senaite.lis2a Documentation

Release 1.0.2

Riding Bytes
Naralabs

Nov 29, 2022

Contents

1 Installation **3**

2 Interpreters **5**

3 Changelog **7**

3.1 2.0.0 (unreleased) 7

3.2 1.0.0 (unreleased) 7

This add-on enables the import of data compliant with the industry supported standard CLSI (Clinical and Laboratory Standards Institute, formerly NCCLS) LIS2-A2 *"Specification for Transferring Information Between Clinical Laboratory Instruments and Information Systems"*, a revision of ASTM E1394-97.

senaite.lis2a makes use of senaite.jsonapi by registering an *IPushConsumer* adapter, in charge of the interpretation of LIS2-A2 incoming data.

This package is not for the transmission or data exchange across RS-232 or TCP/IP, but provides an HTTP route to *push* results data that are compliant with LIS2-A2. For serial binary data exchange compliant with LIS1-A, please look at the command line tool senaite.serial.cli.

For high-demand instances, is strongly recommended to use this add-on together with senaite.queue. senaite.lis2a delegates the import of results to senaite.queue when installed and active. Otherwise, the import of results takes place in a single transaction as soon as the data is received.

Note: In 2001, ASTM Committee E31 decided to restructure its operations, with the intent of focusing on standards-development issues such as security, privacy, and the electronic health record. Part of the reorganization plan was to transfer responsibility for E31.13 standards to NCCLS. Following this transfer, standard ASTM E1394 was redesignated as NCCLS standards LIS2-A2.

Table of Contents:

CHAPTER 1

Installation

To install senaite.lis2a in your SENAITE instance, simply add this add-on in your buildout configuration file as follows, and run *bin/buildout* afterwards:

```
[buildout]

...

[instance]
...
eggs =
    ...
    senaite.lis2a
```

With this configuration, buildout will download and install the latest published release of senaite.lis2a from Pypi.

Note: For high-demand instances, is strongly recommended to use this add-on together with senaite.queue. senaite.lis2a delegates the import of results to senaite.queue when installed and active. Otherwise, the import of results takes place in a single transaction as soon as the data is received.

CHAPTER 2

Interpreters

SENAITE.LIS2A allows to setup custom interpreters to handle messages differently, depending on a user-defined criteria. The most common scenario is to have a different interpreter for each instrument. Multiple interpreters for same instrument but used based on different criteria are also supported.

You need first to create a folder in your add-on and register it as a resources directory for type *senaite.lis2a.interpreters* in *configure.zcml* as follows:

```
<!-- resource directory for LIS2A2 interpreters -->
<plone:static
    directory="interpreters"
    type="senaite.lis2a.interpreters" />
```

Note that in this case, the name of the folder is "interpreters".

Then, you only need to create a file in json format representing the interpreter with your own criteria and mappings. System will only consider files ending with the extension ".json".

For example, given a message like the following:

```
H|\^&|9||cobas infinity^Roche Diagnostics^^3.03.12|||||cobas infinity ^Roche␣
↪Diagnostics||P||20221021184225
P|1||||||||||||||||||||||20221013093955
O|1|221000044W|^^221000002|^^^77140-2\^^^1996-8\^^^59570-
↪2|R|20221013093955|||||X||||SER||||||0||||F
R|1|^^BCRBT^77140-2|44|µmol / l|[44.2 - 132]||||F||||20221021184219
R|2|^^BUN^1996-8|4|mmol / l|[2.2 - 2.7]||||F||||20221021184219
R|3|^^CHLRD^59570-2|5|mmol / l|[3.5 - 7.1]||||P||||20221021184218
L|1|N
```

we can create a file "cobas_infinity.json" representing an interpreter with the criteria and rules for the import of the message above:

```
{
  "id": "COBAS_INFINITY",
  "extends": "LIS2-A2",
```

(continues on next page)

(continued from previous page)

```
  "selection_criteria": {
    "H.SenderName": "cobas infinity"
  },
  "result_criteria": {
    "R.ResultStatus": "F"
  },
  "mappings": {
    "id": [
      "O.InstrumentSpecimenID"
    ],
    "keyword": [
      "R.UniversalTestID_Type",
      "R.UniversalTestID_ManufacturerCode"
    ],
    "result": "R.Measurement",
    "ranges": "R.ReferenceRanges",
    "units": "R.Units",
    "capture_date": "R.DateTimeCompleted",
    "captured_by": [
      "R.OperatorIdentification",
      "R.VerifierIdentification"
    ]
  },
  "O": {
    "InstrumentSpecimenID": [3,2]
  }
}
```

Note that this interpreter *extends* from another interpreter, called LIS2-A2. The LIS2A-2 interpreter provides the basic mappings and positional information from records and values as defined in the standard.

When a LIS2-A2 message is received, the system first checks if the message matches with the criteria set under *selection_criteria* parameter. If all criteria are met, system considers that the given interpreter supports the message and will therefore be used to extract and import results. Otherwise, the system skips the interpreter and keeps searching in resources directory until a suitable interpreter is found.

When the results records from the LIS2-A2 message are processed following the rules defined in the interpreter, the system takes the **result_criteria** into consideration to whether try or not try to import the result record. Going back to the example above, the last results record won't be imported because the value for *ResultStatus* is not *F* (Final), but 'P'(Preliminary):

```
R|3|^^CHLRD^59570-2|5|mmol / l|[3.5 - 7.1]|||P||||20221021184218
```

After results from the message are filtered in accordance with the criteria above, system tries to import the results in accordance with the information provided in *mappings*.

CHAPTER 3

Changelog

3.1 2.0.0 (unreleased)

- Compatibility with SENAITE 2.x and senaite.astm

3.2 1.0.0 (unreleased)

First version