

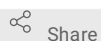


Jun 15, 2021

# Parsing of OAI\_DC metadata in OpenRefine from OJS articles

Alessandra Moi<sup>1</sup>, carlo.bianchini<sup>1</sup>, Andrea Marchitelli<sup>1</sup><sup>1</sup>ATCULT

1 Works for me



Share

[dx.doi.org/10.17504/protocols.io.bvmin44e](https://dx.doi.org/10.17504/protocols.io.bvmin44e)

wikidata and libraries

Alessandra Moi  
ATCULT

## ABSTRACT

### Abstract

Protocols for parsing metadata in OAI\_DC format of articles harvested from Open Journal Systems (OJS) in order to create entities in WikiData

## DOI

[dx.doi.org/10.17504/protocols.io.bvmin44e](https://dx.doi.org/10.17504/protocols.io.bvmin44e)

## PROTOCOL CITATION

Alessandra Moi, carlo.bianchini, Andrea Marchitelli 2021. Parsing of OAI\_DC metadata in OpenRefine from OJS articles. **protocols.io**  
<https://dx.doi.org/10.17504/protocols.io.bvmin44e>

## LICENSE

This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

## CREATED

Jun 08, 2021

## LAST MODIFIED

Jun 15, 2021

## PROTOCOL INTEGER ID

50570

### Attività preliminari

1 Import the OAI identifiers and generate the baseurl list:

1.1 Import the identifiers following the instructions in the protocol

<https://www.protocols.io/view/come-importare-identificatori-oai-pmh-in-openrefin-bp7mmrk6>

1.2 From the list of imported identifiers, generate the list of baseurl for harvesting the metadata of each item according to the indications given in the protocol "[Create baseurl per harvesting di repository OAI con OpenRefine](#)"

N.B. For OAI\_DC metadata collection, the metadataPrefix in the url must be=oai\_dc

2



OpenRefine will create a column containing XML text with the OAI\_DC metadata of the article

Test the operation of one of the generated URLs by clicking on the link and verifying that it opens a page with OAI\_CD metadata in XML format.

Es. [https://aibstudi.aib.it/oai?verb=GetRecord&metadataPrefix=oai\\_dc&identifier=oai:ajs.riviste.aib.it:article/11366](https://aibstudi.aib.it/oai?verb=GetRecord&metadataPrefix=oai_dc&identifier=oai:ajs.riviste.aib.it:article/11366)

#### Harvesting dei metadati

- 3 Click on the metadataURL column , select Edit Column > Add Column by retrieving from URLs, calling the new column metadata\_oai.

Once you have the metadata value for all items, you can move on to parsing

#### Parsing dei dati

- 4 Metadata Parsing

##### 4.1 Publishing data

```
value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')[0].select('dc|source')[0].ownText()
```

The values extracted with the previous grel command usually contain the Journal title (P1433), the Volume, Issue & publication date (P478, P433, P577) and the Pages from/to (P304) in a unique string

To extract these single values, divide using the command Edit Column > Split and choosing the appropriate separator (e.g.: ';')

##### *Total number of pages (P1104)*

From the column obtained by the command Split, derive two more columns (PagStr and PagFin) with the value of the starting page and the value of the ending page.

Create a new column "NumTotPag (P1104)" based on the PagFin column with the grel command: `toNumber(value-cells['PagStr'].value) + 1`

##### 4.2 DOI (P356)

```
value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')[0].select('dc|identifier')[1].ownText()
```

##### 4.3 Title (P1476)

```
value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')[0].select('dc|title')[0].ownText()
```

## 4.4 Authors name and related info

### *Authors data*

```
forEach(value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')  
[0].select('dc|creator'),v,v.ownText()).join("|")
```

### *Author# inverted form*

Once you have the values of all authors in the column "Authors data", derive one column for each name author ("Author#1 inverted form", "Author#2 inverted form", etc...) using the command Edit Column > Split and choosing the separator |

### *First name Author# (P735) - Surname Author# (P734)*

Once you have the values in the column "Author# inverted form", derive the two columns "First name Author# (P735)" and "Surname Author# (P734)" using the command Edit Column > Split and choosing the separator ,

### *Author# (P50) direct form*

Create the new column with the grel command:

```
cells['First name Author# (P735)'].value + ' ' + cells['Surname Author# (P734)'].value
```

## 4.5 Link

### *URL (P2699)*

```
value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')[0].select('dc|identifier')  
[0].ownText()
```

### *PDF Full text (P953)*

```
value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')[0].select('dc|relation')  
[0].ownText()
```

## 4.6 Keyword, per language

Once you have the values in a single cell, divide using the command Edit Column > Split and choosing the appropriate separator (e.g.: '|')

### *KW ita (P921)*

```
value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')  
[0].select('dc|subject[xml:lang="it-IT"]')[0].ownText()
```

### *KW eng (P921)*

```
value.parseXml().select('record')[0].select('metadata')[0].select('oai_dc|dc')  
[0].select('dc|subject[xml:lang="en-US"]')[0].ownText()
```