NADRE tutorial Documentation

Release 1.0

Mario Torrisi

Oct 18, 2018

Contents

This tutorial shows you how to programmatically interact with the Invenio based NADRE Repository.

The tutorial cosists of the following two sections:

- the first one shows some examples on how to search object stored in the digital repository building HTML query;
- the second one show how programmatically submit records to the repository.

The presentation and the video recording of this tutorial are available on NADRE repository and have the following DOIs:

- Presentation 10.20372/ethernet:1533718023.48
- Video tutorial 10.20372/ethernet:1533719889.38

Contents:

CHAPTER 1

Search Engine API

Table of content

1.1 Introduction

Exploiting search Engine API of an Invenio based OAR allows you to create, for example, a new front end for your repository. To search or browse records in the OAR you have to just create HTML queries specifing the list of parameters, that allow users to find the records in which they are interested in, so you can present the results in a nicer interface.

Following the syntax should be used to create the queries:

Syntax

```
GET /search?p=...&of=...&ot=...&jrec=...&rg=...
```

where p is pattern (i.e. your query)

of is output format (e.g. *xm* for MARCXML)

ot is output tags (e.g. ‘‘ to get all fields, *100* to get titles only)

jrec is jump to record ID (e.g. 1 for first hit)

rg is records-in-groups-of (e.g. 10 hits per page)

You can use other parameters as well; the list above mentions the most useful one. For full documentation on these and the other */search* URL parameters, please see section 3.1 of Search Engine API[1].

1.2 XML API

To get results of your queries in XML format you have to set output format parameter to **xm** (`of=xm`). The OAR return the results in MARCXML[2].

1.2.1 Get records

So, to get the first 10 records stored in the OAR in XML format you can use the following query

You can change `jrec` parameter to implement records pagination, for example the query:

returns the next 10 records starting from the eleventh, or use the following

to get records from 21st to 30th, and so on. . .

Warning: Do not set *rg* too high; there is a server-wide safety limit on it.

1.2.2 Look for patterns in fields

To get, for example, the first 10 records that contains the string *‘Hackfest’*, you can use the `p` parameter to specify the pattern you are looking for and `f` parameter to specify the field in which search the patter. See the query below:

Where **p** is pattern (e.g. your query)

f is field to search within (e.g. “title”, “authors”..)

If you want to get, for example, the first 10 records in *‘PRESENTATIONSNADRE’* collection that contains *‘NADRE’* in keyword, you can use:

1.2.3 Filter records and outputs

To get all records uploaded from a given date (e.g. 2018-01-01) to another given date (e.g. 2018-02-22), you can issue:

Where **d1** is the first date in *YYYY-mm-dd* format

d2 is the first date in *YYYY-mm-dd* format

[1] http://nadre.ethernet.edu.et/help/hacking/search-engine-api
[2] http://nadre.ethernet.edu.et/help/admin/howto-marc

1.3 JSON API

Internally, Invenio records are represented in JSON. You can ask for JSON output format, simply, setting `of` to `recjson` (`of=recjson`).

Before proceeding...

You need to have some useful tools used in the rest of this tutorial:

- `curl` a tool to transfer data from or to a server link;
- `jq` a lightweight and flexible command-line JSON processor link.

Note: If you are not on a **NIX* based system, you can use Postman and import this collection `files/postman_collection.json` to perform the queries.

The following are the same example saw In XML API, but this time results are in JSON format. Just copy the command into shell session and see the outputs.

1.3.1 Get records

Get first ten records

```
curl -X GET \
"http://nadre.ethernet.edu.et/search?jrec=1&rg=10&of=recjson" \
| jq .
```

Records from eleventh to twentyth

```
curl -X GET \
"http://nadre.ethernet.edu.et/search?jrec=11&rg=10&of=recjson" \
| jq .
```

From 21st to 30th

```
curl -X GET \
"http://nadre.ethernet.edu.et/search?jrec=21&rg=10&of=recjson" \
| jq .
```

1.3.2 Look for patterns in fields

Get the first 10 records that contains the string “Hackfest” in the title

```
curl -X GET \
'http://nadre.ethernet.edu.et/search?p=Hackfest&f=title&jrec=0&rg=10&of=recjson' \
| jq .
```

Get the first 10 records in ‘PRESENTATIONSNADRE’ collection that contains ‘NADRE’ in keyword

```
curl -X GET \
'http://nadre.ethernet.edu.et/search?p1=collection:PRESENTATIONSNADRE+keyword:NADRE&
↪of=recjson&jrec=1&rg=10' \
| jq .
```

1.3.3 Filter records and outputs

Get all records uploaded from a given date (e.g. 2018-01-01) to another given date (e.g. 2018-02-22)

```
curl -X GET \
'http://nadre.ethernet.edu.et/search?of=recjson&d1=2018-01-01&d2=2018-02-22' \
| jq .
```

Get only the abstract, title and authors of resources

```
curl -X GET \
'http://nadre.ethernet.edu.et/search?of=recjson&ot=abstract,title,authors' \
| jq .
```

CHAPTER 2

Submit

Table of content

2.1 Introduction

This section will show you how to upload records to the NADRE repository through `curl` command and some sample `php` script.

All the examples are colleted in the submit folder, you can find them in the github repository (link). This folder consists of four subfolders:

1. `/asset` folder contains the sample objects used in the examples;
2. `/curl` folder consists of some `.md` files that have the command to perform to upload file to the OAR repository and the expected output;
3. `/php` folder has a set of simple php scripts to upload files to NADRE repository;
4. `/xml` folder contains the MARCXML files that describe the resources are going to be uploaded.

2.2 cURL

The cURL example are stored in the github repository curl subfolder. It has several `<digital-object>-submission-to-OAR.md` files with curl command to upload image, dataset, publication, etc. and the expected output repository will return. Then it contains also a Pyhton script (`request_doi.py`) to generate a DOI based on the system timestamp, is mandatory, before to upload any digital object to the repository, performing this script to reserve a valid DOI and replace it in the corresponding `<digital-object>-submission-to-OAR.xml` inside the `/xml` folder.

2.2.1 Image upload

From the downloaded github repository folder, change to the `/submit/curl/` sub-folders with:

```
nadre-tutorial-master$ cd submit/curl
```

To upload an image, for example, to the NADRE repository, you can refer to the `image-submission-to-OAR.md` file shown below.

~~~
## COMMAND
```bash
curl -T ../xml/image-submission-to-OAR.xml http://nadre.ethernet.edu.et/batchuploader/
↪robotupload/insert -A invenio_webupload -H "Content-Type: application/marcxml+xml"
~~~
EXPECTED OUTPUT
```bash
[INFO] bibupload batchupload --insert /opt/invenio/var/tmp-shared/batchupload_
↪20161122(...)
```
```

That contains the curl command to perform and the expected output.

Step 1: Create new DOI

First of all, you have to create a new DOI executing the `request_doi.py` script, as follows:

```
nadre-tutorial-master/submit/curl$ python request_doi.py
```

Step 2: Edit XML file

Now edit the `../xml/image-submission-to-OAR.xml` file with your favourite editor. As example:

1. edit the placeholder in the `tag="024"` with the DOI generated above

```
  <record xmlns="http://www.loc.gov/MARC21/slim">
    <datafield tag="024" ind1="7" ind2=" "> <!-- DIGITAL OBJECT IDENTIFIER -->
      <subfield code="a">replace-with-valid-DOI</subfield>
      <subfield code="2">DOI</subfield>
    </datafield>
```

2. edit the publication date in the `tag="260"`

```
    <datafield tag="260" ind1=" " ind2=" "> <!-- PUBLICATION, DISTRIBUTION, ETC. -->
      <subfield code="c">2016-11-22</subfield> <!-- c: Date of publication -->
    </datafield>
```
```

3. **the image main author in the `tag="100"` (use the `tag="700"` for other authors), specifing the**
    - Author name in `code="a"`
    - Affiliation in `code="v"`
    - Country in `code="w"`
    - ORCID in `code="j"`

```
    <datafield tag="100" ind1=" " ind2=" "> <!-- MAIN ENTRY PERSONAL NAME -->
      <subfield code="a">John Smith</subfield>
      <subfield code="v">University of City</subfield>
      <subfield code="w">Italy</subfield>
      <subfield code="j">0000-0002-5532-8201</subfield>
    </datafield>
```

3. then specify the publication keywords in the `tag="653"`, you can create several `tag="653"` as many as keywords you want specify

```
    <datafield tag="653" ind1="1" ind2=" "> <!-- KEYWORDS -->
      <subfield code="a">Sci-GaIA</subfield>
    </datafield>
    <datafield tag="653" ind1="1" ind2=" "> <!-- KEYWORDS -->
      <subfield code="a">Science Gateway</subfield>
    </datafield>
```

4. another important filed you have to edit is the `tag="FFT"` that represents where the resource can be found

```
    <datafield tag="FFT" ind1="" ind2=" ">
      <subfield code="a">http://grid.ct.infn.it/hackfest/example-image.png</subfield>
    </datafield>
```

Note: Have a look at this link to know more about MARCXML tag meaning.

Step 3: Submit record

Once you completed to edit the xml file, perform the following command to actually upload the file:

```
curl -T ../xml/image-submission-to-OAR.xml \
http://nadre.ethernet.edu.et/batchuploader/robotupload/insert \
-A invenio_webupload -H "Content-Type: application/marcxml+xml"
```

If everything's went well it returns something like:

```
[INFO] bibupload batchupload --insert /opt/invenio/var/tmp-shared/batchupload_
↪20161122(...)
```

an email will inform NADRE repository administrators of your your new submission then, after they review it, will actually publish the new record on the OAR.

2.3 PHP

This section shows you how to upload file to NADRE repository through simple PHP scripts. PHP examples are stored in the github repository `/php` here. It contains several `<digital-object>-submission-to-OAR.php` files with demostrative php code, that uses the php Client URL library to upload resources. As we saw in the previous section, is mandatory to generate a DOI, to do so you can use `request_doi.py` script in `/curl` folder, while the xml filer that describe the resources are located in the `/xml` one.

2.3.1 Presentation upload

From the downloaded github repository folder, change to the `/submit/php/` sub-folders with:

```
nadre-tutorial-master$ cd submit/php
```

Now we will see how to upload a presentation, to the NADRE repository, so you can refer to the `presentation-submission-to-OAR.php` file shown below.

```php
<?php
if (count($argv) <= 1) {
  exit("API endpoint miss, could not continue.\n");
}
$url_path_str = 'http://'.$argv[1].'/batchuploader/robotupload/insert';
$file_path_str = '../xml/presentation-submission-to-OAR.xml';

$headers = [
    'Content-Type: application/marcxml+xml',
    'User-Agent: invenio_webupload'
];
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, ''.$url_path_str.'');
curl_setopt($ch, CURLOPT_PUT, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
$fh_res = fopen($file_path_str, 'r');
curl_setopt($ch, CURLOPT_INFILE, $fh_res);
curl_setopt($ch, CURLOPT_INFILESIZE, filesize($file_path_str));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$curl_response_res = curl_exec ($ch);
fclose($fh_res);
echo $curl_response_res;
?>
```

As you can see it accept a parameter that is the repository hostname and will notify the user if no hostname is specified.

Step 1: Create new DOI

See *Step 1: Create new DOI*

Step 2: Edit XML file

Now edit the `../xml/presentation-submission-to-OAR.xml` file with your favourite editor. See *Step 2: Edit XML file* to edit the MARCXML tags as you need

Step 3: Submit record

Note: To actually submit the new record we will use PHP form the command line interface, you can install it using the packaged distribution of php-cli for ypu operating system.

As example for an Ubuntu based machine you can install php-cli with the following command

```
sudo apt-get install php5-cli
```

Once you finished to edit the XML file you can submit the new presentation using the following command

```
php presentation-submission-to-OAR.php nadre.ethernet.edu.et
```

The output you'll get, if everything went well, is the same you saw in previous section and at the same way the administrator will receive an email to notify that a new record have to be revised to be published.