



IUNGO BUSINESS DATA INTEGRATION SERVICE

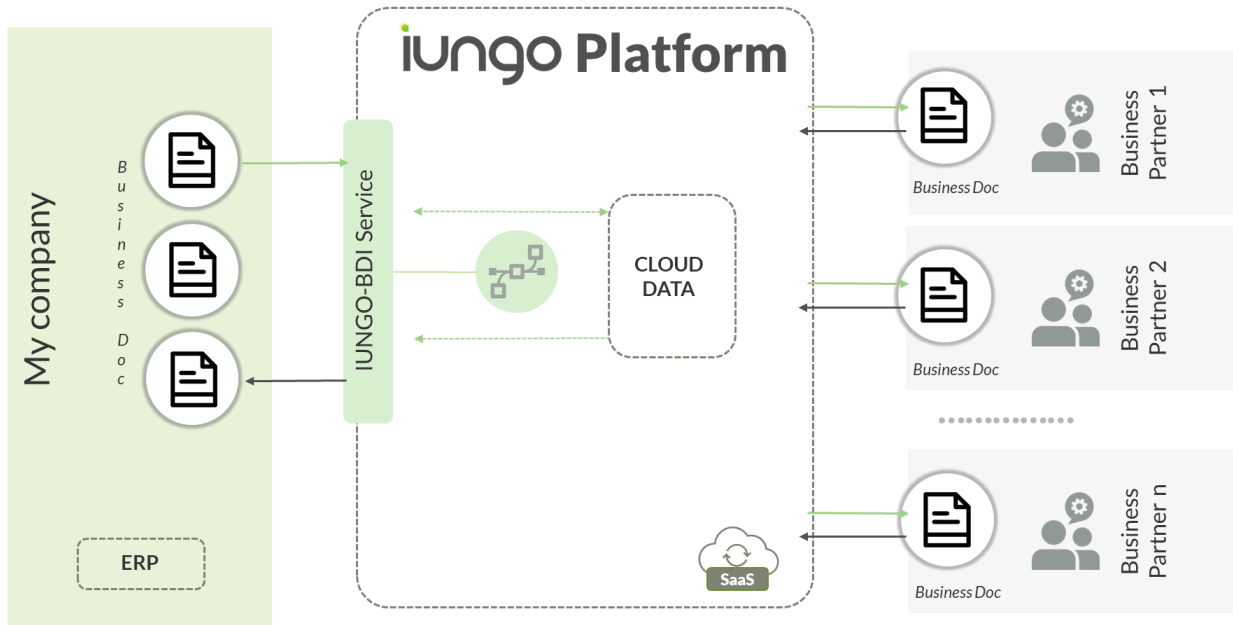
Technical documentation

Summary

1. Introduction	3
1.1. Scope	3
1.2. Out of Scope	3
2. Process Description	5
3. Technical Documentation	7
3.1. Pre-requirements.....	7
3.2. Connection service type	7
3.3. Required control information	7
3.4. Message Structure	8
3.5. TLS requirements needed to access the infrastructure.	8
4. Service description	9
4.1. Security	9
4.2. Message transit time	10
Appendix A – IUNGO BDI API Documentation	11

1. Introduction

The IUNGO BDI (Business Date Integration) service allows Business Partners (clients and suppliers) to automatically exchange Business documents and data.



Particularly the flows related to this service are: order (ORDERS), order response (ORDRSP) and advance shipping notes (DESADV, or ASN) electronic data.

The platform acts as a mediator and unique point of access for Partners, thus reducing the complexity and increasing the efficiency of the order management process.

1.1. Scope

The scope of this document is to describe the technical requirements for Business Partners to connect to and use the IUNGO BDI service. All technical aspects, like the type and form of the electronic record, and functional aspects (related only to the transmission of the document) of the messages will be here described.

1.2. Out of Scope

The following topics are out of scope of the current document and are documented separately:

Standard message structure

The standard XML format used for message exchange is predetermined and is thoroughly described in a separate document. The technical process here described assumes that the Partner can correctly handle (send and receive) a valid IUNGO XML document, according to the provided documentation.

Order data management

IUNGO BDI is a communication enabling platform that acts as a store & forward system for electronic messages. No business specific order management logic will be handled directly by IUNGO BDI. All order workflow related tasks will be implemented either in IUNGO Procurement or directly by the Partner.

2. Process Description

This section provides a brief overview of the of the order communication process.

Purchase/sale orders, order confirmations and advanced shipping notes, after having been created in the **Partner's ERP system**, can be automatically sent to/received by the Business Partner. For this purpose, a Partner can connect to IUNGO BDI to execute three basic tasks:

POP: *download* a document from IUNGO BDI. This allows the Partner to download all documents that were forwarded to him, one at a time;

ACK: *confirm* the download operation. This allows the Partner to confirm the correct download of a document from the IUNGO BDI platform. This acts as a safety mechanism that allows the Partner to re-download a document, should any error occur.

PUSH: *upload* a document to IUNGO BDI. This allows a Partner to connect and send a document to IUNGO BDI. The document contains, among other data, the sender and the receiver's unique ID's that are needed for correctly delivering the document to the Business Partner;

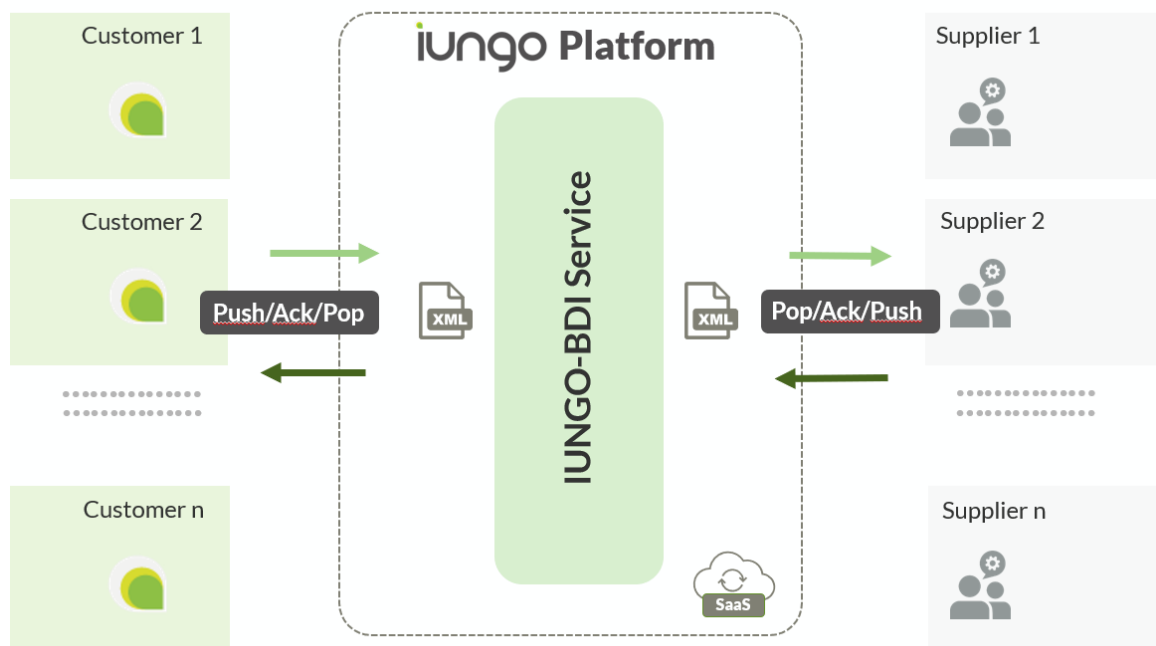


Figure 1 - IUNGO BDI technical architecture

Each Partner (either sender or receiver) may connect to IUNGO BDI independently and asynchronously (according to its own business logic), to perform either one of the three basic tasks.

The process of connecting to the BDI queues, as a Supplier requires:

- POP call
 - ⇒ call to connect to BDI queues and download the document sent by customer (ORDERS flow).
- ACK call
 - ⇒ call made after the download of the document.
- PUSH call
 - ⇒ call with ORDRSP flow, to send back to Customer the data through BDI.

The process of connecting to the BDI queues, as a Customer requires:

- PUSH call
 - ⇒ loading a file of the order in the BDI queues.
- POP call
 - ⇒ call to connect to BDI queues and download the document sent by Supplier (ORDRSP flow).
- ACK call
 - ⇒ call made after the download of the document.

NB: the ACK call must be made individually for each document downloaded from BDI

3. Technical Documentation

This section contains the technical documentation for the communication with IUNGO BDI.

3.1. Pre-requirements

The following are prerequisites for the Partner to communicate with IUNGO BDI:

INTERNET Access

The software client operating at the Partner's side must have internet access to access IUNGO BDI. More specifically, the client must be able to open HTTPs connections (port 443 of the TCP/IP protocol) and to accept all incoming messages of the opened connections.

Note: no incoming connections will be opened from IUNGO BDI towards the Partner; all connections are triggered by the Partner's client software.

Standard XML format management

IUNGO BDI handles only messages in the standard IUNGO XML format defined for the platform (see related documentation). Partners will not be allowed to send non-compliant documents; moreover, IUNGO BDI will deliver only standard compliant XML documents.

3.2. Connection service type

IUNGO BDI offers a REST API service type, where no application layer session is maintained. The interactions with IUNGO BDI are made through single calls (either PUSH, POP or ACK). Every request is atomic and independent from one another.

Requests are accessed at different URLs (to differentiate the type of request) according to the call type defined in Section 2.

3.3. Required control information

The messages can be downloaded and uploaded via authentication. The authentication is made through a unique private key (called `api_key`) that IUNGO provides to each Partner during service activation. The private key must be used each time a Partner connects to IUNGO BDI to ensure that:

- the client that is connecting is acting only on behalf of the Partner it was assigned to;
- other clients cannot access the Partner's private data.

The information needed to forward the message will be contained directly in the document (please refer to the standard XML document description).

3.4. Message Structure

The HTTP request and response messages are composed of a header and a body section (see HTTP standard documentation). The specific content of each section of the request and response messages to/from IUNGO BDI are described in Appendix A.

3.5. TLS requirements needed to access the infrastructure.

Below the TLS requirements for BDI service access:

- **Let's Encrypt CA**
- **TLS-SNI** support (Java 1.7 (2011))
- **TLS >= 1.2:**
- **Cipher Suites:**
 - **Tls 1.3:** *TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256, TLS_AES_128_GCM_SHA256*
 - **Tls 1.2:** *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_DHE_RSA_WITH_AES_128_GCM_SHA256, TLS_DHE_RSA_WITH_AES_256_GCM_SHA384, TLS_DHE_RSA_WITH_AES_128_CCM, TLS_DHE_RSA_WITH_AES_256_CCM, TLS_DHE_RSA_WITH_AES_128_CCM_8, TLS_DHE_RSA_WITH_AES_256_CCM_8*

4. Service description

This section describes qualitative and non-mandatory functional characteristics of the IUNGO BDI service.

4.1. Security

Various security related aspects are directly handled inside the IUNGO BDI platform, mainly *authentication, data privacy, data integrity and data protection*.

Authentication and Data Privacy

Authentication and data privacy are both handled through the same mechanism. During the activation of the service, IUNGO provides each Partner a unique and private key that the Partner will use to connect to the IUNGO BDI. This mechanism has, among other, the following advantages:

- *authenticity* – given that the unique keys are private, only the holder of the private key can certify its own identity; this ensures that only the Partner holding its private key can submit its own documents to the IUNGO BDI service (no other client can submit documents for him);
- *data privacy* – only the Partner presenting its own private key can download its own documents. This ensures that no other client can intentionally download another's Partner documents.

Note: the private key is required at each interaction.

Data Integrity

Various standard methods have been adopted to ensure data integrity. Data consistency is checked both during upload and download to check such characteristics as adherence to the IUNGO XML standard, syntactic integrity and to enforce all security related policies.

Data resilience

The platform relies on a redundant architecture to ensure failover.

Furthermore, data is replicated and backed up regularly to ensure that no intentional or non-intentional loss of data may occur.

Data protection

All connections to the IUNGO BDI service are made through the HTTPs protocol that encrypts data during the transmission. This ensures that private data cannot be intercepted when sent to/received from IUNGO BDI.

4.2. Message transit time

The time needed for a message to transit the IUNGO BDI platform is limited.

The actual time needed for a message to reach its destination Partner, however, depends on the frequency the destination Partner sends/downloads its own messages. Specific Partner related time constraints must be directly agreed by business Partners and are out of scope of the current document.

Appendix A – IUNGO BDI API Documentation

POP – Download available documents.

Untitled Request BUILD

POST https://icn-qty.iungoapps.com/v2.0/webapi/pop/orders Send

Params Authorization **Headers (11)** Body Pre-request Script Tests Settings

Headers 8 hidden

	KEY	VALUE	DESCRIPTION	⋮ Bulk Edit
<input checked="" type="checkbox"/>	Accept	application/xml		
<input checked="" type="checkbox"/>	Content-Type	application/xml		
<input checked="" type="checkbox"/>	Authorization	Bearer d6a6f8ead8fa11e79296cec278b6b50a		
	Key	Value	Description	

Body Cookies Headers Test Results Status: 200 OK Time: -- Size: 22.83 KB Save

Pretty Raw Preview Visualize Text ⌵

1

POP - Parameters

- Protocol: https
- URL-host: icn.iungoapps.com
- Requested type: POST
- Service:
 - v2.0/webapi/**pop**/orders
 - v2.0/webapi/**pop**/ordrsp
 - v2.0/webapi/**pop**/desadv
- Accept: application/xml
- Authorization: Bearer APIKEY
- If order has been downloaded successfully, web-api response with X-Receipt-Handle. This number should be used with **ACK** request to confirm download (see below) and it is unique for each order download, so each order download will be confirmed with single X-Receipt-Handle number

Error Coding

RESPONSE CODE	<ul style="list-style-type: none"> • 200 (OK) – the body of the response contains the first document of the queue; • 204 (No Content) – there are no documents to download; • 401 (Unauthorized) – if incorrect {api-key} is used; • 404 – No document available (que empty); • 415 (Unsupported Media Type) – if the request is badly formatted (e.g., incorrect header attributes); 	<p>For other codes, refer to the HTTP standard.</p>
----------------------	--	---

ACK – Document download acknowledge

Untitled Request BUILD

POST https://icn-qlty.iungoapps.com/v2.0/webapi/ack Send

Params Authorization Headers (12) Body Pre-request Script Tests Settings

Headers 8 hidden

	KEY	VALUE	DESCRIPTION	⋮ Bulk Edit
<input checked="" type="checkbox"/>	Accept	application/xml		
<input checked="" type="checkbox"/>	Content-Type	application/xml		
<input checked="" type="checkbox"/>	Authorization	Bearer d6a6f8ead8fa11e79296cec278b6b50a		
<input checked="" type="checkbox"/>	X-Receipt-Handle	59658ef3-11dd-4b71-af1f-9ace745f570b		
	Key	Value	Description	

Body Cookies Headers Test Results Status: 200 OK Time: 884 ms Size: 170 B Save

Pretty Raw Preview Visualize Text ≡

1

ACK – Parameters

- Protocol: https
- URL-host: icn.iungoapps.com
- Requested type: POST
- Service:
 - v2.0/webapi/ack
- X-Receipt-Handle: receiptHandle is automatically generated when order has been downloaded successfully. This number should be used to confirm the document download
- Authorization: Bearer APIKEY

Error Coding

RESPONSE CODE	<ul style="list-style-type: none"> • 200 (OK) document download <i>confirmed</i>; • 400 (Bad Request) – malformed request header (missing {receiptHandle}); • 401 (Unauthorized) – if incorrect {api-key} is used; • 404 – Invalid {receiptHandle}; 	For other codes, refer to the HTTP standard.
---------------	---	--

PUSH – Document upload

The screenshot shows a REST client interface for an "Untitled Request". The request method is "POST" and the URL is "https://icn.iungoapps.com/v2.0/webapi/push/ordrsp". The "Headers" tab is active, showing a table with the following headers:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Accept	application/xml	
<input checked="" type="checkbox"/> Content-Type	application/xml	
<input checked="" type="checkbox"/> Authorization	Bearer 50b17331d005478594a59222a2e36334	
Key	Value	Description

The "Body" tab is also visible, showing a status of "500 Internal Server Error", a time of "615 ms", and a size of "180 B". The body content is currently empty, and the "Text" tab is selected.

PUSH – Parameters


- Protocol: https
- URL-host: icn.iungoapps.com
- Requested type: POST
- Service:
 - v2.0/webapi/push/orders
 - v2.0/webapi/push/ordrsp
 - v2.0/webapi/push/desadv
- Content-type: application/xml
- Authorization: Bearer APIKEY


Error Coding

RESPONSE CODE	<ul style="list-style-type: none"> • 202 if document upload OK; • 401 (Unauthorized) – if incorrect {api-key} is used; • 400 (Bad Request) – typically badly formatted XML document; • 404 (Not Found) – invalid {documentType} • 415 (Unsupported Media Type) – if the request is badly formatted (e.g., incorrect header attributes); • 409 (Conflict) – Communication Not Enabled; 	For other codes, refer to the HTTP standard.
---------------	---	--



IUNGO S.p.A.

 Via Tacito 7, 41123 Modena

 Tel. 059 251643

 mktg@iungo.it

P.I. 02731600363