

eFaktura Submitter Service

User guide

Versjon 1.3

1. CHANGELOG	3
2. INTRODUCTION	3
2.1. INTENT OF THIS DOCUMENT	3
2.2. TARGET AUDIENCE	3
2.3. OTHER DOCUMENTATION	3
2.4. EXTERNAL RESOURCES	3
3. FUNCTION OVERVIEW	4
3.1. METHOD CALLS	4
3.2. COMMON PARAMETERES FOR ALL FUNCTIONS	5
3.3. OTHER ERRORS	5
4. IMPEMETING THE PROVIDER SERVICE FUNCTIONS	5
4.1. SCENARIO DESCRIPTION	5
4.2. GETBLACKLIST	6
4.3. ADD INVOICE ON BEHALF OF ISSUER	6
4.3.1. OTHER DOCUMENT TYPES	6
4.4. GETTING STATUS ON A CONSIGNMENT	7
4.5. GETTING STATUS ON AN INVOICE	8
4.6. CANCELLING A CONSIGNMENT	8
4.7. CANCELLING AN INVOICE	8
4.8. VALIDATING SECURITY SIGNATURE	8
APPENDIX:	9
I - ALIAS LOOKUP GUIDE	9

1. Changelog

Version	Description	Author	Date
0.1	Created document	Lars Lech-Hanssen	28.05.2016
0.2	Adjustements to WSDL	Morten Gjestad	04.11.2016
0.3	Corrections after technical feedback	Morten Gjestad	14.11.2016
0.4	Added appendix I Alias Lookup Guide	Lars Berg	23.03.2017
1.1	Updated appendix I – Alias lookup guide	Øystein A. Jansen	12.09.2018
1.2	Added validateSecuritySignature	Øystein A. Jansen	18.10.2019
1.3	Added invoiceDocumentType info + cleanup	Øystein A. Jansen	12.03.2020

2. Introduction

2.1. Intent of this document

The eFaktura Submitter Service – documents the online web service that is offered to Submitters in the eFaktura solution. The document describes all function calls and parameters that will necessary to use the eFaktura solution.

This document and the eFaktura Submitter WSDL will be the needed to implement this service.

2.2. Target Audience

Technical staff in companies that is to become a provider in the eFaktura solution.

2.3. Other documentation

WSDL
HTML version of WSDL
Alias Lookup Guide (Appendix I)

2.4. External resources

XML standard : www.w3.org
SOAP : ws.apache.org/soap/

3. Function overview

3.1. Method calls

The following method calls are available in the service.

Functions
getBlackList
getEfakturaIdentifiers
addInvoices
getInvoiceStatus
getConsignmentStatus
cancelConsignment
cancelInvoice
validateSecuritySignature

3.2. Common parameters for all functions

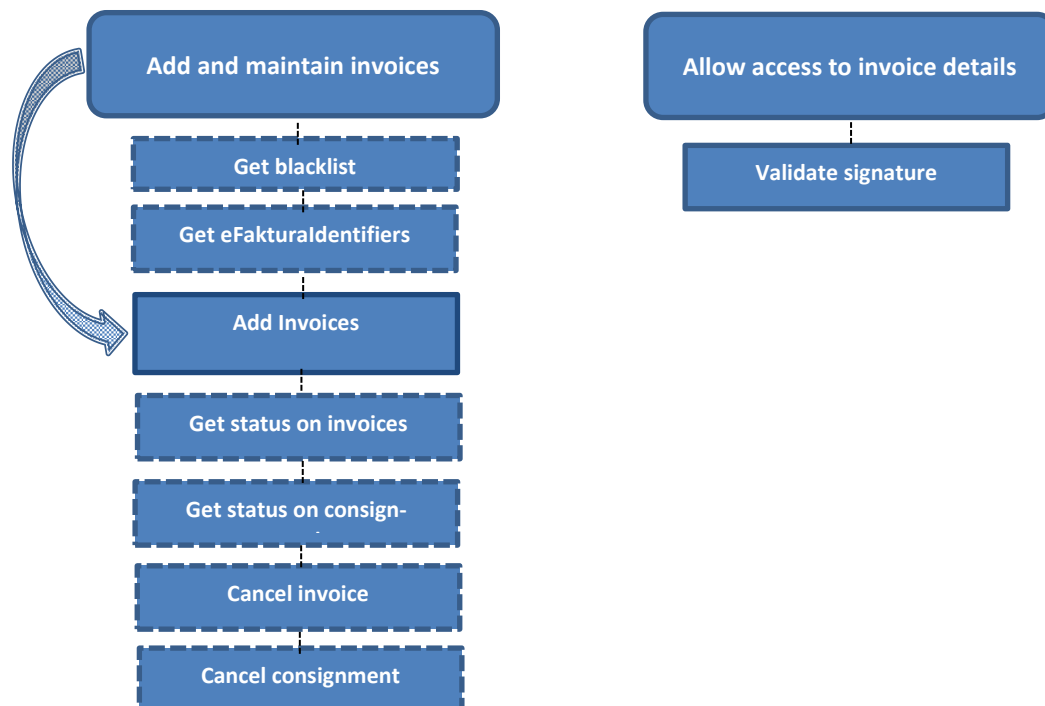
Parameter	Explanation
AuditInformationType	Audit information consists of a MessageIdentifier. This is a string that is used for auditing and logging. It must be between 5 and 50 characters. The provider is free to add the content of the MessageIdentifier, but it is recommended to make it unique for future reference.
UsernameToken	The username token is a part of the WSSE security which is a part of the SOAP header. It must be added for all calls with the username and password that is provided by Nets. If not provided no access will be granted to the service.

3.3. Other errors

If the username and password supplied in the UsernameToken tag in the header is incorrect a ServerFault will be thrown.

4. Implementing the provider service functions

4.1. Scenario description



4.2. *getBlackList*

The submitter is imposed to check whether an Invoice Issuer is blacklisted before adding invoices on behalf of the issuer.

Step	Description	Function call
#1	Return a list of blacklisted invoice issuers.	getBlackList

4.3. *Add invoice on behalf of issuer*

If an issuer is linked to submitter, the submitter may add invoices on behalf of the issuer. Submitter must verify that the consumer accepts invoices and that the issuer is not blacklisted or blocked by the consumer. There are several function calls that can be made to determine this. Based on the information the submitter has about the consumer, the following function calls could be made.

For submitters who has concession to handle SSN (Social security number) and using this to identify the consumer.

Step	Description	Function call
#1	Create invoices on behalf of issuer using SSN for identifying the consumer	addInvoices
#2	By providing the consignment id, a status of all the invoices ca be retrieved. Note that if the consignment is not processed the status will be not found.	getConsignmentStatus

For submitters using alias to identify the consumer.

Step	Description	Function call
#1	If the submitter have the consumers alias, lookup the efakturaIdentifier	getEfakturaIdentifiers
#2	Create invoices on behalf of issuer using efakturaIdentifier for identifying the consumer	addInvoices
#3	By providing the consignment id, a status of all the invoices can be retrieved. Note that if the consignment is not processed the status will be not found.	getConsignmentStatus

4.3.1. *Other Document types*

eFaktura supports several document types in addition to the standard invoice type.

Document type is set using the invoiceDocumentType field in addInvoices.

invoiceDocumentType value	Description
CREDIT_NOTE	Kreditnota
DUNNING	Purring
COLLECTION_NOTICE	Inkassovarsel
PAYMENT_REQUEST	Betalingsoppfordring
REMINDER	Betalingspåminnelse
ENFORCEMENT_WARNING	Varsel om tvangsfullbyrdelse

4.4. Getting status on a consignment

When adding invoices a consignment id must be provided. Based on the consignment id and the issuer a status on each invoice may be retrieved.

Step	Description	Function call
#1	Return the status of the consignment and the shipments. Note that if the consignment is not processed the status will be not found.	getConsignmentStatus

4.5. Getting status on an invoice

When adding invoices a consignment id and an invoice id must be provided. Based on these ids and the issuer a status the invoice may be retrieved.

Step	Description	Function call
#1	Return the status of the invoice. Note that if the invoice is not processed the status will not be found.	getInvoiceStatus

4.6. Cancelling a consignment

When cancelling a consignment, a consignment id and invoice ids must be provided. The consignment id can be used to cancel the invoices in a consignment. Note that invoices that have been processed by the consumer cannot be cancelled.

Step	Description	Function call
#1	Sends a request for cancellation of invoices within a consignment	cancelConsignment

4.7. Cancelling an invoice

When cancelling an invoice, a consignment id and invoice id must be provided. These ids can be used to cancel the invoice. Note that invoices processed by the consumer, cannot be cancelled

Step	Description	Function call
#1	Sends a request for cancellation of an invoice	cancelInvoice

4.8. Validating security signature

Validates the security signature for invoice details.

Step	Description	Function call
#1	Sends a request for validating and decrypting a security signature.	validateSecuritySignature

Appendix:

I - Alias Lookup Guide

This document is meant to serve as a suggested guideline for developers implementing the eFaktura Service through which they will do lookups in the alias register. These guidelines are based on the regulations enforced by Bits and the service parameters defined by Nets. All consumers who can receive eFaktura have been assigned a unique identifier called eFakturaAdresse, which is returned when a match is found through alias lookups.

The aliases assigned to consumers are divided into three categories:

Unique Aliases

- SSN
- eFakturaReferanse + Organization Number
- eFakturaAdresse
 - *Note: Doing lookups using the eFakturaAdresse is made possible so that issuers can verify whether a consumer is an active “eInvoice by default” user.*

Primary Aliases

- Phone number
- Email address

Secondary Aliases

- Zip code (Exact match)
- Birthdate
- Parts of address
- Parts of name

In order to have the eFakturaAdresse returned, a combination of these alias parameters must be used to generate a match to one specific consumer. If the aliases used match with more than one consumer, the service will return an error code stating ”Multiple Receivers”.

It is possible to wildcard score the “Parts of name” and “Part of address” aliases. To do this lookups have to be sent using the first 2 letters of either alias followed by the symbol *.

The following rules outline which alias combinations will return matches when doing lookups:

1. Match on 1 **Unique Alias**.
2. Match on both **Primary Aliases**.
3. Match on 1 **Primary Alias** and 1 **Secondary Alias**.
4. Match on 3 **Secondary Aliases**.

Best practise use of getEfakturaIdentifiers

The method call getEfakturaIdentifiers follows the rules described above and makes prioritized lookup searches to increase the chances for users to get a match and to minimize response time.

Lookups are generated using the following logic:

1. If match on any 1 of the **Unique Aliases** the eFakturaAdresse is returned.
 - a. If both **Unique Aliases** are provided and match different consumers an error is returned.
2. If both **Primary Aliases** are provided and match one specific consumer, the eFakturaAdresse is returned.
 - a. If the **Primary Aliases** match different consumers, an error is returned.
3. If only 1 **Primary Alias** is provided in combination with 1 or more **Secondary Aliases** the method call will first do a lookup using the **Primary Alias** and retrieve one or more consumers. This/these consumers are then matched with the provided **Secondary Aliases** in the following order:
 - a. **Primary Aliases** + Zip Code
 - b. **Primary Aliases** + Birthdate
 - c. **Primary Aliases** + “parts of name” provided
 - d. **Primary Aliases** + “parts of address” provided

If a match is found with one consumer, the eFakturaAdresse is returned.

4. If only **Secondary Aliases** are provided the method will first limit the lookup by using the Zip Code.
Note: Using only Secondary Aliases will result in a long response time.

Developers can choose to either supply all the information they have available when calling the method or supply what information is found to be relevant and of the best quality. If more than the minimum amount of information is provided, the method will return the eFakturaAdresse based on the first unique match following the logic described above.

To ensure the quality of the response given and the performance of the method call, developers are advised to prioritize the alias parameters in the following order:

1. When a **Unique Alias** is available, only this should be used.
2. When both **Primary Aliases** are available, only these should be used.
3. When only one **Primary Alias** is available, the **Secondary Aliases** Zip Code and/or Birthdate should be considered most relevant.
 - a. If Zip Code and Birthdate are not available or won't return a unique match, the “Parts of address” should be applied by wildcard scoring after the first three letters of the first address line. Example: Stensberggata 21 = “Ste*”.
 - b. If “Parts of address” is not available the “Parts of name” should be used by wildcard scoring the first two letters of the last name. Example: Johansen = “Jo*”.
4. Using only **Secondary Aliases** should be avoided if possible as this will result in a long response time.
 - a. **Note: It is not possible to use only Secondary Aliases if the Zip Code parameter is not provided.**

Graphical representation of alias prioritization

Element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SSN	█														
eFakturaRef+Orgnr		█													
eFakturaAdresse			█												
PhoneNumber				█	█	█	█	█							
EmailAddress				█					█	█	█	█			
ZIP					█				█				█	█	█
Birthdate						█				█			█	█	
Parts of Name							█				█		█		█
Parts of Address								█				█		█	█