# geojsplit

***Release 0.1.2***

**Yann-Sebastien Tremblay-Johnston**

**Oct 06, 2019**

# CONTENTS

Version 0.1.2.

Geojsplit is a useful tool for splitting large GeoJSON files into multiple files. Inspired by https://github.com/woodb/geojsplit.

CHAPTER

ONE

# USER GUIDE

## 1.1 Quickstart

### 1.1.1 Install

*Installation instructions*

### 1.1.2 Usage

Although both the library code and the command line tool of geojsplit are relatively simple, there are use cases for both. You may want to use the backend `GeoJSONBatchStreamer` class directly in order to do more sophisticated manipulations with GeoJOSN documents. As a command line tool geojsplit also works well as a preprocessing step for working with large GeoJSON documents i.e. for piping into GDAL's ogr2ogr tool.

#### As a library

Once installed, geojsplit can be imported in like

```
from geojsplit import geojsplit

geojson = geojsplit.GeoJSONBatchStreamer("/path/to/some.geojson")

for feature_collection in geojson.stream():
    do_something(feature_collection)
    ...
```

If the `/path/to/some.geojson` does not exists, `FileNotFound` will be raised.

You can control how many features are streamed into a Feature Collection using the `batch` parameter (Default is 100).

```
>>> g = geojson.stream(batch=2)  # instatiate generator object
>>> data = next(g)
>>> print(data)
{"features": [{"geometry": {"coordinates": [[[-118.254638, 33.7843], [-118.254637,
33.784231], [-118.254556, 33.784232], [-118.254559, 33.784339], [-118.254669,
33.784338], [-118.254668, 33.7843], [-118.254638, 33.7843]]], "type": "Polygon"},
"properties": {}, "type": "Feature"}, {"geometry": {"coordinates": [[[-118.254414,
33.784255], [-118.254232, 33.784255], [-118.254232, 33.784355], [-118.254414,
33.784355], [-118.254414, 33.784255]]], "type": "Polygon"}, "properties": {}, "type":
"Feature"}], "type": "FeatureCollection"}
```

(continues on next page)

(continued from previous page)

```
>>> print(len(data["features"]))
2
```

If your GeoJSON document has a different format or you want to iterate over different elements on your document, you can also pass a different value to the `prefix` keyword argument (Default is `'features.item'`). This is an argument passed directly down to a `ijson.items` call, for more information see https://github.com/ICRAR/ijson.

### As a command line tool

After installing you should have the geojsplit executable in your `PATH`.

```
$ geojsplit -h
usage: geojsplit [-h] [-l GEOMETRY_COUNT] [-a SUFFIX_LENGTH] [-o OUTPUT]
                 [-n LIMIT] [-v] [-d] [--version]
                 geojson

Split a geojson file into many geojson files.

positional arguments:
geojson               filename of geojson file to split

optional arguments:
-h, --help            show this help message and exit
-l GEOMETRY_COUNT, --geometry-count GEOMETRY_COUNT
                        the number of features to be distributed to each file.
-a SUFFIX_LENGTH, --suffix-length SUFFIX_LENGTH
                        number of characters in the suffix length for split
                        geojsons
-o OUTPUT, --output OUTPUT
                        output directory to save split geojsons
-n LIMIT, --limit LIMIT
                        limit number of split geojson file to at most LIMIT,
                        with GEOMETRY_COUNT number of features.
-v, --verbose         increase output verbosity
-d, --dry-run         see output without actually writing to file
--version             show geojsplit version number
```

By default splitted GeoJSON files are saved as `filename_x<SUFFIX_LENGTH characters long>.geojson`. Default SUFFIX_LENGTH is 4, meaning that 456976 unique files can be generated. If you need more use `-a` or `--suffix-length` to increase this value appropriately.

The `--geometry-count` flag corresponds to the `batch` keyword argument for `GeoJSONBatchStreamer.stream` method. Note that if GEOMETRY_COUNT does not divide equally into the number of features in the Feature Collection, the last batch of features will be < GEOMETRY_COUNT.

Finally, to only iterate over the the first n elements of a GeoJSON document, use `--limit`.

## 1.2 Installation of Geojsplit

**Geojsplit** is pure python, with no C dependencies. It does rely on a few useful third party packages

- ijson - Used for streaming JSON documents
- geojson - Used for serializing valid GeoJSON documents

- simplejson - Used for serializing `decimal.Decimal` objects as a result of ijson parsing.

### 1.2.1 With poetry

For an introduction to poetry.

```
$ poetry add geojsplit
```

will add geojsplit to your current virtual environment and update your poetry.lock file. If you would like to contribute or develop geojsplit

```
$ git clone https://github.com/underchemist/geojsplit.git
$ cd geojsplit
$ poetry install
```

**Note:** You may need some extra configuration to make poetry play nice with conda virtual environments.

```
$ poetry config settings.virtualenvs.path $CONDA_ENV_PATH
$ poetry config settings.virtualenvs.create 0
```

See https://github.com/sdispater/poetry/issues/105#issuecomment-498042062 for more info.

### 1.2.2 With pip

Though geojsplit is developed using poetry (and as such does not have a setup.py), pep517 implementation in pip means we can install it directly.

```
$ pip install geojsplit
```

**Note:** Because there is no setup.py `pip install -e .` will not work. You can get similar behavior with `poetry install`.

## 1.3 geojsplit

### 1.3.1 geojsplit package

#### Submodules

#### geojsplit.geojsplit module

Module for geojson streaming logic

Makes use of the excelent ijson library to stream and parse into python objects a JSON document starting at the *features* item. This assumes that a geojson is in the form

```
{
    "type": "FeatureCollection",
    "features": [
        { ... },
        ...
    ],
    "properties
}
```

**class** geojsplit.geojsplit.**GeoJSONBatchStreamer**(*geojson: Union[str, pathlib.Path]*)

Bases: object

Wrapper class around ijson iterable, allowing iteration in batches

**geojson**

Filepath for a valid geojson document.

**Type** Union[str, Path]

**__init__**(*geojson: Union[str, pathlib.Path]*) → None

Constructor for GeoJSONBatchStreamer

**Parameters** **geojson** (*Union[str, Path]*) – Filepath for a valid geojson document. Will attempt to convert to a Path object regardless of input type.

**Raises** **FileNotFoundError** – If *geojson* does not exist.

**stream**(*batch: Optional[int] = None*, *prefix: Optional[str] = None*) → Iterator[geojson.feature.FeatureCollection]

Generator method to yield batches of geojson Features in a Feature Collection.

**Parameters**

- **batch** (*Optional[int], optional*) – The number of features in a single batch. Defaults to 100.
- **prefix** (*Optional[str], optional*) – The prefix of the element of interest in the geojson document. Usually this should be *'features.item'*. Only change this if you now what you are doing. See https://github.com/ICRAR/ijson for more info. Defaults to *'features.item'*.

**Yields** *(Iterator[geojson.feature.FeatureCollection])* – The next batch of features wrapped in a new Feature Collection. This itself is just a subclass of a Dict instance, containing typical geojson attributes including a JSON array of Features. When *StopIteration* is raised, will yield whatever has been gathered so far in the *data* variable to ensure all features are collected.

## Module contents

CHAPTER

TWO

# INDICES AND TABLES

- genindex
- modindex
- search

# PYTHON MODULE INDEX

## g

# INDEX

## Symbols



## G



## S