Appendix 1 - Alias Lookup Guide

This document is ment 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 Mastercard. 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 "MultipleReceivers".

It is possible to wildcard score the "Parts of name" and "Part of address" aliases. To do this lookup it has 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 Alias

Issuers organization number can be added to all calls to check if receiver has blocked issuer.

Best practices for 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 + ZipCode

       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. *Birthdate must always match if sent.*

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 eFakturaIdentifier 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.

Other recommendations:

- Send multiple API-calls in parallel rather than one by one for optimal performance
- Issuers should assess if they need to check all customers or just non-digitized customers before invoicing. If only non-digitized customers are checked automatic exception handling must be in place when invoices are rejected
- Routine for JTTA-lookup should be independent of the actual invoice sending in production.
  - If JTTA-lookup fail this should not stop the invoice production
- eFakturaIdentifier on receiver should be stored is issuers or vendors invoicing system if possible. eFakturaIdentifier is a nine-digit static global value assigned to invoice receivers when they are created in the eFaktura system.

Use of "eFakturaReferanse" and organization number as lookup mechanism

The use of this combination can be used to find the eFakturaIdentifier for a receiver who was previously enrolled using the eFaktura 1.0 solution.

This functionality can also be used to analyze how many invoices that will change receiver if an issuer changes from using both eFaktura 2.0 og 1.0 addressing of eFaktura to "clean" 2.0 addressing with eFakturaIdentifier. This can be done using the following method:

1. Do a call with eFakturareferanse+ organization number
2. Do call with customer data (primary and/or secondary aliases
3. If the same eFakturaIdentifier is returned, the same receiver will receive an invoice if change addressing with eFakturaIdentifier. If not another person will receive the invoice if change to "clean" eFaktura 2.0 addressing.

Graphical representation of alias prioritization

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

Functionality to qualify matches (returnOnlyMatchingKeyParams)

The JTTA search functionality (getEfakturaIdentifiers) attempts to match the provided alias values with a unique eFaktura receiver by running them through a set of allowed alias combinations in a prioritized order.
If a unique match is found for an alias combination, the search stops and a response is provided - alias combinations of lower priority are not considered.

As an example, if the search gets a unique match for the combination phone+email, the match would be returned without the search considering name or address.

In order to give the users of the search the possibility to better assess the quality/certainty of the search response without making the search itself too strict, the returnOnlyMatchingKeyParams functionality was added.

returnOnlyMatchingKeyParams is a Boolean parameter in the getEfakturaIdentifiers API call that controls how the search parameters (getEfakturaIdentifiersKey) in the request are echoed in the response.

False - The response contains all the search parameters provided in the request; no logic applied.

True - The response only contains the search parameters from the request that match the data for the person that was a unique match for the search (following the alias search pattern).

Example:

	User provided info	eFaktura registry
Phone	+4711111111	+4711111111
Email	test@test.no	test@test.no
First name	Bob	Bill
Last name	Test	Test

A user sending in all four alias values in the search would get the following alias values back in the response depending on the returnOnlyMatchingKeyParams value set:

returnOnlyMatchingKeyParams	false	true
Phone	+4711111111	+4711111111
Email	test@test.no	test@test.no
First name	Bob	
Last name	Test	Test

Appendix 2 Submitting invoices using API

Mastercard offers a SOAP-API for submitting eInvoices.

Offered functionality:

- SOAP API for submission of complete invoice specification
  - Options are PDF, Embedded EHF file or fields for mapping a HTML invoice with standard template
    - NB: If submitter has the role as Invoice Hotel (fakturahotell), URL to invoice specification will replace the above-mentioned functionality
  - Option to submit up to 1000 invoices in each call
    - See best practice below
  - Support for submission in multiple parallel threads ensures high processing speed
- Functionality for checking status based on unique IDs that are generated both at shipment and invoice level
  - { 'IN_PROGRESS', 'PROCESSED', 'REJECTED', 'STOPPED' }
  - Replaces file receipts
- Support for new document types
  - Reminders, collection notices etc

Best practice

- For best performance and response time, it is recommended that actors submit several API calls in parallel and not one at a time
- Submit as few invoices as possible per call as this gives the highest processing speed
- Ask about status both at consignment and invoice level (consignment and invoice)
  - Shipment status provides status on whether the entire shipment is OK or failed and lists whether individual invoices have failed
  - Invoice status gives the status of individual invoice
  - Do not resend the invoice if there is no/delayed API response
    - If long delay, contact Mastercards customers service or incident management
- It is recommended that the issuer polls the status of submitted invoices according to a progressive logic
  - For example, after the following intervals: 1 min – 5 min – 15 min – 30 min – 1 hour etc.
- When using embedded EHF, the same rules for content apply as if EHF is submitted as a file via SFTP