|  |
| --- |
|  |



European eInvoicing Standard in Italy

|  |  |
| --- | --- |
| **Project number** | INEA/CEF/ICT/A2017/1560867 2017-IT-IA-0150 |
| **Project acronym** | EeISI |
| **Project title** | European eInvoicing Standard in Italy |
| **Starting date** | 1 May 2018 |
| **Ending date** | 30 June 2019 |
| **Programme** | Connecting Europe Facility (CEF) CEF-TC-2017-3: eInvoicing |

InfoCert’s service for Economic Operators and Public Administrations, software release and documentation

Deliverable D4.6

|  |  |
| --- | --- |
| **Related WP** | WP4 – Implementation – Task 4.4 and Task 4.5 |
| **Deliverable number** | D4.6 |
| **Due date** | 31/01/2019 |
| **Revision date** | 31/01/2019 |
| **Actual date** | **31/01/2019** |

This Page Intentionally Left Blank

Deliverable Info

|  |  |
| --- | --- |
| **Editor (s)** | **InfoCert** |
| **Contributors** | **InfoCert** |
|  |  |
| **Abstract** | **This deliverable aims to provide the InfoCert’s service for Economic Operators and Public Administrations, software release and documentation with relation to EeISI project acitivities** |
| **Keywords** | **eInvoicing, semantic core model, Italian eInvoicing format, LegalInvoice, eDelivery, AS4, access point, SDI** |
|  |  |
| **Acknowledgement** | This work was partially supported by the European Commission (EC) through the Connecting Europe Facility (CEF) programme under project EeISI.(grant agreement no. INEA/CEF/ICT/A2017/1560867 2017-IT-IA-0150) |
| **Disclaimer** | The sole responsibility of this publication lies with the author(s). The European Union is not responsible for any use that may be made of the information contained therein. |
| **Confidentiality** | The information in this document is confidential and restricted only to the members of the EeISI consortium  (including the Commission Services). |
|  |  |
| **Note** | - |
|  |  |

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of change |
| 0.0.1 | 2018-01-31 | InfoCert | First release |
| 0.0.2 | 2018-01-31 | InfoCert | Minor revision |
| 0.0.3 | 2019-01-31 | InfoCert | LegalInvoice Hub Integration |
| 0.1.0 | 2019-01-31 | InfoCert | Final release |
| 0.1.1 | 2020-03-31 | Roberto Reale | Technical review and quality assessment |

This Page Intentionally Left Blank

Table of contents

[Deliverable Info 4](#_Toc6495027)

[Table of contents 6](#_Toc6495028)

[List of figures 6](#_Toc6495029)

[List of tables 7](#_Toc6495030)

[Executive Summary 8](#_Toc6495031)

[Glossary 10](#_Toc6495032)

[1. Normative references 11](#_Toc6495033)

[2. Data entry component 11](#_Toc6495034)

[2.1. Software architecture 11](#_Toc6495035)

[2.1.1. Core module 12](#_Toc6495036)

[2.1.2. Dataentry module 12](#_Toc6495037)

[3. LIHub integration 18](#_Toc6495038)

[3.1. LegalInvoice Hub Integration Overview 18](#_Toc6495039)

[3.2. LegalInvoice Hub Integration details 18](#_Toc6495040)

[3.2.1. Communication layer 18](#_Toc6495041)

[3.2.2. Document Management System 19](#_Toc6495042)

[3.2.3. Validation 22](#_Toc6495043)

[3.2.4. Transformation 22](#_Toc6495044)

[3.2.5. Digital signature 22](#_Toc6495045)

[3.2.6. Communication with the SDI 22](#_Toc6495046)

List of figures

[Figure 1 High level data entry structure 11](#_Toc6495017)

[Figure 2 Data entry UI (pro) 13](#_Toc6495018)

[Figure 3 Data entry UI (base) 14](#_Toc6495019)

[Figure 4 mouse over on info icon6 14](#_Toc6495020)

[Figure 5 info dialog example 15](#_Toc6495021)

[Figure 6 info dialog’s “Go to” functionality 15](#_Toc6495022)

[Figure 7 Data entry export results 16](#_Toc6495023)

[Figure 8 Data entry validation UI 16](#_Toc6495024)

[Figure 9 Data entry validation errors 17](#_Toc6495025)

[Figure 10 Data entry “Import invoice” UI 17](#_Toc6495026)

List of tables

[Table 1 EU invoicing metadata 22](#_Toc6495016)

Executive Summary

This document aims to provide the InfoCert’s service for Economic Operators and Public Administrations, software release and documentation.

The implementation report mainly covers:

* Data Entry module implementation
* LIHub integration

This Page Intentionally Left Blank

Glossary

|  |  |
| --- | --- |
| **AP** | Access Point |
| **AS4** | Applicability Statement 4 |
| **B2B** | Business to Business |
| **B2C** | Business to Consumer/Citizen |
| **B2G** | Business to Government |
| **BII** | Business Interoperability Interfaces |
| **C2G** | Citizen to Government |
| **CCTS** | Core Component Technical Specification |
| **CEF** | Connecting Europe Facility |
| **CEM** | Certified Electronic Mail – Legal Mail (PEC Posta Elettronica Certificata in Italy) |
| **CEN** | European Committee for Standardisation |
| **CII** | Cross Industry electronic Invoice |
| **CIUS** | Core Invoice Usage Specification |
| **DNS** | Domain Name System |
| **DSI** | Digital Service Infrastructures |
| **EDIFACT** | Electronic Data Interchange For Administration, Commerce and Transport |
| **EMSFEI** | European Multi-Stakeholder Forum on eInvoicing |
| **e-SENS** | Electronic Simple European Networked Services |
| **FatturaPA** | Public administration electronic invoice framework (FatturaPubblica Amministrazione) |
| **G2G** | Government to Government |
| **IMR** | Invoice Message Response |
| **INEA** | Innovation and Networks Executive Agency |
| **MLR** | Message Level Response |
| **OASIS** | Organization for the Advancement of Structured Information Standards |
| **PEPPOL** | Pan-European Public Procurement Online |
| **PEPPOL-BIS** | Pan-European Public Procurement Online Business Interoperability Specifications |
| **SDI** | Electronic exchange system in Italy (Sistema Di Interscambio) |
| **SML** | Service Metadata Locator |
| **SMP** | Service Metadata Publisher |
| **UBL** | Universal Business Language |
| **UN/CEFACT** | United Nations Centre for Trade Facilitation and Electronic Business |
| **UNTDID** | UN Trade Data Interchange Directory |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Location |
| **URN** | Uniform Resource Name |
| **XML** | Extensible Mark-up Language |

1. Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application.

* EN 16931-1:2017 Electronic invoicing - Part 1: Semantic data model of the core elements of an electronic invoice

Moreover the following Italian documentation is referenced in this deliverable:

* Schema del file xml FatturaPA versione 1.2 - xsd
* Specifiche tecniche del formato della FatturaPA versione 1.2.1- pdf
* Rappresentazione tabellare del tracciato FatturaPA versione 1.2.1- pdf
* Rappresentazione tabellare del tracciato FatturaPA versione 1.2.1- excel
* Foglio di stile per la visualizzazione della FatturaPA versione 1.2.1 - xslt
* generica Foglio di stile per la visualizzazione della Fattura Ordinaria versione 1.2.1 - xslt
* Elenco modifiche al tracciato FatturaPA - pdf
* Suggerimenti per la compilazione della FatturaPA versione 1.5

1. Data entry component
   1. Software architecture

The software is composed by two modules:

* the **core**, which exposes the APIs, these APIs implement the knowledge and the various mechanisms of editing, conversion and validation of the invoices;
* the **dataentry**, that implements a UI which provides the **core**’s functionalities to the final user in a easy-to-use way.

A high-level structure of the project can be schematized like:

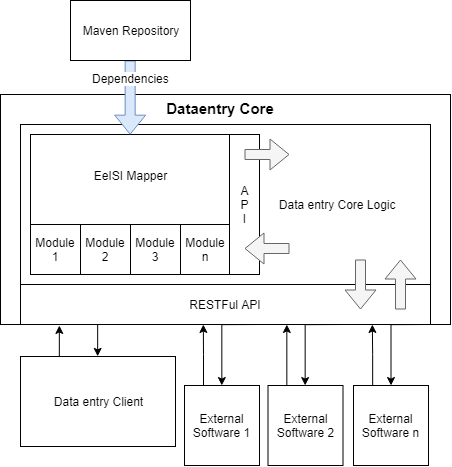


Figure 1 High level data entry structure

The dataentry core envelop the EeISI Mapper’s dependencies (using Maven), that is composed by multiple modules, one of these exposing the API that provide the conversion’s functionalities. The dataentry core implements the logic that uses the Mapper API for conversions and validations; the data entry core logic also contains the processes used to transform data in the way it’s can be manipulated, this feature is used to implement the invoice editor. Finally the dataentry core logic handles the result to be served at the RESTFul API.

* + 1. Core module

The **core** is structured to embed an external software called **mapper** (EeISI mapper) and improve it with the following features:

* provide a layer which allows the invoice editing;
* expose the APIs of editing, conversion and validation which can be used by the **dataentry**, and all the other third party software.

The **core** exposes the APIs with a restful framework, so the APIs are http REST requests.

**APIS**

host: the server identifier (ex. ‘[https://domain.com](https://domain.com/)’)

version: the software’s version (ex. ‘v1’)

base url example: [https://domain.com](https://domain.com/)/v1

* {host}/{version}/json2xml/?type={invoiceType}: POST request that expect a json invoice in the body and a url parameters (see the pattern) representing the invoice type, then return a json containing the xml data of the converted invoice;
* {host}/{version}/json2xml/generate/preview: POST request that expect a json invoice in the body, generate and save an xml static resource server side, then return a json containing the id of that resource;
* {host}/{version}/xml2json/?type={invoiceType}: POST request that expect an xml invoice in the body and a url parameters (see the pattern) representing the invoice type, then return a json containing the json data of the converted invoice;
* {host}/{version}/validate/?type={invoiceType}: POST request that expect an xml invoice in the body and a url parameters (see the pattern) representing the invoice type, then return a json containing the result of the validation\*
* {host}/{version}/validate/ext/schematron: POST request that expect an xml invoice in the body and return a json containing the result of the validation\*

\* Validation result: the validation result says the caller if the validation has been successful, if there any, contains the issues of the validation, the issues can be warnings or errors (also both in the same list).

* + 1. Dataentry module

The **dataentry** is a module written in javascript, using the ReactJS framework, the communication

with the server is implemented with ajax calls, it implements an UI for the following purposes:

* Create CEN invoices;
* Create CIUS-IT invoices;
* Import an invoice in a supported format;
* Export an invoice in a supported format;
* Validate an invoice in a supported format;
* Validate an invoice given an external schematron;

All these features are minded to be user friendly.

Some example of the UI can be the following:

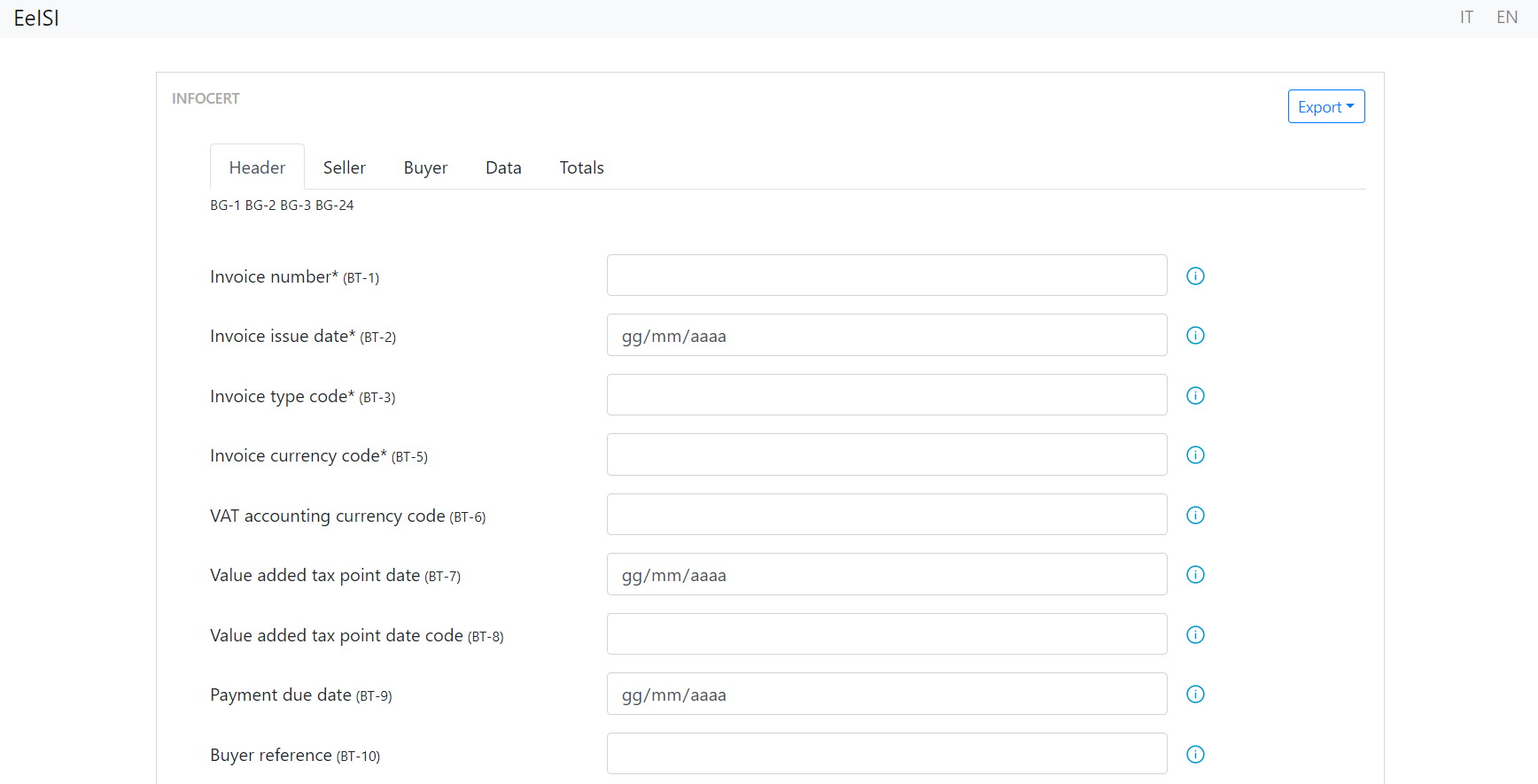
Create invoice

Figure 2 Data entry UI

On the top right of the page there are two selectors, these selectors allow the user to switch between Italian and English version of the application (base feature).

The ‘\*’ character beside a field label indicates that the field is required

The icon on the right of the fields give at the user information about the related field, there are two ways to interact with this icon (base feature):

* The cursor over the icon show a quick description of the related field:

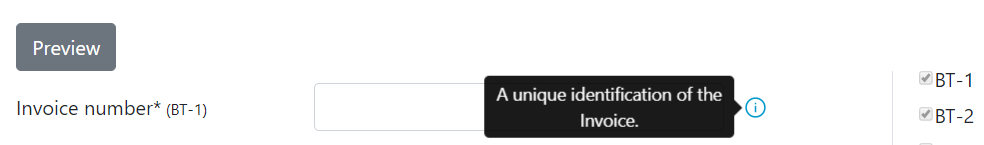


Figure 4 mouse over on info icon6

* Clicking the icon button, a dialog will appear, the content of the dialog is a quick description, some information about the field’s usage, cen BR and cius BR:

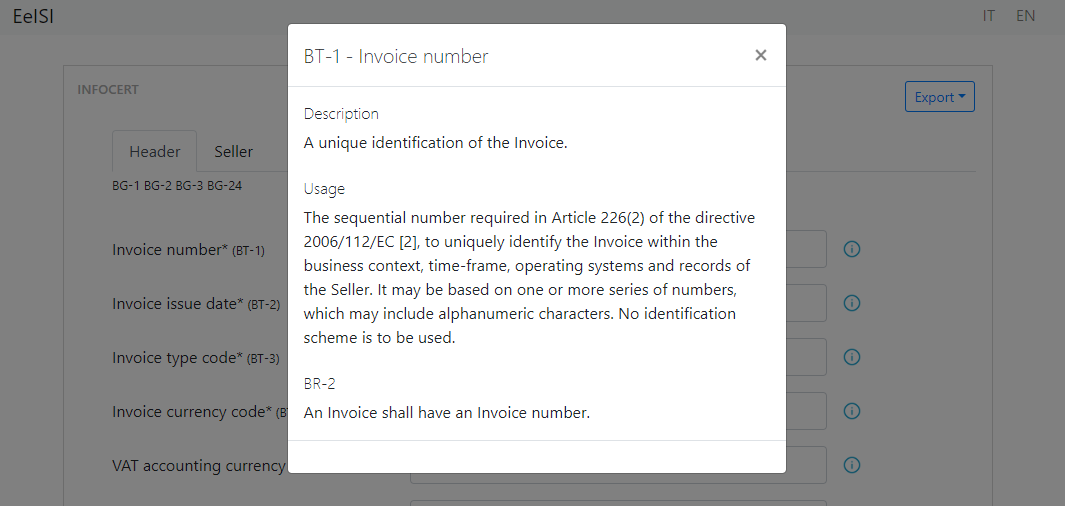


Figure 5 info dialog example

The dialog has a functionality that makes easy reading the sections, this functionality is called “Go to” (visible at the top of the dialog, just below the header), clicking on a section will scroll to the related one (if required, depends of the height of the page), and highlight the section for a few seconds, for example, clicking usage:

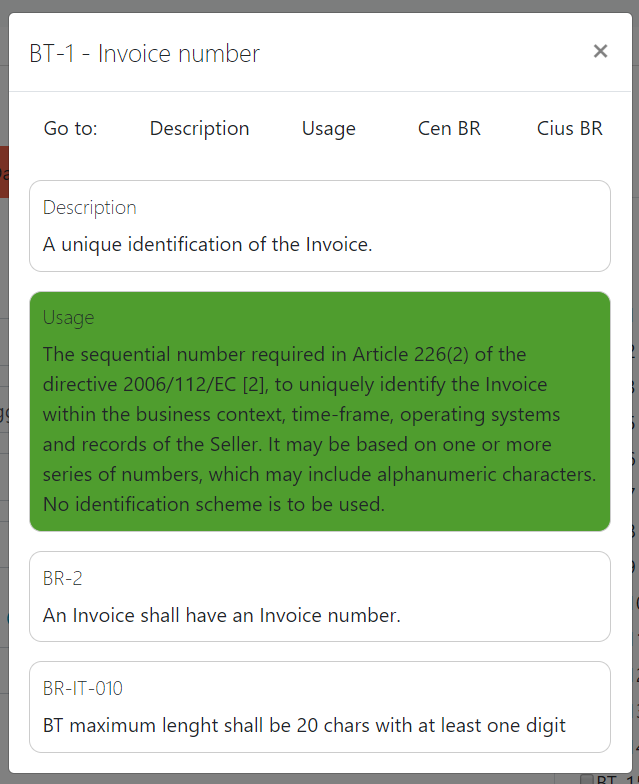
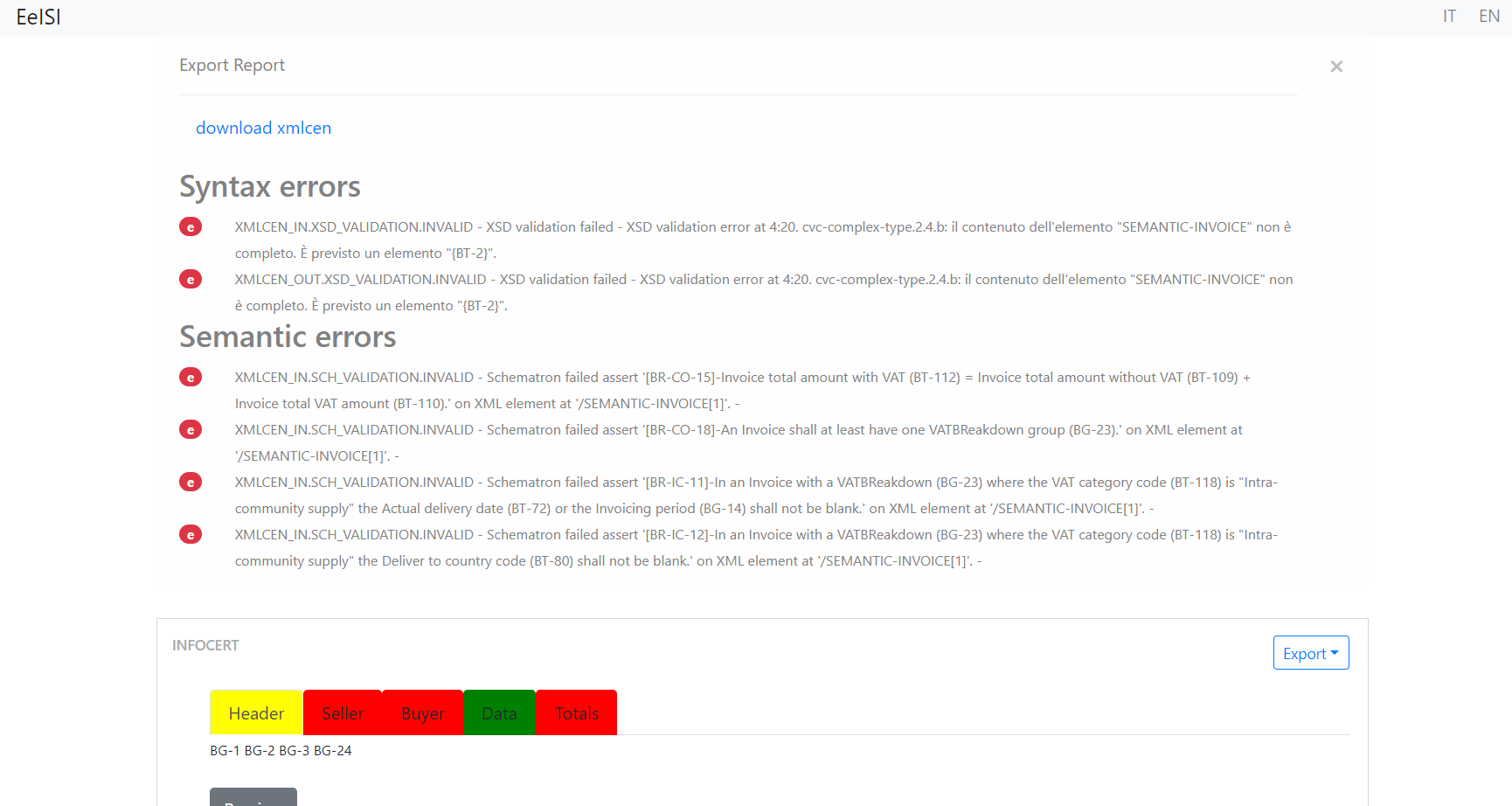


Figure 6 info dialog’s “Go to” functionality

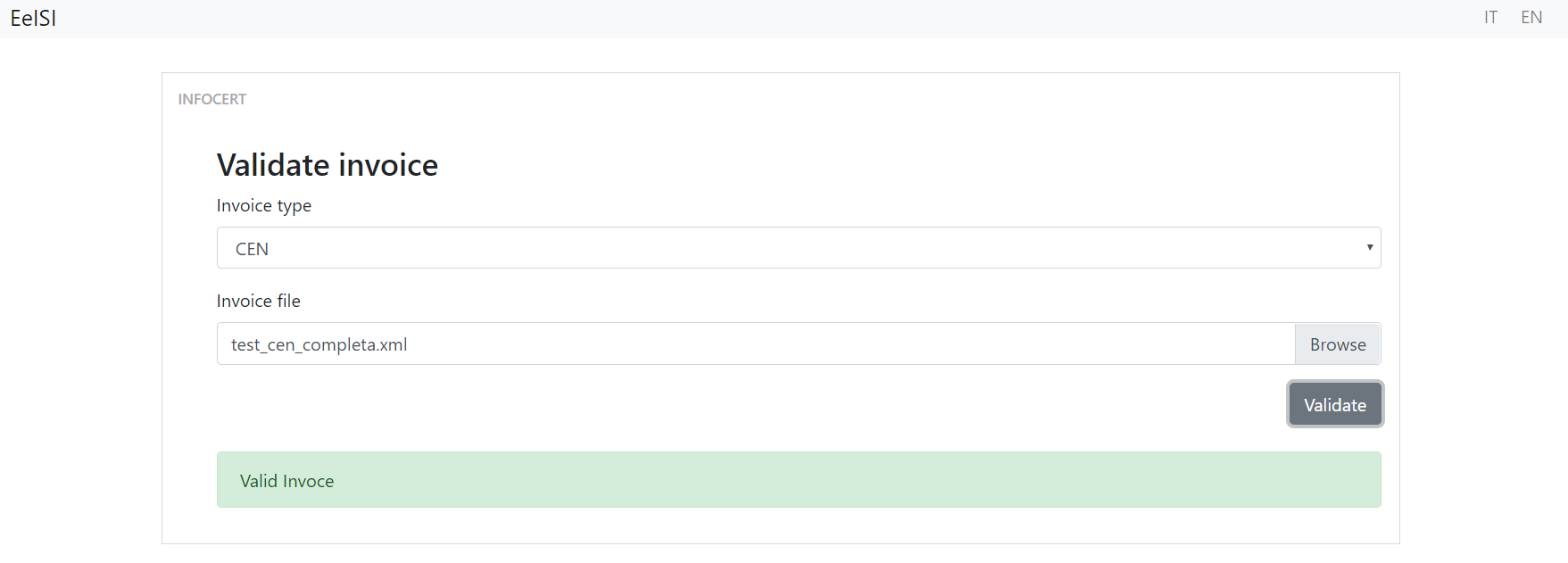
On the right top there is the “Export” button, which allow the user to export the invoice in different supported formats (base feature);

Figure 7 Data entry export results

**Create invoice - export functionality**

In the above screenshot we can see that if we try to export an incomplete invoice, the core generates some issues, the dataentry shows these issues so we can understand and fix them.

**Validation**

Figure 8 Data entry validation UI

Above an example of a validation of a valid cen invoice

**Import invoices (base feature)**

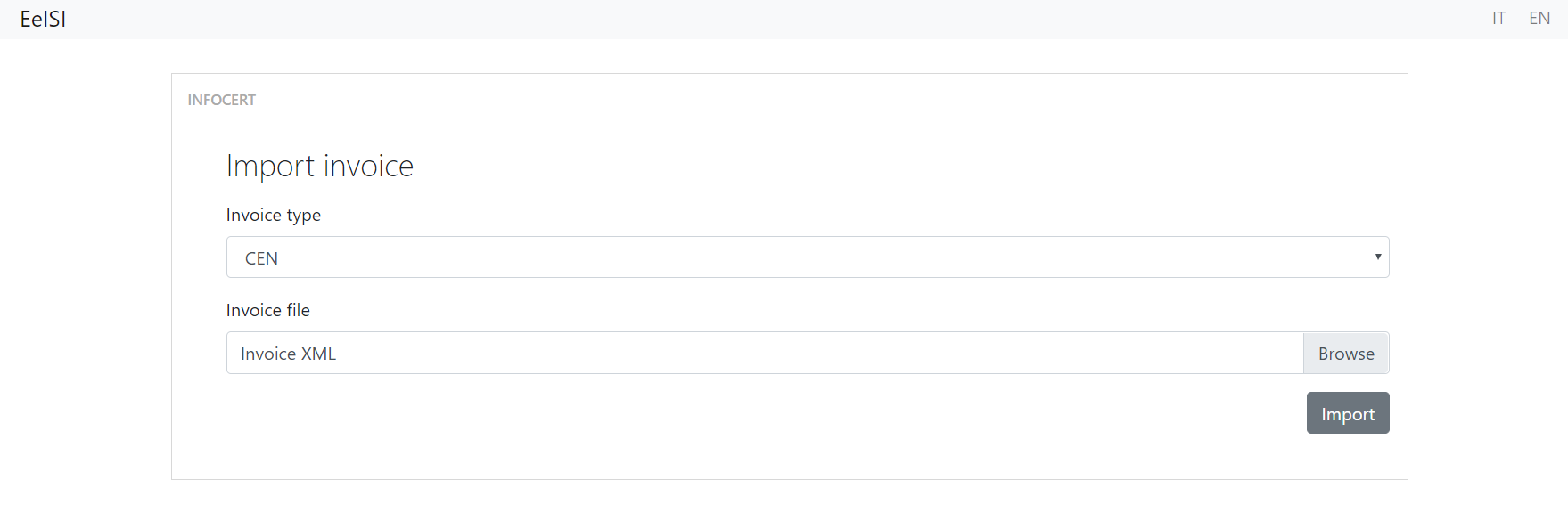
****

Figure 10 Data entry “Import invoice” UI

Above there is the “Import invoice” UI, it allows to select the invoice type to import, after the user has selected the invoice type can select the invoice file, than clicking the “Import” button the application redirects to the editing form, with the fields filled with the imported values.

1. LIHub integration
   1. LegalInvoice Hub Integration Overview

The LegalInvoice Hub system currently provides a set of functionalities related to the eInvoicing area exposed through different communication channels.

The standard processes that are managed by LIHub are composed by three steps:

1. document acquisition,

2. document processing,

3. information export.

These steps are performed by application modules that provide specific and atomic functionalities.

Some of these modules have been updated to manage the invoices in European format.

* 1. LegalInvoice Hub Integration details
     1. Communication layer

**Introduction**

Batch input processing is one of the main functionalities of LegalInvoice Hub.

The ingestion of documents and any return data (status notifications, etc.) can transit in a channel based on FTP transfer.

Files transferred in batch mode are further processed by the LegalInvoice Hub component dedicated to file system import.

Internal invoice management follows a standard and regulated process that is independent of the channel from which the invoice was imported.

**Technical specifications**

The Infocert server dedicated to managing files over FTP has currently the following characteristics:

• Dedicated FTP box

• protocol: FTPS (Explicit FTP over TLS)

• listening port: 990 TCP

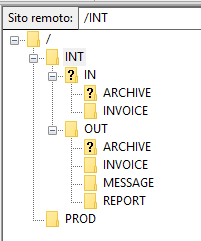
• data port: 989 TCP

• data port range(passive mode): 35000 – 35030

Processing of the received files takes place immediately via a file system trigger that triggers the invoice import.

LegalInvoice Hub FTP layer provides also, as an alternative channel, a sFTP asset that can be configured with public key authentication or password only authentication.

Within the ftp space there is a fixed folder structure that allows the transfer of information in two environments: test and production:

The new European formats require a new folder for the invoice ingestion of the kind:

/<env\_name>/IN/EU\_INVOICE

and a new folder for the message responses:

/<env\_name>/OUT/EU\_MESSAGE

**File naming conventions**

It is possible to name invoices in different ways:

1. Entering any name, as long as it is unique.
2. Choosing a name containing metadata to be added to the invoice according to the following scheme:

INVOICE\_NAME%KEY=VALUE%KEY1=VALUE%.xml

The rules to be respected are:

1. The invoice name must precede the custom attributes and must not contain the character %
2. I custom attributes must be delimited by character %
3. The key and the value must be separated from the character =
4. The custom attributes are optional

eg (for european invoices):

CII00802050153\_00023%exercise=2018%.xml

Invoices are taken over by the system and then deleted from the folder.

A unique InfoCert identifier is assigned to each invoice (invoiceId).

* + 1. Document Management System

The DMS that provides the data persistency of the documents is based on a RDBMS structure for the metadata and a Filesystem system for the document content.

The European invoices have been added as new documents classes named:

* InvoiceUbl21Cen
* InvoiceCIICen

both with the following metadata:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **nullable** | **multiple** |
| anno-fiscale | INTEGER | 1 | 0 |
| bulk-data | BASE64BINARY | 1 | 0 |
| canale-sdi | STRING | 1 | 0 |
| codice-cig | STRING | 1 | 0 |
| codice-cup | STRING | 1 | 0 |
| codice-destinatario | STRING | 1 | 0 |
| codice-fiscale | STRING | 1 | 0 |
| codice-fiscale-emittente | STRING | 1 | 0 |
| company-id | STRING | 1 | 0 |
| company-pec-address | STRING | 1 | 0 |
| custom-attributes | BASE64BINARY | 1 | 0 |
| data-documento | DATE | 1 | 0 |
| data-inizio-numerazione | DATE | 1 | 0 |
| data-inizio-transazione | DATETIME | 1 | 0 |
| denominazione | STRING | 1 | 0 |
| denominazione-emittente | STRING | 1 | 0 |
| email-destinatario | STRING | 1 | 0 |
| errore-sogei | STRING | 1 | 1 |
| errore-validazione-fattura | STRING | 1 | 1 |
| esito | STRING | 1 | 0 |
| esito-descrizione | STRING | 1 | 0 |
| esportato | BOOLEAN | 1 | 0 |
| external-anagrafica | STRING | 1 | 0 |
| external-reference | STRING | 1 | 0 |
| fonte | STRING | 1 | 0 |
| formato-fattura | STRING | 1 | 0 |
| hash | STRING | 1 | 0 |
| id-anagrafica | STRING | 0 | 0 |
| id-anagrafica-master | STRING | 1 | 0 |
| id-anagrafica-parent | STRING | 1 | 0 |
| IdentificativoSdi | STRING | 1 | 0 |
| id-paese | STRING | 1 | 0 |
| id-paese-emittente | STRING | 1 | 0 |
| id-processo | STRING | 0 | 0 |
| id-sottotipologia | STRING | 1 | 0 |
| id-supporto | INTEGER | 1 | 0 |
| id-tipologia | STRING | 0 | 0 |
| indice-fascicolo | STRING | 1 | 0 |
| is-conserved | BOOLEAN | 1 | 0 |
| is-signed | BOOLEAN | 1 | 0 |
| lbhub-soap-operation | STRING | 1 | 0 |
| localita | STRING | 1 | 0 |
| ndoc-bucket | STRING | 1 | 0 |
| ndoc-classe-documentale | STRING | 1 | 0 |
| ndoc-correlation-id | STRING | 1 | 0 |
| ndoc-error-code | STRING | 1 | 0 |
| ndoc-error-desc | STRING | 1 | 0 |
| ndoc-path | STRING | 1 | 0 |
| ndoc-policy | STRING | 1 | 0 |
| ndoc-time-reference | DATETIME | 1 | 0 |
| ndoc-token | STRING | 1 | 0 |
| nome-file-originale | STRING | 1 | 0 |
| nome-file-sogei | STRING | 1 | 0 |
| nome-supporto | STRING | 1 | 0 |
| note | STRING | 1 | 0 |
| notificato | BOOLEAN | 1 | 0 |
| numero-documento | STRING | 1 | 0 |
| partita-iva | STRING | 1 | 0 |
| partita-iva-emittente | STRING | 1 | 0 |
| partita-iva-pa | STRING | 1 | 0 |
| pec-casella-cc | STRING | 1 | 0 |
| pec-casella-destinatario | STRING | 1 | 0 |
| pec-destinatario | STRING | 1 | 0 |
| pec-messaggio | BASE64BINARY | 1 | 0 |
| pec-oggetto | STRING | 1 | 0 |
| profilo-sogei | STRING | 1 | 0 |
| progressivo-invio | STRING | 1 | 0 |
| progressivo-invio-int | STRING | 1 | 0 |
| provincia | STRING | 1 | 0 |
| request-id | STRING | 1 | 0 |
| resend | INTEGER | 1 | 0 |
| riferimento-amministrazione | STRING | 1 | 0 |
| sdi-pec-address | STRING | 1 | 0 |
| serie | STRING | 1 | 0 |
| sign-cname | STRING | 1 | 0 |
| sign-error-code | STRING | 1 | 0 |
| sign-error-desc | STRING | 1 | 0 |
| sign-issuer | STRING | 1 | 0 |
| sign-issuer-cnname | STRING | 1 | 0 |
| sign-subject | STRING | 1 | 0 |
| sign-timestamp-date | DATETIME | 1 | 0 |
| stato | STRING | 1 | 0 |
| tipo-documento | STRING | 1 | 0 |
| tipo-fattura | STRING | 1 | 0 |
| totale-importo | STRING | 1 | 0 |
| via | STRING | 1 | 0 |
| webrainbow-content | BOOLEAN | 1 | 0 |
| webrainbow-export-error | STRING | 1 | 0 |
| ws-output | STRING | 1 | 0 |
| ws-request-id | STRING | 1 | 0 |
| x-riferimento-message-id | STRING | 1 | 0 |

Table 1 EU invoicing metadata

* + 1. Validation

The invoices that are imported into the LegalInvoiceHub system must be validated by the emitting company.

Infocert system provides a validation with a xsd artifact due to the further necessity of extraction of some specific attributes from the invoice (VAT code, …).

* + 1. Transformation

Some information are extracted from the invoice due to guarantee its correct routing.

For instance, the main attribute of an invoice that denote the owner of the document is the VAT code.

This value is extracted with a xPath rule that depends on the syntax:

* UBL: "//AccountingSupplierParty/Party/PostalAddress/Country/IdentificationCode/text()";
* CII: "//AccountingSupplierParty/Party/PartyTaxScheme/CompanyID/text()";

The transformation module, besides, generates the correct file name that must be set according to the SDI naming conventions. The prefix of the invoice file name has to denote the invoice syntax:

- II for CII invoices

- UB for UBL invoices

examples:

UB12345678901\_00001.xml

II99999999999\_00002.xml

* + 1. Digital signature

After the transformation module the invoice is routed to a sub-process that differs from the standard (Italian) process.

The European invoces aren’t digitally signed, this implies a specific routing that skips the signing phase of the processment.

* + 1. Communication with the SDI

Infocert owns a certified communication channel with the SDI; within this channel, due to the abstract packaging format (archive files in zip format), the European invoices can be added without modifications of then application.

The only difference is the file name of the single invoice contained into the archive that has to be set as described in the previous paragraph.