REJECTED, COMPLETED)
totalAmount <i>required</i>		OptionalAmountWithCurrency

6.16. InstalmentBill

A business case of type InstalmentBill

Polymorphism : Inheritance

Discriminator : type

Name	Description	Schema
id <i>required</i>	A unique id for this business case. Property must not be given when creating a new business case. Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
type <i>required</i>	the type of the business case	enum (Bill, InstalmentBill, Advice, CreditNote, Reminder, DonationInquiry)
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
referenceNumber <i>required</i>	A business case reference given by the biller. Must be unique in combination with the billerId. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Length : 1 - 120 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "2018-123456-22"	string
referencedBill <i>optional</i>		ReferencedBill
businessCaseDate <i>required</i>	The business case date (ISO-8601 format), can not be more than 90 days in the past on the date it was created. Cannot be in the future. Example : "2017-12-22"	string (date)
status <i>optional</i>	the status of the business case	enum (OPEN, APPROVED, REJECTED, COMPLETED)
totalAmount <i>required</i>		OptionalAmountWithCurrency

6.17. CreditNote

A business case of type CreditNote

Polymorphism : Inheritance

Discriminator : type

Name	Description	Schema
id <i>required</i>	A unique id for this business case. Property must not be given when creating a new business case. Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
type <i>required</i>	the type of the business case	enum (Bill, InstalmentBill, Advice, CreditNote, Reminder, DonationInquiry)
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
referenceNumber <i>required</i>	A business case reference given by the biller. Must be unique in combination with the billerId. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Length : 1 - 120 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "2018-123456-22"	string
referencedBill <i>optional</i>		ReferencedBill
businessCaseDate <i>required</i>	The business case date (ISO-8601 format), can not be more than 90 days in the past on the date it was created. Cannot be in the future. Example : "2017-12-22"	string (date)
status <i>optional</i>	the status of the business case	enum (OPEN, APPROVED, REJECTED, COMPLETED)
totalAmount <i>required</i>		OptionalAmountWithCurrency

6.18. Reminder

A business case of type Reminder

Polymorphism : Inheritance

Discriminator : type

Name	Description	Schema
id <i>required</i>	A unique id for this business case. Property must not be given when creating a new business case. Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
type <i>required</i>	the type of the business case	enum (Bill, InstalmentBill, Advice, CreditNote, Reminder, DonationInquiry)
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
referenceNumber <i>required</i>	A business case reference given by the biller. Must be unique in combination with the billerId. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Length : 1 - 120 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "2018-123456-22"	string
referencedBill <i>optional</i>		ReferencedBill
businessCaseDate <i>required</i>	The business case date (ISO-8601 format), can not be more than 90 days in the past on the date it was created. Cannot be in the future. Example : "2017-12-22"	string (date)
status <i>optional</i>	the status of the business case	enum (OPEN, APPROVED, REJECTED, COMPLETED)
totalAmount <i>required</i>		OptionalAmountWithCurrency

6.19. Advice

A business case of type Advice

Polymorphism : Inheritance

Discriminator : type

Name	Description	Schema
id <i>required</i>	A unique id for this business case. Property must not be given when creating a new business case. Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
type <i>required</i>	the type of the business case	enum (Bill, InstalmentBill, Advice, CreditNote, Reminder, DonationInquiry)
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
referenceNumber <i>required</i>	A business case reference given by the biller. Must be unique in combination with the billerId. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Length : 1 - 120 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "2018-123456-22"	string
referencedBill <i>optional</i>		ReferencedBill
businessCaseDate <i>required</i>	The business case date (ISO-8601 format), can not be more than 90 days in the past on the date it was created. Cannot be in the future. Example : "2017-12-22"	string (date)
status <i>optional</i>	the status of the business case	enum (OPEN, APPROVED, REJECTED, COMPLETED)
totalAmount <i>required</i>		OptionalAmountWithCurrency

6.20. DonationInquiry

A business case of type donation inquiry. Please note for donation inquiries ReferencedBill is not supported and must not to be specified

Polymorphism : Inheritance

Discriminator : type

Name	Description	Schema
id <i>required</i>	A unique id for this business case. Property must not be given when creating a new business case. Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
type <i>required</i>	the type of the business case	enum (Bill, InstalmentBill, Advice, CreditNote, Reminder, DonationInquiry)
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
referenceNumber <i>required</i>	A business case reference given by the biller. Must be unique in combination with the billerId. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Length : 1 - 120 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "2018-123456-22"	string
referencedBill <i>optional</i>		ReferencedBill
businessCaseDate <i>required</i>	The business case date (ISO-8601 format), can not be more than 90 days in the past on the date it was created. Cannot be in the future. Example : "2017-12-22"	string (date)
status <i>optional</i>	the status of the business case	enum (OPEN, APPROVED, REJECTED, COMPLETED)
totalAmount <i>required</i>		OptionalAmountWithCurrency

6.21. BusinessCase

the abstract business case object containing the shared properties

Name	Description	Schema
id <i>required</i>	A unique id for this business case. Property must not be given when creating a new business case. Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
type <i>required</i>	the type of the business case	enum (Bill, InstalmentBill, Advice, CreditNote, Reminder, DonationInquiry)
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
referenceNumber <i>required</i>	A business case reference given by the biller. Must be unique in combination with the billerId. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues. Length : 1 - 120 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "2018-123456-22"	string
referencedBill <i>optional</i>		ReferencedBill
businessCaseDate <i>required</i>	The business case date (ISO-8601 format), can not be more than 90 days in the past on the date it was created. Cannot be in the future. Example : "2017-12-22"	string (date)
status <i>optional</i>	the status of the business case	enum (OPEN, APPROVED, REJECTED, COMPLETED)
totalAmount <i>required</i>		OptionalAmountWithCurrency

6.22. ReferencedBill

The business case can only reference bills or instalment bills.

Name	Description	Schema
businessCaseId <i>optional</i>	business case id Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
referenceNumber <i>required</i>	The reference number of the referenced bill. Length : 1 - 120 Example : "2018-123456-22"	string

6.23. Sector

Name	Description	Schema
id <i>optional</i>	A unique id for this sector. Maximal length : 14 Pattern : "SCID[0-9]{10}" Example : "SCID0000000000"	string
localizedData <i>optional</i>		localizedData

localizedData

Name	Schema
ger <i>optional</i>	LocalizedSectorData
fre <i>optional</i>	LocalizedSectorData
ita <i>optional</i>	LocalizedSectorData
eng <i>optional</i>	LocalizedSectorData

6.24. LocalizedSectorData

Name	Description	Schema
name <i>required</i>	name of the sector Length : 1 - 36 Example : "Transport"	string

6.25. Event

contains common fields for all events

Name	Description	Schema
eventId <i>required</i>	event id Maximal length : 36 Pattern : "EVID[0-9A-Z]{32}" Example : "EVID82A65938766547EBBBA39BA6F7B07F24"	string
timestamp <i>required</i>	timestamp of the event Example : "2015-01-01T10:00:00.000Z"	string (date-time)

6.26. BusinessCaseStatusChangedEvent

An event describing the status change of a business case.

These events are sent for bills, reminders and donation inquiries.

The approved amount does always contain a value and a currency, but is only provided for the status APPROVED.

Polymorphism : Composition

Name	Description	Schema
eventId <i>required</i>	event id Maximal length : 36 Pattern : "EVID[0-9A-Z]{32}" Example : "EVID82A65938766547EBBBA39BA6F7B07F24"	string
timestamp <i>required</i>	timestamp of the event Example : "2015-01-01T10:00:00.000Z"	string (date-time)
billerId <i>optional</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
businessCaseId <i>optional</i>	business case id Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
newStatus <i>optional</i>	The new status of the business case. CHARGEBACK_INITIATED is only used in conjunction with bills containing an eBill Direct Debit payment.	enum (OPEN, APPROVED, REJECTED, COMPLETED, CHARGEBACK_INITIATED)
approvedAmount <i>optional</i>		ApprovalAmountWithCurrency

Name	Description	Schema
externalDonationPurposeId <i>optional</i>	optional field only to be used for donation inquiries to represent a potential selection of a donation purpose by the bill recipient, note the connection to the field externalDonationPurposeId from eBill business case specification	string

6.27. BillRecipientEmailAddressChangedEvent

An Event describing the change of the email address of a bill recipient.

Polymorphism : Composition

Name	Description	Schema
eventId <i>required</i>	event id Maximal length : 36 Pattern : "EVID[0-9A-Z]{32}" Example : "EVID82A65938766547EBBBA39BA6F7B07F24"	string
timestamp <i>required</i>	timestamp of the event Example : "2015-01-01T10:00:00.000Z"	string (date-time)
oldEmailAddress <i>optional</i>	the old email address of the bill recipient which has been used in the submission of a business case Length : 1 - 256 Example : "peter@muster.ch"	string (email)
newEmailAddress <i>optional</i>	the new email address of the bill recipient Length : 1 - 256 Example : "peter_new@muster.ch"	string (email)
triggeredBy <i>optional</i>		triggeredBy

triggeredBy

Name	Description	Schema
businessCaseId <i>required</i>	business case id Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
billerId <i>required</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string

6.28. InstalmentStatusChangedEvent

An Event describing the status change of an instalment.

These events are only sent for instalment bills.

The approved amount does always contain a value and a currency, but is only provided for the status APPROVED.

Polymorphism : Composition

Name	Description	Schema
eventId <i>required</i>	event id Maximal length : 36 Pattern : "EVID[0-9A-Z]{32}" Example : "EVID82A65938766547EBBBA39BA6F7B07F24"	string
timestamp <i>required</i>	timestamp of the event Example : "2015-01-01T10:00:00.000Z"	string (date-time)
billerId <i>optional</i>	biller id Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
businessCaseId <i>optional</i>	business case id Maximal length : 36 Pattern : "BCID[0-9A-Z]{32}" Example : "BCID0FB909852BBC4D06AD8336AAE87D7FC9"	string
externalPaymentByInstalmentsId <i>optional</i>	external id of the respective paymentByInstalment Length : 1 - 36 Example : "298031-2999"	string
externalInstalmentId <i>optional</i>	external id of the instalment Length : 1 - 36 Example : "298031-2999-ACX01"	string
newStatus <i>optional</i>	the new status of the instalment	enum (OPEN, APPROVED, REJECTED, COMPLETED)
approvedAmount <i>optional</i>		ApprovalAmountWithCurrency

6.29. BillRecipientSubscriptionStatusChangedEvent

An Event describing the status change of a bill recipient subscription.

Polymorphism : Composition

Name	Description	Schema
eventId <i>required</i>	event id Maximal length : 36 Pattern : "EVID[0-9A-Z]{32}" Example : "EVID82A65938766547EBBBA39BA6F7B07F24"	string
timestamp <i>required</i>	timestamp of the event Example : "2015-01-01T10:00:00.000Z"	string (date-time)
billerId <i>required</i>	Maximal length : 14 Pattern : "BIID[0-9]{10}" Example : "BIID0000123456"	string
billRecipient <i>required</i>		BillRecipient
accountNumber <i>optional</i>	Account number of the biller (e.g. iban), if provided from the financial institution Maximal length : 21	string
referenceStructured <i>optional</i>	QR or creditor reference number, if provided from the financial institution. Maximal length : 27 Pattern : "([a-zA-Z0-9])*" Example : "123456123456789012345678901"	string
billRecipientSubscriptionFormFields <i>optional</i>		< billRecipientSubscriptionFormFields > array
newStatus <i>optional</i>	the new status of the bill recipient subscription, see Section 4.3, “Subscriptions and subscription cancellations” for further information	enum (INACTIVE, REQUESTED, ACTIVE)
allowedEbillDebitSubmissions <i>optional</i>	Defines what kind of eBill Direct Debit submissions are allowed.	< AllowedEbillDebitSubmission > array

billRecipientSubscriptionFormFields

Name	Description	Schema
technicalId <i>required</i>	The identifying token of this subscription form field. Corresponds to the property technicalId of BillerSubscriptionFormField. Length : 1 - 35 Example : "customerNumber"	string
value <i>required</i>	The value of this subscription form field as entered by the user. If a pattern is defined, it has been checked against it. Length : 1 - 256	string

6.30. BillRecipientSubscriptionForm

Contains additional information regarding a bill recipient subscription. It contains the users input to the BillerSubscriptionForm.

Name	Schema
billRecipientSubscriptionFormFields <i>optional</i>	< billRecipientSubscriptionFormFields > array

billRecipientSubscriptionFormFields

Name	Description	Schema
technicalId <i>required</i>	The identifying token of this subscription form field. Corresponds to the property technicalId of BillerSubscriptionFormField. Length : 1 - 35 Example : "customerNumber"	string
value <i>required</i>	The value of this subscription form field as entered by the user. If a pattern is defined, it has been checked against it. Length : 1 - 256	string

6.31. OptionalAmountWithCurrency

An amount whose value may be omitted.

Name	Schema
value <i>optional</i>	AmountValue
currencyCode <i>required</i>	CurrencyCode

6.32. ApprovalAmountWithCurrency

Amount provided by status changed events if the new status is APPROVED. The value is always greater than zero.

Name	Schema
value <i>required</i>	AmountValue
currencyCode <i>required</i>	CurrencyCode

6.33. AmountValue

The amount value. Take care when using JavaScript libraries to parse this value - it should be treated as a financial amount and therefore not as a floating point number but rather using a precise decimal representation (like BigDecimal in Java).

Maximum value: 99'999'999.99

Maximal length: 10

Type : number

Example: 99.99

6.34. EbillDebitCurrenyCode

The amount currency code according to ISO-4217 used for eBill Direct Debit.

Type : string

Example: "CHF"

6.35. CurrencyCode

The amount currency code according to ISO-4217.

Type : string

Example: "CHF"

6.36. AccountNumber

For the account number, an IBAN has to be specified.

Name	Description	Schema
iban <i>optional</i>	<p>credit account</p> <p>- The requirements for IBAN usage are limited to the country codes CH and LI, otherwise the business case will be rejected.</p> <p>- The IBAN should match an already existing credit account of the biller, otherwise the business case will be rejected.</p> <p>- See also ISO-13616-1</p> <p>Maximal length : 21</p> <p>Pattern : "(CH LI)\d{7}[0-9A-Z]{12}"</p> <p>Example : "CH100023000A109822346"</p>	string

6.37. Problem

Name	Description	Schema
type <i>optional</i>	An absolute URI that identifies the problem type. We may provide human-readable documentation for the problem type in the future, when the URI is dereferenced. For now consult the network partner API documentation for further information. The URI consists of the /problems endpoint and the documented problem type, see example. Default : "about:blank" Example : "/problems/REQUEST_BODY_VALIDATION_FAILED"	string (uri)
title <i>optional</i>	A short, human readable summary of the problem type. Example : "Payload has missing or invalid values"	string
status <i>optional</i>	The HTTP status code generated by the origin server for this occurrence of the problem. Minimum value : 100 Maximum value (exclusive) : 600 Example : 400	integer (int32)
detail <i>optional</i>	A human readable explanation specific to this occurrence of the problem. Example : "The submitted request contains invalid or missing data which can not be processed."	string
instance <i>optional</i>	An absolute URI that identifies the specific occurrence of the problem. It may or may not yield further information if dereferenced. Example : "/api/pns/networkpartner/v5/billers/errors/NWID0090000001/provided-x-correlation-id"	string (uri)
technicalReason <i>optional</i>	The Technical Description/Reason for engineers might contain addition system information on the problem. Example : "Some field validations failed"	string
fieldErrors <i>optional</i>		< fieldErrors > array

fieldErrors

Name	Description	Schema
fieldName <i>optional</i>	the name of the field with the error Example : "localizedData.ger.address.city"	string

Name	Description	Schema
message <i>optional</i>	the message describing the error Example : "size must be between 1 and 35"	string
rejectedValue <i>optional</i>	the provided value which was rejected if available Example : "Very Long Invalid City Name Which Is Rejected"	string

6.38. HealthCheckRequest

Name	Description	Schema
message <i>required</i>	expected response message from health check Length : 1 - 100 Example : "any string"	string

6.39. HealthCheckResponse

Name	Description	Schema
message <i>required</i>	response message from health check Maximal length : 100 Example : "The healthcheck GET request was successfully received and processed"	string
requestDateTime <i>required</i>	according to RFC3339, section 5.6 in ISO 8601 with timezone and milliseconds Example : "2018-10-03T16:03:09.101+02:00"	string (date-time)
receivedHeaders <i>required</i>		< receivedHeaders > array
environmentStage <i>required</i>	the instance which the request was sent to Example : "XE"	string
applicationVersion <i>required</i>	the version of the eBill infrastructure Example : "1.4.3.0-desire-20180927091004161-71-5e3ca91"	string
apiVersion <i>required</i>	the version of the network partner api Example : "1.0.23"	string
maintenanceWindows <i>optional</i>	Information about the upcoming maintenance windows. If no maintenance is planned, an empty list is returned.	< MaintenanceWindow > array

receivedHeaders

Name	Description	Schema
headerName <i>optional</i>	the name of the provided header Example : "x-correlation-id"	string
headerValue <i>optional</i>	As received Example : "9bcd4351-4b7b-4017-9b63-9613414c6ff1"	string

6.40. DefaultLanguage

From the provided localizedData, one has to be marked as the default language.

The eBill infrastructure will use this localizedData in case a user requests a language where no localizedData have been provided.

Type : enum (ger, fre, ita, eng)

6.41. BillRecipientSubscriptionStatus

If allowed, the biller can be found and bill recipients can subscribe with this biller.

Type : enum (ALLOWED, NOT_ALLOWED)

6.42. BillerSearchFilter

Filter object for the search of billers. Filter attributes may be omitted or empty. All filter parameters are combined by AND. See also [Section 3.3.10, "Search operations"](#)

Name	Schema
filter <i>required</i>	filter

filter

Name	Description	Schema
name <i>optional</i>	Search pattern applied on legal and display names of all languages. All names conforming to or containing the pattern are matches. The search is case insensitive.	string
enterpriseIdentificationNumber <i>optional</i>	Search pattern applied on the swiss enterprise identification number (UID). The search argument may not contain dashes, dots or extensions. The pattern must be an exact match.	string
iban <i>optional</i>	Search pattern applied on the IBAN credit account. The pattern must be an exact match.	string
billerId <i>optional</i>	Search pattern applied on biller id. Only exact matches, will always return maximum one record.	string

6.43. BillerSearchResult

Name	Description	Schema
totalCount <i>required</i>	Total count of matching billers, regardless of limit and offset applied.	number
items <i>required</i>	An array of found items that might be empty.	< Biller > array

6.44. SubscriptionInitiationEmailAddress

Name	Description	Schema
emailAddress <i>required</i>	Email address the user entered for subscription initiation Length : 1 - 256 Example : " hansmuster@muster.info "	string (email)

6.45. SubscriptionInitiationToken

Name	Description	Schema
subscriptionInitiationToken <i>required</i>	subscription initiation token which will be used to confirm the subscription initiation Length : 36 Example : " 0dc2ff79-db4c-4635-a4aa-f93f36ab5dbf "	string

6.46. SubscriptionInitiationActivationCode

Name	Description	Schema
activationCode <i>required</i>	activation code provided by the user Length : 6 Example : " 123456 "	string

6.47. MaintenanceWindow

Name	Description	Schema
start <i>required</i>	start time of the maintenance window Example : " 2021-01-01T10:00:00.000Z "	string (date-time)
end <i>required</i>	end time of the maintenance window Example : " 2021-01-01T14:00:00.000Z "	string (date-time)

6.48. BillerSubscriptionForm

A custom subscription form of a biller.

Name	Description	Schema
defaultLanguage <i>required</i>		DefaultLanguage
infoText <i>optional</i>		LocalizedBillerSubscriptionFormInfoText
billerSubscriptionFormFields <i>optional</i>	An array of custom subscription form fields. The number of fields within the context of a bill recipient type may be zero and may not exceed three (See the description of the property "applyToBillRecipientType" for more details). Specified fields are mandatory to successfully complete the subscription process to the biller.	< BillerSubscriptionFormField > array

6.49. BillerSubscriptionFormField

An abstract biller subscription form field. Depicts all common properties. Not used directly but rather subclassed by several concrete variants.

Name	Description	Schema
type <i>required</i>	Designates the type of form field. The individual form field types behave as follows: - CUSTOM: The biller pre-defines and delivers all field-properties. Localized data is mandatory for this type. A constraint may be given but remains optional. - BIRTHDATE: Static definition data is kept and maintained by the eBill infrastructure. No constraint or localized data is allowed for this type. - CHOICE: The biller has to specify localized data for the field as well as an array of possible choices. A constraint is not allowed for this type. For example, if the biller is a non-profit organisation a subscription form field of this type can represent the different proposed donation purposes.	enum (CUSTOM, BIRTHDATE, CHOICE)
applyToBillRecipientType <i>required</i>	Establishes the type of bill recipient for which this field shall be used. For example, for bill recipients of type COMPANY only form fields of type COMPANY and ALL will be shown. Fields of type ALL will be included in every form. The maximum of three fields may not be exceeded for any bill recipient type. For example, if one field is specified for the bill recipient type "ALL", a maximum of two other fields per bill recipient type "PRIVATE" and "COMPANY" are allowed to be specified. Default : "ALL"	enum (ALL, PRIVATE, COMPANY)

Name	Description	Schema
technicalId <i>required</i>	The identifying name of a subscription form field. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 35 Pattern : <code>"^[\u0021-\u007E\u0085\u00A0-\u00D7FF\uE000-\uFDCF\uFDF0-\uFFFD]+\$"</code> Example : <code>"customerNumber"</code>	string
constraint <i>optional</i>	Constraints to be applied on the input of this field.	constraint
localizedData <i>optional</i>		localizedData
choices <i>optional</i>	An array of possible choices for this form field. Only allowed for form fields of type CHOICE, for which it is mandatory.	< BillerSubscriptionFormChoice > array

constraint

Name	Description	Schema
pattern <i>optional</i>	Regex pattern limiting possible input values. Adheres to the Java regex syntax and must conform to these restrictions: * No unbounded quantifiers <code>\d+</code> or <code>\d*</code> or <code>\d{4,}</code> and no more than 30 repetitions. Use exact cardinalities <code>\d{5}</code> or ranges <code>\d{4,7}</code> * No alternations within a repeated group <code>'((abc){3} (def){2}){30}'</code> . * No partial or multiple matches <code>'[a-f]{2}'</code> . Use only exact matches <code>^[a-f]{7}\$</code> . * No longer than 100 characters. The string to define the pattern must be a subset of XML1.0 conform characters to avoid downstream issues If a pattern is given to validate the user input, a concise explanation and example for the end user must be provided in the description field of <code>LocalizedBillerSubscriptionFormField</code> . Length : 1 - 100 Pattern : <code>"[\\u0009\\u000A\\u000D\\u0020-\\u007E\\u0085\\u00A0-\\u00D7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*"</code> Example : <code>"^\\d{3}-\\w{4}-\\d{3}\$"</code>	string
minLength <i>optional</i>	If given, specifies the minimum length of the field. Must be between 1 and 256. Minimum value : 1 Maximum value : 256	number

Name	Description	Schema
maxLength <i>optional</i>	If given, specifies the maximum length of the field. Must be between 1 and 256. Minimum value : 1 Maximum value : 256	number

localizedData

Name	Schema
ger <i>optional</i>	LocalizedBillerSubscriptionFormField
fre <i>optional</i>	LocalizedBillerSubscriptionFormField
ita <i>optional</i>	LocalizedBillerSubscriptionFormField
eng <i>optional</i>	LocalizedBillerSubscriptionFormField

6.50. LocalizedBillerSubscriptionFormField

Textual properties of a subscription form field in a specific locale.

Name	Description	Schema
label <i>required</i>	The display label of a subscription form field. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 35 Pattern : "[\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "Kundennummer"	string
description <i>optional</i>	An additional field description to a subscription form field. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues Length : 1 - 256 Pattern : "[\\u0009\\u000A\\u000D\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFD]*" Example : "Kundennummer nach diesem Beispiel eingeben: \\\"123-abcd-456\\\"."	string

6.51. BillerSubscriptionFormChoice

Textual representation of a subscription form field of type CHOICE in specific locales.

Name	Description	Schema
choiceId <i>required</i>	The choiceId, which the user has selected, will be delivered as "value" along with the technicalId in the object billRecipientSubscriptionFormFields of the bill-recipient-subscription-status-changed-event. The pattern restricts to word characters of the US-ASCII Length : 1 - 36 Pattern : "[\\w]*"	string
defaultChoice <i>required</i>	Exactly one of the subscription form field choices must be marked as a pre-selected default choice.	boolean
localizedData <i>required</i>	Translations of one particular choice into potential target languages of a recipient. At least the default language of the subscription form must be provided.	localizedData

localizedData

Name	Schema
ger <i>optional</i>	LocalizedBillerSubscriptionFormChoice
fre <i>optional</i>	LocalizedBillerSubscriptionFormChoice
ita <i>optional</i>	LocalizedBillerSubscriptionFormChoice
eng <i>optional</i>	LocalizedBillerSubscriptionFormChoice

6.52. LocalizedBillerSubscriptionFormChoice

Textual properties of a subscription form choice in a specific locale.

Name	Description	Schema
label <i>required</i>	The display label of a subscription form choice. The pattern restricts to any latin letter and any digit. Length : 1 - 36 Pattern : "[\\p{IsLatin} \\d]+"	string

6.53. LocalizedBillerSubscriptionFormInfoText

Introductory text presented to the user as part of a biller subscription form.

Name	Schema
localizedData <i>required</i>	localizedData

localizedData

Name	Description	Schema
ger <i>optional</i>	Maximal length : 500 Pattern : "[\\u0009\\u000A\\u000D\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "Diese Anmeldung gilt für \"xy Unfallversicherungen\", für \"xy Lebensversicherungen\" müssen Sie sich gesondert anmelden. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues"	string
fre <i>optional</i>	Maximal length : 500 Pattern : "[\\u0009\\u000A\\u000D\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "Cette inscription est valable pour \"xy assurance accident\", pour \"xy assurance vie\" vous devez vous inscrire séparément. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues"	string
ita <i>optional</i>	Maximal length : 500 Pattern : "[\\u0009\\u000A\\u000D\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "Questa registrazione è valida per \"xy assicurazione infortuni\", per \"xy assicurazione sulla vita\" dovete registrarvi separatamente. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues"	string
eng <i>optional</i>	Maximal length : 500 Pattern : "[\\u0009\\u000A\\u000D\\u0020-\\u007E\\u0085\\u00A0-\\uD7FF\\uE000-\\uFDCF\\uFDF0-\\uFFFF]*" Example : "This registration is valid for \"xy accident insurance\", for \"xy life insurance\" you must register separately. The pattern restricts to a subset of XML1.0 conform characters to avoid downstream issues"	string

6.54. AnomalyDetectionConfig

Configuration of the anomaly detection for biller submissions.

Name	Description	Schema
monthlySubmissionLimit <i>required</i>	<p>Biller submissions are only allowed up to this monthly count limit.</p> <p>Minimum value : 0</p> <p>Maximum value : 2147483647</p>	integer

6.55. AllowedEbillDebitSubmission

Defines the relevant information what type of eBill Direct Debit submission the biller is allowed to do.

Name	Schema
currencyCode <i>required</i>	EbillDebitCurrenyCode
chargebackMode <i>required</i>	ChargebackMode

6.56. EbillDebitProposal

eBill Direct Debit proposal for the bill recipient.

Name	Description	Schema
id <i>optional</i>	A unique id for this eBill Direct Debit proposal. Property must not be given when creating a new object. Length : 36	string
billRecipientIdentification <i>required</i>		BillRecipientIdentification
currencyCode <i>required</i>		EbillDebitCurrenyCode
chargebackMode <i>required</i>		ChargebackMode

6.57. ChargebackMode

If supported, the bill recipient can initiate a chargeback of the corresponding payment.

Type : enum (SUPPORTED, NOT_SUPPORTED)

6.58. Certification

Name	Description	Schema
id <i>optional</i>	A unique id for this certification. Maximal length : 14 Pattern : "CEID[0-9]{10}" Example : "CEID0000000000"	string

Name	Description	Schema
name <i>optional</i>	Name of the certification. Length : 2 - 70 Example : "ZEWO"	string

Chapter 7. Problem Descriptions Overview

7.1. Basic Validations

UNAUTHORIZED

Status code: 401

Request failed authorization validation

All requests require a combination of a valid client certificate and an associated x-networkpartner-id header.

For more information about authentication, see network partner API documentation in [Section 3.2.2, “Authentication”](#).

REQUEST_URL_MALFORMED

Status code: 400

Request failed url parsing

The request URL must be a valid normalized URL.

Request URLs must not contain path traversal sequences or multiple forward slashes.

NETWORK_PARTNER_UNKNOWN

Status code: 401

Network partner is unknown

The x-networkpartner-id header of the request has to reference a known and active network partner in the system.

This also requires an associated and valid client certificate, see network partner API documentation in [Section 3.2.2, “Authentication”](#).

NETWORK_PARTNER_NOT_ACTIVE

Status code: 401

Network partner is not active

The x-networkpartner-id header of the request has to reference a known and active network partner in the system.

This also requires an associated and valid client certificate, see network partner API documentation in [Section 3.2.2, “Authentication”](#).

REQUEST_CONTENT_TYPE_UNSUPPORTED

Status code: 415

The content type of the request is unsupported

Content type of a POST or PUT request should be set to the allowed content type for this resource, see network partner API documentation for details.

For JSON content this has to be application/json, an image asset must be one of image/png, image/jpeg or image/gif and for PDF content it has to be application/pdf. Other content types are not supported.

REQUEST_METHOD_NOT_ALLOWED

Status code: 405

HTTP method of the request is not allowed

HTTP method of a request is validated and has to be one of the supported methods for this resource, see network partner API documentation for details.

INVALID_RESOURCE

Status code: 404

Resource not found

The requested resource or endpoint does not exist.

REQUEST_PAYLOAD_EXCEEDS_MAXIMUM_SIZE

Status code: 413

Payload of the request is too large

Submitted POST or PUT request payload data should be smaller than 10'000'000 bytes, otherwise it is discarded without further analysis.

REQUEST_PAYLOAD_IS_EMPTY

Status code: 400

Payload of the request is empty

Empty payload content is not allowed for POST or PUT.

REQUEST_BODY_VALIDATION_FAILED

Status code: 400

Payload has missing or invalid values

The submitted request contains invalid or missing data in its payload and can not be processed.

REQUEST_HEADER_VALIDATION_FAILED

Status code: 400

Header is missing or has invalid value

The submitted request contains an invalid or missing header and can not be processed.

HTTP_MEDIA_TYPE_NOT_ACCEPTABLE

Status code: 406

No acceptable representation for requested http media-type

The request URL ends with an unsupported file-extension or the content-type header of the request specifies an unsupported media-type value.

7.2. Shared Validations

NETWORK_PARTNER_OPERATION_NOT_ALLOWED

Status code: 403

Requested operation not allowed for network partner

There are restrictions regarding access to some resources. See network partner API documentation in [Section 2.3, “Primary network partners”](#) for details.

REQUEST_QUERY_PARAMETER_VALIDATION_FAILED

Status code: 400

Request query parameter has missing or invalid values

The submitted request contains invalid or missing query parameters which can not be processed. This problem type is used in case of missing or invalid input values.

INVALID_EMAIL

Status code: 400

Payload contains not well formed email address

The submitted request contains a not well formatted email address which can not be processed.

INVALID_CURRENCY_CODE

Status code: 400

Payload contains invalid currency code

The submitted request contains a currency code which can not be processed.

The currency code must be according to ISO-4217.

INVALID_ASSET_ID

Status code: 400

The assetId is invalid

The provided assetId is invalid as it doesn't match the existing assetId, which was generated by the system.

ASSET_ID_MUST_NOT_BE_PROVIDED

Status code: 400

The assetId must not be provided by the client

The assetId must not be provided by the client, it will be generated by the system.

LOCALIZED_DATA_FOR_DESIRED_DEFAULT_LANGUAGE_MISSING

Status code: 400

Payload does not contain the localized data for the given default language

For the specified default language no localized data could be found in the request. Make sure for the referenced localizedData.language you also provide the data set.

7.3. Bill Recipient Validations

BILL_RECIPIENT_IDENTIFICATION_INVALID

Status code: 400

Provided bill recipient identification is invalid

The provided bill recipient identification object is invalid. Exactly one identifying property must be set.

BILL_RECIPIENT_EMAIL_NOT_FOUND

Status code: 404

Provided bill recipient email not found

The bill recipient email must be currently or historically available in the system.

An e-mail address is considered to be historically available if it was present in the period of 15 months up to the submission time.

For further details see network partner API documentation in [Section 5.3.4, "Find events for bill recipients email address changes"](#).

BILL_RECIPIENT_ID_NOT_FOUND

Status code: 404

Provided bill recipient id not found

The bill recipient id must be available in the system.

BILL_RECIPIENT_HRUID_NOT_FOUND

Status code: 404

Provided bill recipient enterprise identification number (UID) not found

The provided swiss enterprise identification number (UID) of the bill recipient must be available in the system.

7.4. Biller Validations

BILLER_ACCOUNT_COMBINATION_ALREADY_EXISTS

Status code: 400

Biller account number combination already exists in the system

The combination of biller account and optional qrIbanAccountSupplement must not conflict with any existing account information.

BILLER_ACCOUNT_CURRENCY_CODE_COMBINATION_REJECTED

Status code: 400

Biller account / currency code combination is not allowed

The currency code of the biller account has to match the allowed currencies for the account type. Account type QR-IBAN only supports payment type 3 and therefore currency code is limited to CHF or EUR.

See the definition of payment types in <https://www.six-interbank-clearing.com/dam/downloads/en/standardization/iso/swiss-recommendations/implementation-guidelines-ct.pdf>.

More information about QR in

<https://www.paymentstandards.ch/dam/downloads/ig-qr-bill-en.pdf>.

BILLER_NAME_HRUID_COMBINATION_ALREADY_EXISTS

Status code: 400

Biller name / UID combination already exists in the system

The combination of display name and swiss enterprise identification number (UID) from the commercial register (Handelsregister) must be unique for every biller in the system.

BILLER_NAME_POSTAL_CODE_COMBINATION_ALREADY_EXISTS

Status code: 400

Biller name / postal code combination already exists in the system

Display name and postal code combination must be unique for every biller in the system.

BILLER_REFERENCED_SECTOR_DOES_NOT_EXIST

Status code: 400

One or more referenced sectorIds do not exist

One or more referenced sectorIds do not exist. Make sure you use only existing sectorIds.

BILLER_INVALID_IBAN_NUMBER

Status code: 400

Payload contains invalid IBAN number

The submitted request contains an IBAN number which can not be processed.

BILLER_ACCOUNT_QR_IBAN_ACCOUNT_SUPPLEMENT_UNSUPPORTED

Status code: 400

IBAN can not be combined with a qrIbanAccountSupplement

Only QR-IBAN can be used in combination with qrIbanAccountSupplement.

BILLER_ACCOUNT_WITH_INVALID_QR_IBAN_ACCOUNT_SUPPLEMENT_COMBINATION

Status code: 400

Invalid constellation of QR-IBAN using account supplement

If an account supplement is provided with a QR-IBAN, there must be no empty account supplement entries for the same QR-IBAN. The same applies if there is already a QR-IBAN without an account supplement, then no QR-IBAN with an account supplement may be used.

7.5. Asset Validations

ASSET_IMAGE_INVALID

Status code: 400

Asset payload does not seem to be a valid image

The submitted binary data has to be a valid image.

ASSET_MIME_TYPE_DOES_NOT_CORRESPOND_TO_CONTENT_TYPE

Status code: 415

Asset mime type does not correspond to content type

The mime type of the uploaded asset does not correspond to the provided content type.

ASSET_CONTENT_TYPE_MUST_BE_IMAGE

Status code: 400

Only images are allowed for the given asset

The asset is connected to an image. Only the content types image/jpeg, image/gif or image/png are allowed.

ASSET_CONTENT_TYPE_MUST_BE_PDF

Status code: 400

Only PDFs are allowed for the given asset

The asset is connected to a PDF. Only the content type application/pdf is allowed for the given asset.

ASSET_IMAGE_EXCEEDS_MAXIMUM_SIZES

Status code: 413

Image asset maximum sizes exceeded

Image asset size must be at maximum 100'000 bytes and 1024x1024 pixels.

7.6. Custom Subscription Form Validations

CUSTOM_SUBSCRIPTION_FORM_CONSTRAINT_INVALID

Status code: 400

Custom subscription form contains invalid field constraint

The custom subscription form contains a field with an invalid constraint.

The constraint pattern of fields in a biller's custom subscription form must comply to the restrictions defined in [Section 6.49, "BillerSubscriptionFormField"](#).

CUSTOM_SUBSCRIPTION_FORM_TOO_MANY_FIELDS

Status code: 400

Custom subscription form contains too many fields

The amount of form fields within the context of a bill recipient type may not exceed three.

BOTH_CUSTOM_SUBSCRIPTION_FORM_AND_BILL_RECIPIENT_SUBSCRIPTION_URL_DEFINED

Status code: 400

Both a custom subscription form and a bill recipient subscription URL are defined

The definitions of a custom subscription form and a bill recipient subscription URL are mutually exclusive. There can not be both.

CUSTOM_SUBSCRIPTION_FORM_INVALID

Status code: 400

Custom subscription form invalid

The custom subscription form is invalid.

The biller's custom subscription form contains invalid fields or field combinations.

CUSTOM_SUBSCRIPTION_FORM_CHOICE_ID_NOT_UNIQUE

Status code: 400

The identifications of different choices in custom subscription form are not unique

The different choices must be uniquely identified within a biller subscription form field of type 'CHOICE'.

7.7. Custom Subscription Form Data Validations

BILL_RECIPIENT_SUBSCRIPTION_FORM_INVALID

Status code: 400

The bill recipient subscription form data is invalid

The provided bill recipient subscription form data is invalid.

7.8. Business Case Validations

BC_PDF_ATTACHMENT_VALIDATION_FAILED

Status code: 400

Business case PDF does not contain a valid attachment

PDF file contains exactly one embedded attachment with the exact name **eBill-SIX_v5.xml** and none of any other version.

More attachments are allowed - and ignored - as long as they have different names.

BC_FILENAME_HEADER_EXCEEDS_MAXIMUM_LENGTH

Status code: 400

HTTP header x-filename exceeds maximum length

The request can contain a custom HTTP header **x-filename** of max length 99.

BC_ANOMALY_DETECTION_HEADER_INVALID

Status code: 400

HTTP header x-anomaly-detection is not set correctly

For more information, see the network partner API documentation in [Section 5.1.12, "Create business case in PDF/A-3b-format"](#).

BC_EBILL_SIX_XML_SCHEMA_VALIDATION_FAILED

Status code: 400

XML schema validation of the eBill-SIX_v5.xml attachment failed

The eBill-SIX_v5.xml file has to be schema valid.

BC_BILLER_NOT_ACTIVE

Status code: 400

Biller is not active

The request has to refer to an active biller in the system.

BC_BILLER_SUBMISSION_FORBIDDEN

Status code: 403

Biller is currently suspended from submitting business cases

Submissions are not permitted for suspended billers.

Due to detected anomaly the biller has been suspended from submitting business cases.

For further details please contact the infrastructure provider.

BC_BILLER_SUBMISSION_LIMIT_EXCEEDED

Status code: 403

Biller has reached the specified submission limit

Submissions are not permitted after reaching a specified submission limit.

Due to reached submission limit the submission of the biller has been rejected. Check whether the 'x-anomaly-detection' header should be set.

BC_BILLER_ID_INCONSISTENT

Status code: 400

Business case has inconsistent biller id

The biller id in url must match the value in the submitted xml file.

BC_BILLER_NAME_INCONSISTENT

Status code: 400

Business case has inconsistent biller legal name

The biller legal name must match the value from the system.

The submitted biller legal name must match the value from the biller data managed in system. Otherwise the business case will be rejected.

BC_BILLER_BILLRECIPIENT_RELATION_INSUFFICIENT

Status code: 400

Biller and bill recipient do not have the required relation

Check for a sufficient biller - bill recipient relation, see network partner API documentation for details.

If none of the following conditions are met, the business case is rejected:

- . There is an active biller - bill recipient relation (say, the subscription process was completed beforehand).
- . There is a biller - bill recipient relation in 'initiated' status (say, a registration process was at least started by the bill recipient).
- . There is no biller - bill recipient relation in 'inactive' status and the bill recipient has enabled to be found.

BC_INVALID_DATE

Status code: 400

Business case has invalid date

On the submission date, the business case date can not be more than 90 days in the past and it can not be in the future.

BC_INVALID_REFERENCE_NUMBER

Status code: 400

Business case has invalid reference number

The reference number must be unique per biller

BC_REFERENCED_BILL_IS_NOT_A_BILL

Status code: 400

Provided business case referenced bill is not a bill

The business case must not reference other business cases types but bill or instalment bill.

BC_REFERENCE_NUMBER_FOR_REFERENCED_BILL_REQUIRED

Status code: 400

Referenced number for referenced bill is missing

The reference number of the referenced bill is mandatory in case of reminder.

BC_INVALID_REFERENCE_NUMBER_FOR_REFERENCED_BILL

Status code: 400

Referenced business case has invalid reference number

The reference number of the referenced business case has to be valid.

BC_INVALID_TOTAL_AMOUNT

Status code: 400

Business case has invalid total amount

The total amount of the business case must fulfill the requirements for minimum and maximum value.

See eBill-SIX_v5.xsd for details about total amount validation.

7.9. Business Case Payment Validations

BC_PAYMENT_INFORMATION_INVALID_PAYMENT_DUE_DATE

Status code: 400

Business case has invalid payment due date

On the submission date, the payment due date cannot be more than 3 years (1095 days) in the future for payment mode ebill and cannot be more than 30 days in the future for payment mode ebill debit and cannot be more than 90 days in the past for both payment modes.

BC_INCONSISTENT_CURRENCY_CODES

Status code: 400

Business case has inconsistent currency codes

In the entire business case only one currency is allowed.

It is not allowed to submit instalments, total amount or workflow specifications with different currencies.

BC_PAYMENT_INFORMATION_INVALID_CURRENCY_CODE_FOR_QR_IBAN

Status code: 400

The business case contains a payment information with an invalid currency code for the QR-IBAN

payment type

Payment type QR-IBAN only supports payment type 3 and therefore currency code is limited to CHF or EUR.

See the definition of payment types in

<https://www.six-interbank-clearing.com/dam/downloads/en/standardization/iso/swiss-recommendations/implementation-guidelines-ct.pdf>.

More information about QR in

<https://www.paymentstandards.ch/dam/downloads/ig-qr-bill-en.pdf>.

BC_PAYMENT_INFORMATION_ACCOUNT_INFORMATION_INCONSISTENT

Status code: 400

The business case contains a biller account that has inconsistent information

The biller account in the business case 'payment information' must match the account information in the biller master data.

BC_PAYMENT_INFORMATION_INVALID_REFERENCE_TYPE

Status code: 400

The business case has an invalid reference type

In case of structured reference the biller account has to contain a valid reference type. See definitions and documentation in eBill-SIX_v5.xsd for more details.

The allowed values for reference type are: QRR: QR-Reference, SCOR: Creditor Reference (according to ISO 11649), NON: no reference, IPI: IPI-Reference (according to ISO 7064) (Note: The IPI-Reference must be withdrawn from 31.03.2020 at the latest.)

BC_PAYMENT_INFORMATION_INVALID_STRUCTURED_REFERENCE

Status code: 400

The business case contains invalid structured reference

Validation of the structured reference failed for the given referenceType.

QRR: QR-Reference, the submitted QR-Reference must be numeric and 27 characters long. The 27th digit (check digit) must be in accordance with Modulo 10, recursive.

SCOR: Creditor Reference (according to ISO 11649), it must be alphanumeric and its maximal length is 25.

NON: no reference, only reference type NON allows empty structured reference.

IPI: IPI-Reference, the submitted IPI-Reference must be alphanumeric and its maximal length is 20. The first two digit are check digits, according to Modulo 97-10 (ISO 7064). Note: The IPI-Reference must be withdrawn from 31.03.2020 at the latest.

BC_PAYMENT_INFORMATION_INVALID_FOR_QR_IBAN_ACCOUNT_SUPPLEMENT

Status code: 400

Payment information contains QR reference which is not valid for the existing qrIbanAccountSupplement

The submitted QR reference number must begin with the qrIbanAccountSupplement of the matching biller account.

7.10. Business Case Instalment Validations

BC_INVALID_TOTAL_PAYMENT_OPTION

Status code: 400

Payload contains invalid total payment option for instalments

Validation of the total payment option failed.

BC_EXTERNAL_PAYMENT_BY_INSTALLMENTS_ID_NOT_UNIQUE

Status code: 400

The identifications for the instalment options are not unique

The instalment options must be uniquely identified within a business case.

BC_EXTERNAL_INSTALLMENT_ID_NOT_UNIQUE

Status code: 400

The identifications for the instalments are not unique

The instalments must be uniquely identified within a business case.

BC_INVALID_INSTALLMENT_AMOUNT

Status code: 400

Business case has instalment with invalid amount

The instalment amount must be positive.

7.11. Business Case Donation Inquiry Validations

BC_BILLER_DONATION_INQUIRIES_NOT_ALLOWED

Status code: 403

Biller is not allowed to submit donation inquiries

Submission of donation inquiries is only allowed for billers who have been granted special permission to do so.

Only billers that have been verified to be non-profit organisations (NPO) may be granted the permission to submit donation inquiries.

For further details see network partner API documentation in [Section 4.1, “Biller management”](#).

BC_DONATION_INQUIRY_AMOUNT_INVALID

Status code: 400

Donation inquiry has invalid donation amount(s)

The donation amount(s) must be greater or equal to the minimum amount value specified for donation inquiries.

The minimum amount for donation inquiries is CHF 5.

BC_DONATION_INQUIRY_AMOUNT_INCONSISTENT

Status code: 400

The donation inquiry has inconsistent amount(s)

If proposed donation amounts are specified, the total and the payment information amount value must not be specified and vice versa. It is also not allowed to have proposed donation amounts with the same values.

BC_DONATION_INQUIRY_REFERENCED_BILL_UNSUPPORTED

Status code: 400

Submission of donation inquiry with referenced bill is not supported

A donation inquiry must not reference any other donation inquiry or business case of other types.

BC_EXTERNAL_DONATION_PURPOSE_ID_NOT_UNIQUE

Status code: 400

The identifications of different donation purposes are not unique

The different donation purposes must be uniquely identified within a donation inquiry.

7.12. Business Case eBill Direct Debit Validations

BC_EBILL_DEBIT_SUBMISSION_NOT_ALLOWED**Status code: 403**

Submission of a bill with eBill Direct Debit payment not allowed

The submission of a bill with eBill Direct Debit payment requires the permission of the bill recipient.

BC_EBILL_DEBIT_REFERENCED_BILL_UNSUPPORTED**Status code: 400**

Submission of a bill with eBill Direct Debit payment and referenced bill is not supported

eBill Direct Debit does not support referenced bills.

BC_EBILL_DEBIT_DONATION_INQUIRY_UNSUPPORTED**Status code: 400**

Submission of a donation inquiry with eBill Direct Debit payment is not supported

eBill Direct Debit does not support donation inquiries.

BC_EBILL_DEBIT_REMINDER_UNSUPPORTED**Status code: 400**

Submission of a reminder with eBill Direct Debit payment is not supported

eBill Direct Debit does not support reminders.

BC_EBILL_DEBIT_BILLER_ACCOUNT_UNSUPPORTED**Status code: 400**

Biller account does not support eBill Direct Debit

Biller account has not been enabled for eBill Direct Debit.

BC_EBILL_DEBIT_SUBMISSION_LIMIT_UNDEFINED**Status code: 400**

Submission rejected due to missing eBill Direct Debit submission limit

No eBill Direct Debit submission limit defined for biller, financial institution and currency.

BC_EBILL_DEBIT_SUBMISSION_LIMIT_EXCEEDED**Status code: 400**

Submission exceeds the defined eBill Direct Debit submission limit

eBill Direct Debit submission limit exceeded for biller, financial institution and currency.

7.13. PDF Validations

PDF_INVALID**Status code: 400**

Payload does not seem to be a valid PDF

The submitted binary data has to be a valid PDF document.

PDF_IS_NOT_PDFA3B

Status code: 400

PDF is not PDF/A-3b

PDFs must be provided in a valid PDF/A-3b format.

PDF_INVALID_SIGNATURE

Status code: 400

PDF does not contain a valid signature

The PDF must contain a valid PAdES-B-B-level signature.

For more information, see network partner API documentation in [Section 4.2.1.4, “File specification and signatures”](#).

7.14. Subscription Validations

INVALID_TOKEN_AND_ACTIVATION_CODE_COMBINATION

Status code: 400

The subscription initiation token / activation code combination is invalid

The subscription initiation token and activation code combination should be valid, can only be consumed once and are only valid for a limited time.

The initiation token has a 60 minute lifetime and a maximum of 3 tries for the activation code before it is invalidated.

7.15. eBill Direct Debit Validations

EBILL_DEBIT_NETWORK_PARTNER_NOT_ALLOWED

Status code: 403

Network partner is not allowed for eBill Direct Debit actions

eBill Direct Debit has not been enabled for the network partner.

EBILL_DEBIT_BILLER_NOT_ALLOWED

Status code: 403

Biller is not allowed for eBill Direct Debit actions

eBill Direct Debit has not been enabled for the biller. Neither submissions of bills with eBill Direct Debit payment nor eBill Direct Debit proposals are allowed.

EBILL_DEBIT_SUPPORT_READ_ONLY_FOR_BILLER_ACCOUNT

Status code: 400

ebillDebitSupport is 'read only' for biller accounts

ebillDebitSupport for the biller's accounts can not be set by the network partner, neither for creating nor for updating a biller.

EBILL_DEBIT_PROPOSAL_NOT_ALLOWED

Status code: 403

Biller is not allowed to propose eBill Direct Debit to the bill recipient

The submission of an eBill Direct Debit proposal requires the permission of the bill recipient.

EBILL_DEBIT_PROPOSAL_REDUNDANT

Status code: 400

The same eBill Direct Debit proposal is already known to the system

eBill Direct Debit proposals that exactly match an existing eBill Direct Debit standing approval or an existing proposal are rejected.

EBILL_DEBIT_CHARGEBACK_MODE_UNSUPPORTED

Status code: 400

Provided chargeback mode is not supported

The chargeback mode NOT_SUPPORTED is only valid for business eBill users.

7.16. Technical Errors

TECHNICAL_ERROR

Status code: 500

Technical error on server side

Processing yielded a technical error. This is a probable server-side bug and was automatically flagged for review.

Appendix A: Roadmap API versions

Version	Go Live	End of Life
Network Partner API V1	September 2019	March 2021
Network Partner API V2.0	September 2020	October 2022
Network Partner API V2.3	March 2021	October 2022
Network Partner API V2.4	June 2021	October 2022
Network Partner API V2.5	September 2021	October 2022
Network Partner API V2.7	April 2022	October 2022
Network Partner API V3.3	April 2022	December 2023
Network Partner API V3.4	June 2022	December 2023
Network Partner API V4.0	June 2023	November 2024
Network Partner API V5.0	July 2024	not planned

Appendix B: Changelog

B.1. Changes between v4 and v5

B.1.1. Overview

- Removed ESR as it is EOL
- Introduced new bill recipient subscription option 'Subscription at the eBill infrastructure'

B.1.2. Changes of the eBill XSD

- removed ESR as possible debit account for the payment information
- added new element 'unstructuredAddress' to type 'accountHolderType'
- make address (structured or unstructured) mandatory in 'billRecipient' and 'accountHolderType' if provided
- make countryCode, addressLine1, and addressLine2 mandatory in 'unstructuredAddressType'

B.1.3. Changes of the Swagger definitions

- added new definition 'BillRecipientURLSubscription'
- added new optional property 'birthDate' to 'BillRecipientIdentification'
- added new optional property 'postalCode' to 'BillRecipientIdentification'
- added new optional property 'certificationIds' to 'Biller'
- added new problem types:
 - 'BILLER_REFERENCED_CERTIFICATION_DOES_NOT_EXIST'
 - 'BILL_RECIPIENT_SUBSCRIPTION_FORM_INVALID'
- Removed ESR and BESR definitions from BillerAccount definition
- Removed ESR and BESR definitions from the filter for the biller search
- removed problem types used in conjunction with ESR:
 - 'BILLER_ACCOUNT_TYPE_ESR_NOT_ALLOWED'
 - 'BILLER_ACCOUNT_MISSING_BESRID'
 - 'BILLER_ACCOUNT_BESRID_NOT_ALLOWED'
 - 'BILLER_ACCOUNT_BESRID_COMBINATION_ALREADY_EXISTS'
 - 'BILLER_ACCOUNT_ESR_PARTICIPANT_NUMBER_CURRENCY_INCONSISTENCY'
 - 'BILLER_ACCOUNT_INVALID_ACCOUNT_NUMBER_COMBINATION'
 - 'BILLER_INVALID_ESR_PARTICIPANT_NUMBER'
 - 'BILLER_ACCOUNT_WITH_INVALID_BESR_ACCOUNT_SUPPLEMENT_COMBINATION'
 - 'BC_PAYMENT_INFORMATION_INVALID_CURRENCY_CODE_FOR_ESR'

- 'BC_PAYMENT_INFORMATION_ESR_INVALID_FOR_BESR'
- 'BC_PAYMENT_INFORMATION_INVALID_ESR_REFERENCE_NUMBER'

B.1.4. Changes of the Documentation

- added description to Biller about the biller certifications

B.1.5. Changes of the Swagger operations

- introduced new operation 'certifications' to retrieve all available certifications
- introduced new operation 'bill-recipient-subscription-initiations-url' for subscription option 'Subscription at the eBill infrastructure'

B.2. Changes between 5.0.5 and 5.0.10

B.2.1. Changes of the Documentation

- removed all the information regarding biller attachments
- adapted documentation for 'BC_PAYMENT_INFORMATION_INVALID_PAYMENT_DUE_DATE' to reflect ebill debit payment mode due date restrictions

B.2.2. Changes of the eBill XSD

- adapted documentation for due date of payment information to reflect ebill debit payment mode

B.2.3. Changes of the Swagger operations

- Operation /billers/{billerId}/attachments was removed
- Operation /billers/{billerId}/attachments/{attachmentId} was removed
- mime type application/pdf was removed from /billers/{billerId}/assets/{assetId}

B.2.4. Changes of the Swagger definitions

- correspondenceLanguage was removed from BillRecipientURLSubscription