



EDIGURUS-Data-Format

Version 1

Scope

This document defines the EDIGURUS-Data-Format in the Version 1. It is intended as a guideline for the exchange of business documents between companies within a non-hierarchical business network by electronic communication. Since each company has a multitude of business relationships in its network it is important to keep communication to many partners easy to handle. Especially for small and medium sized enterprises a multilateral approach will simplify electronic business communication as today the efforts to handle bilateral communication channels are too heavy.

Terms and Definitions

Within this document the following terms and definitions apply:

- "Workflow of documents"
Flow of all documents relating to each other within a business process (All documents within a workflow of documents represent a business process in reality)
- "Customer"
Representation of the participant that receives goods or services within a business process
- "Supplier"
Representation of the participant that provides goods or services within a business process

List of abbreviations

- EDI Electronic Data Interchange
- ERP Enterprise Resource Planning



- JSON Java Script Object Notation
- JSONPath JSON Path Language
- MIME Multipurpose Internet Mail Extensions
- PDF Portable Document Format
- URL Unified Resource Location
- W3C World wide Web Consortium
- XML Extended Markup Language

Order management process and messages

The EDIGURUS-Data-Format is used for electronic data interchange in the communication between customers and suppliers within non-hierarchical business networks. It covers most of the communication when customers order goods or services (e.g. transportation) at suppliers.

Each business document exchanged between companies consists of a PDF which is readable for humans. Embedded in this PDF is a file in the EDIGURUS-Data-Format, which contains the data of the PDF as JSON and can thus be processed by applications. The name of this file must end with "_edigurus.json" (or "_edigurusreceipt.json" if the receipt for the message is contained too).

If placed directly by an ERP system, the data of the order has to be translated from the ERP system to a PDF including the file in the EDIGURUS-Data-Format. Afterwards it is transmitted to the providers of the ordered goods or services via e-mail. There the file in the EDIGURUS-Data-Format is translated again into a format that can be understood by the receiver's ERP system. Afterwards it will be processed directly in the ERP system of the receiver.

Workflow of documents

A business process between a customer and a supplier is accompanied by many business documents which are exchanged parallel to production and delivery of goods or services.

With the first type of document sent in a workflow of documents the roles of the participants within this workflow are defined. All subsequent types of documents in this particular workflow of documents may be issued by the participants according to their role:

Types of documents issued by customers	Types of documents issued by suppliers
--	--



Forecast	Master data
Request for quotation	Quotation
Order	Order confirmation
Change order	
Transport order	Transport confirmation
Transport change	
Inventory (for Consignment)	Request for Inventory (for Consignment)
Withdrawal (for Consignment)	
Returns (for Consignment)	
Goods receipt	Dispatch Notification
Complaint	Transport Status
Returns	Proof of Delivery
Credit Memo	Invoice
Receipt of Customer	Receipt of Supplier

Within a workflow of documents any type of document may follow any other. Nevertheless in business exists a typical workflow of documents starting with the types of documents are listed below:

Types of documents issued by customers when starting a new workflow	Types of documents issued by suppliers when starting a new workflow
Forecast	Master data
Request for quotation	
Order	
Transport order	Request for Inventory (for Consignment)

The link of one document to another within the workflow of documents is established on line item level within the body of the documents by the "Parent"-element. Each following document will contain the identifier of the previous message (contained in the data field under JSONPath "Message-



Key") and the document type of the message (contained in the data field under the JSONPath "Type"). By the type of the document it is clear which company was the sender of this message, because the role within a single workflow of documents is fixed after the first message. In the case that more than one document of the sender has the same identification of the message the following document is assumed to be related to the last known document which was sent with this identification.

Receipt management

The receiver shall send exactly one receipt for each document a sender issues, which is not a receipt itself. The receipt shall contain at least the identification of the sender, the identification of the receiver and the identification of the document this receipt refers to. Receipts may be "positive", which means that no errors are contained, or "negative", if at least one error is contained. An error is indicated, if the data field "Code" (JSONPath: "Receipt.Log[].Code" contains a value which is equal or greater than 200. The list of error codes is contained in Annex H. Further hints can be found in the other fields within the "Log" data element.

As long as no receipt is transmitted to the sender of a document, it is assumed juristically that the document has not been sent to the receiver and the sender shall take appropriate actions within a reasonable time frame, if no receipt is received. If a receipt for a document is "negative", it is clear to the sender that the document could not be transmitted or technically processed at the receiver's side. In this case the document is juristically deemed as not been sent and the sender shall take appropriate actions. The "Log" element (JSONPath: "Receipt.Log[]") may give further hints to solve the problem. If the sender gets a "positive" receipt, the message could be successfully processed at the receivers; only in this case the document has been juristically transmitted to the receiver. However a "positive" receipt does not mean that the receiver accepts the content of the document, it indicates only the successful transmission.

Before a message is transmitted, it shall be checked for completeness and formal correctness. Messages that do not pass this test shall not be handed over to the receiver and raise a "negative" receipt containing the occurred error.

Data Format

This chapter contains the complete specification of all data elements and data fields for the EDIGURUS-Data-Format. This chapter shows the maximum structure of the which is the basis for all document types, but not all the document types require the whole range of elements (e.g. a dispatch notification may not contain any price information).

All documents consist of hierarchically structured data elements which on the lowest level contain data fields. The content of the data fields along with the structure of the document defines the EDIGURUS-Data-Format.



Data types

All the data of the EDIGURUS-Data-Format is encoded in JSON. Each data field has a certain data type, which restricts its contents. The data types are:

- String (Length)

Fields of this variable type may contain any character or digit. The maximum length of the content is indicated in brackets.

The following characters need a special treatment:

"	:	\"	Quotes
<	:	\u003c	Less than
>	:	\u003e	Greater than
&	:	\u0026	Ampersand
CR/LF	:	\r\n	Carriage return

Characters with an ASCII code of 0 (zero) are not allowed

- Boolean

A field of this type has two possible values: "true" and "false".

If the field is missing or the content is empty "false" is assumed as value for this field.

- Integer

Values of this type have a range from -2147483648 to 2147483647.

Negative values have a leading minus ("-").

If the field is missing or the content is empty "0" is assumed as value for this field.

- Integer ≥ 0

Values of this type have a range from 0 to 2147483647.

If the field is missing or the content is empty "0" is assumed as value for this field.

- Float

Values of this type are floating point numbers. The decimal separator is the dot ("."). Negative values have a leading minus ("-").

If the field is missing or the content is empty "0" is assumed as value for this field.



- Float ≥ 0

Values of this type are floating point numbers with a value greater or equal to zero. The decimal separator is the dot (".").

If the field is missing or the content is empty "0" is assumed as value for this field.

- Timestamp

A time stamp contains a date and always a time too; it has the following form:

yyyy-MM-dd'T'HH:mm:ssZ (e.g. "2012-12-24T18:30:15+01:00")

where:

yyyy	:	is the year ("0000" to "9999")
MM	:	is the month ("01" (January) to "12" (December))
dd	:	is the day within the month ("01" to "31")
HH	:	is the hour of the day ("00" to "23")
mm	:	are the minutes within the hour ("00" to "59")
ss	:	are the seconds within the minute ("00" to "59")
Z	:	is the time zone with the following values: "+00:00" for "Coordinated Universal Time" (UTC) "+01:00" for "Central European Time" (CET) other time zones are coded in the same manner.

Some data fields are restricted in their contents to values from enumerations, if not noted differently. In the description of these data fields the possible values are listed.

Most of the fields within a single message have no restriction regarding the size of their contents. Nevertheless the overall size of the whole message (including JSON-Tags) may not exceed 2 Megabytes. The reason for this restriction lays in the way JSON is processed on current computer systems: Large JSON documents may slow down the processing very much. The documents will only exceed this size, if many line items (JSON-Path: Body.Item[]) are contained. To send such large documents the line items shall be split into some chunks, and then each chunk shall be embedded in one independent "Body/Item" element, so the chunks can be processed one by one.



Structure & Example

Within the EDIGURUS-Data-Format the order of elements and data fields is irrelevant as long as the hierarchical structure (in JSON established by "{" and}") is consistent.

Required data fields are marked with " required " in the column of the data type.

References of one data field to another are defined using the JSONPath (e.g. the JSONPath of "Body.Item[].ItemKey" refers to the data field containing the identifier of a line item).

{		The document starts here.
"Version": "1",	String (∞) required	This field contains the version of the EDIGURUS-Data-Format used for the document. Values are restricted to: 1 : EDIGURUS-Data-Format Version 1
"Type": "ORDER",	String (36) required	This field contains the type of the document. Values are restricted to (C indicates document sent by the customer, S indicates documents sent by the supplier or the forwarder): MASTERDATACUSTOMER : C Master data for purchased articles MASTERDATASUPPLIER : S Master data for sold articles REQUESTFORQUOTATION : C Request for quotation QUOTATION : S Quotation FORECAST : C Forecast about future purchases ORDER : C Order ORDERCONFIRMATION : S Confirmation of an order CHANGEORDER : C Change of an order TRANSPORTORDER : C Transport order advising the forwarder to transport



		goods
TRANSPORTCONFIRMATION	: S	Confirmation of a transport order
TRANSPORTCHANGE	: C	Change of a transport order
MOVEMENTNOTIFICATION	: C	Notification of the incoming or outgoing goods for a recipient, who is not in charge of the transport
MOVEMENTCONFIRMATION	: S	Confirmation of incoming or outgoing goods from the sender, who was not in charge of the transport
STOCKREQUEST	: C	Request of the customer to send the quantities on stock for the listed items
STOCKINVENTORY	: S	Inventory of the stock
CONSIGNMENTREQUEST	: S	Request to send an inventory of a consignment stock of the listed articles
CONSIGNMENTINVENTORY	: C	Inventory of a consignment stock
CONSIGNMENTDELIVERY	: C	Confirmation of the delivery of goods from the supplier to a



		consignment storage
CONSIGNMENTWITHDRAWAL	: C	Withdrawal by the customer from a consignment stock
CONSIGNMENTRETURNS	: C	Return of goods by the customer to a consignment stock
CONSIGNMENTREDELIVERY	: C	Redelivery of goods from a consignment storage to the supplier
DISPATCHNOTIFICATION	: S	Notification of the dispatch of goods
TRANSPORTSTATUS	: S	Status of a transport
PROOFOFDELIVERY	: S	Confirmation of the arrival of the goods sent by the forwarder
GOODSRECEIPT	: C	Confirmation of the arrival of the goods sent by the customer
COMPLAINT	: C	Complaint about delivered goods
RETURNS	: C	Announcement of the return of goods
INVOICE	: S	Invoice
CREDITMEMO	: C	Credit memo
RECEIPTCUSTOMER	: C	Receipt of the customer for a document sent by



the supplier

RECEIPTSUPPLIER

: S

Receipt of the supplier for a document sent by the customer

"CustomerKey": "customer.edigurus.org",

String (36) ^{required} This field contains the identifier of the customer within the EDI communication. It is used for routing the document over communication channels. Depending on the sender of the type of this document, the content of this field contains the sender or the receiver of the document.

"SupplierKey": "supplier.edigurus.org",

String (36) ^{required} This field contains the identifier of the supplier within the EDI communication. It is used for routing the document over communication channels. Depending on the sender of the type of this document, the content of this field contains the receiver or the sender of the document.

"MessageKey": "M990003",

String (36) ^{required} This field contains the identification of the message in the verbal communication between customer and supplier. This identification is chosen by the sender of the document and may occur more than once, if it is inevitable (nevertheless best practice requires that the identification is unique).

"TransmissionKey": "M990003-201612",

String (72) This field contains the identification of the message in machine-to-machine-communication. This identification is chosen by the sender of the document and may occur only once in all the documents the sender issues.

"Sent": "2016-12-24T12:00:00+01:00",

Timestamp This field contains the time and date when the document was sent.

"Urgent": "false",

Boolean This field indicates with the content of "true" that the processing of this message at the receiver should have a higher priority than the processing of other messages from the sender.

"Test": "false",

Boolean If the message is for testing only, the value of this field should be "true".



"Language": "EN",	String (2)	This field contains the uppercase 2-digit code of the language that is used all over the message. Values are restricted to the values defined in ISO 639-1:2002: DE : German EN : English ES : Spanish FR : French ... (see ISO 639-1:2002 for all possible values)
"Subject": "An important message",	String (∞)	This field contains the Subject of the message.
"Body": {	sometimes required	This element contains the "Body" of the document. It must be included, if the message contains a business document. It is not included in receipts, which means that the type of the document (JSONPath: "Type") may not be "RECEIPTCUSTOMER" or "RECEIPTSUPPLIER".
"Deadline": "2016-12-27T23:59:59+01:00",	Timestamp	This field contains the date until when an answer to this document is requested from the business partner e.g. in a request for quotation the date, when the quotation must have been sent; or in an order the date when the order confirmation must have been sent.
"Preface": "A remark at the top",	String (∞)	This field contains the remark about the whole document printed at the top of the document.
"Note": "A remark at the bottom",	String (∞)	This field contains the remark about the whole document printed at the bottom of the document.
"Customer": {		This element contains the company data (address & contact data) of the customer. This element is structured exactly like the other elements that contain company data too (e.g. the supplier (JSONPath: "Body.Supplier") etc.). The element itself is optional, if the field with the identification of the



		customer (JSONPath: "CustomerKey") is sufficient. Nevertheless would it be better to fill this element to achieve a better readability for humans.
"CompanyKey": "1234",	String (36)	This field contains the identifier of the company which is used outside the EDI communication. For example it might contain the identifier of the company within the ERP system of the sender. This way it is possible to transmit the exact identification of the subsidiary of the company.
"Name": "Demo-Company",	String (∞) <small>required</small>	This field contains the name of the company.
"Department": "Demo-Department",	String (∞)	This field contains the department of the company.
"Street": "Demo-Street 123",	String (∞)	This field contains the street and house number or p.o. box of the company.
"City": "Demo-City",	String (∞)	This field contains the city of the company.
"ZipCode": "12345",	String (∞)	This field contains the zip code of the company.
"Region": "Bavaria",	String (∞)	This field contains the identification of the state or region of the company.
"Country": "DE",	String (2)	This field contains the uppercase 2-digit code of the country of the company. Values are restricted to the values defined in ISO 3166:2001: AT: Austria, CH: Swiss, DE: Germany, ES: Spain, FR: France, GB: Great Britain, US: USA, ... (see ISO 3166:2001 for all possible values).
"GeoLat": "48.39094",	Float	This field contains the latitude of the GPS location of the company in degrees.
"GeoLon": "9.9569813",	Float	This field contains the longitude of the GPS location of the company in degrees.
"GeoAlt": "480",	Float	This field contains the altitude of the GPS location of the company in meters.
"EmployeeKey": "1234-D",	String (36)	This field contains the identification of the contact person within the company.



"FirstName": "Demo",	String (∞)	This field contains the first name of the contact.
"Surname": "Demonstratos",	String (∞)	This field contains the surname of the contact.
"Email": "somemail@edigurus.org",	String (∞)	This field contains the e-mail address of the contact or, if none is available, the e-mail address of the company.
"Phone": "+49 123 456789",	String (∞)	This field contains the telephone number of the contact or, if none is available, the number of the company.
"TaxPayerKey": "DE12345678",	String (36)	This field contains the identification of the company when paying taxes.
"Legal": "Demo legal statement"	String (∞)	In this field all the necessary legal statements of the company are included (e.g. the number of the company in the commercial register or the name of the CEO).
},		The company data of the customer ends here.
"CustomerBilling": {		This element contains the company data (address & contact data) of the party that will be billed and is responsible for the taxes from it. This element is optional. If it is not existing, the customer (JSONPath: "Body.Customer") is assumed to take this role. Best practice is to skip this element, if the content is exactly like the content of the customers company data.
 (see above)		Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.
},		The company data of the billed party ends here.
"EndCustomer": {		This element contains the company data (address & contact data) of the party finally receiving the product or the benefit for the services described in this document. This element is optional. If it is not existing, the customer (JSONPath: "Body.Customer") is assumed to take this role. Best practice is to skip this element, if the content is exactly like the content of the customers company data.



(see above)

},

"Supplier":{

(see above)

},

"SupplierBilling":{

(see above)

},

"Origin":{

Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.

The company data of the end customer ends here.

This element contains the company data (address & contact data) of the supplier. The element itself is optional, if the field with the identification of the supplier (JSONPath: "SupplierKey") is sufficient. Nevertheless would it be better to fill this element to achieve a better readability for humans.

Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.

The company data of the supplier ends here.

This element contains the company data (address & contact data) of the party that will issue the invoice and is responsible for the taxes from it.

This element is optional. If it is not existing, the supplier (JSONPath: "Body.Supplier") is assumed to take this role. Best practice is to skip this element, if the content is exactly like the content of the suppliers company data.

Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.

The company data of the party issuing the invoice ends here.

This element contains the company data (address & contact data) of the location from which the goods in this document will be dispatched. This element is optional. If it is not existing, the supplier (JSONPath: "Body.-Supplier") is assumed to take this role. Best practice is to skip this ele-



(see above)

},

"Location":{

sometimes required

(see above)

},

"Destination":{

(see above)

},

ment, if the content is exactly like the content of the suppliers company data.

Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.

The company data of the location from which the goods will be dispatched ends here.

This element contains the company data (address & contact data) of the location of the goods when sending this document. This element is required only when sending a document of the type "TRANSPORTSTATUS" (see JSONPath: "Type") in all other cases this element is optional.

Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.

The company data of the current location of the goods ends here.

This element contains the company data (address & contact data) of the destination to which the goods in this document will be delivered. This element is optional. If it is not existing, the customer (JSONPath: "Body.-Customer") is assumed to take this role. Best practice is to skip this element, if the content is exactly like the content of the customers company data.

Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.

The company data of the destination of the goods ends here.



"Forwarder":{

(see above)

},

"Item":[

{

"ItemKey":"10",

"Description":"A beautiful Zeppelin",

"ArticleCustomer":"Z21-D40",

"ArticleSupplier":"LZ21-R20",

"Unit":"PCE",

required

required

Integer ≥ 0 required

String (∞)

String (∞)

String (∞)

String (∞) required

This element contains the company data (address & contact data) of the forwarder. This element may only be filled, if the forwarder is not the sender or the receiver of this message - in this case the forwarder would be the supplier (JSONPath: "Body.Supplier").

Because this element is structured exactly like the company data of the customer (JSONPath: "Body.Customer") the structure is not repeated here.

The company data of the forwarder ends here.

This element contains the list of the line items.

Here starts the first line item. The first line items is required in every document containing a body. All further line items are optional.

This field contains the identification of the line item; this number must be unique within this message.

This field contains the description of the line item.

This field contains the article number at the customers company.

This field contains the article number at the suppliers company.

This field contains the unit in which the article is ordered. Values are restricted to the following values:

- CMT : cm Centimeters
- DAY : Days
- GRM : g Gramm
- HUR : h Hours
- KGM : kg Kilograms
- KMT : km Kilometers
- KWH : kWh Kilo Watt Hours
- LTR : l Liters



- MIN : min Minutes
- MMT : mm Millimeters
- MTK : m² Square meters
- MTQ : m³ Cubic meters
- MTR : m Meters
- PCE : pcs. Pieces
- SET : A set of articles indicates that the number and type of the articles is described in the description of the unit (JSONPath: "Body.Item[].UnitName"), the description of the article (JSONPath: "Body.Item[].Description") or it is clear between customer and supplier
- TNE : t Tons

"UnitName": "",	String (∞)	This field contains the verbal description of the unit. It is only necessary to fill this field, if the verbal description differs from the unit (JSONPath: "Body.Item[].Unit"). E.g. if "pairs" are ordered. This is especially helpful when using the unit "SET".
"Quantity": "1.0",	Float ≥ 0 <small>required</small>	This field contains the quantity of the article in the unit defined above. The quantity of 0 ("zero") indicates that this line item is either canceled by the customer or rejected by the supplier (depending on the type of the document).
"CumulatedQuantity": "0",	Float ≥ 0	This field contains the cumulated quantity of the article in the unit defined above. The quantity of 0 ("zero") indicates that a cumulated quantity is not defined for this item.
"CumulatedReset": "",	Timestamp	This field contains the date when the cumulation was reset.



"CumulatedResetQuantity": "0",	Float ≥ 0	This field contains the cumulated quantity of the article in the unit defined above before the cumulation was reset.
"Delivered": "false",	Boolean	If "true" this field indicates that the quantity above is already delivered. Thus this line item has to be ignored in an order or change order. It is just an information for the supplier. Often this flag is used together with "Cumulative" in a request for delivery (e.g. in the automotive industry).
"Announced": "false",	Boolean	If "true" this field indicates that this line item is only announced but not ordered yet. In this case the line item is only intended to tell the supplier about planned purchases.
"Arrival": "2016-12-24T23:59:59+01:00",	Timestamp	This field contains the latest date when the goods will be arriving at the destination or the services will be delivered. Dates in the future are requested dates of the customer or - if the supplier is the sender of the document - confirmed dates of the supplier. Dates in the past indicate the time, when the arrival took place.
"ArrivalEarliest": "",	Timestamp	This field contains the earliest date when the goods will be arriving at the destination or the services will be delivered. Dates in the future are requested dates of the customer or - if the supplier is the sender of the document - confirmed dates of the supplier. Dates in the past are skipped.
"Dispatch": "2016-12-23T23:59:59+01:00",	Timestamp	This field contains the latest date when the goods will be dispatched at the origin. Dates in the future are requested dates of the customer or - if the supplier is the sender of the document - confirmed dates of the supplier. Dates in the past indicate the time, when the dispatch took place.
"DispatchEarliest": "",	Timestamp	This field contains the earliest date when the goods are dispatched at the origin. Dates in the future are requested dates of the customer or - if



		the supplier is the sender of the document - confirmed dates of the supplier. Dates in the past are skipped.
"NetWeight": "8.5",	Float ≥ 0	This field contains the net weight of the whole item in kilograms (kg).
"Drawing": "DW-Z21",	String (∞)	This field contains the identification of a drawing this line item relates to.
"DrawingVersion": "2016-09-01",	String (∞)	This field contains the version of this drawing.
"PrintForCustomer": "Plant A/23",	String (∞)	The content of this field has to be printed on every paper relating to this line item that is sent to the customer (e.g. it contains the location of the construction site or the (human readable) content of a label). The customer is responsible to fill this field in the communication before a paper is sent to him.
"PrintForSupplier": " Facility-3434",	String (∞)	The content of this field has to be printed on every paper relating to this line item that is sent to the supplier (e.g. it contains the (human readable) label on the items when they are returned). The supplier is responsible to fill this field in the communication before a paper is sent to him.
"BarcodeForCustomer": "A23",	String (∞)	The content of this field has to be printed as barcode on every label applied on goods relating to this line item that are sent to the customer. The customer is responsible to fill this field in the communication before a label is sent to him.
"BarcodeForSupplier": "3434F",	String (∞)	The content of this field has to be printed as barcode on every label applied on goods relating to this line item that are returned to the supplier. The supplier is responsible to fill this field in the communication before a label is sent to him.
"StorageOrigin": "AB12",	String (∞)	This field contains the place within the location of the origin of the goods in this item.
"StorageDestination": "XY89",	String (∞)	This field contains the place within the location of the destination of the goods in this item.



"CommodityGroup": "11/22/33/44",	String (∞)	This field contains the commodity group of the article. Sub-groups may be separated by a slash "/".
"CommodityCode": "12345",	String (∞)	This field contains the Commodity code that the government of the supplier has assigned to the goods.
"CountryOfOrigin": "DE",	String(2)	This field contains the uppercase 2-digit code of the country of the origin of the goods. Values are restricted to the values defined in ISO 3166:2001 (see JSONPath: "Body.Customer.Country").
"CustomsTariff": "54321",	String (∞)	This field contains the customs tariff of the goods in this line item.
"PreferenceIndicator": "65432",	String (∞)	This field contains the preference indicator of the goods for the customs administration.
"ExportCertificate": "76543",	String (∞)	This field contains the export certificate of the goods for the customs administration.
"Note": "An important remark",	String (∞)	The field contains a remark about this line item.
"Price": {	sometimes required	This element contains information about the price of this item. The "Price" element is required for the following document types (JSONPath: "Type"): "ORDERCONFIRMATION", "INVOICE" and "CREDITMEMO".
"Unit": "KGM",	String (∞)	This field contains the unit the supplier uses to calculate the price of the article. Best practice is to leave this field empty, if the unit does not differ from the unit of the line item (JSONPath: "Body.Item[.].Unit"). Values are restricted to the values listed in the unit of the line item.
"UnitName": "",	String (∞)	This field contains the verbal description of the unit the supplier uses to calculate the price of the article. It is only necessary to fill this field, if the verbal description differs from the unit (JSONPath: "Body.Item[.].Price.Unit"). E.g. if "pairs" are ordered. This is especially helpful when using the unit "SET".



"Quantity": "8.5",	Float ≥ 0	This field contains the quantity of the article in the unit the supplier uses to calculate the price of the article. Best practice is to leave this field empty, if the unit to calculate the price does not differ from the unit of the line item (JSONPath: "Body.Item[.].Unit").
"BasePrice": "10.0",	Float <small>required</small>	This field contains the base price in the currency of the document (JSONPath: "Body.Total.Currency") for the quantity of articles defined in the base quantity of this element (JSONPath: "Body.Item[.].Price.Base-Quantity"). The unit on which the price is calculated is the suppliers unit (JSONPath: "Body.Item[.].Price.Quantity"), if it is not empty, otherwise it is the unit of the line item (JSONPath: "Body.Item[.].Quantity").
"BaseQuantity": "1.0",	Float ≥ 0	This field contains the base quantity for the base price; If this field is omitted or the content is 0 (zero) then then content of 1 is assumed as base quantity. The unit of this quantity is the unit on which the price is calculated (see above).
"Value": "76.5",	Float <small>required</small>	This field contains the total amount of this line item including all discounts and surcharges without any taxes in the currency of the message (JSONPath: "Body.Total.Currency"); it is calculated by adding the calculated amount of every discount and surcharge (JSONPath: "Body.Item[.].Price.Addition[.].Value") to the base value of this line item, which is calculated by multiplying the base price with the quantity and divided by the base quantity (see above).
"TaxKey": "VAT",	String (36)	This field contains the identification of the tax that has to be applied on the base value (JSONPath: "Body.Item[.].Price.BasePrice") of this line item. This identification must be found in the list of taxes (JSONPath: "Body.Total.Tax[.].TaxKey"). If no tax identification is filled in, no tax is applied.
"Addition": [This element contains the list of discounts and surcharges.



{			Here starts the first discount or surcharge.
"AdditionKey":"DISCOUNT",	String (36)		This field contains the identification of this discount or surcharge. This identification may be used to find the type of this discount or surcharge.
"Description":"For nice people",	String (∞)		This field contains the description of this discount or surcharge.
"Percent":"-10.0",	Float		This field contains the percentage that is used to calculate the value of this discount or surcharge; if 0.0 this surcharge or discount is an absolute value; if it is less than 0.0 it is a discount (which reduces the total amount of this line item); if it is more than 0.0 it is a surcharge (which is added to the total amount of this line item). If the value calculated with this percentage does not match the calculated amount (below), the calculated amount is the more important value.
"Value":"-8.5",	Float <small>required</small>		This field contains the calculated amount of this discount or surcharge for the whole line item, if no percentage is included or it is 0.0 it contains the absolute value of this discount or surcharge for the whole line item; If it is less than 0.0 it is a discount (which reduces the total amount of this line item); If it is more than 0.0 it is a surcharge (which is added to the total amount of this line item).
"TaxKey":"VAT"	String (36)		This field contains the identification of the tax that has to be applied on the calculated amount of this discount or surcharge. This identification must be found in the list of taxes (JSONPath: "Body.Total.Tax[].TaxKey"). If no tax identification is filled in, no tax is applied.
}			The first discount or surcharge ends here.
{			Here starts the second discount or surcharge. Every line item can contain an unlimited number of discounts and surcharges which are all included in this list.



(see above)

```
    }  
  
  ]  
},  
"Batch":[  
  {  
    "BatchKey":"LZ21-201612",  
  
    "Description":"Zeppelin 2016/12",  
    "Quantity":"1.0",  
  
    "Production": "",  
    "Expiration": "",  
  
    "PackageKey": ""  
  },  
  {
```

String (36)

String (∞)

Float ≥ 0 required

Timestamp

Timestamp

String (36)

Because this element is structured exactly like the first discount or surcharge (JSONPath: "Body.Item[].Price.Addition[]") the structure is not repeated here.

The second discount or surcharge ends here. Add a subsequent discount or surcharge, if more discounts or surcharges should be included in this line item.

The list of discounts and surcharges ends here.

The "price" element ends here.

This element contains the list of batches of the line item.

Here starts the first batch of this line item.

This field contains the identification of the batch, if the quantity (JSON-Path: "Body.Item[].Batch[].Quantity") is larger than 1 ("one") or the serial number, if just a single article is contained in this batch.

This field contains the description of this batch.

This field contains the quantity of items in this batch; The unit for this quantity is found in the line item (JSONPath: "Body.Item[].Unit").

This field contains the date when the goods were produced.

This field contains the date when the goods in this batch expire or nothing, if there is no such date.

This field contains the identification of the package that contains this batch; it references to a package with the same identification (JSONPath: "Body.Package[].PackageKey").

The first batch ends here.

Here starts the second batch. Every line item can contain an unlimited number of batches which are all included in this list.



```
(see above)
}
],
"Parent":[
{
  "Type":"QUOTATION",
```

Because this element is structured exactly like the first batch (JSONPath: "Body.Item[.].Batch[.]") the structure is not repeated here. The second batch ends here. Add a subsequent batch, if more batches should be included in this line item.

The list of batches ends here.

This element contains the list of all line items in previous documents to which this line item relates to.

Here starts the first line item in a previous document to which this line item relates to.

String (36) ^{required} This field contains the document type of the previous message. It relates to the field with the JSONPath "Type" in the previous document or to additional relations listed below. Values are restricted to the list of document types found in the type field of this document (see JSONPath: "Type"). Additionally it is possible to fill in some more types which relate to the communication outside the electronic exchange of documents:

COMMISSION	:	A commission of the customer
CONTRACT	:	A contract between customer and supplier
ENDCUSTOMERORDER	:	The order of the end customer of the products (e.g. in a transport order the order number of the destination of the goods)
ORIGINALORDER	:	Points to the original order and item. This is used only in the order itself and if the current



		item has been split (otherwise "ORDER" will do)
	MESSAGECUSTOMER	: An unspecified message of the customer
	MESSAGESUPPLIER	: An unspecified message of the supplier
"MessageKey": "Q230045",	String (36) ^{required}	This field contains the identification of the previous document in verbal communication that the sender of it gave to that document. It relates to the field under the JSONPath "MessageKey" in the previous document.
"TransmissionKey": "Q230045-5",	String (72)	This field contains the identification of the previous document in machine-to-machine communication that the previous sender gave to the previous document. It relates to the field under the JSONPath "TransmissionKey" in the previous document.
"ItemKey": "5"	Integer ≥ 0	This field contains the identification of the line item (JSONPath: "Body.Item[].ItemKey") in the previous document to which the current line item relates to.
}		The first line item in a previous document ends here.
{		Here starts the second line item in a previous document to which this line item relates to. Every line item can contain an unlimited number of relations which are all included in this list.
(see above)		Because this element is structured exactly like the first line item in a previous document (JSONPath: "Body.Item[].Parent[]") the structure is not repeated here.
}		The second line item in a previous document ends here. Add a subsequent line item in a previous document, if more line items in a previous documents should be included in this line item.



"Feature":[This element contains the list of features of the line item. Features contain additional values in the communication between customer and supplier which are not covered by the EDIGURUS-Data-Format and which should be transmitted between both parties in a way that can be interpreted by IT systems.
{			Here starts the first feature of this line item.
"FeatureKey":"COLOR",	String (36) <small>required</small>		This field contains the identification of the feature in a form that can be identified by another IT system.
"Value":"26",	String (∞)		This field contains the content of this feature in a form that can be read by another IT system.
"Description":"The color is grey"	String (∞)		This field contains the verbal description of this feature. This is provided to make a feature understandable for humans even if the other fields of this element cannot be processed automatically.
},			The first feature ends here.
{			Here starts the second feature. Every line item can contain an unlimited number of features which are all included in this list.
(see above)			Because this element is structured exactly like the first feature (JSON-Path: "Body.Item[].Feature[]") the structure is not repeated here.
}			The second feature ends here. Add a subsequent feature, if more features should be included in this line item.
]			The list of features ends here.
},			The first line item ends here.
{			Here starts the second line item. The document can contain an unlimited number of line items which are all included in this list of line items.



(see above)		
}		Because this element is structured exactly like the first line item (JSON-Path: "Body.Item[]") the structure is not repeated here.
],		The second line items ends here. Add a subsequent line item, if more line items should be included in this document.
"Total":{	sometimes required	The list of line items ends here.
		This element contains the total of this document. The "Total" element is required for the following document types (JSONPath: "Type"): "ORDER-CONFIRMATION", "INVOICE" and "CREDITMEMO".
"Currency":"EUR",	String (3) required	This field contains the currency that is used for the whole message. Values are restricted to the values defined in ISO 4217:2008: CHF : Swiss Franks EUR : € European Euros GBP : £ Great Britain Pounds USD : \$ United States Dollars ... (see ISO 4217:2008 for all possible values)
"Value":"76.5",	Float	This field contains the value of the whole document without taxes.
"TaxValue":"14.54",	Float	This field contains the value of all the taxes of the whole document.
"Tax":[This element contains the list of taxes in this document.
{		Here starts the first tax.
"TaxKey":"VAT",	String (36) required	This field contains the identification of the tax.
"Description":"Value added tax",	String (∞)	This field contains the description of this tax.
"Percent":"19.0",	Float	The percentage of a value that has to be added for this tax.
"Value":"14.54"	Float	This field contains the value of this tax across the whole document.
}		The first tax ends here.



```
{  
  (see above)  
}  
  
],  
"Condition": [  
  {  
    "ConditionKey": "C30",           String (36)  
    "Description": "30 days, 3% off", String (∞)  
    "Days": "30",                   Float  
  
    "Percent": "-3.0"               Float  
  }  
}  
  
  (see above)
```

Here starts the second tax. The document can contain an unlimited number of taxes which are all included in this list.

Because this element is structured exactly like the first tax (JSONPath: "Body.Total.Tax[]") the structure is not repeated here.

The second tax ends here. Add a subsequent tax, if more taxes should be included in this document.

The list of taxes ends here.

This element contains the list of payment conditions in this document.

Here starts the first payment condition.

This field contains the identification of the condition.

This field contains the description of this condition.

This field contains the number of days (incl.) until which this payment condition can be applied. The days are counted from the time when this message was sent (JSONPath: "Sent").

This field contains the percentage of the total amount (JSONPath: "Body.Total.Value") for this payment condition. If it is less than 0.0, it is a discount (which reduces the amount that must be paid by the customer). If it is greater than 0.0 it is a surcharge (which is added to the amount that must be paid by the customer).

The first payment condition ends here.

Here starts the second payment condition. The document can contain an unlimited number of payment conditions which are all included in this list.

Because this element is structured exactly like the first payment condition (JSONPath: "Body.Total.Condition[]") the structure is not repeated here.



<pre> }</pre>		The second payment condition ends here. Add a subsequent payment condition, if more payment conditions should be included in this document.
<pre>]</pre>		The list of payment conditions ends here.
<pre>},</pre>		The total ends here
<pre>"Transport":{</pre>		This element contains information about the transport of the goods.
<pre> "TransportKey":"RAIL",</pre>	String (36)	This field contains the identification of the type of the transport.
<pre> "Description":"Transported by rail",</pre>	String (∞)	This field contains the description of the transport.
<pre> "FreightTerms":"CFR/UIm"</pre>	String (∞)	This field contains the freight terms of this transport. For example the Incoterms, if they are used.
<pre>},</pre>		The "Transport" element ends here.
<pre>"Package":[</pre>		This element contains the list of the packages.
<pre>{</pre>		Here starts the first package.
<pre> "PackageKey":"P543",</pre>	String (36) ^{required}	This field contains the (at least within this document unique) identification of the package (e.g. the number of the shipment at the forwarder). Best practice requires a globally unique identification.
<pre> "ParentPackageKey":"","</pre>	String (36)	If this package is contained in another package, the identification of the so called parent package is contained in this field. If the field is empty, this package is not contained in another package or the packing hierarchy is omitted.
<pre> "Description":"Zeppelin package",</pre>	String (∞)	This field contains the verbal description of this package.
<pre> "GrossWeight":"8.8",</pre>	Float ≥ 0	This field contains the gross weight of this package in kilograms (kg).
<pre> "NetWeight":"8.5",</pre>	Float ≥ 0	This field contains the net weight of this Package in kilograms (kg).
<pre> "Length":"100.0",</pre>	Float ≥ 0	This field contains the length of this package in meters (m).



<pre>"Width": "40.0", "Height": "42.0" }, { (see above) }], "Attachment": [(see above)], "Feature": [(see above)] }, "Receipt": {</pre>	<p>Float ≥ 0</p> <p>Float ≥ 0</p> <p>sometimes required</p>	<p>This field contains the width of this package in meters (m).</p> <p>This field contains the height of this package in meters (m).</p> <p>The first package ends here.</p> <p>Here starts the second package. A document can contain an unlimited number of packages which are all included in this list of packages.</p> <p>Because this element is structured exactly like the first package (JSON-Path: "Body.Package[]") the structure is not repeated here.</p> <p>The second package ends here. Add a subsequent package, if more packages should be included in this document.</p> <p>The list of packages ends here.</p> <p>Contains the list of attachments for the whole document.</p> <p>Because this list is structured exactly like the attachments on line item level (JSONPath: "Body.Item[].Attachment[]") the structure is not repeated here.</p> <p>The list of attachments for the whole document ends here.</p> <p>Contains the list of features for the whole document.</p> <p>Because this list is structured exactly like the features on line item level (JSONPath: "Body.Item[].Feature[]") the structure is not repeated here.</p> <p>The list of features for the whole document ends here.</p> <p>The "Body" element ends here.</p> <p>This element contains the "Receipt". It is only included, if a receipt for an other document is the content of this document. It is only included in receipts, which means that the type of the document (JSONPath: "Type") needs be "RECEIPTCUSTOMER" or "RECEIPTSUPPLIER". It contains in-</p>
--	---	---



		formation about the status of the processing of this document on the receivers side.
"ParentType": "ORDER",	String (36) ^{required}	This field contains the document type of the previous document this receipt refers to. It relates to the field with the JSONPath "Type" in the previous document. Values are restricted to the list of document types found in the type field of this document (see JSONPath: "Type").
"ParentMessageKey": "M990003",	String (36) ^{required}	This field contains the identification of the previous document in verbal communication this receipt refers to. It relates to the field under the JSONPath "MessageKey" in the previous document.
"ParentTransmissionKey": "M990003-201612",	String (72)	This field contains the identification of the previous document in machine-to-machine communication this receipt refers to. It relates to the field under the JSONPath "TransmissionKey" in the previous document.
"Log": [This element contains the list of the log relating to the document this receipt refers to.
{		Here starts the first log.
 "Code": "100",	Integer ≥ 0 ^{required}	This field contains the code of the severity of this log. The values are restricted to: 100 ... 199 : A general information 200 ... 299 : A warning 300 ... 399 : An error indicating the processing could not be finished (thus the document is deemed as if not transmitted)
 "Description": "Veni vedi transmitti",	String (∞)	This field contains the verbal description of this log.
 "Path": "Body.Item[0].ItemKey",	String (∞)	This field contains the JSONPath in the previous document which yielded this log. If this log contains the reason for the reception of the document, this field should start with an exclamation mark ("!"). If this log contains



<pre> "Issuer": "edigurus.org",</pre>	String (∞)	information about the processing of the previous document, this field should start with a hash ("#"). This field contains the origin that issued this log (normally it is the name of the user).
<pre> "Issued": "2016-12-24T12:00:00+01:00"</pre>	Timestamp	This field contains the time and date when the log occurred.
<pre> },</pre>		The first log ends here.
<pre> {</pre>		Here starts the second log. A receipt can contain an unlimited number of logs which are all included in this list of logs.
<pre> (see above)</pre>		Because this element is structured exactly like the first log (JSONPath: "Body.Package[]") the structure is not repeated here.
<pre> }</pre>		The second log ends here. Add a subsequent log, if more logs should be included in this receipt.
<pre>]</pre>		The list of logs here.
<pre> }</pre>		The "Receipt" element ends here
<pre> }</pre>		The "Message" element (and thus the message) ends here