



# NSR Workflow

Benjamin Shu  
National Nuclear Data Center

October 30th, 2025



# Effects of Current Events

- Budget constraints
  - No contractors for NSR as of April 2025
  - Reduced ability to add/update entries

# Effects of Current Events

- Budget constraints
  - No contractors for NSR as of April 2025
  - Reduced ability to add/update entries
- Shifting focus towards AI-readiness
  - Like it or not, generative AI is here to stay

# Effects of Current Events

- Budget constraints
  - No contractors for NSR as of April 2025
  - Reduced ability to add/update entries
- Shifting focus towards AI-readiness
  - Like it or not, generative AI is here to stay
- **NSR's format, data model, and workflow need to adapt**
  - Updating the workflow gives us opportunities to leverage AI
  - Updating the data model makes NSR more AI-friendly

# Exchange Format

- *Not used anywhere other than NSR*
  - Requires custom parsing code to read
  - Makes processing needlessly difficult

<KEYNO	>	2023WIZZ	Unique ID
<HISTORY	>	A20230605	Entry/modification dates
<CODEN	>	CONF Sacramento...	Source type and information
<REFERENCE>		Proc.15th.Intern...	Citation information
<AUTHORS	>	D.Wiarda, ...	List of authors
<TITLE	>	Modernization...	Formatted title
<KEYWORDS>		[N/A]	Sentences about content
<SELECTRS>		[N/A]	Structured search keys

# Exchange Format (contd.)

- Still uses "fake LaTeX" for Greek/math symbols
  - *Not used by any other database*
  - Requires further translation after reading exchange format

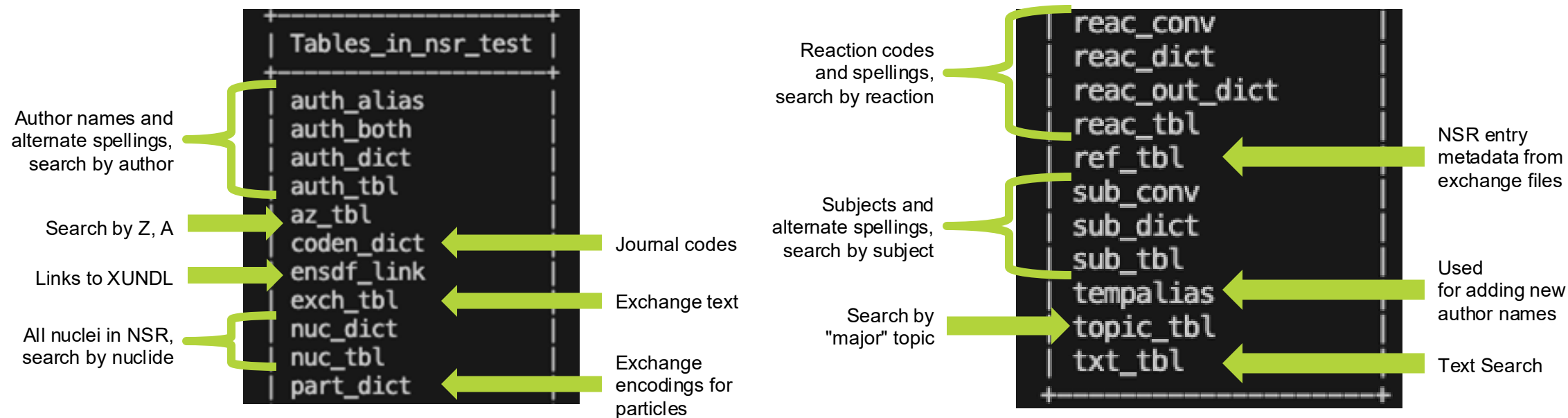
```
<KEYWORDS>RADIOACTIVITY {+37},{+38}Al(|b{+-}),(|b{+-}n)[from {+48}Ca({+9}Be,X),E
=140 MeV/nucleon]; measured E|g, I|g, I|b, |g|g-coin, |b|g-coin; deduced T{-1/2
}. {+37}Si; deduced levels, J, |p, T{-1/2} of isomeric states at 68- and 156-keV
, B(M1), matrix elements for low-lying ground-state M1 transitions. {+36},{+38}S
i; deduced E|g, I|g, transition intensities. Comparison to other experimental da
ta and shell model calculations using the SDPF-MU, SDPF-U-SI, and FSU interactio
ns. CeBr{-3} based implantation detector with position-sensitive photomultiplier
tube surrounded by 16 segmented Ge detectors (SeGa) and 15 LaBr{-3} detectors a
t Coupled Cyclotron Facility (NSCL, MSU)
```

<https://www.nndc.bnl.gov/nsr/?search-type=keynumber&key=2023OG04>



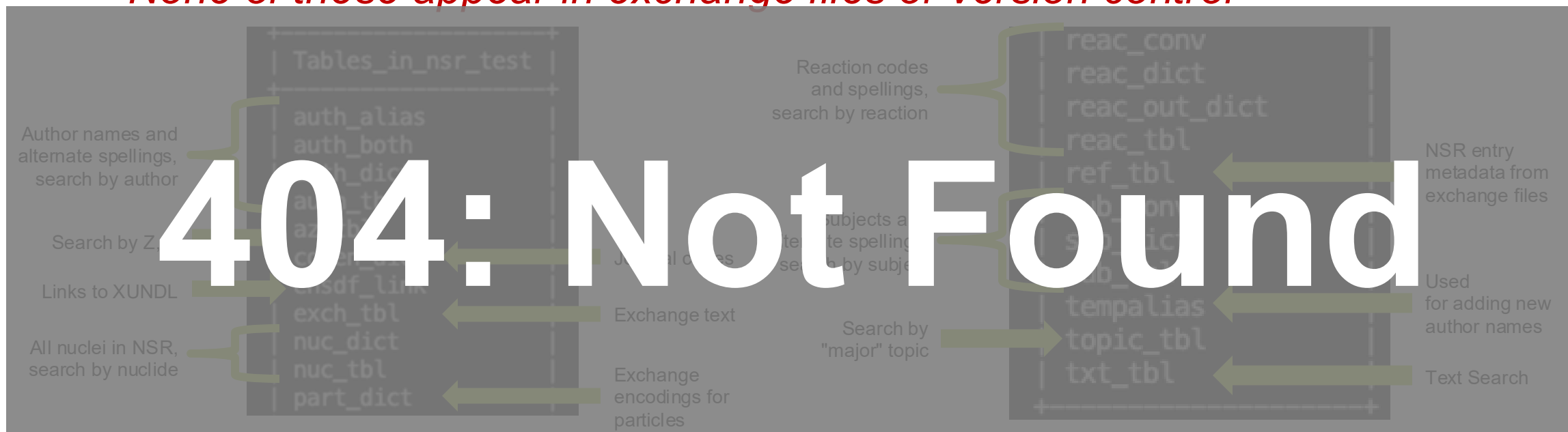
# Exchange "Data Model"

- NSR's exchange format *does not store all of its data*
  - Searching requires SQL tables of dictionaries and links



# Exchange "Data Model"

- NSR's exchange format *does not store all of its data*
  - Searching requires SQL tables of dictionaries and links
  - *None of these appear in exchange files or version control*





# A Better Data Model...

- ...would define *every* database component using consistent structures and formats
  - NSR's current model (exchange + SQL tables) does not do this!

# A Better Data Model...

- ...would define *every* database component using consistent structures and formats
  - NSR's current model (exchange + SQL tables) does not do this!
- Changing the data model:
  - Enables translation while preserving original data
  - Greatly simplifies parsing
  - Makes NSR easier for AI to ingest

# A Data Model for NSR

- How do we build an NSR data model?



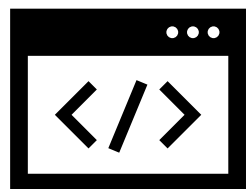
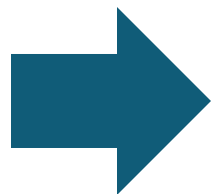
NSR 80-column  
text files

# A Data Model for NSR

- How do we build an NSR data model?
  - *Separate data from its original format*



NSR 80-column  
text files



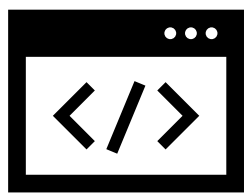
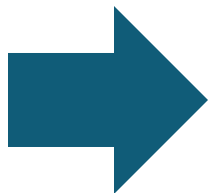
NSR concepts  
defined using  
strings, enums,  
numbers, etc.

# A Data Model for NSR

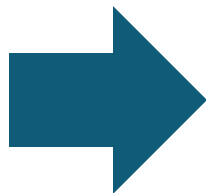
- How do we build an NSR data model?
  - *Separate data from its original format*
    - Enables translation into other formats (like JSON)



NSR 80-column  
text files



NSR concepts  
defined using  
strings, enums,  
numbers, etc.



Executable code  
(Java, Python) for  
reading/writing  
NSR data

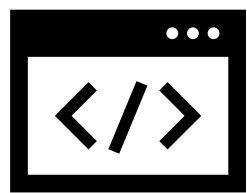
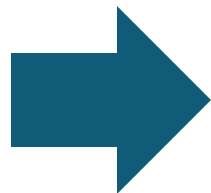
# A Data Model for NSR

- How do we build an NSR data model?
  - *Separate data from its original format*
    - Enables translation into other formats (like JSON)
    - Also enables **Object-Relational Mapping (ORM)**

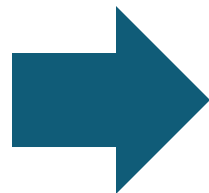
**Note:** We already have a copy of NSR in JSON!



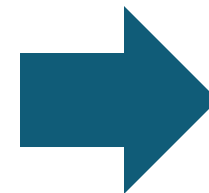
NSR 80-column  
text files



NSR concepts  
defined using  
strings, enums,  
numbers, etc.



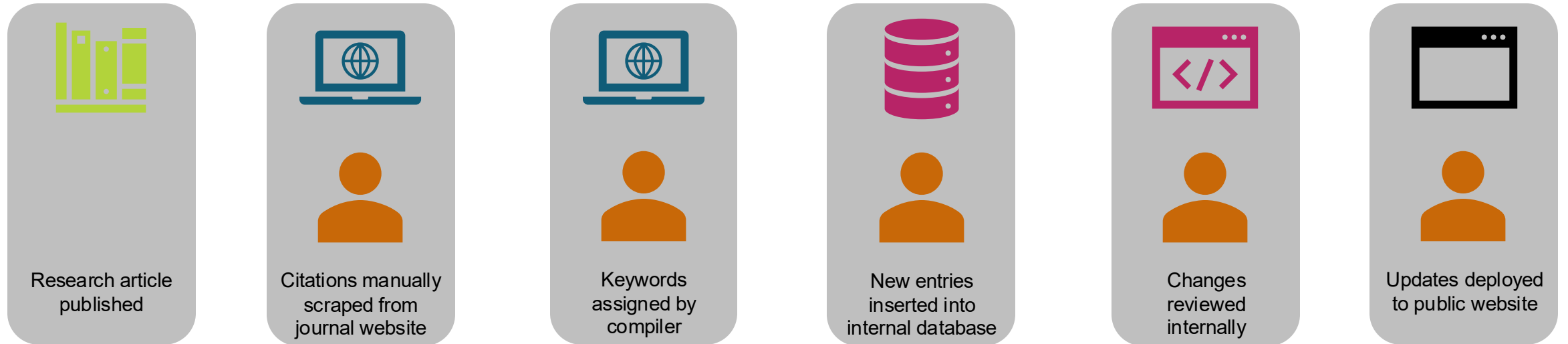
Executable code  
(Java, Python) for  
reading/writing  
NSR data



Ability to recreate  
NSR database in  
different formats



# Previous Workflow



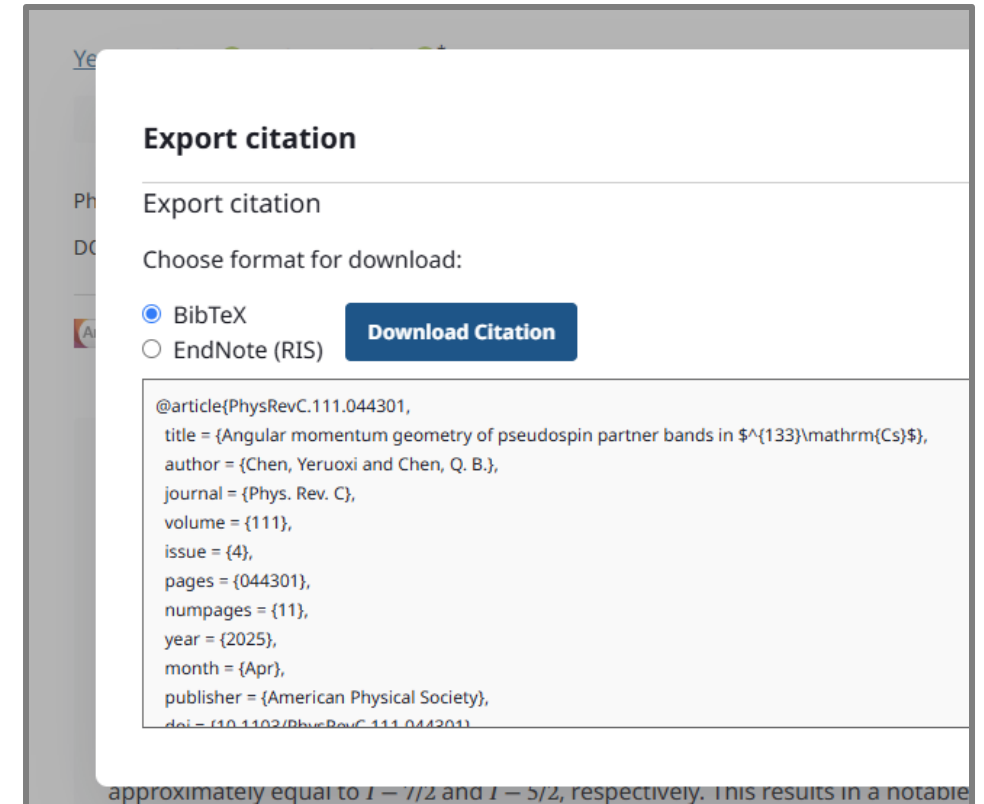
- All steps handled by human beings
  - See previous point about having no contractors

# Manual Web Scrapping

- Article citations formerly obtained by people searching in websites

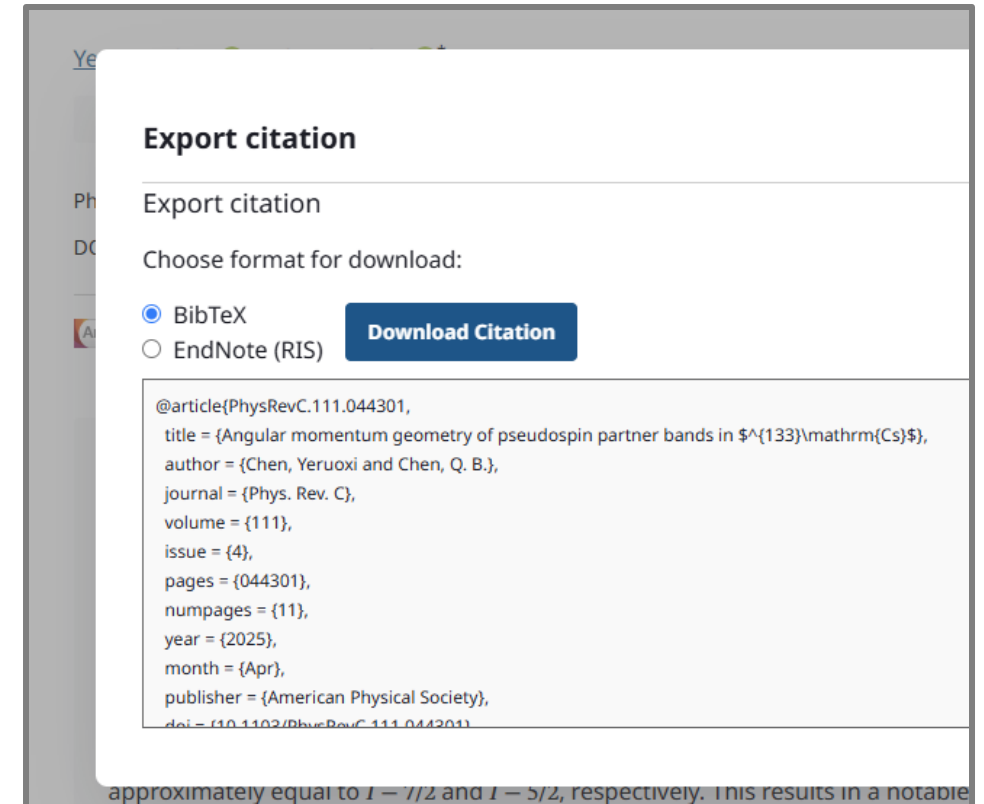
# Manual Web Scrapping

- Article citations formerly obtained by people searching in websites
  - **Ex. Physical Review C (PRC)**
    - Click on one article, download BibTeX
    - Rinse and repeat for an entire issue
    - Vol 11, Issue 3 – 85 articles added

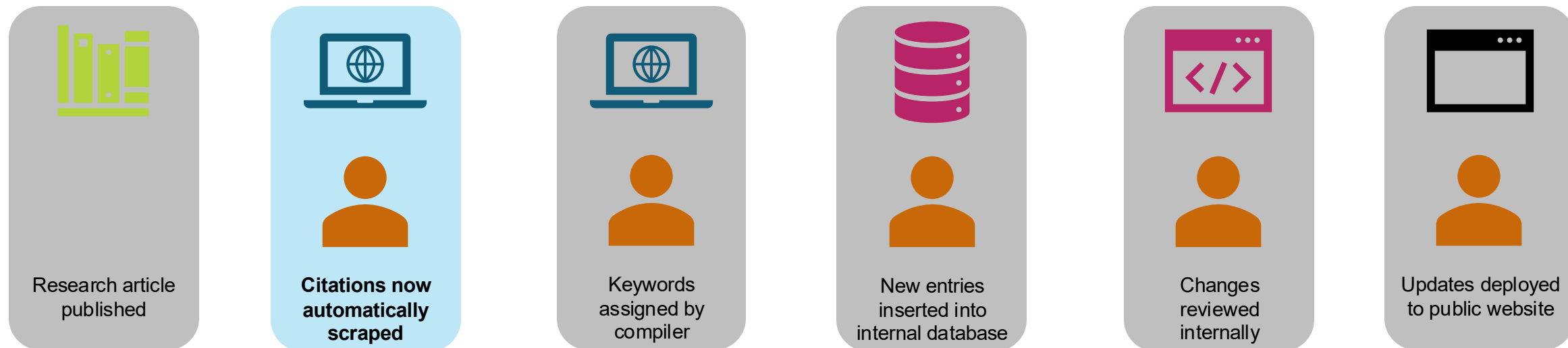


# Manual Web Scrapping

- Article citations formerly obtained by people searching in websites
  - **Ex. Physical Review C (PRC)**
    - Click on one article, download BibTeX
    - Rinse and repeat for an entire issue
    - Vol 11, Issue 3 – 85 articles added
- **This can be automated**  
(and we already have a scraper!)



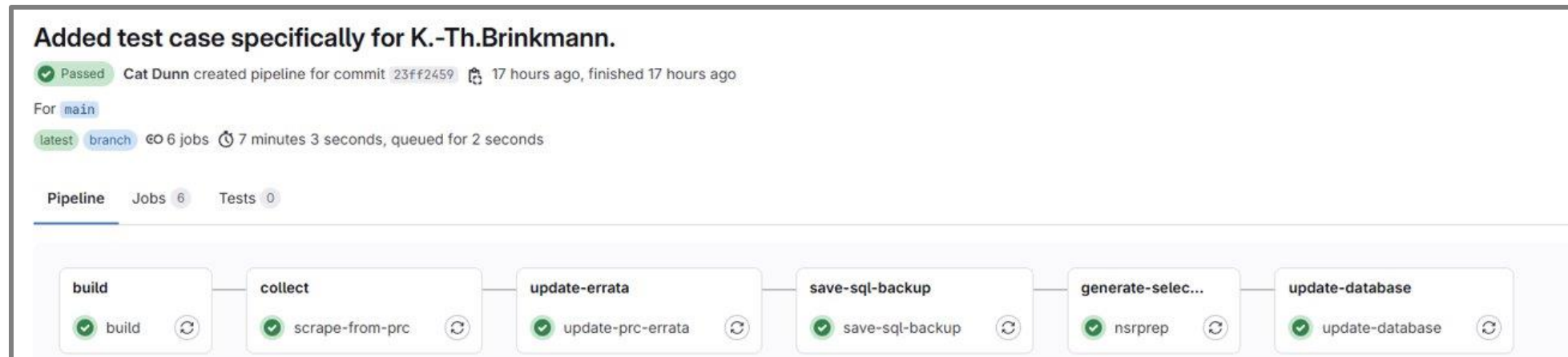
# Workflow With Automation



- *Automation saves time/effort that can be used elsewhere*
  - What other steps can we automate?

# Proof of Concept Workflow

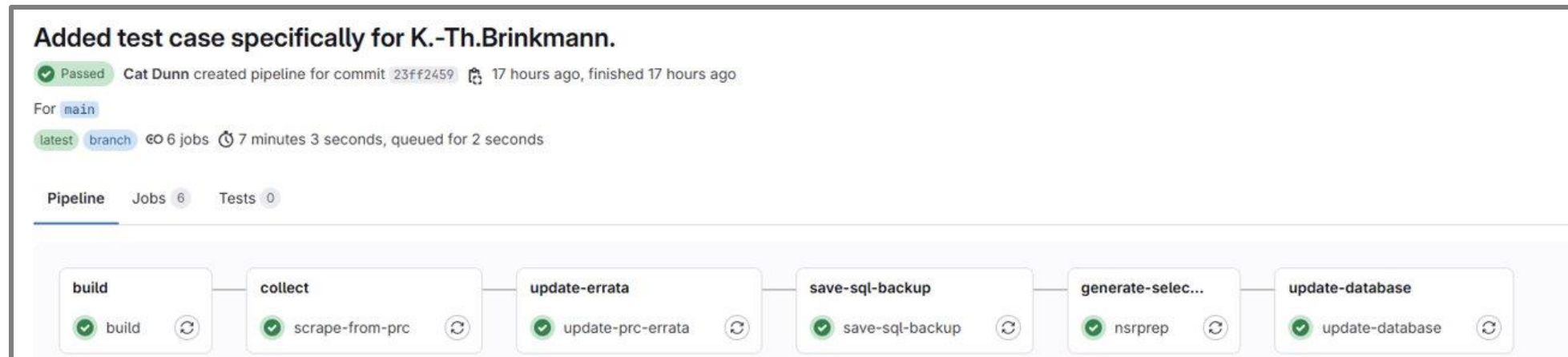
- **update-nsr-sql: we have a pipeline!** (based on legacy workflow)





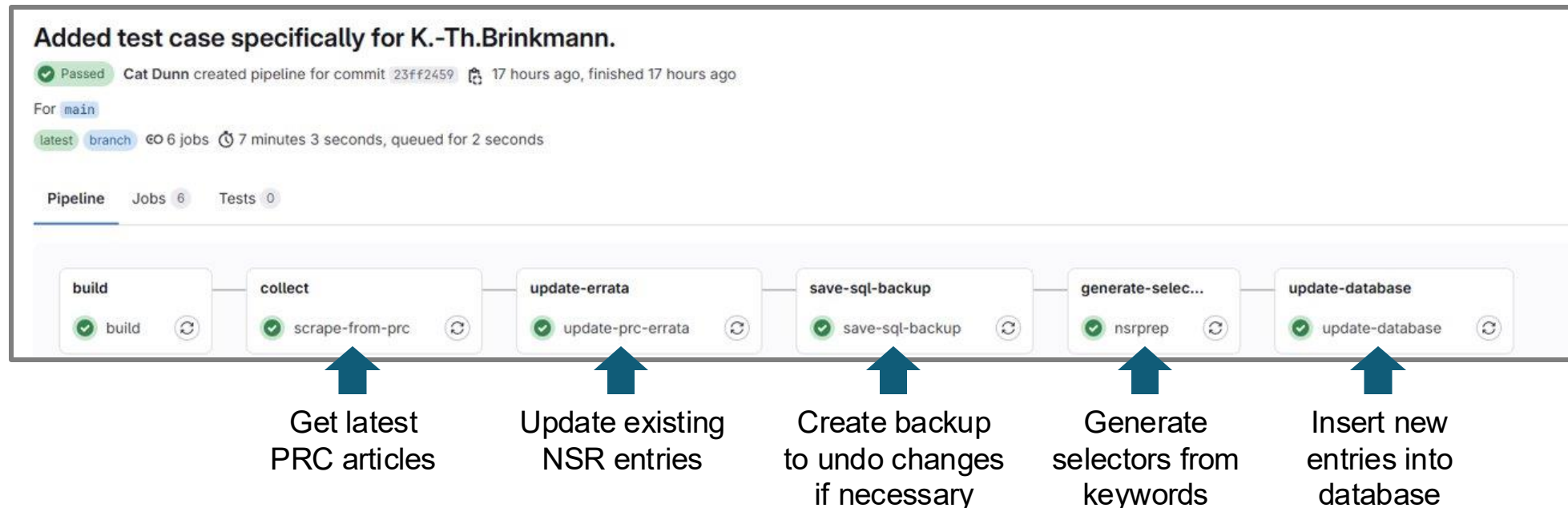
# Proof of Concept Workflow

- **update-nsr-sql: we have a pipeline!** (based on legacy workflow)
  - Currently only scrapes from Physical Review C



# Proof of Concept Workflow

- **update-nsr-sql: we have a pipeline!** (based on legacy workflow)
  - Currently only scrapes from Physical Review C



# Things to Fix

- Issues with **update-nsr-sql**:

# Things to Fix

- Issues with **update-nsr-sql**:
  - Uses old exchange format and previous SQL tables

# Things to Fix

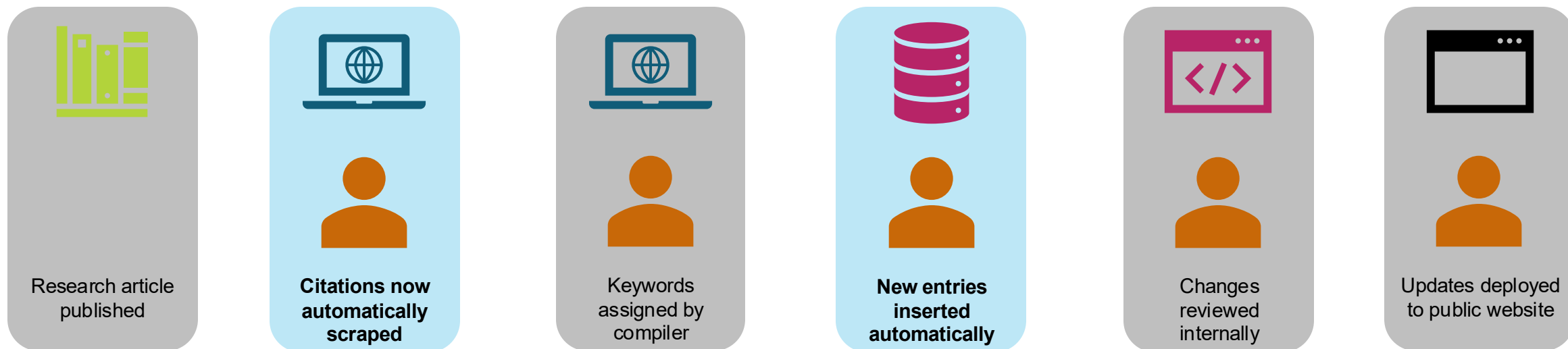
- Issues with **update-nsr-sql**:
  - Uses old exchange format and previous SQL tables
- **nsrprep**
  - does not separate format validation and selector generation
  - cannot generate selectors without keywords

# Things to Fix

- Issues with **update-nsr-sql**:
  - Uses old exchange format and previous SQL tables
- **nsrprep**
  - does not separate format validation and selector generation
  - cannot generate selectors without keywords
- No tools for correcting mistakes
  - typos, name conventions, etc.

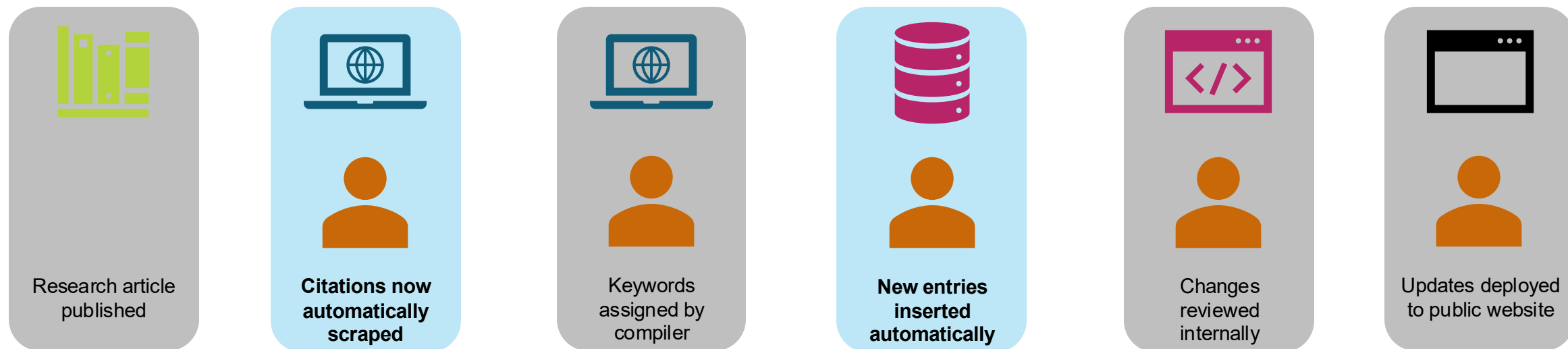


# Workflow With Automation



- **update-nsr-sql** shows that NSR's workflow can be streamlined
  - We could deploy this framework today, if we wanted to

# Workflow With Automation



- **update-nsr-sql** shows that NSR's workflow can be streamlined
  - We could deploy this framework today, if we wanted to
  - See Ramon Arcilla's work on **chatNSR** for automated keywording

# What Next?

- How do we prepare NSR for "AI-readiness"?

# What Next?

- How do we prepare NSR for "AI-readiness"?
  - ✓ **Create a data model to store existing entries**
    - ✓ Include the SQL search tables!
  - ✓ **Translate from exchange format to JSON**
  - ✓ **Automate data collection and insertion**

# What Next?

- How do we prepare NSR for "AI-readiness"?
  - ✓ **Create a data model to store existing entries**
    - ✓ Include the SQL search tables!
  - ✓ **Translate from exchange format to JSON**
  - ✓ **Automate data collection and insertion**
- **Create and enforce a schema to validate data files**
- **Build an editor for adding/editing NSR entries**

# Schema Validation

- Validator implemented using **Pydantic**

2 validation errors for NSREntryReader

**publicationYear**

Input should be a valid integer [type=int\_type, input\_value=None, input\_type=NoneType]

For further information visit [https://errors.pydantic.dev/2.11/v/int\\_type](https://errors.pydantic.dev/2.11/v/int_type)

**referenceType**

Input should be 'JOURNAL', 'CONFERENCE', 'REPORT', 'BOOK', 'PRIVATE\_COMMUNICATION', 'THESIS' or 'PREPRINT' [type=enum, input\_value='', input\_type=str]

For further information visit <https://errors.pydantic.dev/2.11/v/enum>



# Schema Validation

- Validator implemented using **Pydantic**
  - Built into Python data model classes
  - Enforces rules for data consistency

2 validation errors for NSREntryReader

**publicationYear**

Input should be a valid integer [type=int\_type, input\_value=None, input\_type=NoneType]

For further information visit [https://errors.pydantic.dev/2.11/v/int\\_type](https://errors.pydantic.dev/2.11/v/int_type)

**referenceType**

Input should be 'JOURNAL', 'CONFERENCE', 'REPORT', 'BOOK', 'PRIVATE\_COMMUNICATION', 'THESIS' or 'PREPRINT' [type=enum, input\_value='', input\_type=str]

For further information visit <https://errors.pydantic.dev/2.11/v/enum>

# Schema Validation

- Validator implemented using **Pydantic**
  - Built into Python data model classes
  - Enforces rules for data consistency

2 validation errors for NSREntryReader

**publicationYear**

Input should be a valid integer [type=int\_type, input\_value=None, input\_type=NoneType]

For further information visit [https://errors.pydantic.dev/2.11/v/int\\_type](https://errors.pydantic.dev/2.11/v/int_type)

**referenceType**

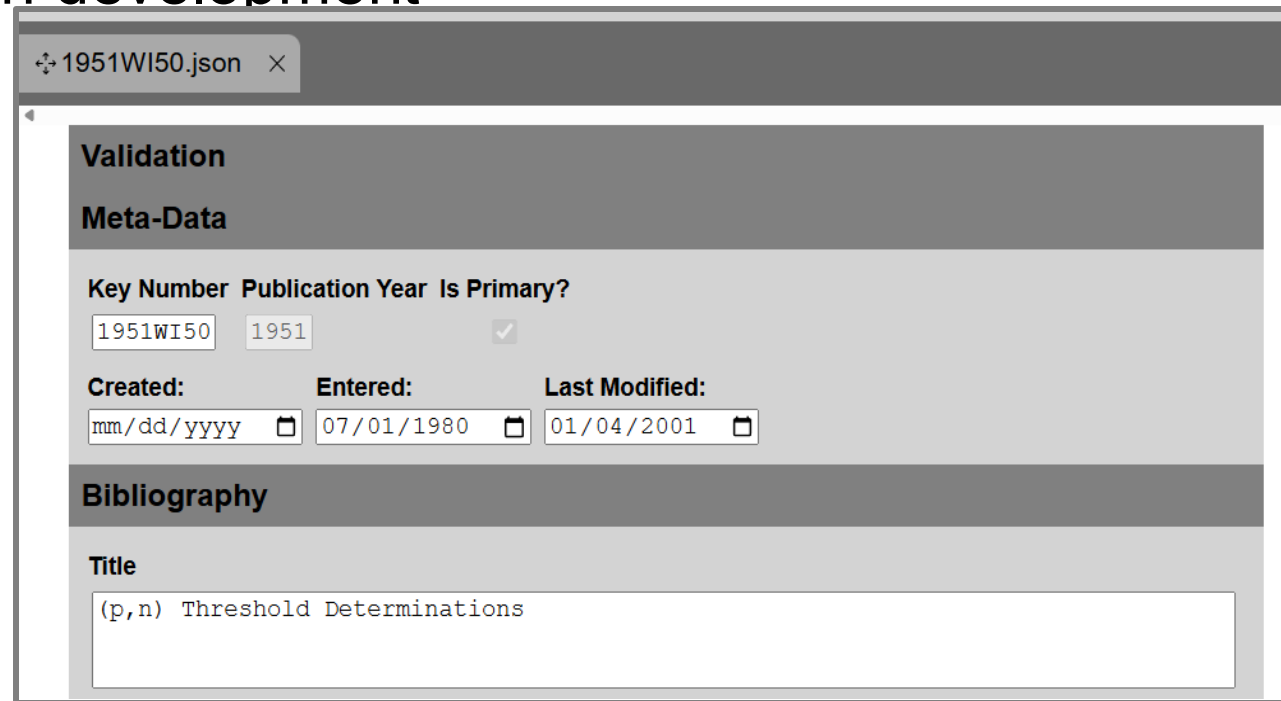
Input should be 'JOURNAL', 'CONFERENCE', 'REPORT', 'BOOK', 'PRIVATE\_COMMUNICATION', 'THESIS' or 'PREPRINT' [type=enum, input\_value='', input\_type=str]

For further information visit <https://errors.pydantic.dev/2.11/v/enum>

- *Unlike **nsrprep**, this can run offline without connecting to NSR*

# NSR Editor

- An application for creating/editing NSR JSON files
  - Currently in development

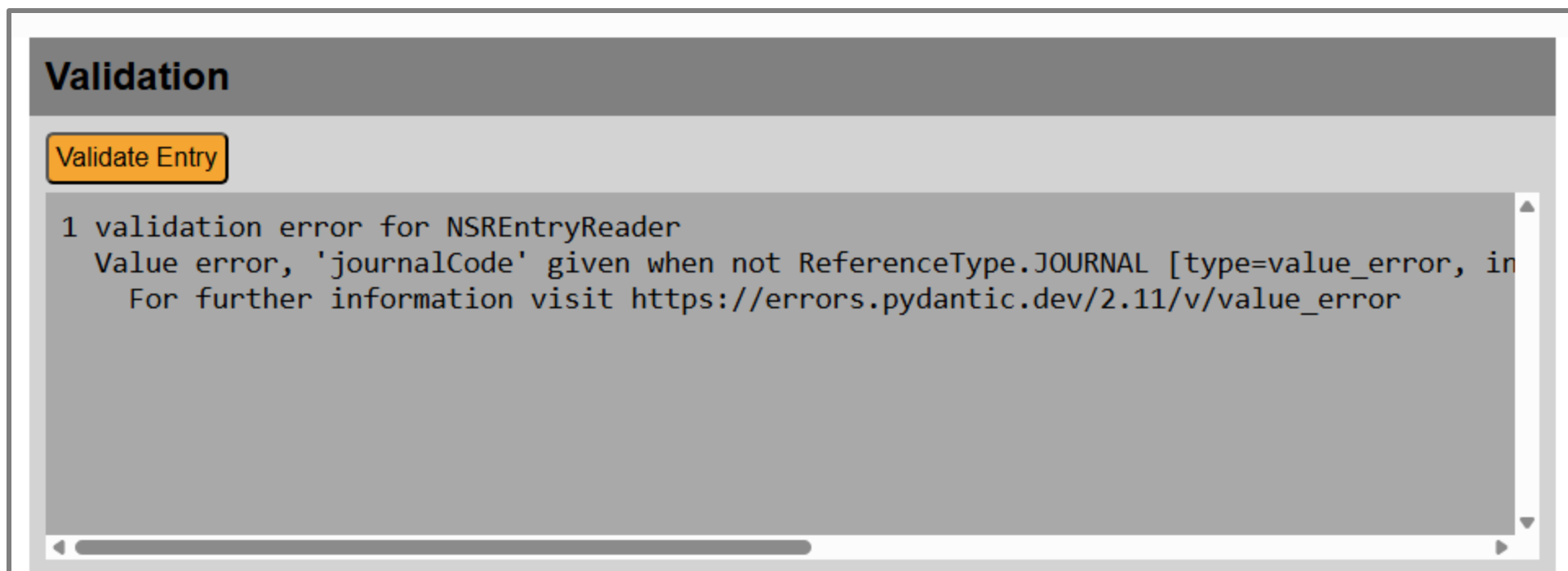


The screenshot shows a web-based application window titled "1951WI50.json". The interface is divided into several sections:

- Validation**: A header section.
- Meta-Data**: A section containing form fields for:
  - Key Number**: A text input field containing "1951WI50".
  - Publication Year**: A text input field containing "1951".
  - Is Primary?**: A checkbox that is checked.
  - Created:**: A date input field with a calendar icon, showing "mm/dd/yyyy".
  - Entered:**: A date input field with a calendar icon, showing "07/01/1980".
  - Last Modified:**: A date input field with a calendar icon, showing "01/04/2001".
- Bibliography**: A section containing a text input field for the **Title**, which contains the text "(p,n) Threshold Determinations".

# NSR Editor (contd.)

- Built-in validator to catch format errors






# NSR Editor (contd.)

- Has inputs for identifying information

**Meta-Data**

Key Number	Publication Year	Is Primary?
<input type="text" value="1951WI50"/>	<input type="text" value="1951"/>	<input checked="" type="checkbox"/>

Created:	Entered:	Last Modified:
<input type="text" value="mm/dd/yyyy"/> 	<input type="text" value="07/01/1980"/> 	<input type="text" value="01/04/2001"/> 

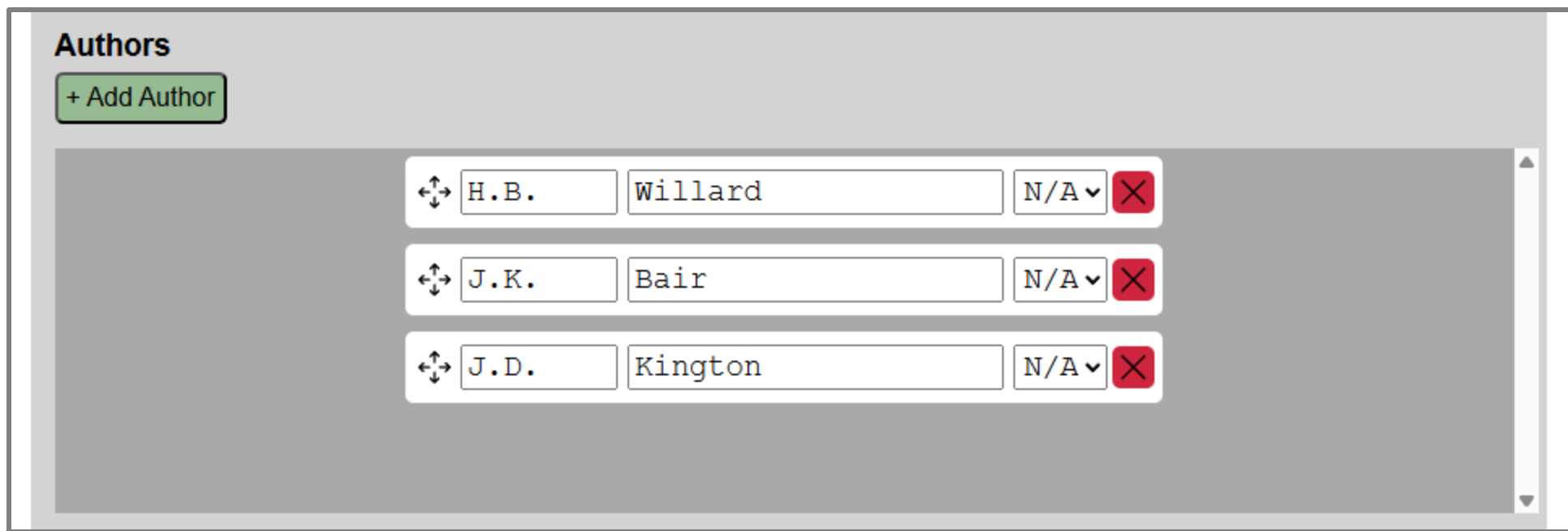
# NSR Editor (contd.)

- Also has inputs for bibliographic data

Bibliography
<b>Title</b>
<input type="text" value="(p,n) Threshold Determinations"/>
<b>DOI</b>
<input type="text"/>
<b>CODEN</b>
<input type="text" value="REPT ORNL-1278,P5,Willard"/>
<b>Reference Type</b>
<input type="text" value="Report"/>
<b>Reference Text</b>
<input type="text" value="ORNL-1278, p.5 (1951)"/>

# NSR Editor (contd.)

- Enables adding and re-ordering authors/collaborations



The screenshot shows the 'Authors' section of the NSR Editor. It features a green '+ Add Author' button at the top left. Below it is a list of three authors, each with a four-way arrow icon for re-ordering. The authors are: H.B. Willard, J.K. Bair, and J.D. Kington. Each author entry has a text field for initials, a text field for the name, a dropdown menu with 'N/A' selected, and a red 'X' button for deletion. The list is contained within a scrollable area.

Initials	Name	Dropdown	Action
H.B.	Willard	N/A	X
J.K.	Bair	N/A	X
J.D.	Kington	N/A	X

# NSR Editor (contd.)

- Enables adding links to supporting datasets

**Library Connections**

PDF Available? XUNDL Data Available?

☒ ☐

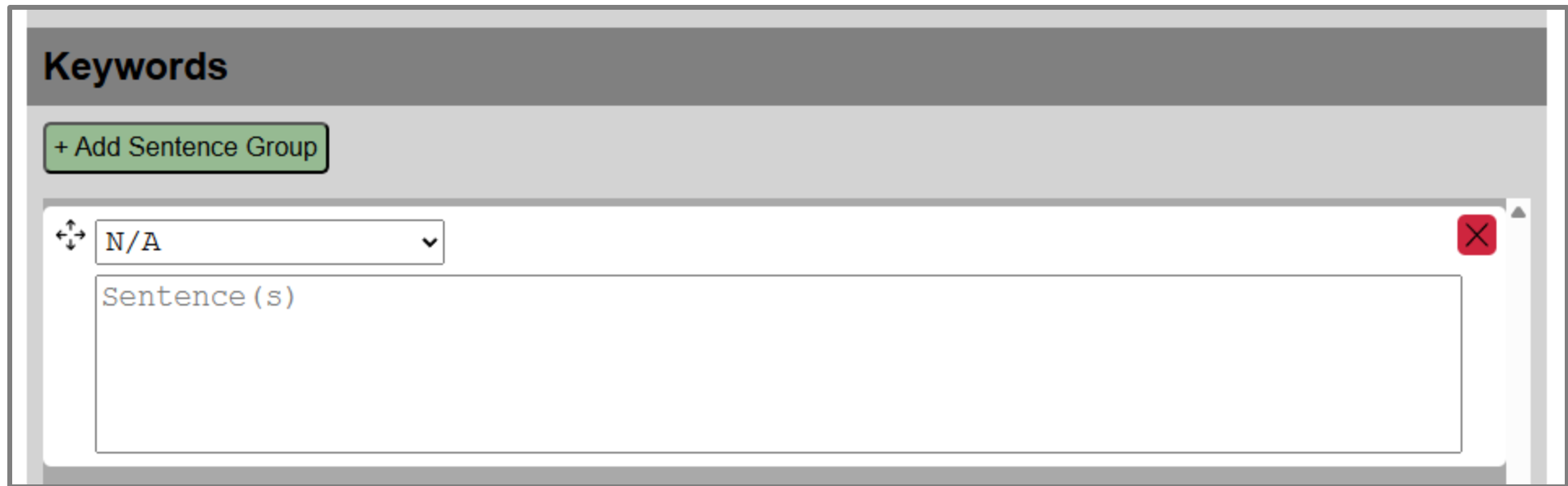
**EXFOR Keys**

Key1, Key2, etc.



# NSR Editor (contd.)

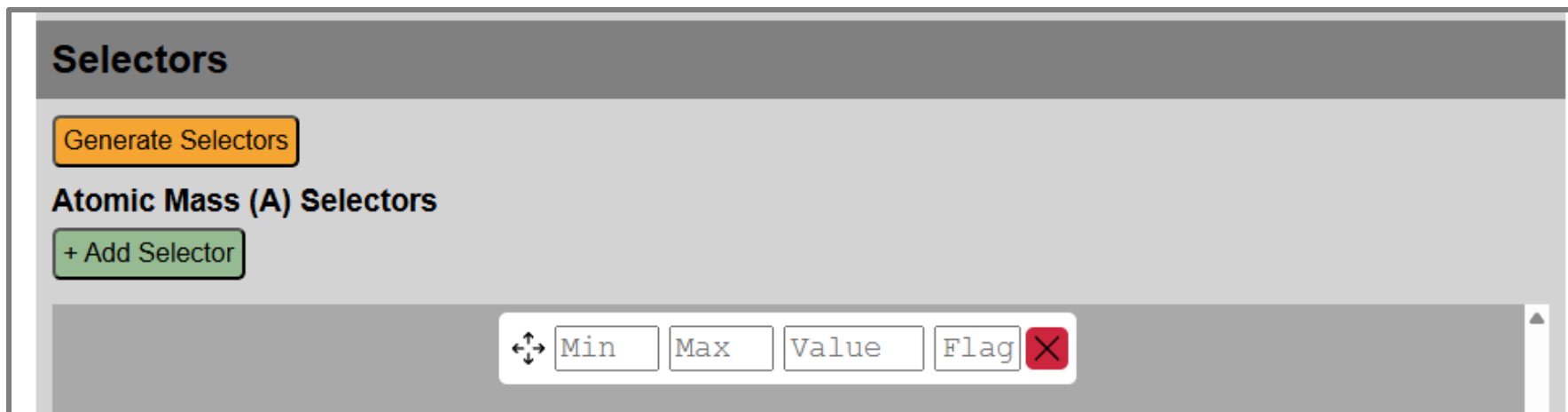
- Includes text inputs for adding keywords



The screenshot shows a web interface titled "Keywords". At the top, there is a green button labeled "+ Add Sentence Group". Below this, there is a section with a red double-headed arrow icon on the left and a dropdown menu containing "N/A". To the right of the dropdown is a red square button with a white "X". Below the dropdown is a large text input area with the placeholder text "Sentence (s)".

# NSR Editor (contd.)

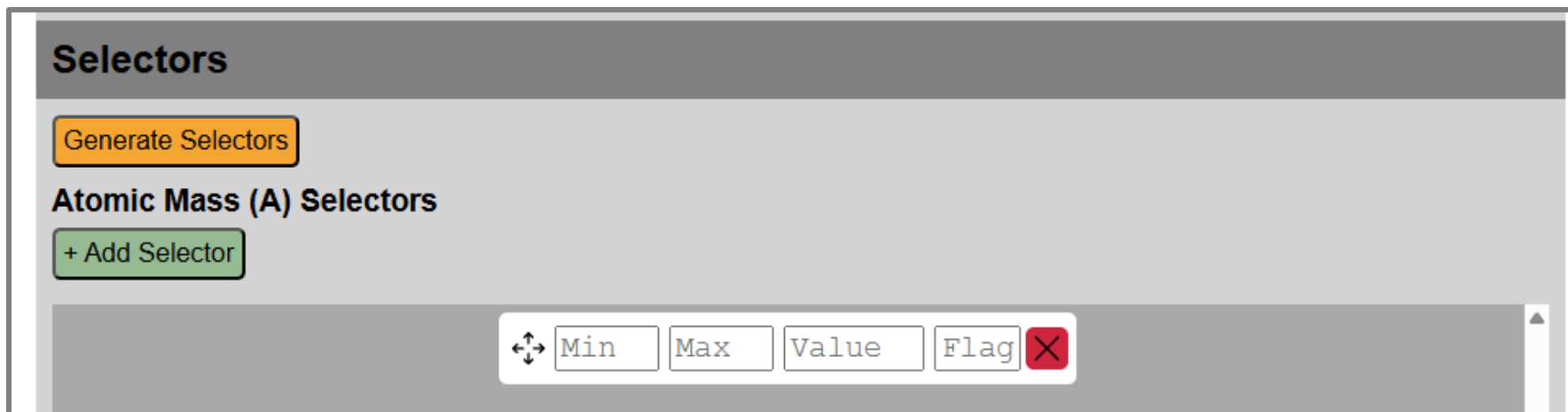
- Also includes space for manually adding selectors



The screenshot displays the 'Selectors' section of the NSR Editor. It features a dark gray header with the title 'Selectors'. Below the header, there is an orange button labeled 'Generate Selectors'. Underneath this, the text 'Atomic Mass (A) Selectors' is displayed. A green button labeled '+ Add Selector' is positioned below the text. At the bottom of the section, there is a row of input fields: a dropdown menu with a double-headed arrow icon, a text box containing 'Min', a text box containing 'Max', a text box containing 'Value', and a text box containing 'Flag' followed by a red square button with a white 'X' icon. A vertical scrollbar is visible on the right side of the section.

# NSR Editor (contd.)

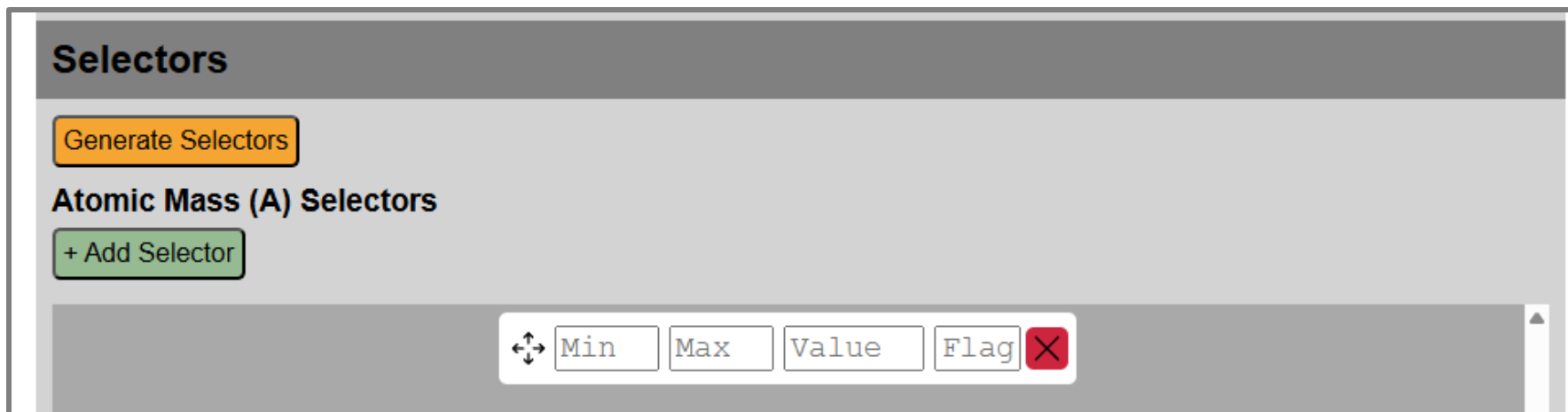
- Also includes space for manually adding selectors
  - **Reminder:** *these power NSR's search functions!*



The screenshot displays the 'Selectors' section of the NSR Editor. It features a grey header with the title 'Selectors'. Below the header, there is an orange button labeled 'Generate Selectors'. Underneath, the text 'Atomic Mass (A) Selectors' is displayed, followed by a green button labeled '+ Add Selector'. At the bottom of the section, there is a row of input fields: a dropdown menu with a double-headed arrow icon, a text box labeled 'Min', a text box labeled 'Max', a text box labeled 'Value', a text box labeled 'Flag', and a red square button with a white 'X' icon.

# NSR Editor (contd.)

- Also includes space for manually adding selectors
  - **Reminder:** *these power NSR's search functions!*
  - In development: automatic selector generation from keywords



The screenshot displays the 'Selectors' section of the NSR Editor. It features a grey header with the title 'Selectors'. Below the header, there is an orange button labeled 'Generate Selectors'. Underneath this, the text 'Atomic Mass (A) Selectors' is displayed. A green button labeled '+ Add Selector' is positioned below the text. At the bottom of the section, there is a row of input fields: a double-headed arrow icon, a 'Min' field, a 'Max' field, a 'Value' field, a 'Flag' field, and a red 'X' icon.

# Where To From Here?

- Full JSON workflow not yet complete

# Where To From Here?

- Full JSON workflow not yet complete
  - Need to scrape from more journals
    - Nuclear Physics A
    - European Physics Journal

# Where To From Here?

- Full JSON workflow not yet complete
  - Need to scrape from more journals
    - Nuclear Physics A
    - European Physics Journal
  - Need more keyword assignment
    - (see Ramon's presentation on **chatNSR**)

# Where To From Here?

- Full JSON workflow not yet complete
  - Need to scrape from more journals
    - Nuclear Physics A
    - European Physics Journal
  - Need more keyword assignment
    - (see Ramon's presentation on **chatNSR**)
  - NSR editor needs further testing



# Where To From Here?

- End goal: transition to full JSON database
  - *NSR exchange format has not been replaced*

# Where To From Here?

- End goal: transition to full JSON database
  - *NSR exchange format has not been replaced*
  - Changes will be made to **nsr-update** first:
    - <https://www.nndc.bnl.gov/nsr-update/>

# Where To From Here?

- End goal: transition to full JSON database
  - *NSR exchange format has not been replaced*
- Changes will be made to **nsr-update** first:
  - <https://www.nndc.bnl.gov/nsr-update/>
- Already completed:
  - Re-building SQL database from JSON using ORM
  - Searching for entries from ORM database
  - Displaying LaTeX titles/keywords on web