DoJSON Documentation

Release 1.0.0

Invenio collaboration

January 14, 2016

Contents

DoJSON is a simple Pythonic JSON to JSON converter.

CHAPTER 1

Installation

DoJSON is on PyPI so all you need is:

```
$ pip install dojson
```

CHAPTER 2

Example

A simple example on how to convert MARCXML to JSON:

```
from dojson.contrib.marc21.utils import create_record, split_stream
from dojson.contrib.marc21 import marc21
[marc21.do(create_record(data)) for data in split_stream(open('/tmp/data.xml', 'r'))]
```

CHAPTER 3

API

DoJSON API.

class dojson.**Overdo**(*bases=None*, *entry_point_group=None*)
Translation index.

build()
Build.

do(*blob*, *ignore_missing=True*)
Translate blob values and instantiate new model instance.

Raises MissingRule when no rule matched and ignore_missing is False.

Parameters

- **blob** – dict-like object on which the matching rules are going to be applied.
- **ignore_missing** – Set to False if you prefer to raise an exception MissingRule for the first key that it is not matching any rule.

New in version 1.0.0: ignore_missing allows to specify if the function should raise an exception.

missing(*blob*)
Return keys with missing rules.

over(*name*, **source_tags*)
Register creator rule.

3.1 Command line interface

DoJSON command line interface.

This script is installed using the magic console_script section in entry_points parameters thanks to setuptools. In order to get help from console one can run:

```
$ dojson --help
```

Extensions

New extensions with loaders, dumpers, or rules can be provided via entry points.

- dojson.cli.load functions expecting a stream and returning Python dict or iterator;
- dojson.cli.dump functions expecting a Python object and returning str;
- dojson.cli.rule instances of *dojson.Overdo* with loaded rules.

dojson.cli.**open_entry_point**(*group_name*)
Open entry point.

3.2 Contrib

Define model for transforming MARC21.

CHAPTER 4

Changes

4.1 Version 1.0.0 (released 2016-01-14):

4.1.1 Incompatible changes

- Removes support for single key matching multiple rules. Please make your rules mutually exclusive!
- controlfields 00x are expected to be the element or a list of multiple elements.

4.1.2 New features

- Adds new keyword argument *ignore_missing* to *Overdo.do* method to specify if method should raise *MissingRule* exception when there is no matching rule for a key.
- Adds new CLI option *–strict* to the *do* command that sets the *ignore_missing* argument to *False*. (#51)
- MARC XML serialization from to_marc21.

4.1.3 Improved features

- Adds support for Python 3+.
- Uses an OrderedDict to let the external tools working on *dict* (like json) behave correctly.
- All results from rules using *for_each_value* decorator are being automatically extended. This is useful for repeatable MARC21 fields with different indicators. (#53)
- Record are stored in an immutable sorted structure which enables to keep the intended order while offering easy ways to access, index and manipulate.
- Adds two records to be tested.
- Reorders some of the assertion: *expected* == *actual*.

4.2 Version 0.4.0 (released 2015-11-18):

4.2.1 New features

- Improves dojson.contrib.marc2.utils.load() to read the input by iterating of the open stream, rather than loading it all in memory in one go. (#45) (#46)

- Renames OverUndo to Underdo following same name convention as for Overdo.

4.2.2 Bug fixes

- Fixes indicator extraction from value in *Underdo* model.

4.3 Version 0.3.0 (released 2015-11-09):

4.3.1 New features

- Adds **experimental** rules for converting human readable JSON into a JSON representation of the MARC21 Format.
- Adds *do* and *missing* commands for *dojson* command line interface (see *dojson –help* for more information).

4.3.2 Improved features

- Adds missing mapping for the first indicator of field 856.

4.4 Version 0.2.0 (released 2015-10-07):

4.4.1 New features

- Adds the posibility to use base DoJSON model so the rules are "inherited" from them.
- Adds new decorator *ignore_value* that remove the key in the resulting json for None value.

4.4.2 Improved features

- Uses entry points instead of plain imports to load the creator rules.

4.4.3 Bug fixes

- Removes calls to PluginManager consider_setuptools_entrypoints() removed in PyTest 2.8.0.

4.5 Version 0.1.1 (released 2015-07-27):

- Sorts and removes duplicated enum values.
- Swaps wrongly defined repeatable and non-repeatable subfields. (#23)
- Addresses issue when allowed indicators where defined as a range. (#22)

4.6 Version 0.1.0 (released 2015-07-03):

- Initial public release.

CHAPTER 5

Contributing

Bug reports, feature requests, and other contributions are welcome. If you find a demonstrable problem that is caused by the code of this library, please:

1. Search for already reported problems.
2. Check if the issue has been fixed or is still reproducible on the latest *master* branch.
3. Create an issue with **a test case**.

If you create a feature branch, you can run the tests to ensure everything is operating correctly:

```
$ python setup.py test

...

====== 31 passed, 23 skipped in 1.37 seconds ======
```

You can also test your feature branch using Docker:

```
$ docker-compose build
$ docker-compose run web python setup.py test
$ docker-compose run web python setup.py build_sphinx
$ docker-compose run web pep257 --match-dir='dojson'
```

CHAPTER 6

Authors

DoJSON is developed for the Invenio digital library software.

Contact us at info@invenio-software.org.

Active contributors:

- Jiri Kuncar <jiri.kuncar@cern.ch>
- Esteban J. G. Gabancho <esteban.jose.garcia.gabancho@cern.ch>
- Sami Hiltunen <sami.mikael.hiltunen@cern.ch>
- Tibor Simko <tibor.simko@cern.ch>

CHAPTER 7

License

DoJSON is free software; you can redistribute it and/or modify it under the terms of the Revised BSD License quoted below.

Copyright (C) 2015 CERN.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

In applying this license, CERN does not waive the privileges and immunities granted to it by virtue of its status as an Intergovernmental Organization or submit itself to any jurisdiction.

d

B

D

M

O