Invenio master

An Invenio tutorial based on the master branch
About this tutorial

This tutorial is dedicated to new developers who are starting to develop on Invenio master branch. Invenio is currently (Feb 2014) being rewritten using modern Python frameworks and packaging styles into the Invenio next (or pu) branch. Another tutorial is dedicated to explore that branch.

It is assumed the developer knows Python and has succeeded in installing Invenio Atlantis Demo site and has live installation similar to the one available at http://invenio-demo.cern.ch/
Invenio is a digital library, i.e. it is specialized in storing digital objects, preserve them and make them available. These objects can be anything: abstract information (e.g. about an institute) or real documents (e.g. the PDF representation of a scientific article).

The strong points of Invenio are its focus on metadata, i.e. information about the digital object itself.
Records

Take a look at: http://invenio-demo.cern.ch/record/93

That’s a record: it’s identified by an integer (93 in this case) and it’s about a digital object in this case a scientific article about: “A new PPN parameter to test Chern-Simons gravity”.

What you saw when you followed the last link is simply a detailed HTML representation of the record metadata. The real metadata are stored natively in a format called MARC.
Briefly MARC is a standard that Invenio adopted as metadata representation. Every piece of information is structured into “datafields” identified by a “tag” (3 digits) and 2 characters called “indicators” (ind1 and ind2). Each datafield is structured into several “subfields” which are identified by a “code” (1 character).

See for example: http://invenio-demo.cern.ch/record/93/export/xm

This is the XML serialization of the MARC metadata of the previously mentioned record. You will notice special fields called “controlfields” these are meta-metadata about the record itself (e.g. the modification timestamp, the integer identifier, etc.).
... Exercise

Look at http://invenio-demo.cern.ch/record/15/export/xm:

- Can you spot where the title of the article is stored?
- Can you identify all the authors?
- Can you guess what is the difference between datafield 100 and 700?

See: http://invenio-demo.cern.ch/help/admin/howto-marc
Other metadata formats

From MARC data it is possible to write a *mapping* towards other metadata formats:

E.g. if you have ever written your CV in LaTeX or you have written a scientific article you might know what BibTeX is:

http://invenio-demo.cern.ch/record/15/export/hx
Our 1st record

OK. It is now time to create our first record and have it digested by Invenio!
We’re Python developers, right? So let’s open “ipython” and create our 1st record with Pythonic tools.
Our 1st record

In [1]: from invenio.bibrecord import record_add_field, record_xml_output
In [2]: rec = {} ## Our record start empty
In [3]: record_add_field(rec, tag="100", subfields=[('a', 'Doe, John'), ('u', 'Never never land')]) ## We are just adding the first author!
1
In [4]: record_add_field(rec, tag="245", subfields=[('a', 'This is my first record')]) ## And here comes a dumb title
2
In [5]: record_add_field(rec, tag="980", subfields=[('a', 'ARTICLE')]) ## Let’s declare it as an article. We’ll see later what that means
3
Our 1st record

In [6]: rec
{'245': [[('a', 'This is my first record')], '', '', '', 2]}, '100': [[('a', 'Doe, John'), ('u', 'Never never land')], '', '', '', 1], '980': [[('a', 'ARTICLE')], '', '', '', 3]}

## This is how our MARC record is represented Pythonically into Invenio

## Let’s have the XML serialization

In [7]: xml = record_xml_output(rec)
Our 1st record

[8]: print xml
<record>
  <datafield tag="100" ind1=" " ind2=" ">
    <subfield code="a">Doe, John</subfield>
    <subfield code="u">Never never land</subfield>
  </datafield>
  <datafield tag="245" ind1=" " ind2=" ">
    <subfield code="a">This is my first record</subfield>
  </datafield>
  <datafield tag="980" ind1=" " ind2=" ">
    <subfield code="a">ARTICLE</subfield>
  </datafield>
</record>

## Our record has taken shape! Let’s save it into a file
In [9]: open("/tmp/1st.xml", "w").write(xml)
BibUpload is one of Invenio bibliographic tasks (a.k.a. bibtasks). This tool is the entry point for adding, manipulating or deleting records in Invenio.

- Check: bibupload --help to see how many things you can do with it.

Let’s ingest our 1st record!
$ bibupload --insert /tmp/1st.xml

Note: Invenio commands are typically executed with the same user of the Apache web server, so you might need to prefix the above command with:
$ sudo -u www-data /opt/invenio/bin/bibupload ...
(or apache instead of www-data)
If you had not yet started the bibliographic scheduler (a.k.a. bibsched) then nothing has yet happened to your record. It’s ingestion is simply scheduled and waiting to be performed.

$ bibsched

will open a the bibsched monitor interface. You should find a line corresponding to the bibupload task being scheduled. Just press “a” to switch the scheduler to automatic mode. (Press “h” to see all the commands.)
You will see some activity! Your task is going to be executed and will hopefully disappear.

- Just press “1” to move to the list of “done” tasks.
- Move to the line corresponding to your task.
- Press “l” to see the log of the task and actually see what happened.
- You will find a line similar to:
  2014-02-28 10:08:48 --> Record 123 DONE
- That is the ID of your brand new record. Yay!
Where is your record?

If 123 was the identifier of your record, just go to it from the web interface:

- Visit: http://.../record/123
- The system is prompting you for authentication... why is that so? Because Invenio wants to protect data. It doesn’t know (yet) what the record 123 is, so, in order to play safe it is protecting it.
- Go to: http://.../collection/Articles
- Can you find your record? ... No?
Webcoll

- A collection, in Invenio, is based on
  - a query on record metadata
  - the union of daughter collections
- When new records arrive, Invenio needs to attribute them to the correct collections. This happens through another bibliographic task called webcoll

$ webcoll$

- If you have left bibsched in automatic mode, the webcoll will be executed automatically.
- Go to the bibsched monitor and see if it’s still being executed or it has already finished
Where is your record?

- Can you now find your record in the Articles collection?
- Note that, on Invenio master, webcoll is responsible both for assigning records to collections and for creating collection splash pages.
- Now, let’s try to search for it:
  - Go to the home page on Atlantis Invenio and search for:
    - title:”This is my first record”
- Did your record appeared?
Booh! Not yet! And guess why? There is another bibliographic task that need to further elaborate the record

\$ \texttt{bibindex}

- This will populate search indexes so that words from your record are properly associated with it.
- Try again to search for it.
Where to go from here?

- Login as admin (no password) to explore all the administrative interfaces available in Invenio!

- **See under** `$ cdvirtualenv bin what actual tools exist and try to understand what they are for :)`

- Invenio devals are always there for helping!