

Technical interface – Application guidelines 2020

Incomes Register Unit

Version history

Version	Date	Description
1.0	5/11/2019	Published a new version of the document containing the annual changes for 2020.
		Added the allowed root element WageReportsOnePayerOneIncomeEarnerRequestToIR to the DataRequestToIR schema. This relates to the inclusion of the service Earnings payment reports - one payer. one income earner.
1.01	21/4/2020	Added information on how to delete files that have been named incorrectly to the Section 4.1.2 Uploading files into the SFTP In directory.

CONTENTS

1	Foreword	4
2	Identification and access right to the interface	4
2.1	TLS encryption setting.....	4
3	Structure and signatures of XML messages.....	5
3.1	Structure	5
3.2	Signature	5
3.2.1	Specifications related to signature generation.....	5
3.2.2	Description of the signature element's data contents	7
4	Service channels	9
4.1	SFTP channel	9
4.1.1	SFTP home directory.....	9
4.1.2	Uploading files into the SFTP In directory.....	12
4.1.3	File distribution via the SFTP Out directory	13
4.1.4	File naming	14
4.2	Web Service channel	15
4.2.1	Sending messages in the Web Service channel	16
4.2.2	Web Services	17
5	Record size and allowed root elements	21
6	Versioning.....	23
7	Error feedback.....	24
7.1	Web Service channel	24
7.1.1	HTTP errors.....	24
7.1.2	SOAP errors	24
7.1.3	Business errors	24
7.2	SFTP channel	24
7.3	Error structures of returned message.....	25
7.3.1	Message-level errors (MessageErrors)	25
7.3.2	Record-level errors (DeliveryErrors)	25
7.3.3	Item-level errors (ItemErrors).....	26

1 FOREWORD

This document describes guidelines related to the implementation of the technical interface of the Incomes Register from the system integrator's perspective. Message structures, validation rules and the use of the services are described in separate documents.

The purpose of this document is to specify the guidelines for the technical implementation, with a sufficient level of detail to allow the parties to ensure that the technologies they choose meet the requirements of the Incomes Register's service interface. The parties can also specify and implement their own systems on the basis of this document.

2 IDENTIFICATION AND ACCESS RIGHT TO THE INTERFACE

The identification of a party invoking the Incomes Register's technical interface is implemented either with a certificate or a PKI key pair connected to a certificate, depending on the service channel. Practices related to obtaining a certificate are described on the incomesregister.fi website under certificate service. In the Web Service channel, the party is identified using an SSL/TLS customer certificate.

In addition to a certificate, a username has also been delivered to a party using the SFTP channel. Identification is handled using SSH key identification, with the party using the username and the private key of the key pair generated during the certificate application process. The Incomes Register has the public key of the key pair in its possession.

A party must agree on access rights to the technical interface with the Incomes Register authority. Access rights are agreed service-specifically. The details on agreeing on access rights will be further specified at a later date.

2.1 TLS encryption setting

The Incomes Register's Web Service channel offers an HTTPS connection for the use of parties using the technical interface. TLS protocol version 1.2 must be used to establish the connection. The party's system must support one of the following cipher suites (combinations of encryption algorithms) of the TLS 1.2 protocol:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
- TLS_DHE_DSS_WITH_AES_128_CBC_SHA256

3 STRUCTURE AND SIGNATURES OF XML MESSAGES

3.1 Structure

The structure of the records in XML format submitted to and distributed from the Incomes Register are described using XSD schemas (XML Schema Definition Language, <http://www.w3.org/TR/xmlschema11-1>).

3.2 Signature

The integrity and non-repudiation of the data submitted to and distributed from the Incomes Register are ensured with an electronic signature. The signature is implemented using the XML Enveloped Signature mechanism; its processing rules and structure are described in the document, "XML Signature Syntax and Processing" (<http://www.w3.org/TR/xmlsig-core/>).

The party signs the records it delivers to the Incomes Register with the certificate received from the Incomes Register's certificate service. The Incomes Register signs the records and response messages it generates with its own certificate. If the party so wishes, it can verify the XML signature created by the Incomes Register and the signature certificate used by the Incomes Register with the help of the certificate chain. The verification of the signature and the signature certificate allows the party to ensure that the message was sent by the Incomes Register and that no-one has altered the message after signing. The issuer certificates and the related certificate chains for the certificates issued by the certificate service can be downloaded from the www.incomesregister.fi website.

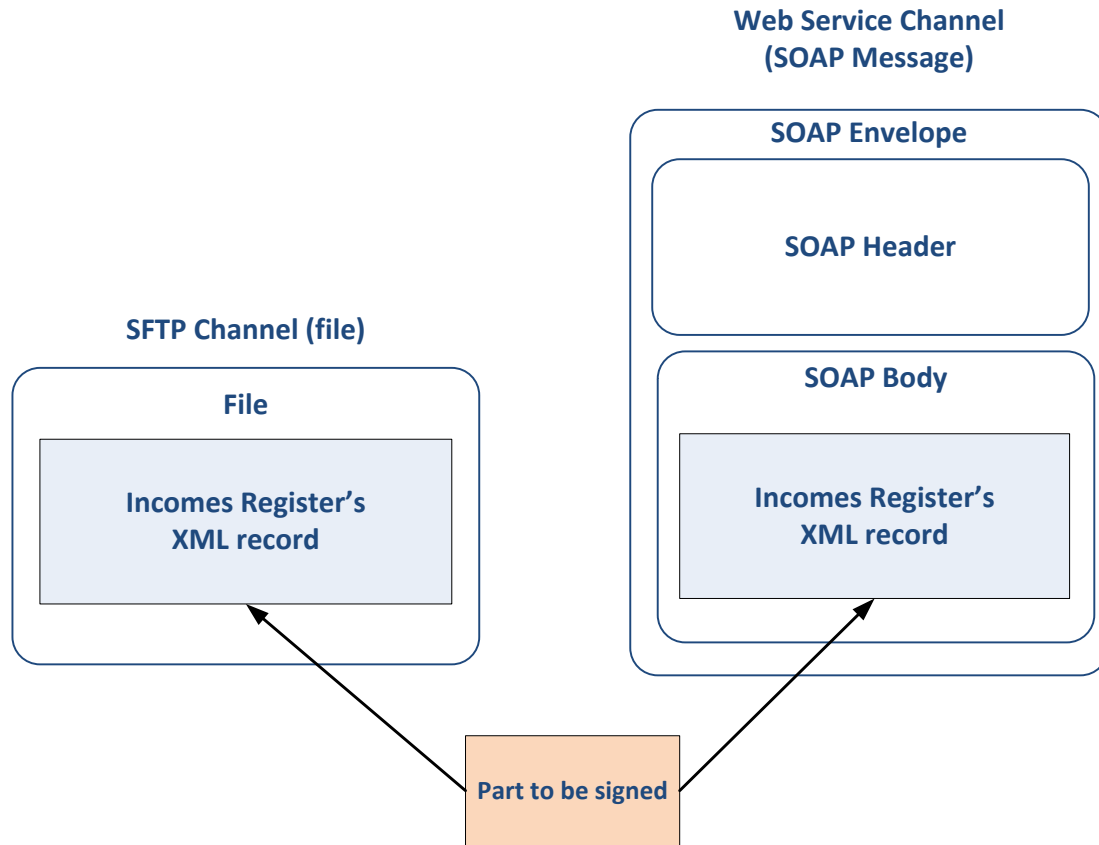
The subsections of this section define the procedure for generating a signature approved by the Incomes Register, and the required and allowed data contents of the signature. The use of canonicalisation, digest value calculation and signature calculation methods and 'transforms' not described in this document will lead to a failed signature verification and the rejection of the record.

3.2.1 Specifications related to signature generation

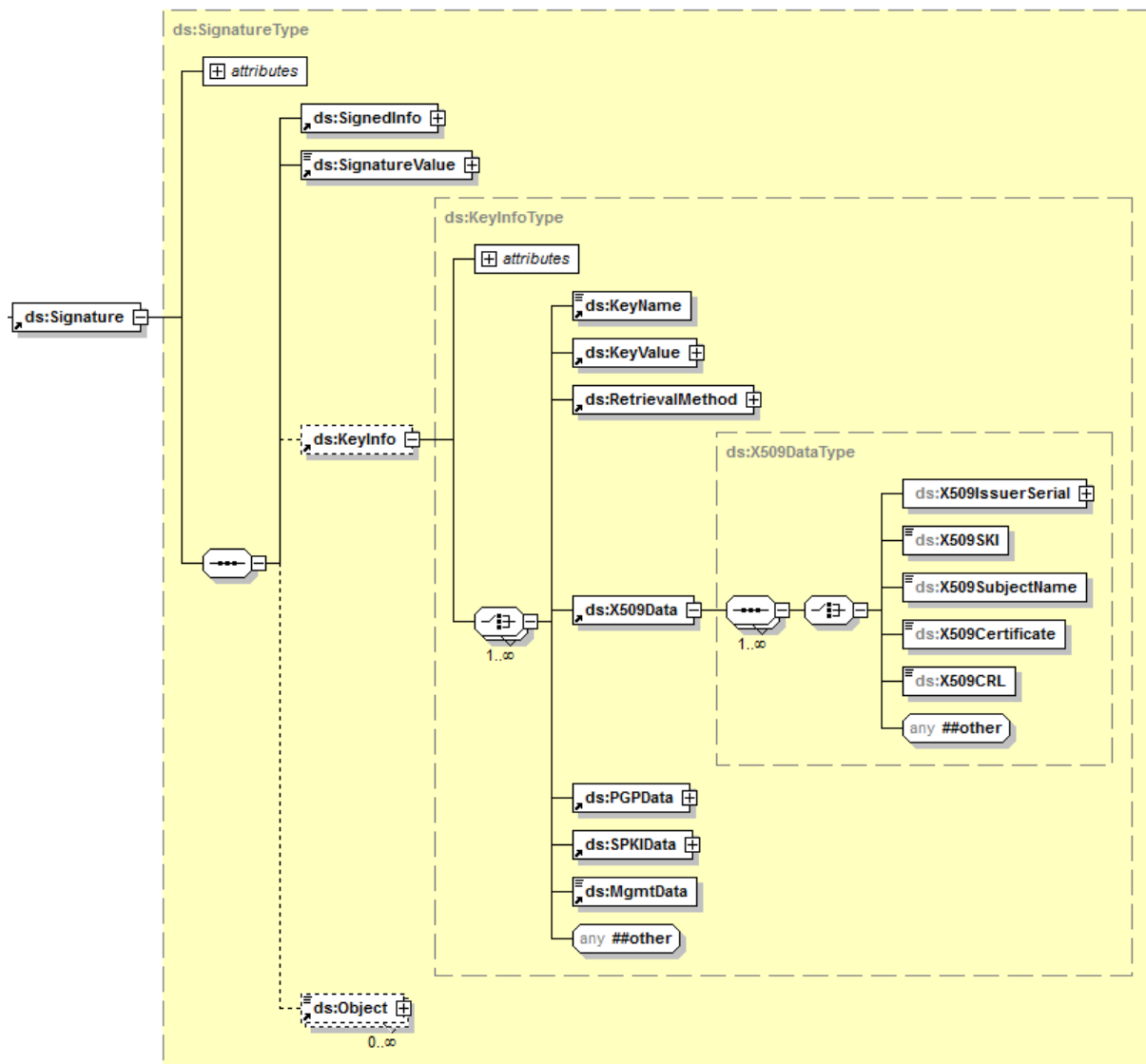
The signature of a record in XML format submitted to the Incomes Register must meet the following requirements:

1. The signature must be an Enveloped Signature, placed as the last child element of the document's root element. This place is also specified in the XML schema of each record type. The signature must be generated from the entire document.
2. A private key related to the X.509 certificate and allocated by the Incomes Register must be used as the signature key.
3. The SHA256 algorithm must be used in the generation of the digest calculated during signature generation.
4. The signature must be generated using the RSA-SHA256 algorithm.
5. The canonicalisation of the signature must use "Exclusive XML Canonicalization Version 1.0".
6. The signature certificate (with the public key) must be delivered with the signature in the KeyInfo/X509Data/X509Certificate element.

Regardless of the service channel, the record to be delivered to the Incomes Register is always what is signed, not the frame. When the Web Service channel is used, the signature is generated from the element that is located inside the Envelope/Body element and conforms to the Incomes Register schema. This element, signed according to the documentation, is then placed inside a SOAP Body element before it is sent to the Incomes Register. The signed record must not be changed during this stage.



3.2.2 Description of the signature element's data contents



Element data:

The data required by the Incomes Register to be included in the signature element is described in the table below.

Data designation	Type	V/ M	Label
Signature	xsig:SignatureType	M	Element containing the signature.
SignedInfo	ds:SignedInfoType	M	Element containing the description of the signature generation method.
CanonicalizationMethod	ds:CanonicalizationMethodType	M	Element defining the canonicalisation algorithm. An example of the value of the Algorithm attribute: <ul style="list-style-type: none"> • http://www.w3.org/2001/10/xml-exc-c14n#
SignatureMethod	ds:SignatureMethodType	M	Element defining the algorithm generating the signature. There can be only one of these elements. An example of the value of the Algorithm attribute: <ul style="list-style-type: none"> • http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Reference	ds:ReferenceType	M	Element defining the algorithm generating the signature digest. There can be only one instance of this element. The value of the URI attribute must be blank (URI = "") which means that the signature applies to the entire document.
Transforms	ds:TransformsType	M	Element defining the transforms performed before the signature digest is calculated. There can be only one instance of this element.
Transform	ds:TransformType	M	Elements defining the transform performed before the digest calculation. There must be at least one element, the first of which must be a Transform element conforming to the Enveloped Signature specification. An example of the value of the Algorithm attribute: <ul style="list-style-type: none"> • http://www.w3.org/2000/09/xmldsig#enveloped-signature
DigestMethod	ds:DigestMethodType	M	Element defining the algorithm used in the digest calculation. There can be only one instance of this element. An example of the value of the Algorithm attribute: <ul style="list-style-type: none"> • http://www.w3.org/2001/04/xmenc#sha256
DigestValue	ds:DigestValueType	M	Element containing the digest.
SignatureValue	ds:SignatureValueType	M	Element containing the actual signature.
KeyInfo	ds:KeyInfoType	M	Element containing information on the signature key. Although this element is not mandatory according to the schema, the Incomes Register requires the presence of this element. Only an X509Data element may be entered as its contents.
X509Data	ds:X509DataType	M	Element containing information on the signature key. At a minimum, an X509Certificate element must be entered as its contents.
X509Certificate	base64Binary	M	Element containing the Base64-encoded signature certificate.

4 SERVICE CHANNELS

The services in the technical interface of the Incomes Register can be used via three different channels:

- SFTP, deferred;
- Web Service, real-time; and
- Web Service, deferred.

The choice of which channel to use is based on need.

4.1 SFTP channel

The SFTP (SSH File Transfer Protocol) channel is primarily intended for use in situations where there is a large amount of data to be submitted to the Incomes Register. The data is transferred in XML files in such a manner that a single file may contain an XML message of only one schema.

4.1.1 SFTP home directory

An SFTP home directory is created in the Incomes Register for each party submitting and/or receiving files, containing In and Out directories for the party's files. The party uploads the files into the In directory and downloads the files created by the Incomes Register for the party from the Out directory.

For uploading and downloading files, the party needs a user account and an SFTP certificate that is linked to the party's SFTP home directory in the Incomes Register. The Incomes Register has SFTP certificates of different types for earnings payment data providers, benefits payment data providers and data users (see the table below).

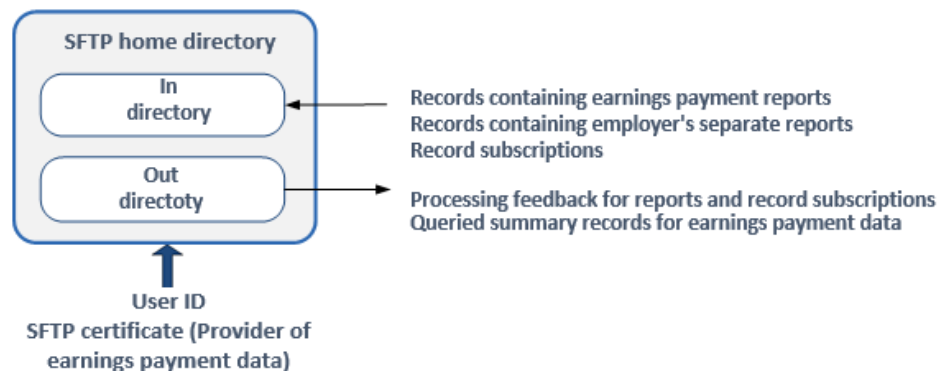
Purpose of use	Certificate type
Provider of earnings payment data	Data Providers SFTP Issuing CA v1
Provider of benefits payment data	IR Benefit Data Providers SFTP Issuing CA v1
Data user	IR Income Data Users SFTP Issuing CA v1

A party may have a maximum of three SFTP home directories in the Incomes Register, one for each of the purposes described in the table above.

When a party sends a record subscription to the Incomes Register via the SFTP channel, the record will be distributed into the SFTP home directory into which the record subscription was uploaded. When a party submits a signed record subscription via the upload service, the SFTP home directory used for the distribution of the record is selected based on the type of the certificate used in the record subscription's signature. When a party submits a record subscription via the upload service without a signature and requests the record to be delivered via the SFTP channel, the SFTP home directory is selected based on the Party type (PartyType) data item specified in the record subscription. When a party submits a record subscription via the e-service and requests the record to be delivered via the SFTP channel, the record is delivered to the SFTP home directory matching the subscriber's user role.

Example: Party acts as a provider of earnings payment data

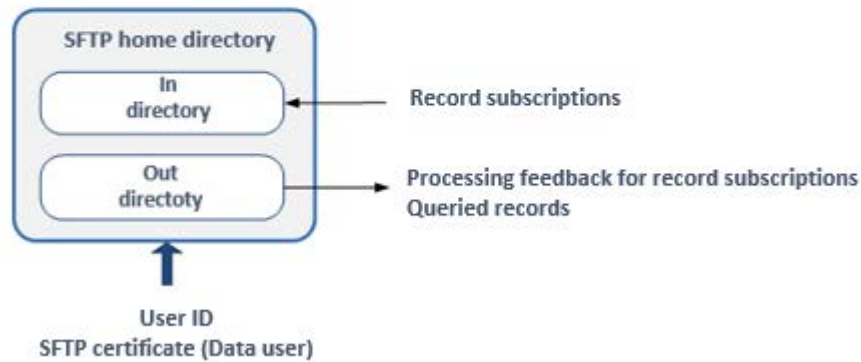
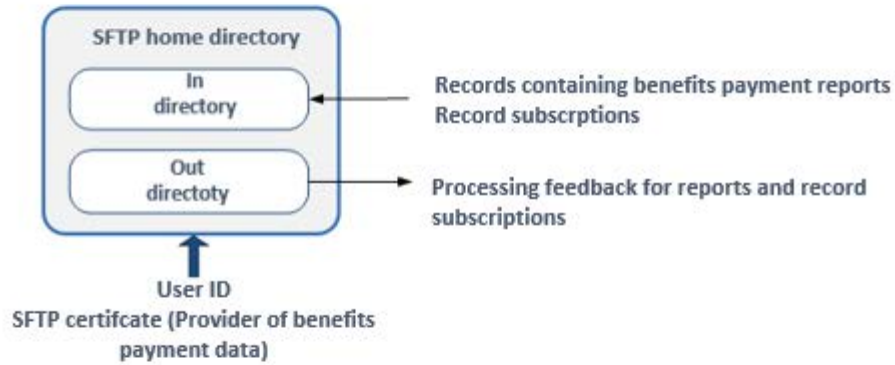
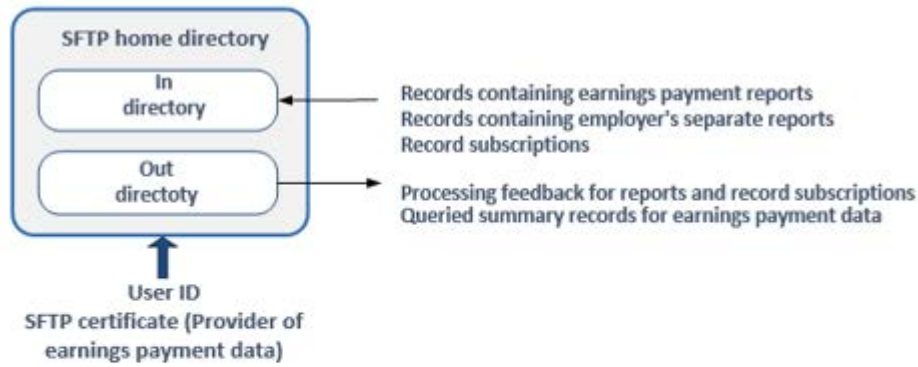
If a party (payer) submits earnings payment records to the Incomes Register and requests summary records of the payer's earnings payment data in XML format with a record subscription submitted via the technical interface, the records delivered and distributed via the party's SFTP home directory are as depicted in the figure below:



If the payer submits earnings payment reports for several suborganisations to the Incomes Register, all records are handled via the same In and Out directories. The files submitted to the Incomes Register must be named in such a manner that the processing feedback files intended for each individual suborganisation can be identified when retrieving the files from the Incomes Register. Files containing processing feedback can be identified using <FileId> strings appearing in the name of the record submitted to the Incomes Register, as the <FileId> string is copied to the name of the file containing the processing feedback (see Section File naming later in this document). Records generated for a party based on a record subscription (such as a summary record of a payer's earnings payment data) can be identified based on the <MainSubscriptionId> string appearing in the file name (the subscriber's primary subscription reference in the record subscription).

Example: A party acts as a provider of earnings and benefits payment data and as a user of Incomes Register data

If a party submits both earnings payment records (the organisation's own earnings payment data) and benefits payment records to the Incomes Register and uses Incomes Register data as a data user, the party will have three SFTP home directories in the Incomes Register. The records submitted and distributed via the directories are as depicted in the figure below:



4.1.2 Uploading files into the SFTP In directory

When a party uploads a file into the In directory, the party must name the file using the "tmp" file extension (see Section File naming later in this document). After a successful file transfer, the party changes the file extension to "xml". This ensures that the file is not processed by the Incomes Register while the upload is still incomplete.

The Incomes Register generates feedback on record processing that the party can download after the processing delay following the time of the upload. The processing feedback contains information on the correct and incorrect data in the record, as well as an itemised list of any errors detected during processing. The feedback file is saved in the party's Out directory.

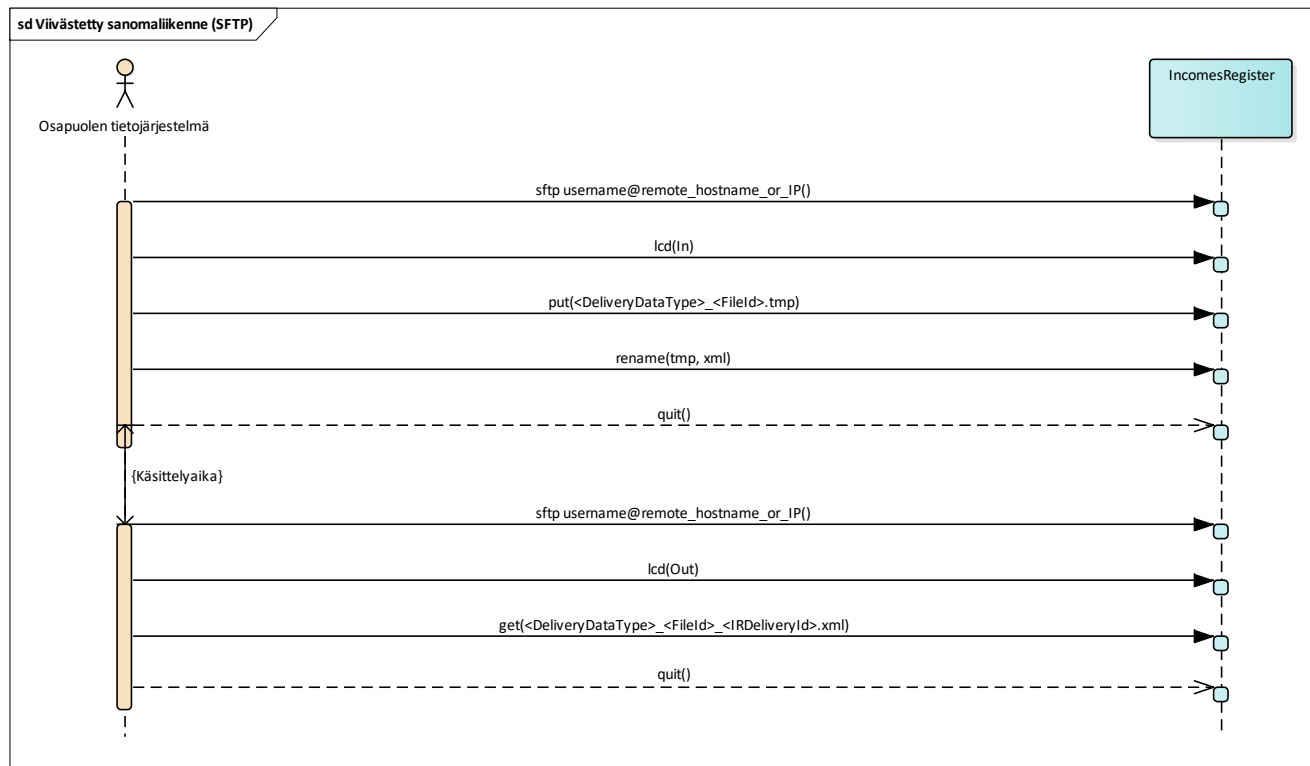


Figure 1. Sequence diagram of the messaging model in the SFTP channel.

A file saved in the In directory of the SFTP interface is removed from the directory as soon as its processing begins in the Incomes Register.

If the party does not name the file using the "xml" file extension within 7 days of the file's generation, the Incomes Register will delete the file. If the file has been named otherwise against the instructions, processing feedback will be generated containing an error message, provided that forming processing feedback is technically possible.

4.1.3 File distribution via the SFTP Out directory

The Incomes Register will deliver files distributed via the SFTP interface into the party's SFTP Out directory. Processing feedback distributed via the SFTP channel and records generated based on a record subscription and distributed via the SFTP interface are delivered into the directory. When the Incomes Register uploads the files related to a record being distributed into the Out directory, it names the files using the "tmp" file extension. After a successful upload, the Incomes Register renames the files using the "xml" file extension. The party downloads files with the "xml" file extension from the Out directory.

The Incomes Register will split the records in XML format, generated based on a record subscription and distributed via the SFTP interface, into a multipart file. The splitting ensures that a query does not generate a file that is difficult to handle due to technical issues arising from its large size.

The multipart file is generated in compliance with the schema of the distributed record, which means that the schema of an individual file being distributed is the same regardless of whether a single file or a multipart file is generated by the query. Each part of the multipart file is signed as a separate file.

The parts of the multipart file are numbered sequentially starting from 1. The total number of parts in the multipart file and the sequential number of the part file are added at the end of the name of the file being distributed from the Incomes Register. After renaming, the file names are of the following format:

<QueryDataType>_<MainSubscriptionId>_<SubscriptionId>_<sequential number of the query>_<IRQueryId>_<total number of part files>_<sequential number of the part file>.xml (see Section File naming later in this document). If only a single file is generated by the query, its name is of the following format:
<QueryDataType>_<MainSubscriptionId>_<SubscriptionId>_<sequential number of the query>_<IRQueryId>_1_1.xml.

The maximum size of one part of a multipart file is 100 MB. The part files may be of different sizes, and a part file may be smaller in size than 100 MB.

If a distributed record contains reports, they are sorted into the part files according to the report version's creation timestamp (CreatedTimestamp), starting from the oldest. If a record subscription has requested all versions of a report to be queried, they may be placed in different part files in accordance with the order of their timestamps. If a distributed record contains messages, they are sorted into the part files according to the times the messages were sent (Timestamp). If a distributed record contains log data, it is sorted into the part files according to the creation time of the log event (Timestamp).

All parts of a multipart file generated by one query have the same query metadata. In the distribution schemas, the Number of queried items value distributed in the Query summary data group is the total number of queried items in all of the part files.

The Incomes Register will upload the part files generated during the query into the party's SFTP Out directory with "tmp" file extensions one by one as described above. Once the entire multipart file has been successfully uploaded, the Incomes Register changes the file extension(s) to "xml". Because the part files are uploaded and renamed one by one, they are not available in the Out directory at exactly the same time. The party must download the number of part files indicated in the name of the file in the <total number of part files> section from the Incomes Register.

The timestamp of a part file is the time when the part file was saved in the Out directory. The timestamps are not necessarily in the order of the sequential numbering of the part files, i.e. a part file with a larger sequential number may have been saved in the Out directory earlier than a part file with a smaller sequential number.

It is recommended that the party delete files saved in the Out directory. If the party does not delete the files, they are deleted by the Incomes Register after their retention period is over.

4.1.4 File naming

File naming is described in Tables 1 and 2.

Record	Directory	File name format	Example
Data to be submitted to the Incomes Register	IN	<DeliveryDataType>_<FileId>.xml	100_87765434543.xml
Processing feedback distributed from the Incomes Register	OUT	<DeliveryDataType>_<FileId>_<IRDeliveryId>.xml	100_87765434543_850166cc02fa4a038da5ee36b990b07a.xml
Records distributed from the Incomes Register	OUT	<QueryDataType>_<MainSubscriptionId>_<SubscriptionId>_<sequential number of the query>_<IRQueryId>_<total number of part files>_<sequential number of the part file>.xml	300_2367756AC4_SUB1_12_d41f67294769429db2891693a2b84055_7_5.xml

Table 1. Naming of the files submitted to and distributed from the Incomes Register.

File name part	Description
DeliveryDataType	Value from the code set "Record type, submitted to Incomes Register (DeliveryDataType)".
FileId	A free-form reference assigned by the record sender to the file, uniquely identifying the file in the In directory. The maximum length is 40 characters. The allowed characters for this value are the numerals 0-9, the letters a-z and A-Z, and the special characters "_" and "-".
IRDeliveryId	Incomes Register's record reference that is a unique identifier assigned by the Incomes Register to the record.
QueryDataType	Value from the code set "Record type, distributed from the Incomes Register (QueryDataType)".
MainSubscriptionId	Subscriber's primary subscription reference.
SubscriptionId	Subscriber's secondary subscription reference.
IRQueryId	Incomes Register's query reference that is a unique identifier assigned by the Incomes Register to the queried record.
Sequential number of the query	The sequential number of queries for the primary subscription. The numbering begins from 1.
Total number of part files	The total number of parts in a multipart file generated during the record query.
Sequential number of the part file	The sequential number of a part file generated during the record query. The numbering begins from 1.

Table 2. Parts of the file name.

4.2 Web Service channel

The Web Service channel of the Incomes Register is implemented as a SOAP 1.1 interface (Simple Object Access Protocol, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>). The implementation is divided into deferred (asynchronous) and real-time (synchronous) services.

Real-time services are intended for the processing of individual data items (e.g. a single earnings or benefits payment report), when the usage situation requires an immediate processing response (e.g. a customer service situation).

Deferred services are intended for use when uploading data sets containing multiple items. In this context, deferred service means a dialogue between the information systems, where the execution of a function requires two separate service invocations. In the first service invocation, the party submits the record to the Incomes Register. Feedback for record processing is retrieved with the second service invocation. The individual service invocations are all synchronous in nature.

The services are described using the WSDL 1.1 description language (Web Service Description Language, <http://www.w3.org/TR/wsdl>). The WSDL description can be utilised to define the service and message references of the client application invoking the service.

When sending records via the Web Service channel, the following HTTP Headers must be set:

- SOAPAction: "enter the value of the soapAction field defined in the WSDL description of the service in question"
- Content-Type: text/xml;charset=UTF-8

4.2.1 Sending messages in the Web Service channel

In **deferred data transfer**, a party submits data to the Incomes Register and receives an acknowledgement message (AckFromIR). The acknowledgement message contains information on any errors that occurred during the reception of the data. If no errors occurred in message reception, the processing feedback for the reception can be queried after the processing delay, using the GetDeliveryDataStatus service.

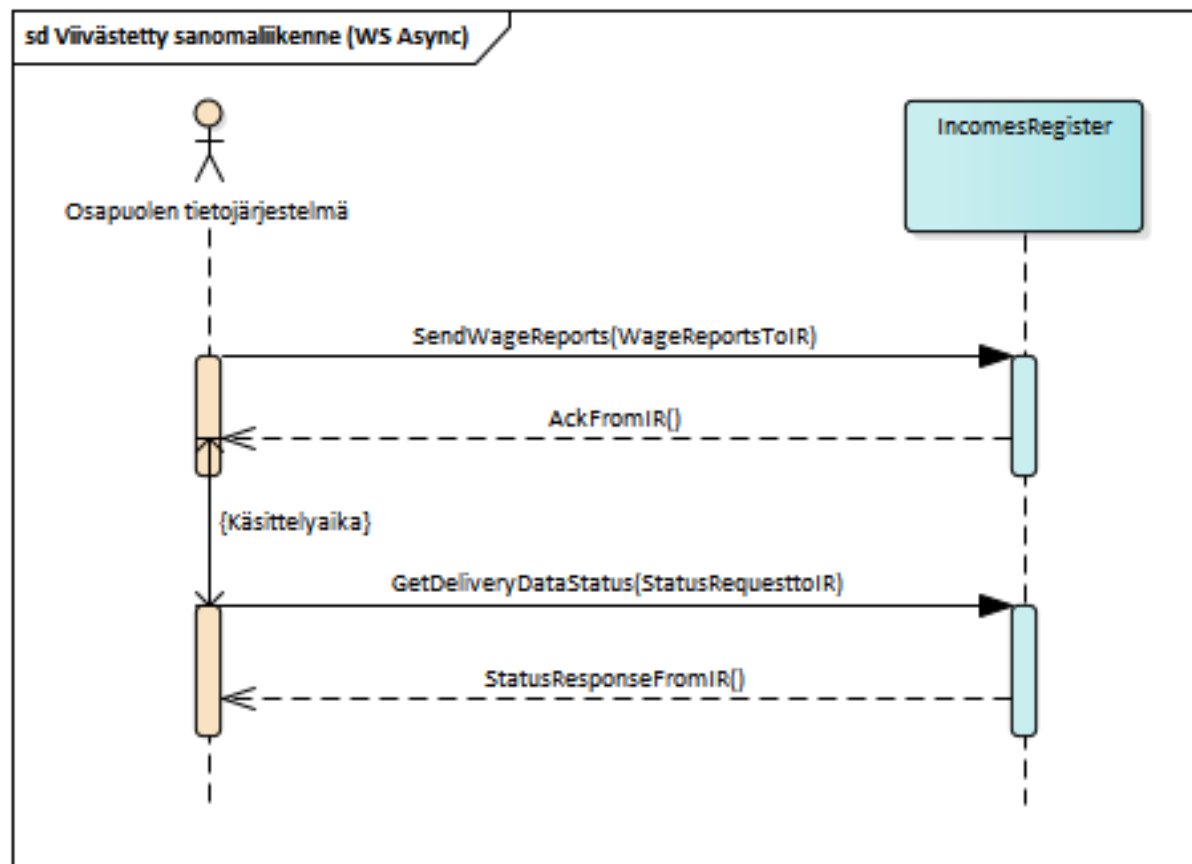


Figure 2. Sequence diagram of the messaging model in the deferred Web Service channel.

The above figure presents the sequence diagram of the deferred Web Service channel on a general level when earnings payment reports are submitted to the Incomes Register. The messaging model is similar when benefits payment reports are submitted (operation SendBenefitReports, schema BenefitReportsToIR).

In **real-time Web Services**, the Incomes Register responds to a service invocation with processing feedback (StatusResponseFromIR). The processing feedback contains information on the correct and incorrect data in the record, as well as an itemised list of any errors detected during processing.

4.2.2 Web Services

See below for all Web Services of the Incomes Register. Each service is described in its own WSDL description.

BenefitReportQueryService:

Operation	Deferred / real-time	Request message	Response message	Description
GetBenefitReportsOneIncomeEarner	Real-time	DataRequestToIR	BenefitReportsFromIR	Returns the benefits payments reports for a single income earner in accordance with the request message.
GetBenefitReportsOnePayerOneIncomeEarner	Real-time	DataRequestToIR	BenefitReportsFromIR	Returns the benefits payment reports generated by a single payer for a single income earner in accordance with the request message.
GetBenefitReportsOneIRReportId	Real-time	DataRequestToIR	BenefitReportsFromIR	Returns the benefits payment report related to a single Incomes Register report reference in accordance with the request message.

BenefitReportService:

Operation	Deferred / real-time	Request message	Response message	Description
SendBenefitReport	Real-time	BenefitReportsToIR	StatusResponseFromIR	Processing of a record containing a single benefits payment report.
SendBenefitReports	Deferred	BenefitReportsToIR	AckFromIR	Processing of a record containing several benefits payment reports.

EchoService:

Operation	Deferred / real-time	Request message	Response message	Description
SendEcho	Real-time	Echo	Echo	A service that can be used to test the operation of the connection and authentication. The service returns the received message to the service invoker as the response.

InvalidationService:

Operation	Deferred / real-time	Request message	Response message	Description
SendInvalidation	Real-time	InvalidationsToIR	StatusResponseFromIR	Processing of a record containing an individual cancellation.
SendInvalidations	Deferred	InvalidationsToIR	AckFromIR	Processing of a record containing several cancellations.

MessageService:

Operation	Deferred / real-time	Request message	Response message	Description
GetMessages	Real-time	DataRequestToIR	MessagesFromIR	Returns the messages sent to a single party.
SendMessage	Real-time	MessagesToIR	StatusResponseFromIR	Processing of a record containing an individual message.
SendMessages	Deferred	MessagesToIR	AckFromIR	Processing of a record containing several messages.

PayerSummaryReportQueryService:

Operation	Deferred / real-time	Request message	Response message	Description
GetPayerSummaryReportsOnePayer	Real-time	DataRequestToIR	PayerSummaryReportsFromIR	Returns the employer's separate reports from one payer in accordance with the request message.
GetPayerSummaryReportsOnePolicyNo	Real-time	DataRequestToIR	PayerSummaryReportsFromIR	Returns the employer's separate reports related to a single pension policy number in accordance with the request message.

PayerSummaryReportService:

Operation	Deferred / real-time	Request message	Response message	Description
SendPayerSummaryReport	Real-time	PayerSummaryReportsToIR	StatusResponseFromIR	Processing of a record containing a single employer's separate report.
SendPayerSummaryReports	Deferred	PayerSummaryReportsToIR	AckFromIR	Processing of a record containing several employer's separate reports.

StatusService:

Operation	Deferred / real-time	Request message	Response message	Description
GetDeliveryDataStatus	Deferred	StatusRequestToIR	StatusResponseFromIR	Returns processing feedback containing information on the correct and incorrect data in the record, as well as an itemised list of any errors detected during processing.

SubscriptionService:

Operation	Deferred / real-time	Request message	Response message	Description
ProcessSubscription	Real-time	SubscriptionsToIR	StatusResponseFromIR	Processing of a record containing a single record subscription.
SendSubscription	Deferred	SubscriptionsToIR	AckFromIR	Processing of a record containing a single record subscription.

WageReportQueryService:

Operation	Deferred / real-time	Request message	Response message	Description
GetWageReportsOneIncomeEarner	Real-time	DataRequestToIR	WageReportsFromIR	Returns the earnings payments reports for a single income earner in accordance with the request message.
GetWageReportsOnePayer	Real-time	DataRequestToIR	WageReportsFromIR	Returns the earnings payment reports generated by a single payer in accordance with the request message.
GetWageReportsOnePayerOneIncomeEarner	Real-time	DataRequestToIR	WageReportsFromIR	Returns the earnings payment reports generated by a single payer for a single income earner in accordance with the request message.
GetWageReportsOnePolicyNo	Real-time	DataRequestToIR	WageReportsFromIR	Returns the earnings payment reports related to a single pension policy number in accordance with the request message.

WageReportService:

Operation	Deferred / real-time	Request message	Response message	Description
SendWageReport	Real-time	WageReportsToIR	StatusResponseFromIR	Processing of a record containing a single earnings payment report.
SendWageReports	Deferred	WageReportsToIR	AckFromIR	Processing of a record containing several earnings payment reports.

5 RECORD SIZE AND ALLOWED ROOT ELEMENTS

The records to be submitted to the Incomes Register and records distributed from the Incomes Register have the size and number limitations described below. Also see below for a description of the root elements allowed in the different channels.

- An http-level error is returned in the Web Service channel if maximum record size is exceeded.
- An error message is returned in the processing feedback if maximum data number is exceeded (MessageError).
- Maximum size has been determined for the size of the SFTP channel transfer directory (operator-specific). An sftp transfer error is returned when submitting a record to the Incomes Register if maximum directory size is exceeded.

Service channel	Direction	Schema	Allowed root elements	Maximum record size	Maximum number of data (pcs)
SFTP	IN	InvalidationsToIR	InvalidationsRequestToIR	50 MB	10,000 cancellation data
SFTP	IN	PayerSummaryReportsToIR	PayerSummaryReportsRequestToIR	50 MB	10,000 employer's separate reports
SFTP	IN	SubscriptionsToIR	SubscriptionsRequestToIRAsync	50 MB	1 record subscription, containing no more than 10,000 secondary subscription query criteria (total number of all secondary subscriptions related to the primary subscription)
SFTP	IN	WageReportsToIR	WageReportsRequestToIR	50 MB	10,000 earnings payment reports
SFTP	IN	BenefitReportsToIR	BenefitReportsRequestToIR	50 MB	10,000 benefits payment reports
SFTP	IN	MessagesToIR	MessagesRequestToIR	50 MB	10,000 messages One message can contain no more than 5 appendices with a maximum size of 45 MB (base64 converted appendices in total).
SFTP	OUT	LogDataFromIR	LogDataFromIR	no limitation	no limitation
SFTP	OUT	MessagesFromIR	MessagesFromIR	no limitation	no limitation
SFTP	OUT	PayerSummaryReportsFromIR	PayerSummaryReportsFromIR	no limitation	no limitation
SFTP	OUT	StatusResponseFromIR	StatusResponseFromIR	no limitation	no limitation
SFTP	OUT	WageReportsFromIR	WageReportsFromIR	no limitation	no limitation
SFTP	OUT	BenefitReportsFromIR	BenefitReportsFromIR	no limitation	no limitation
Web Service real-time	IN	InvalidationsToIR	InvalidationRequestToIR	1 MB	1 cancellation data
Web Service real-time	IN	PayerSummaryReportsToIR	PayerSummaryReportRequestToIR	1 MB	1 employer's separate report
Web Service real-time	IN	SubscriptionsToIR	SubscriptionsRequestToIR	1 MB	1 record subscription, containing no more than 100 secondary subscription query criteria (total number of all secondary subscriptions related to the primary subscription)
Web Service real-time	IN	WageReportsToIR	WageReportRequestToIR	1 MB	1 earnings payment report
Web Service real-time	IN	BenefitReportsToIR	BenefitReportRequestToIR	1 MB	1 benefits payment report
Web Service real-time	IN	MessagesToIR	MessageRequestToIR	50 MB	1 message The message contains no more than 5 appendices with a maximum size of 45 MB (base64 converted appendices in total).
Web Service real-time	IN	DataRequestToIR	MessagesRequestToIR	1 MB	1 "record subscription" with no more than 20 query criteria

			PayerSummaryReportsOnePayerRequestToIR PayerSummaryReportsOnePolicyNoRequestToIR WageReportsOneIncomeEarnerRequestToIR WageReportsOnePayerRequestToIR WageReportsOnePayerOneIncomeEarnerRequestToIR WageReportsOnePolicyNoRequestToIR		
Web Service real-time	OUT	StatusResponseFromIR	StatusResponseFromIR	no limitation	no limitation
Web Service real-time	OUT	PayerSummaryReportsFromIR	PayerSummaryReportsFromIR	no limitation	150 employer's separate reports
Web Service real-time	OUT	WageReportsFromIR	WageReportsFromIR	no limitation	150 earnings payment reports
Web Service real-time	OUT	BenefitReportsFromIR	BenefitReportsFromIR	no limitation	150 benefits payment reports
Web Service real-time	OUT	MessagesFromIR	MessagesFromIR	no limitation	150 messages
Web Service deferred, upload service	IN	InvalidationsToIR	InvalidationsRequestToIR	50 MB	10,000 cancellation data
Web Service deferred, upload service	IN	PayerSummaryReportsToIR	PayerSummaryReportsRequestToIR	50 MB	10,000 employer's separate reports
Web Service deferred, upload service	IN	SubscriptionsToIR	SubscriptionsRequestToIRAsync	50 MB	1 record subscription, containing no more than 10,000 secondary subscription query criteria (total number of all secondary subscriptions related to the primary subscription)
Web Service deferred, upload service	IN	WageReportsToIR	WageReportsRequestToIR	50 MB	10,000 earnings payment reports
Web Service deferred, upload service	IN	BenefitReportsToIR	BenefitReportsRequestToIR	50 MB	10,000 benefits payment reports
Web Service deferred, upload service	IN	MessagesToIR	MessagesRequestToIR	50 MB	10,000 messages One message can contain no more than 5 appendices with a maximum size of 45 MB (base64 converted appendices in total).
Web Service deferred	IN	StatusRequestToIR	StatusRequestToIR	10 kB	1 record query
Web Service deferred	OUT	AckFromIR	AckFromIR	no limitation	no limitation
Web Service deferred	OUT	StatusResponseFromIR	StatusResponseFromIR	no limitation	no limitation

6 VERSIONING

The aim is to implement the versioning of the Incomes Register's technical interface in such a manner that the system retains backwards compatibility.

A change to the technical interface means one or more of the following:

- Change to the service
 - service addition
 - service operation addition
- Change to the data content
 - change to a message structure
 - addition of a message structure.

Changes to the service involve the addition of a new service or service operation to the existing WSDL namespace. Such changes can also be implemented directly for a new WSDL namespace. The URLs for the services are based on the following model: <https://incomes-register-address/service-version/service-name.svc>.

Changes to the data content are implemented, if possible, in such a way that an existing schema is complemented with new data that is non-mandatory for the schema. This keeps the namespace unchanged, allowing a party to submit data in accordance with the old schema.

If the change applies to a schema used in data distribution, data users must update their systems to utilise the new schema, if they require the new data.

7 ERROR FEEDBACK

This section describes the general principles of the technical interface's error messages returned by the Incomes Register as a response to a message sent to the Incomes Register (a message containing a record or a service request sent to the Incomes Register).

7.1 Web Service channel

7.1.1 HTTP errors

Errors detected before the processing of the SOAP framework are returned to the invoker in the HTTP status code ([HTTP/1.1: Status Code Definitions](#)).

The use of the Incomes Register's Web Services requires authentication with a customer certificate. If authentication fails, the Incomes Register returns the HTTP error code 401 (Unauthorized).

7.1.2 SOAP errors

The Incomes Register returns SOAP-level errors in accordance with the SOAP 1.1 Fault structure (https://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383507) and with the HTTP status code 500 (Internal Server Error).

A SOAP Fault can be returned, for example, in situations where the SOAP framework is invalid. A SOAP-level error is returned, for example, when the received message cannot be parsed into an XML document, or the document fails the schema check. The recommendation is that the sending system should perform a schema validation on the message before sending the message to the Incomes Register.

In all identified situations, the SOAP Fault error feedback will include the error code assigned by the Incomes Register to the error, and the description of the error code. The error codes for identified situations and their descriptions will be published as part of the error codes of the Incomes Register's technical interface.

7.1.3 Business errors

In business error situations, the Incomes Register returns the error messages in the error structures of the message returned to the sender. The error structure contains the error code, an English description of the error, and the identifying information of the element containing the error. For a more detailed description of the error structure, see the Section "Error structures of returned message".

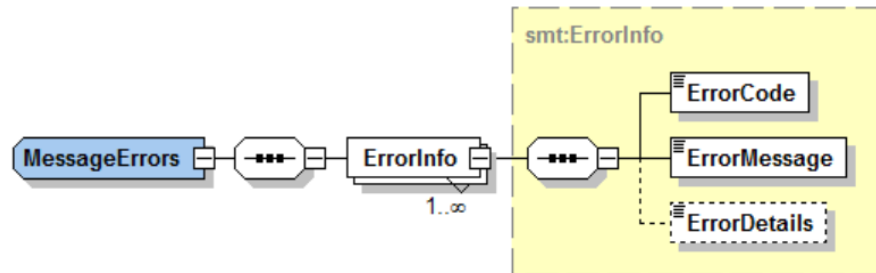
7.2 SFTP channel

Errors occurring at SFTP protocol level in accordance with the standard are not discussed in this document.

The Incomes Register generates processing feedback (StatusResponseFromIR.xsd) for all records received via the SFTP channel, containing the detected technical and business errors. The processing feedback can be retrieved via the SFTP channel after the record processing has finished. Over the SFTP channel, the Incomes Register returns error messages in the error structures of the processing feedback, described in the Section "Error structures of returned message".

7.3 Error structures of returned message

7.3.1 Message-level errors (MessageErrors)



This data group is used to return technical and authorisation errors. The error code (ErrorCode) and error code description (ErrorMessage) are returned for message-level errors. Identifying information on the error (ErrorDetails) will not be returned in connection with message-level errors.

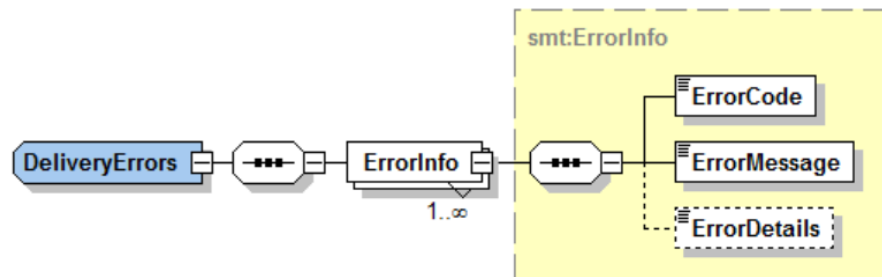
If message-level errors are detected in a record submitted to the Incomes Register, the record is not processed further in the Incomes Register. In addition to message-level errors, the processing feedback cannot therefore contain any other errors (record-level errors, and rejected items and the errors related to them).

Message-level errors include:

- Record is not a valid XML document.
- Schema check failed.
- The electronic signature of the record is invalid.
- Other unexpected error.

In the Web Service channel, message-level errors are also returned as part of the SOAP Fault error message, in which case no separate feedback message will be generated of them.

7.3.2 Record-level errors (DeliveryErrors)

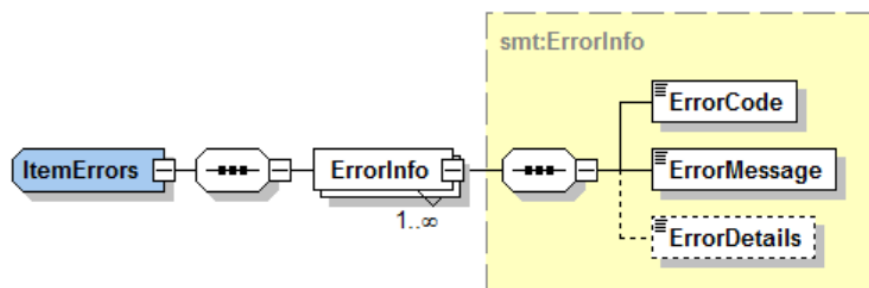


This data group is used to return errors related to the contents of record-level data. The error code (ErrorCode) and error code description (ErrorMessage) are returned for record-level errors. Identifying information on the error (ErrorDetails) will not be returned in connection with message-level errors.

Record-level errors apply to all items in the record. If there are record-level errors in the record, all items of the record are invalid and will not be saved in the Incomes Register. In such a case, the invalid items are not separately delivered in the "Rejected items" data group.

7.3.3 Item-level errors (ItemErrors)

An item means an item submitted within a record; it can be, for example, an individual earnings payment report/benefits payment report or an individual record subscription. Invalid items are returned inside the InvalidItems data group. The data of each item includes the Item-level errors data group, in which information on the errors the item contains is returned.



This data group is used to return errors related to the contents of rejected items. The error code (ErrorCode), error code description (ErrorMessage) and the identifying information on the error (ErrorDetails) are returned for item-level errors.

An XPath referring to the element that caused the error in the request message (in a record submitted to the Incomes Register) is returned in the feedback message element containing the identifying information on the error (ErrorDetails).

The element (an individual data item or a data group) that caused the error is identified in the feedback message in such a manner that, with the help of the identifying information, a party's system is able to unambiguously identify the invalid element in the message submitted to the Incomes Register. For example, the presentation method enables a party's system to indicate the invalid item to the end user, via the party's user interface, on the basis of the error message.

XPath offers a path to each element in the tree structure of the XML message. An example of a path referring to the payer's last name in an employer's separate report:

```
/psrtir:PayerSummaryReportRequestToIR/DeliveryData/Payer/PayerBasic/LastName
```

If an element can be repeated in a message, the path presentation will include an index. An example of a path referring to the first report in an earnings payment record, with the income earner's identifier type specified second:

```
/wrtir:WageReportRequestToIR/DeliveryData/Reports/Report[1]/IncomeEarner/IncomeEarnerIds/Id[2]/Type
```

It is recommended that the same tree structure be used in the user interface of the party's system, in which case the paths can be directly used to refer to data displayed on the screen. If the tree structure is not used in the user interface as such, each UI element should at least include information on its location in the structure of the XML message, so that the required references are easy to create.