
warn-transformer

unknown

Sep 15, 2023

CONTENTS

1	Installation	3
2	Table of contents	5
3	Links	19
4	About	21
	Python Module Index	23
	Index	25

Consolidate, enrich and republish the data gathered by [warn-scrapers](#)

INSTALLATION

Use `pip` or `pipenv` to install the Python library and command-line interface.

```
pipenv install warn-transformer
```


TABLE OF CONTENTS

2.1 Contributing

Clone the repository to your computer and navigate into the directory in your terminal.

The `pipenv` package manager can install all of the Python tools necessary to run and test our code.

```
pipenv install --dev
```

Now install `pre-commit` to run a battery of automatic quick fixes against your work.

```
pipenv run pre-commit install
```

Create a `.env` file and define your `BLN_API_TOKEN`.

```
BLN_API_TOKEN=<yourtoken>
```

Download the raw source files from biglocalnews.org.

```
make download
```

Consolidate the raw files into a single file.

```
make consolidate
```

Integrate the consolidated file with our latest polished dataset.

```
make integrate
```

2.2 Releasing

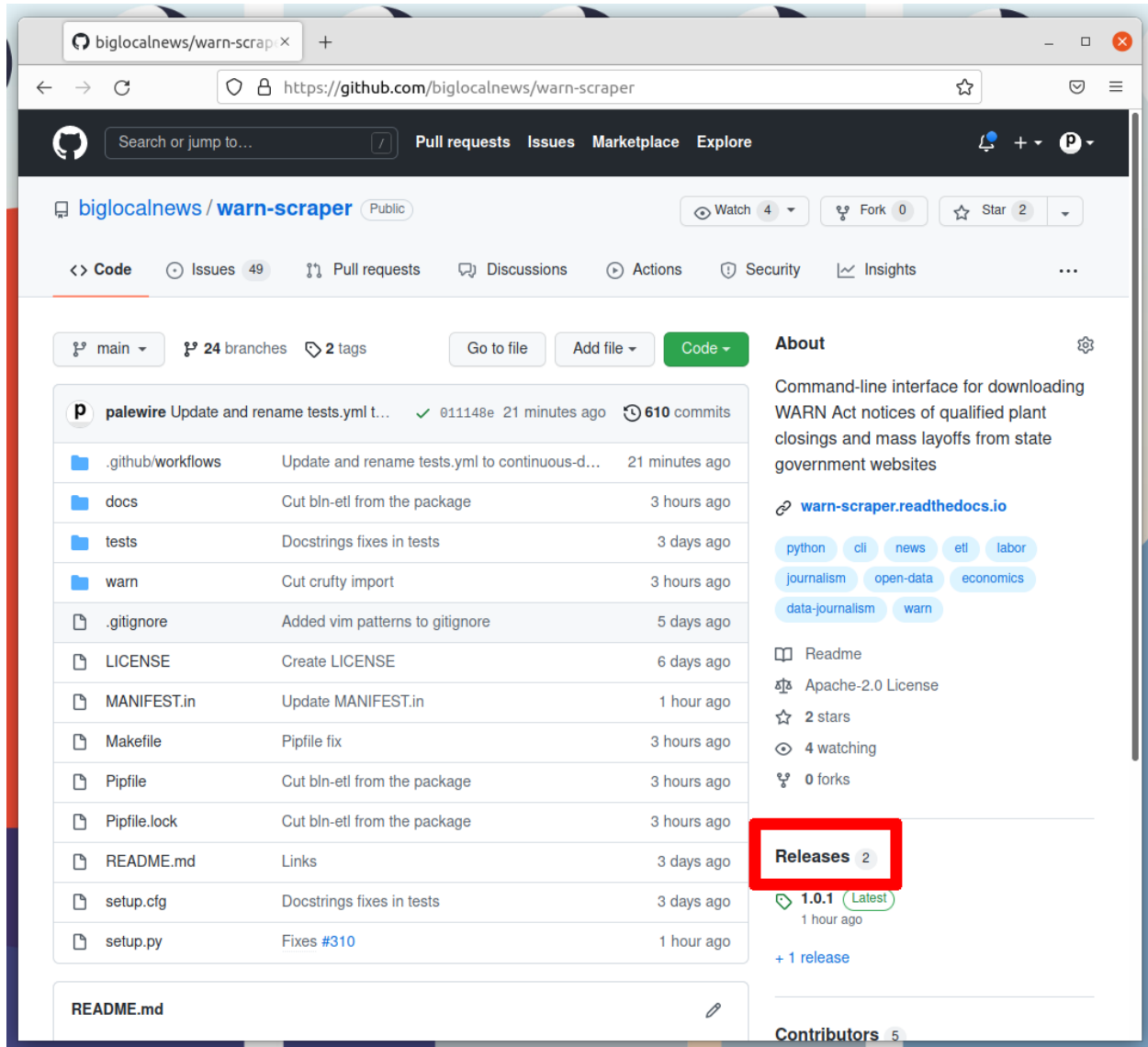
Our release process is automated as a [continuous deployment](#) via the [GitHub Actions](#) framework. The logic that governs the process is stored in the [workflows](#) directory.

That means that everything necessary to make a release can be done with a few clicks on the GitHub website. All you need to do is make a tagged release at biglocalnews/warn-transformer/releases, then wait for the computers to handle the job.

Here's how it's done, step by step. The screenshots are from a different repository, but the process is the same.

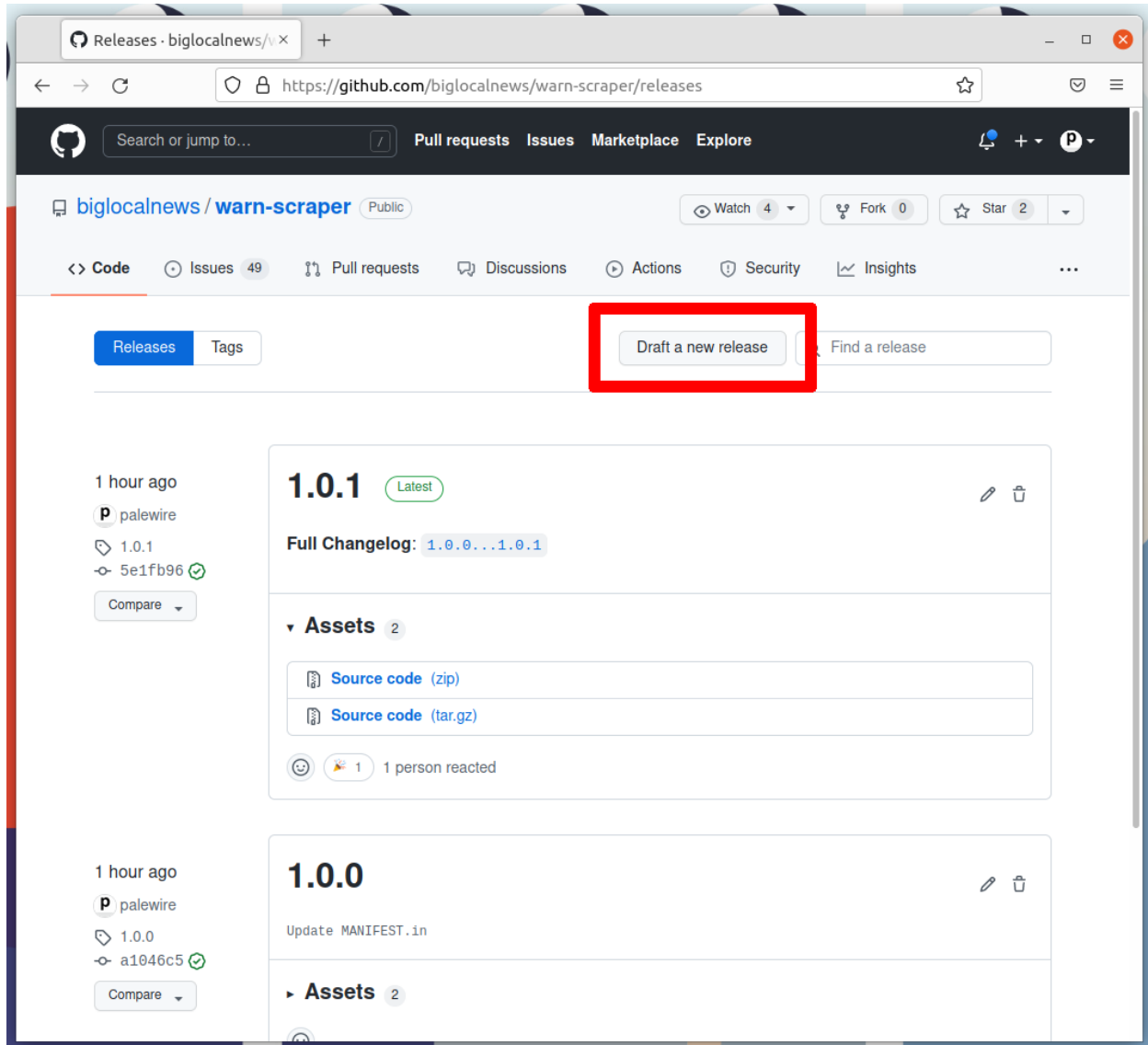
2.2.1 1. Go to the releases page

The first step is to visit our repository's homepage and click on the “releases” headline in the right rail.



2.2.2 2. Click ‘Draft a new release’

Note the number of the latest release. Click the “Draft a new release” button in the upper-right corner. If you don’t see this button, you do not have permission to make a release. Only the maintainers of the repository are able to release new code.



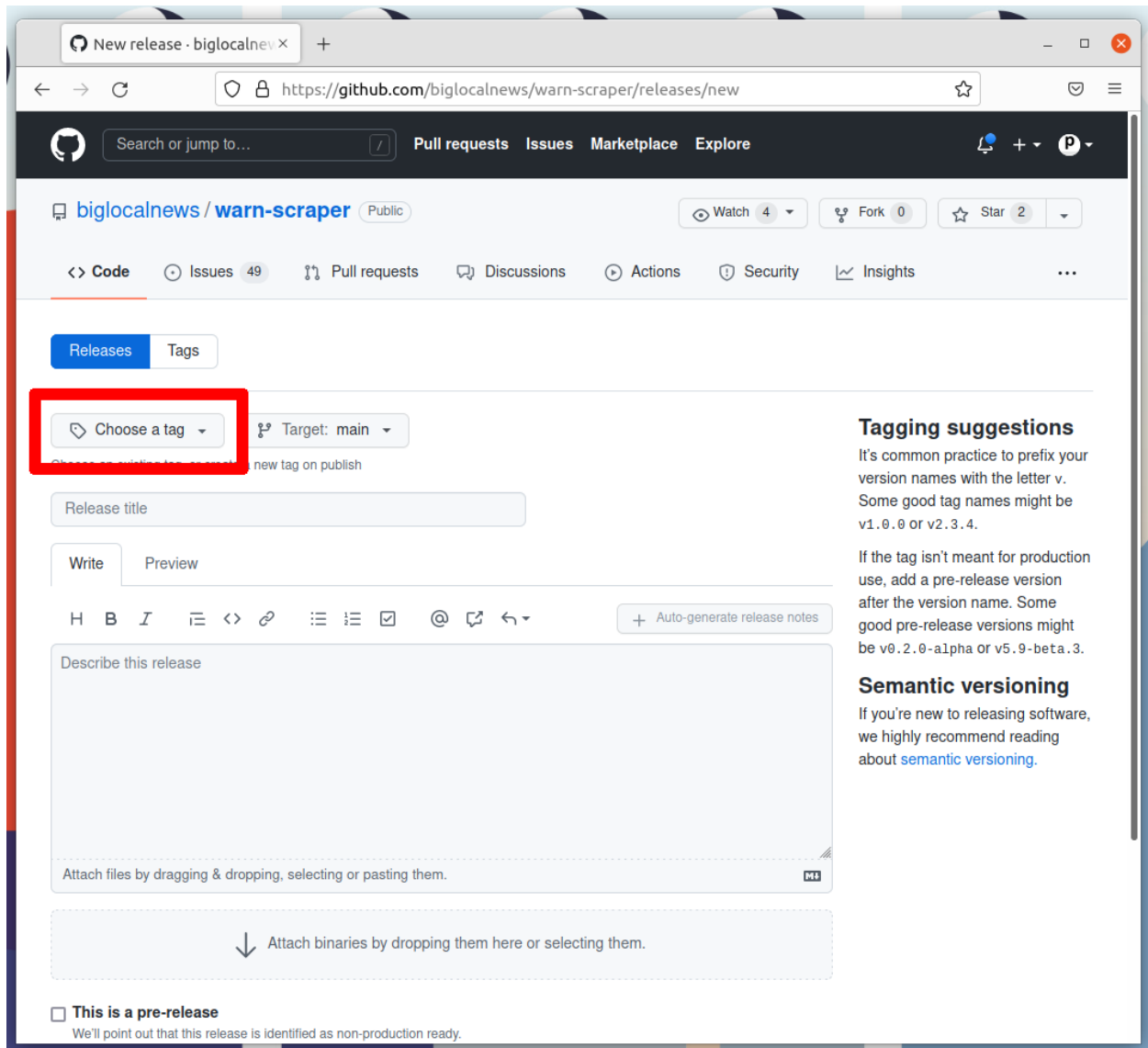
2.2.3 3. Create a new tag

Think about how big your changes are and decide if you’re a major, minor or patch release.

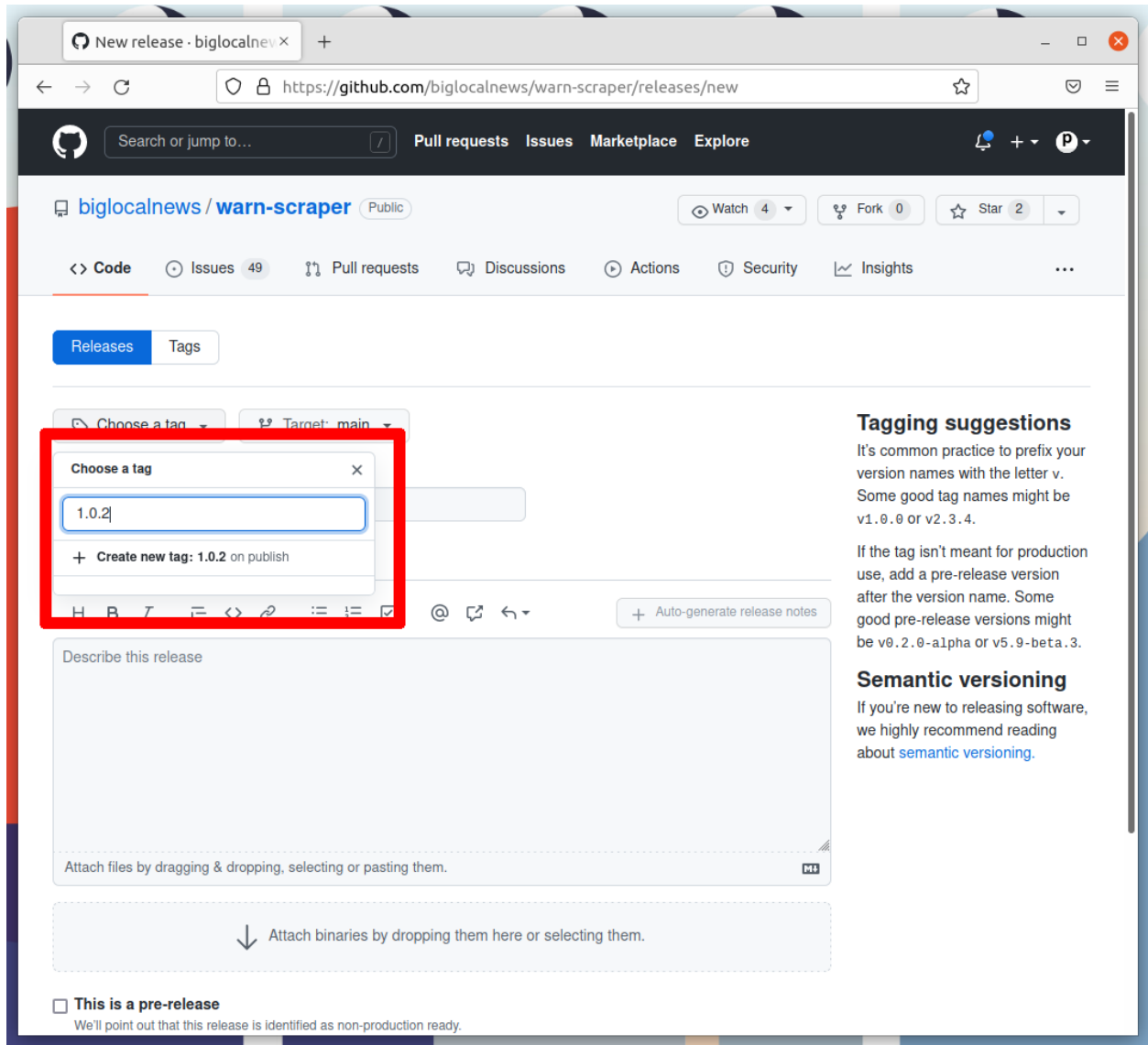
All version numbers should feature three numbers separated by the periods, like 1.0.1. If you’re making a major release that isn’t backwards compatible, the latest release’s first number should go up by one. If you’re making a minor release by adding a feature or making a large change, the second number should go up. If you’re only fixing bugs or making small changes, the third number should go up.

If you’re unsure, review the standards defined at semver.org to help make a decision. In the end don’t worry about it too much. Our version numbers don’t need to be perfect. They just need to be three numbers separated by periods.

Once you've settled on the number for your new release, click on the "Choose a tag" pull down.

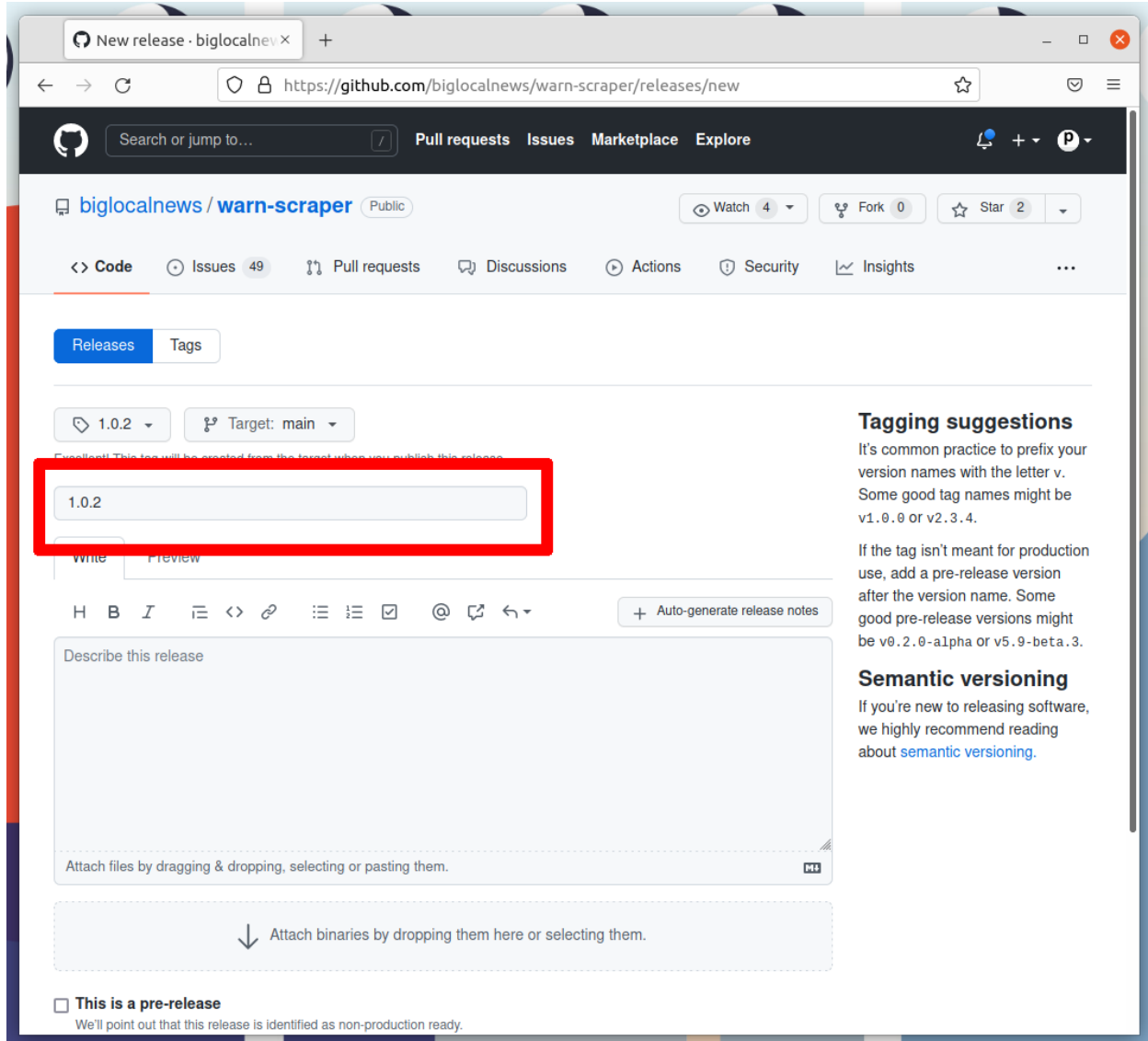


Enter your version number into the box. Then click the "Create new tag" option that appears.



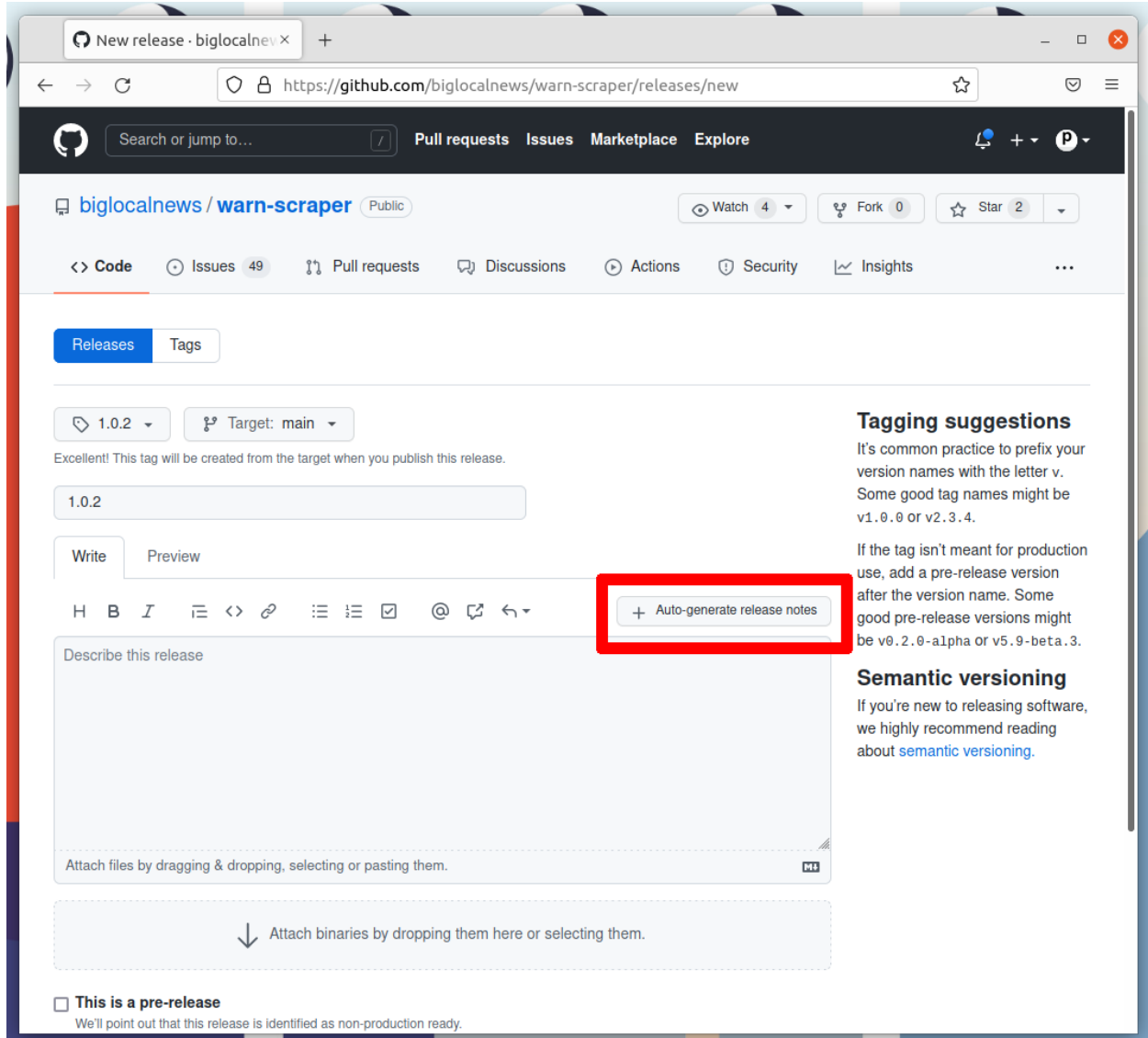
2.2.4 4. Name the release

Enter the same number into the “Release title” box.



2.2.5 5. Auto-generate release notes

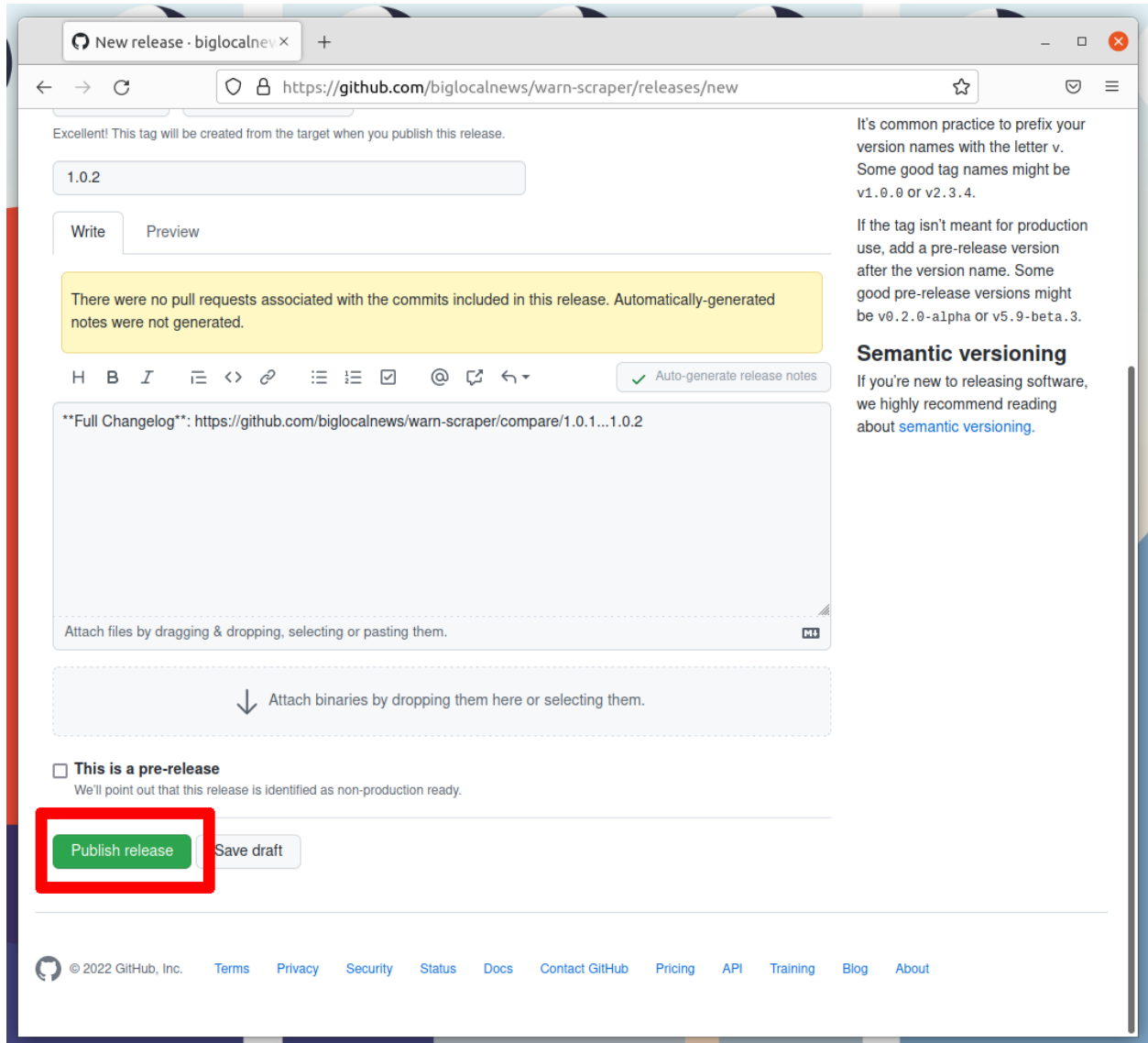
Click the “Auto-generate release notes” button in the upper right corner of the large description box.



That should fill in the box below. What appears will depend on how many pull requests you've merged since the last release.

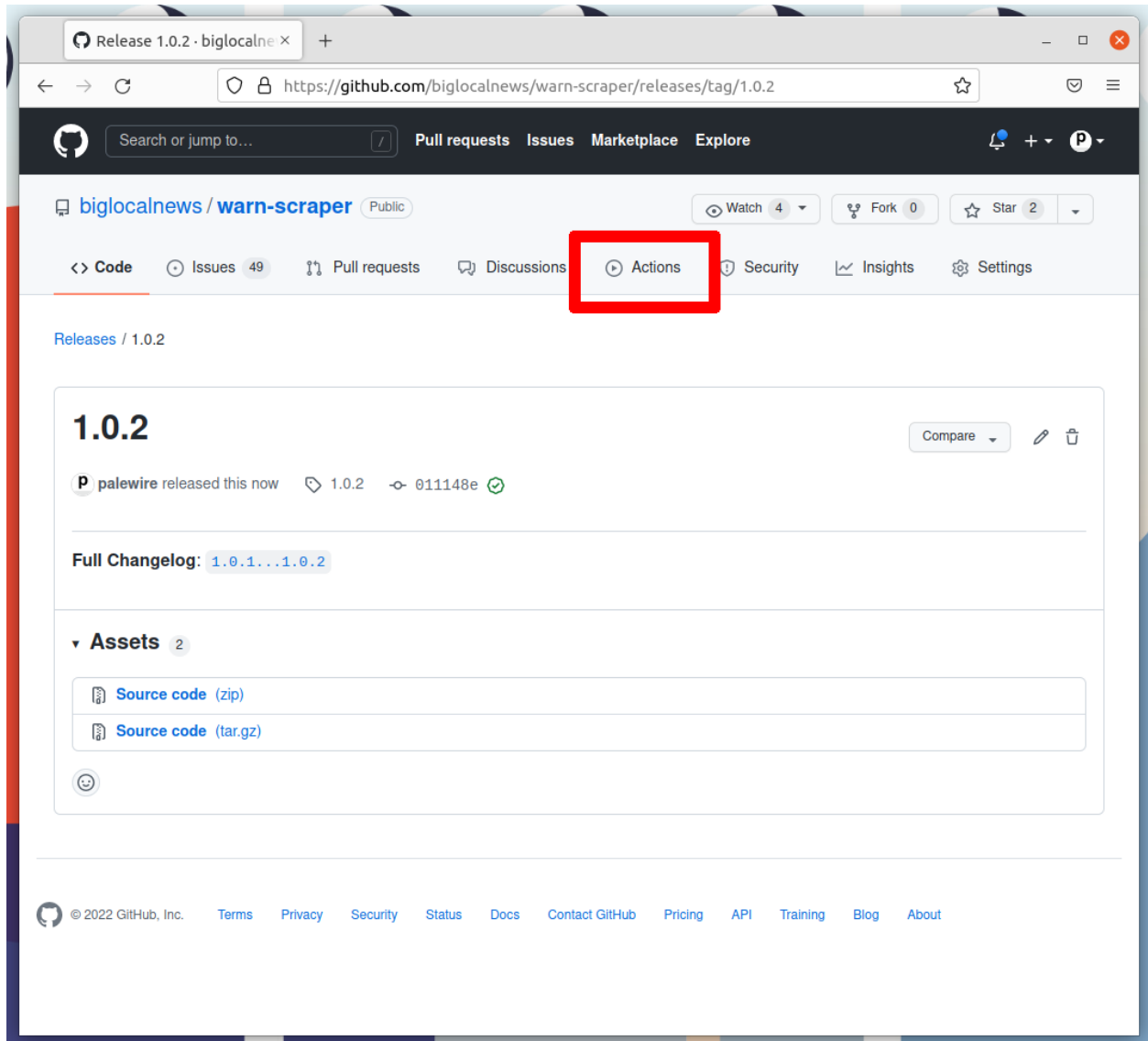
2.2.6 6. Publish the release

Click the green button that says “Publish release” at the bottom of the page.

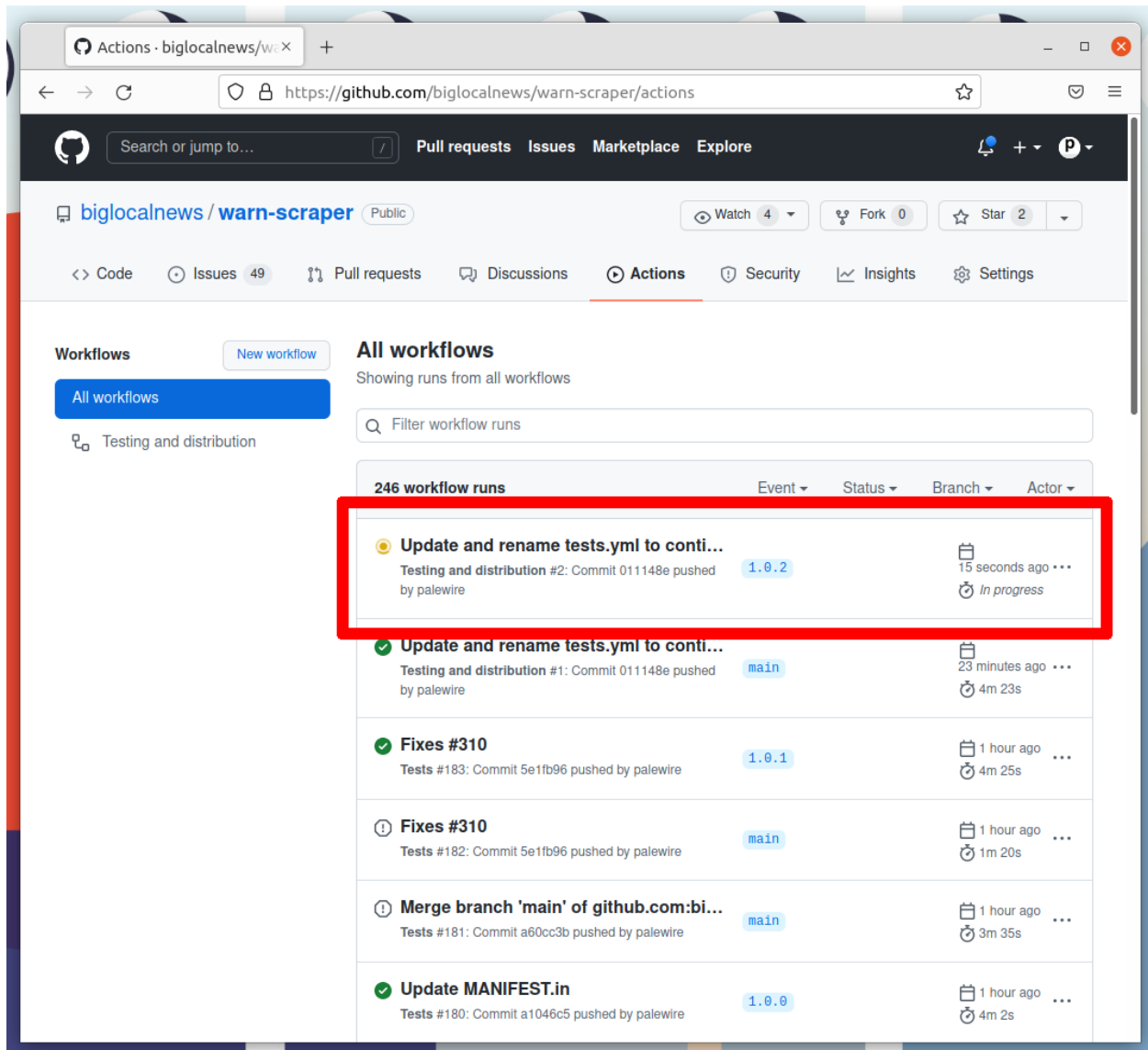


2.2.7 7. Wait for the Action to finish

GitHub will take you to a page dedicated to your new release and start an automated process that release our new version to the world. Follow its progress by clicking on the [Actions](#) tab near the top of the page.



That will take you to the Actions monitoring page. The task charged with publishing your release should be at the top.



After a few minutes, the process there should finish and show a green check mark. When it does, visit [warn-transformer's page on PyPI](#), where you should see the latest version displayed at the top of the page.

If the action fails, something has gone wrong with the deployment process. You can click into its debugging panel to search for the cause or ask the project maintainers for help.

2.3 Reference

Sections

- *Commands*
- *Schema*
- *Utils*

2.3.1 Commands

warn-transformer

Consolidate, enrich and republish the data gathered by warn-scraper.

```
warn-transformer [OPTIONS] COMMAND [ARGS] ...
```

consolidate

Consolidate raw data using a common data schema.

```
warn-transformer consolidate [OPTIONS]
```

Options

--input-dir <input_dir>

The Path were the raw files results are located

--source <source>

The source to download. Default is all sources.

-l, --log-level <log_level>

Set the logging level

Options

DEBUG | INFO | WARNING | ERROR | CRITICAL

download

Download all the CSVs in the WARN Notice project on biglocalnews.org.

```
warn-transformer download [OPTIONS]
```

Options

--download-dir <download_dir>

The Path were the results will be downloaded

--source <source>

The source to download. Default is all sources.

-l, --log-level <log_level>

Set the logging level

Options

DEBUG | INFO | WARNING | ERROR | CRITICAL

integrate

Integrate the latest consolidated data with the current database.

```
warn-transformer integrate [OPTIONS]
```

Options

--input-dir <input_dir>

The Path were the new results are located

--init

-l, --log-level <log_level>

Set the logging level

Options

DEBUG | INFO | WARNING | ERROR | CRITICAL

2.3.2 Schema

class warn_transformer.schema.BaseTransformer(*input_dir: Path*)

Transform a state's raw data for consolidation.

check_if_amendment(*row: Dict*) → bool

Determine whether a row is an amendment or not.

Parameters

row (*dict*) – The raw row of data.

Returns: A boolean

check_if_closure(*row: Dict*) → Optional[bool]

Determine whether a row is a closure or not.

Parameters

row (*dict*) – The raw row of data.

Returns: A boolean or null

check_if_temporary(*row: Dict*) → Optional[bool]

Determine whether a row is a temporary or not.

Parameters

row (*dict*) – The raw row of data.

Returns: A boolean or null

get_hash_id(*data: Dict*) → str

Convert the row into a unique hexdigest to use as a unique identifier.

Parameters

data (*dict*) – One raw row of data from the source

Returns: A unique hexdigest string computed from the source data.

get_raw_data() → List[Dict]

Get the raw data from our scraper for this source.

Returns: A list of raw rows of data from the source.

get_raw_value(*row*, *method*)

Fetch a value from the row that for transformation.

Parameters

- **row** – One raw row of data from the source
- **method** – The technique to use to pull data. If a string method is provided, it is used to fetch a key of that name from the row. If a callable function is provided, the row is run through it.

Returns: A value ready for transformation.

handle_amendments(*row_list*: List[Dict]) → List[Dict]

Remove amended filings from the provided list of records.

Parameters

row_list (*list*) – A list of clean rows of data.

Returns: A list of cleaned data, minus amended records.

prep_row_list(*row_list*: List[Dict]) → List[Dict]

Make necessary transformations to the raw row list prior to transformation.

Parameters

row_list (*list*) – A list of raw rows of data from the source.

Returns: The row list minus empty records

schema

alias of [WarnNoticeSchema](#)

transform() → List[Dict]

Transform prepared rows into a form that's ready for consolidation.

Returns: A validated list of dictionaries that conform to our schema

transform_company(*value*: str) → str

Transform a raw company name.

Parameters

value (*str*) – The raw company string provided by the source

Returns: A string object ready for consolidation.

transform_date(*value*: str) → Optional[str]

Transform a raw date string into a date object.

Parameters

value (*str*) – The raw date string provided by the source

Returns: A date object ready for consolidation. Or, if the date string is invalid, a None.

transform_jobs(*value*: str) → Optional[int]

Transform a raw jobs number into an integer.

Parameters

value (*str*) – A raw jobs number provided by the source

Returns: An integer number ready for consolidation. Or, if the value is invalid, a None.

transform_location(value: str) → str

Transform a raw location.

Parameters

value (str) – The raw location string provided by the source

Returns: A string object ready for consolidation.

transform_row(row: Dict) → Dict

Transform a row into a form that's ready for consolidation.

Parameters

row (dict) – One raw row of data from the source

Returns: A transformed dict that's ready to be loaded into our consolidated schema.

```
class warn_transformer.schema.WarnNoticeSchema(*, only: types.StrSequenceOrSet | None = None,  
                                                exclude: types.StrSequenceOrSet = (), many: bool =  
                                                False, context: dict | None = None, load_only:  
                                                types.StrSequenceOrSet = (), dump_only:  
                                                types.StrSequenceOrSet = (), partial: bool |  
                                                types.StrSequenceOrSet = False, unknown: str | None  
                                                = None)
```

An standardized instance of a WARN Act Notice.

2.3.3 Utils

warn_transformer.utils.get_all_transformers() → List[str]

Get all the states and territories that have scrapers.

Returns: List of lower-case post abbreviations.

LINKS

- Documentation: warn-transformer.readthedocs.io
- Repository: github.com/biglocalnews/warn-transformer
- Packaging: pypi.org/project/warn-transformer

CHAPTER FOUR

ABOUT

The project is sponsored by [Big Local News](#), a program at Stanford University that collects data for impactful journalism. The code is maintained by [Ben Welsh](#), a visiting data journalist from the Los Angeles Times.

PYTHON MODULE INDEX

W

`warn_transformer.schema`, [16](#)

`warn_transformer.utils`, [18](#)

Symbols

--download-dir
warn-transformer-download command line option, 15

--init
warn-transformer-integrate command line option, 16

--input-dir
warn-transformer-consolidate command line option, 15
warn-transformer-integrate command line option, 16

--log-level
warn-transformer-consolidate command line option, 15
warn-transformer-download command line option, 15
warn-transformer-integrate command line option, 16

--source
warn-transformer-consolidate command line option, 15
warn-transformer-download command line option, 15

-l
warn-transformer-consolidate command line option, 15
warn-transformer-download command line option, 15
warn-transformer-integrate command line option, 16

B

BaseTransformer (class in warn_transformer.schema), 16

C

check_if_amendment()
(warn_transformer.schema.BaseTransformer method), 16

check_if_closure() (warn_transformer.schema.BaseTransformer method), 16

check_if_temporary()
(warn_transformer.schema.BaseTransformer method), 16

G

get_all_transformers() (in module warn_transformer.utils), 18

get_hash_id() (warn_transformer.schema.BaseTransformer method), 16

get_raw_data() (warn_transformer.schema.BaseTransformer method), 16

get_raw_value() (warn_transformer.schema.BaseTransformer method), 17

H

handle_amendments()
(warn_transformer.schema.BaseTransformer method), 17

M

module
warn_transformer.schema, 16
warn_transformer.utils, 18

P

prep_row_list() (warn_transformer.schema.BaseTransformer method), 17

S

schema (warn_transformer.schema.BaseTransformer attribute), 17

T

transform() (warn_transformer.schema.BaseTransformer method), 17

transform_company()
(warn_transformer.schema.BaseTransformer method), 17

transform_date() (warn_transformer.schema.BaseTransformer method), 17

transform_jobs() (warn_transformer.schema.BaseTransformer method), 17

`transform_location()`
 (*warn_transformer.schema.BaseTransformer*
 method), 18

`transform_row()` (*warn_transformer.schema.BaseTransformer*
 method), 18

W

`warn_transformer.schema`
 module, 16

`warn_transformer.utils`
 module, 18

`warn-transformer-consolidate` command line
 option
 --input-dir, 15
 --log-level, 15
 --source, 15
 -l, 15

`warn-transformer-download` command line
 option
 --download-dir, 15
 --log-level, 15
 --source, 15
 -l, 15

`warn-transformer-integrate` command line
 option
 --init, 16
 --input-dir, 16
 --log-level, 16
 -l, 16

`WarnNoticeSchema` (class in
 warn_transformer.schema), 18