



BibField

Record abstraction definition

Non-Marc21 formats

Virtual field infrastructure

Easy API to access record info

Esteban J. G. Gabancho
Invenio AAHEP Meeting 2012

BibField API

A screenshot of a terminal window titled "bash" at the top. The window has a menu bar with "File", "Edit", "View", "Bookmarks", "Settings", and "Help". The main area of the terminal shows the following Python code:

```
from invenio import bibrecord
from invenio.search_engine import get_record
rec = get_record(12)
bibrecord.record_get_field_value(rec, '245', ' ', ' ', 'a')
'Physics at the front-end of a neutrino factory...'
bibrecord.record_get_field_value(rec, '100', ' ', ' ', 'a')
'Mangano, M L'
bibrecord.record_get_field_values(rec, '700', ' ', ' ', 'a')
['Alekhin, S I',
 'Anselmino, M',
 'Ball, R D',
 ... ]
```

BibField API



A screenshot of a terminal window titled "bash" at the top. The window has a menu bar with "File", "Edit", "View", "Bookmarks", "Settings", and "Help". The main area of the terminal displays the following Python code:

```
from invenio.bibfield_engine import record_factory
record = record_factory(12)
record['title']
'Physics at the front-end of a neutrino factory...'
record['authors']
[{'ln':'Mangano', 'fn':'M L', ...}, ...]
record['authors[0]']
{'ln':'Mangano', 'fn':'M L', ...}
record['authors[0].ln']
'Mangano'
record['authors[1:]']
record['authors.ln']
```

BibField API



A screenshot of a terminal window titled '~ : bash'. The window contains the following text:

```
bibreCORD.record_get_field_value(rec, '245','','','a')
1000000 loops, best of 3: 1.29 us per loop
record['title']
1000000 loops, best of 3: 602 ns per loop
record['authors[0]']
100000 loops, best of 3: 18.5 us per loop
record['authors'][0]['ln']
1000000 loops, best of 3: 654 ns per loop
```

BibField internals: The configuration file



```
3 authors[0], creator:
4   creator:
5     @legacy(("100_a", "first author name", "authors[0].full_name"),
6       ("100_e", "authors[0].relator_name"),
7       ("100_h", "authors[0].CCID"),
8       ("100_i", "authors[0].INSPIRE_number"),
9       ("100_u", "first author affiliation", "authors[0].affiliation"))
10      marc, "100_", {'full_name':value['a'], 'fn':get_item splitted(value['a'], ',', 1), 'ln':get_item splitted(value['a'], ',', 0), 'relator_name':value['e']}
11      unimarc, "700[0,1]", {'full_name': "".join((value['a'], value['b']))} 'fn':value['b'], 'ln':value['a'], 'qualifier':value['c'], 'roman_numeral':value['d']
12 checker:
13   existence(0,1)
14   type('string')
15 documentation:
16   "Main Author"
17   @subfield fn: "First name"
18   @subfield ln: "Last name"
19
20 authors[n], contributor:
21   creator:
22     @legacy(("700_a", "additional author name", "authors[1:].full_name"),
23       ("700_u", "additional author affiliation", "authors[1:].affiliation"))
24     marc, "700_", {'full_name': value['a'], 'fn':get_item splitted(value['a'], ',', 1), 'ln':get_item splitted(value['a'], ',', 0), 'relator_name':value['e']}
25     unimarc, "70[1,2]_[0,1]", {'full_name': "".join((value['a'], value['b']))} 'fn':value['b'], 'ln':value['a'], 'qualifier':value['c'], 'roman_nu
26 checker:
27   existence(0,'n')
28   type('string')
29 documentation:
30   "Authors"
31
32 corporate_name[0]:
33   creator:
34     @legacy(("110_a", "corporate_name.name"),
35       ("110_b", "corporate_name.subordinate_unit"),
36       ("110_g", "corporate_name.collaboration"))
37     marc, "110_", {'name':value['a'], 'subordinate_unit':value['b'], 'collaboration':value['g']}
38     unimarc, "7100[,0,1,2]", {'name':value['a'], 'subdivision':value['b'], 'qualifier':value['c'], 'address':value['p'], 'authority_record_number':value['r']}
39 checker:
40   existence(0,1)
41
42 corporate_name[n]:
43   creator:
44     @legacy(("710_a", "corporate_name.name"),
45       ("710_b", "corporate_name.subordinate_unit"),
46       ("710_g", "collaboration", "corporate_name.collaboration"))
47     marc, "710_", {'name':value['a'], 'subordinate_unit':value['b'], 'collaboration':value['g']}
48     unimarc, "71[1,2]0[,0,1,2]", {'name':value['a'], 'subdivision':value['b'], 'qualifier':value['c'], 'address':value['p'], 'authority_record_num
49 checker:
50   existence(0,'n')
51 ...
```

BibField internals: The configuration file



A screenshot of a vim editor window displaying configuration code for BibField. The code defines a configuration object with sections for title, creator, checker, documentation, and a subfield mapping. The vim status bar at the bottom shows the file is named "vim" and has 1,1 lines.

```
title:
creator:
    marc, "245__", { 'title':value['a'], ... }
    unimarc, "2001__", { 'title':value['a'], ..... }
checker:
    existence(0,1)
    type('string')
documentation:
    "Main Title"
@subfield title: "Actual title"
```

BibField internals: The configuration file



A screenshot of a vim editor window displaying configuration code for BibField. The code defines two main sections: 'authors[0], creator:' and 'authors[n], contributor:'. Each section contains a 'creator' field with a marc value and a 'get_item_splitter' function call, and a 'checker' field with an 'existence' and 'type' check. The 'documentation' field provides a description of the field, and an '@subfield fn' entry specifies the first name. The vim status bar at the bottom shows the file is named '[Sin nombre]' and has 1,1 lines.

```
~ : vim
File Edit View Bookmarks Settings Help
~ authors[0], creator:
  creator:
    marc,"100__",__fn__:get_item_splitter(value['a'],' ',' ',1), ...
  checker:
    existence(0,1)
    type('string')
  documentation:
    "Main Author"
    @subfield fn: "First name"

authors[n], contributor:
  creator:
    marc,"700__",__fn__:get_item_splitter(value['a'],' ',' ',1), ...
  checker:
    existence(0,'n')

[Sin nombre] [+]
-- INSERTAR --
~ : vim
1,1 Todo
```

BibField internals: The configuration file



A screenshot of a vim editor window displaying configuration code for BibField. The code defines two fields: n_authors and _copies. The n_authors field has a derived value calculated from the length of the authors list, and a checker function that ensures it exists and is a number. The _copies field has a calculated value based on the number of copies for a specific record ID, and a similar checker function. The vim status bar at the bottom shows the file is named 'n_authors' and '_copies' respectively, and indicates the current line is 1, column 1.

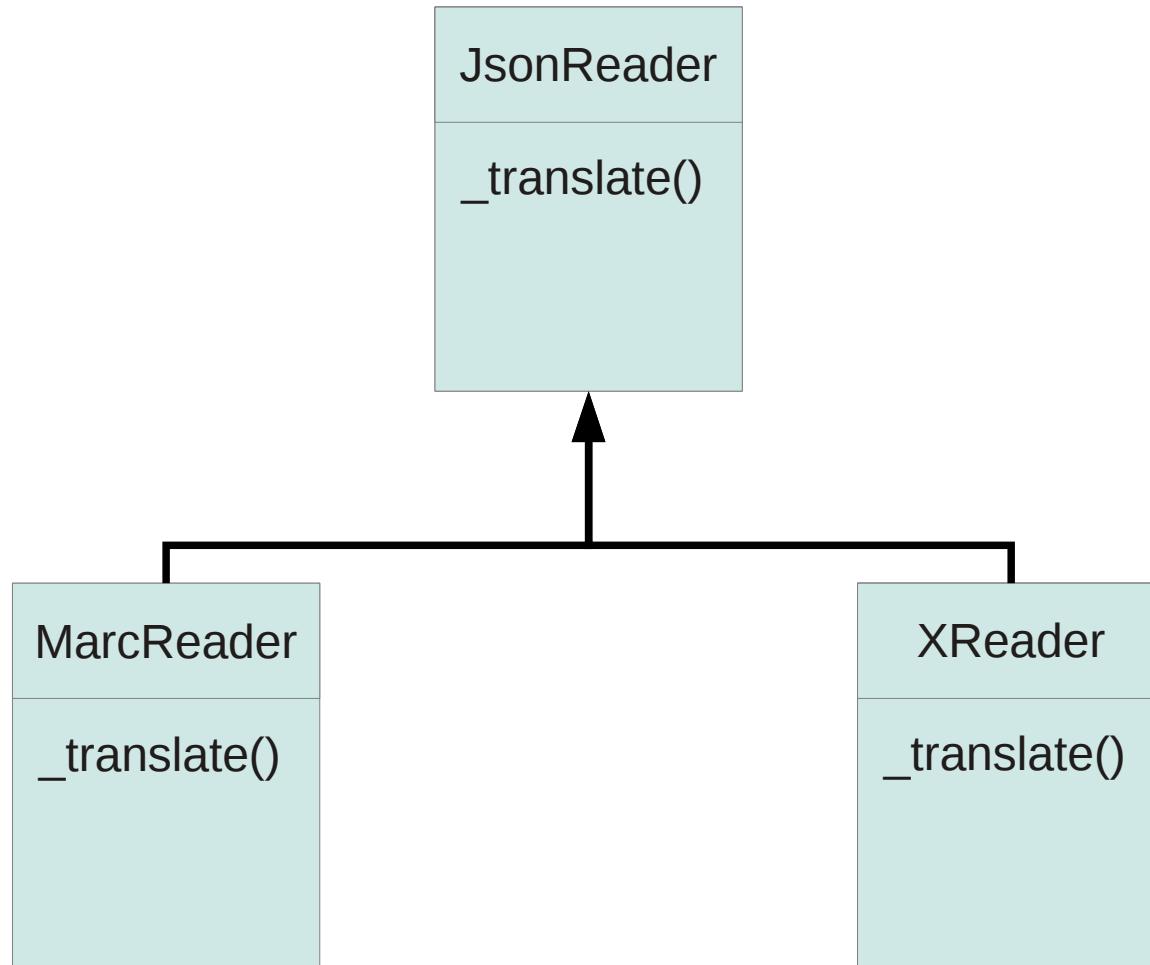
```
n_authors:
    derived :
        @dependsOn(['authors'])
        len(self['authors'])

checker:
    existence(0, 1)
    type('nums')

_copies:
    calculated:
        @dependsOn(['recid', 'collection.primary'])
        @onlyIf(['BOOK' in self['collection.primary'], ])
        number_of_copies(self['recid'])

checker:
    existence(0, 1)
    type('nums')
```

BibField internals: Extending to non-MARC



What's next



- Regular expressions inside configuration file
- Improve the translation phase
- Improve the checker phase
- Update invenio modules to use BibField

THE END