Invenio Development Practices

How to Contribute

Tibor Šimko

`@tiborsimko`

Invenio User Group Workshop 2015

Geneva, Switzerland, 12–15 October 2015

Life cycle of a bug report

Step 1 of 6: Reporting

An issue in the morning
A feature request in the evening
What else?

— after Ekon

invenio-unapi

invenio-records

Here is how to reproduce . . .

BibUpload: investigate non-indexing of a record with 005
#2693   Edit   New issue

🚫 Closed   **tiborsimko** opened this issue on 21 Jan · 3 comments

TS   **tiborsimko** commented on 21 Jan   Owner   ✏️

There seems to be a bug leading to non-indexing of some records. Here is how to reproduce the problem.

Let us recreate demo site and index any pending OAI repository jobs:

```
$ invenio-recreate-demo-site --yes-i-know
$ sudo -u www-data /opt/invenio/bin/bibindex -u admin -v9
$ sudo -u www-data /opt/invenio/bin/bibindex 10
$ sudo -u www-data /opt/invenio/bin/bibindex -u admin -v9
$ sudo -u www-data /opt/invenio/bin/bibindex 11
```

Let's insert new record looking like record ID 1: (note missing 001 but keeping 005)

```
$ wget -O /tmp/z.xml http://pcuds06.cern.ch/record/1/export/xm
$ grep -v '"001"' /tmp/z.xml > /tmp/zz.xml
$ colordiff -u /tmp/z.xml /tmp/zz.xml
--- /tmp/z.xml   2015-01-21 14:08:47.772553827 +0100
```

Labels   ⚙
c_BibIndex
c_BibUpload
legacy
p_critical
s_TIND
t_bug

Milestone   ⚙
v1.2.0

Assignee   ⚙
TS tiborsimko

Notifications

Reporting tips

1. If it ain't on GitHub, it does not exist.
2. Watch notifications in interested repositories.
3. Check existing issues before submitting.
4. Describe concrete steps how to reproduce a problem.
   - what you did
   - what was the outcome
   - what you expected instead
5. Submit a test case. Submit a use case.
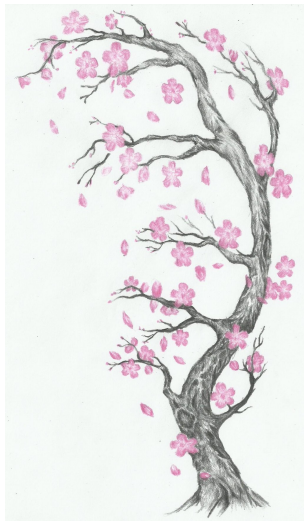6. Get involved! Comment and advise on other issues.
7. Join Invenio chat room.

Step 2 of 6: Triaging

Step 2 of 6: Triaging

No milestone
No assignee
A lonely feeling

— after Yosa Buson (1716-1784)

Issues, milestones, labels, humans

Triage process

- who?
  - 9 persons reachable at `@inveniosoftware/triagers`
  - distributed triage team representing "technology" and "services"
  - community presence and volunteering

- when?
  - ongoing triage of incoming issues
  - once-a-week live triage brainstorms

- how?
  - github labels, milestones, assigness
  - priorities and roadmaps

Triaging tips

1 Every issue should have an assignee.
2 Every issue should have a milestone.
3 Use additional type and priority labels.
4 Use additional service labels to provide overview.
5 Silent since a month? Ping responsible.
6 Nobody in sight to work on this? Discuss and triage.
7 No time to realistically tackle this? "Someday" or close.
8 No reply? Ping `@inveniosoftware/triagers`.

Step 3 of 6: Developing

Step 3 of 6: Developing

Mutable defaults
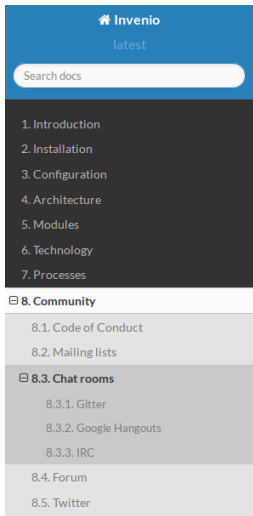*The Master*
Is full of regrets

— after Yosa Buson (1716-1784)

Contributing documentation

http://invenio.readthedocs.org/

Contributing translations

```
#: modules/bibauthorid/lib/bibauthorid_templates.py:2246
msgid "most recent documents:"
msgstr "Neueste Dokumente:"

#: modules/bibauthorid/lib/bibauthorid_templates.py:2255
msgid "Sorry, there are no documents known for this person"
msgstr "Sorry, es gibt keine Dokumente für dieser Person gekennzeichnet"

#: modules/bibauthorid/lib/bibauthorid_templates.py:2325
msgid "The following profile is the one you were viewing before logging in: "
msgstr ""
"Das folgende Profil ist dasjenige, die Sie sahen, bevor Sie sich anmeldeten: "
```

```
❶  % 614k de.po   PO                              unix | 2102: 0   16%
Sorry, es gibt keine Dokumente für dieser Person gekennzeichnet<
```

```
❷  * 64 *de.po*   Text   Ⓚ                        unix | 1:21   All
```

Contributing code

```python
from invenio_ext.sqlalchemy import db
from invenio_utils.datastructures import SmartDict

from .models import RecordMetadata
from .registry import functions
from .signals import (after_record_insert, after_record_update,
                      before_record_insert, before_record_update)


class Record(SmartDict):

    @property
    def __key_aliases__(self):
        return cfg['RECORD_KEY_ALIASES']

    def __getitem__(self, key):
        try:
            return super(Record, self).__getitem__(key)
        except KeyError:
            if key in self.__key_aliases__:
                if callable(self.__key_aliases__[key]):
```

`- 3.9k api.py  Python  ●13  Ⓢ Ⓥ Ⓟ ⓐ Ⓚ  Git-master       unix | 36: 0  26%`

The Zen of Python / Tim Peters

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Python Anti-Patterns

http://docs.quantifiedcode.com/python-anti-patterns/

Code with style

```
from ..api import Record

logger = get_task_logger(__name__)


@celery.task
def create_record(json, force=False):
    from invenio_ext.sqlalchemy import db
    try:
        return Record.create(json).get('recid')
    except exc.IntegrityError:
        if force:
            current_app.logger.warning(
```

```
❶  - 1.4k api.py  Python  •1 •3  Ⓢ Ⓥ Ⓟ ⓐ Ⓚ  Git-master    unix | 34:30  67%
Line  Col Level     ID      Message (Checker)
   0    1 warning   I001    found unsorted imports... (python-flake8)
   1    1 warning   D100    Missing docstring in public module... (python-flake8)
  34    1 warning   D103    Missing docstring in public function... (python-flake8)
  40   13 error     F821    undefined name 'current_app'... (python-flake8)

❷  % 304 *Flycheck errors* for buffer api.py  Flycheck errors    utf-8 | 3: 0  All
```

pep8 · pep257 · flake8 · isort

Sensible commit history


```
Jiri Kuncar          2015-06-24 16:04 global: preparation for new data model
         from .api import get_record
Esteban J. G. Gabancho 2014-06-04 18:03 JsonAlchemy: initial release
         merged_recid = None
Jiri Kuncar          2015-06-24 16:04 global: preparation for new data model
         # FIXME
         for val in get_record(recid).get("970__d", []):
Esteban J. G. Gabancho 2014-06-04 18:03 JsonAlchemy: initial release
             try:
                 merged_recid = int(val)
                 break
             except ValueError:
                 pass
```

```
● % 4.8k models.py │ Python │ •4 │ Blame Ⓢ Ⓨ Ⓟ ⓐ Ⓚ │ Git-master │ unix | 80:12  51
Commits in master
00e0a78 * models: version counter for concurrent update  Jiri Kuncar    4 months
90db4ab * global: support for jsonpatch                  Jiri Kuncar    4 months
a1dc0a1 * installation: disabled Invenio dependency      Jiri Kuncar    4 months
e59d151 * global: preparation for new data model         Jiri Kuncar    4 months
a685b1d * initial package separation                     Jiri Kuncar    4 months
490EdFa × linkbacks: module removal                      Jiri Kuncar    4 months
❷ % 13k *magit-log: invenio-records  Magit Log          utf-8 | 76:26  29%
```

Local kwalitee check 1 of 2

```
<kwalitee> (M 34310) ~/private/src/invenio-records> kwalitee check message master..upstream/pr/31
commit 2c5dfd523e60d2a29711fd56668361270843ac1e

    global: simplification of access right check

    * BETTER Uses cache containing permitted restricted collections for
      given user and cached list of restricted collections to perform access
      right check.

    Signed-off-by: Jiri Kuncar <jiri.kuncar@cern.ch>

Everything is OK.

<kwalitee> (M 34310) ~/private/src/invenio-records> |

 ●  * 505 *ansi-term*   Term  Ⓚ : char run                      utf-8 | 14:53  All
```
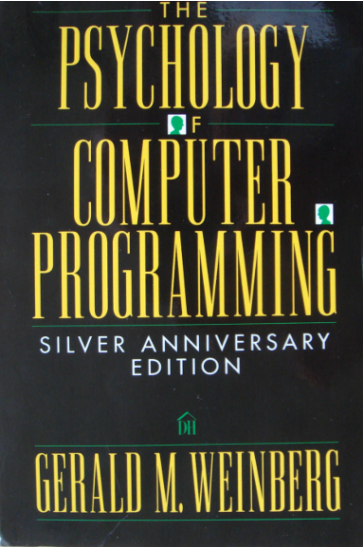
```
$ kwalitee check message <sha1a..sha1b>
```

Local kwalitee check 2 of 3

```
<kwalitee> (M:34310) ~/private/src/invenio-records> kwalitee check files master..upstream/pr/31
commit 2c5dfd523e60d2a29711fd56668361270843ac1e

    global: simplification of access right check

    * BETTER Uses cache containing permitted restricted collections for
      given user and cached list of restricted collections to perform access
      right check.

    Signed-off-by: Jiri Kuncar <jiri.kuncar@cern.ch>

invenio_records/access.py
22:1: F401 'warnings' imported but unused
34: D103 Missing docstring in public function
142: D202 No blank lines allowed after function docstring (found 1)

1  <kwalitee> (M:34310) ~/private/src/invenio-records>
```

```
 ⓘ  * 669 *ansi-term*  Term  Ⓚ : char run                              utf-8 | 16: 0  All
```

$ kwalitee check files <sha1a..sha1b>

Local kwalitee check 3 of 3

```
<kwalitee> (M 34310) ~/private/src/invenio-records> kwalitee prepare release ..upstream/pr/31
Improved features
-----------------

- Uses cache containing permitted restricted collections for given
  user and cached list of restricted collections to perform access
  right check.

<kwalitee> (M 34310) ~/private/src/invenio-records> |
```

```
* 335 *ansi-term*   Term  Ⓚ : char run                              utf-8 | 9:53  All
```

$ kwalitee prepare release <sha1a..sha1b>

TPoCP / Gerald Weinberg

Egoless Programming

1. Understand and accept that you will make mistakes.
2. You are not your code.
3. No matter how much "karate" you know, someone else will always know more.
4. Don't rewrite code without consultation.
5. Treat people who know less than you with respect, deference, and patience.
6. The only constant in the world is change.
7. The only true authority stems from knowledge, not from position.
8. Fight for what you believe, but gracefully accept defeat.
9. Don't be "the guy in the room".
10. Critique code instead of people.

Re-read CONTRIBUTING.rst

1. Before starting any wider-impact changes, submit a ticket and use a discussion channel.
2. Publish your code under GNU General Public License.
3. Choose good starting point to build your topical branch upon.
4. Make things easily configurable and reusable by others.
5. Create logically separate commits for logically separate things.
6. Use sensible commit messages and stamp them with QA and ticket directives.
7. Include test cases with the code.
8. Include documentation with the code.
9. Check the overall code kwalitee.
10. Send the code for review and integration.
11. Help improving overall development, deployment, and operational bandwidth.
12. Share thoughts, needs, requirements, solutions, "howto" recipes with others.

Step 4 of 6: Reviewing

Step 4 of 6: Reviewing

Ecce pull request!
Undocumented features
Still shivering

— after Akutagawa Ryunosuke (1892-1927)

Code reviews

Code coverage

Thinking forward

Reviewing tips

1. Every PR should preserve or increase code coverage.
   If it ain't green, it ain't finished.
2. Check it as a black box. Input, magic, output.
   If it ain't documented, it ain't finished.
3. Check it as a white box. Implementation details.
   If it ain't styled, it ain't finished.
4. Check it as a release news. Messages only.
   If it does not announce anything, it may not be finished.
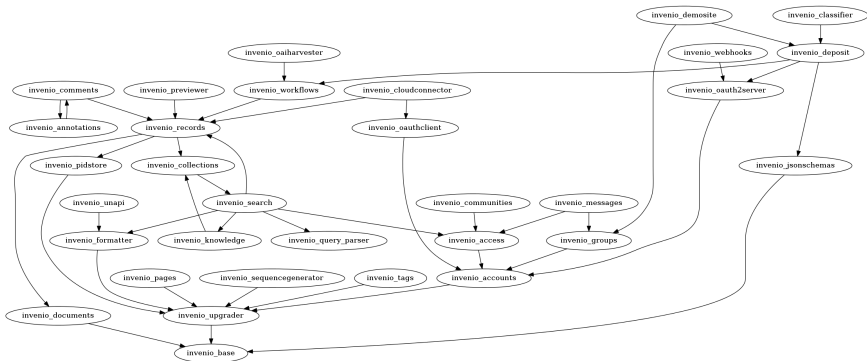
Step 5 of 6: Integrating

Step 5 of 6: Integrating

Covered with tests
Instantly I'd like to merge
In this dream of ours!

— after Ochi Etsujin (1656?-1739?)

Inter-module relations

Inter-service relations

Integrating tips

1. Check the signature karma.

   If it ain't signed, it ain't finished.

2. Check the counter-signature karma.

   If it ain't counter-signed, it ain't finished.
   - `Reviewed-by`
   - `:shipit:`
   - cross-team signatures

3. Check it from the helicopter.

   If it ain't green, it ain't finished.

   If it ain't understandable, it ain't documented.

4. Check its neighbourhoods.

   If it changes pre-existing tests, beware of compatibility.

   If it removes API-like functions, check outside usage.

Step 6 of 6: Releasing

Step 6 of 6: Releasing

The old repository
A tag jumps in
Plop

— after Matsuo Basho (1644-1694)
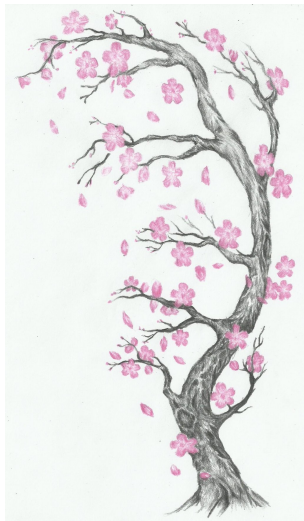
Structured release notes

Invenio v1.2.1 is released

Invenio v1.2.1 was released on May 21, 2015.

About

Invenio is a digital library framework enabling you to build your own
digital library or document repository on the web.

This stable release update is recommended to all Invenio sites using
v1.2.0 or previous releases.

Security fixes

- BibAuthorID:

    - Improves URL redirecting by properly quoting all URL parts, in order to better protect
      against possible XSS attacks.

- WebStyle:

    - Adds back the `HttpOnly` cookie attribute in order to better protect against potential XSS
      vulnerabilities. (#3064)

Structured release notes

commit label	release notes section
SECURITY	Security fixes
INCOMPATIBLE	Incompatible changes
NEW	New features
BETTER	Improved features
FIX	Bug fixes
NOTE	Notes
(AMENDS)	(amending past messages)
(missing)	(developers only)

Releasing tips

1. Cross-check CI green lights.
2. Cross-check Read The Docs documentation builds.
3. Cross-check demo site builds.
4. `check-manifest`
5. Check author list. Add newcomers.
6. Update I18N message catalogs.
7. Update version number. Generate release notes.
8. Push a pre-release to testpypi. Try a test install from there.
9. Tag it. Push it. Bump it.
10. Add release notes on GitHub. Tweet it. Post it.

Community

#IUGW2012

If you want to go fast, go alone.

If you want to go far, go together.

(after `projecthydra.org`)

Thanks for flying Invenio

Happy hacking

`inveniosoftware.org`