Mercantile Documentation

Release 1.0.0

Sean C. Gillies

Oct 22, 2021

Contents

Mercantile is a module of utilities for working with XYZ style spherical mercator tiles (as in Google Maps, OSM, Mapbox, etc.) and includes a set of command line programs built on these utilities.

Project repository: https://github.com/mapbox/mercantile

CHAPTER 1

Contents

1.1 Quick start

In the XYZ tiling system, the region of the world from 85.0511 (more precisely: `arctan(sinh(π))`) degrees south of the Equator to 85.0511 degrees north is covered at zoom level 0 by a single tile. The number of tiles at each zoom level is `4 ** Z`. At zoom level 1, we have 4 tiles.

```
    +-------------+-------------+  85.0511 deg N
    |             |             |
    |    x: 0     |    x: 1     |
    |    y: 0     |    y: 0     |
    |    z: 1     |    z: 1     |
    |             |             |
    +-------------+-------------+   0.0 deg N
    |             |             |
    |    x: 0     |    x: 1     |
    |    y: 1     |    y: 1     |
    |    z: 1     |    z: 1     |
    |             |             |
    +-------------+-------------+  85.0511 deg S

180.0 deg W               180.0 deg E
```

You can get the tile containing a longitude and latitude pair from the `mercantile.tile` function.

```
>>> import mercantile
>>> mercantile.tile(-105.0, 40.0, 1)
Tile(x=0, y=0, z=1)
```

You can get the geographic (longitude and latitude) bounds of a tile from the `mercantile.bounds` function.

```
>>> mercantile.bounds(mercantile.Tile(x=0, y=0, z=1))
LngLatBbox(west=-180.0, south=0.0, east=0.0, north=85.0511287798066)
```

1.2 Installation

Mercantile is easy to install with pip

```
pip install mercantile
```

or with conda.

```
conda install -c conda-forge mercantile
```

1.3 API reference

1.3.1 mercantile package

Module contents

Web mercator XYZ tile utilities

class mercantile.**Bbox**(*left*, *bottom*, *right*, *top*)
Bases: tuple

A web mercator bounding box

Attributes

left, bottom, right, top [float] Bounding values in meters.

Methods

count(value, /)	Return number of occurrences of value.
index(value[, start, stop])	Return first index of value.

bottom
Alias for field number 1

left
Alias for field number 0

right
Alias for field number 2

top
Alias for field number 3

class mercantile.**LngLat**(*lng*, *lat*)
Bases: tuple

A longitude and latitude pair

Attributes

lng, lat [float] Longitude and latitude in decimal degrees east or north.

Methods

count(value, /)	Return number of occurrences of value.
index(value[, start, stop])	Return first index of value.

lat
Alias for field number 1

lng
Alias for field number 0

class mercantile.**LngLatBbox**(*west*, *south*, *east*, *north*)
Bases: tuple

A geographic bounding box

Attributes

west, south, east, north [float] Bounding values in decimal degrees.

Methods

count(value, /)	Return number of occurrences of value.
index(value[, start, stop])	Return first index of value.

east
Alias for field number 2

north
Alias for field number 3

south
Alias for field number 1

west
Alias for field number 0

class mercantile.**Tile**
Bases: *mercantile.Tile*

An XYZ web mercator tile

Attributes

x, y, z [int] x and y indexes of the tile and zoom level z.

Methods

count(value, /)	Return number of occurrences of value.
index(value[, start, stop])	Return first index of value.

mercantile.**bounding_tile**(**bbox*, ***kwds*)
Get the smallest tile containing a geographic bounding box

NB: when the bbox spans lines of lng 0 or lat 0, the bounding tile will be Tile(x=0, y=0, z=0).

Parameters

bbox [sequence of float] west, south, east, north bounding values in decimal degrees.

Returns

Tile

mercantile.**bounds**(**tile*)
Returns the bounding box of a tile

Parameters

tile [Tile or tuple] May be be either an instance of Tile or 3 ints (X, Y, Z).

Returns

LngLatBbox

mercantile.**children**(**tile*, ***kwargs*)
Get the children of a tile

The children are ordered: top-left, top-right, bottom-right, bottom-left.

Parameters

tile [Tile or sequence of int] May be be either an instance of Tile or 3 ints, X, Y, Z.

zoom [int, optional] Returns all children at zoom *zoom*, in depth-first clockwise winding order. If unspecified, returns the immediate (i.e. zoom + 1) children of the tile.

Returns

list

Raises

InvalidZoomError If the zoom level is not an integer greater than the zoom level of the input tile.

Examples

```
>>> children(Tile(0, 0, 0))
[Tile(x=0, y=0, z=1), Tile(x=0, y=1, z=1), Tile(x=1, y=0, z=1), Tile(x=1, y=1,␣
↪z=1)]
>>> children(Tile(0, 0, 0), zoom=2)
[Tile(x=0, y=0, z=2), Tile(x=0, y=1, z=2), Tile(x=0, y=2, z=2), Tile(x=0, y=3,␣
↪z=2), ...]
```

mercantile.**feature**(*tile*, *fid=None*, *props=None*, *projected='geographic'*, *buffer=None*, *precision=None*)
Get the GeoJSON feature corresponding to a tile

Parameters

tile [Tile or sequence of int] May be be either an instance of Tile or 3 ints, X, Y, Z.

fid [str, optional] A feature id.

props [dict, optional] Optional extra feature properties.

projected [str, optional] Non-standard web mercator GeoJSON can be created by passing 'mercator'.

buffer [float, optional] Optional buffer distance for the GeoJSON polygon.

precision [int, optional] GeoJSON coordinates will be truncated to this number of decimal places.

Returns

dict

mercantile.**lnglat**(*x*, *y*, *truncate=False*)

Convert web mercator x, y to longitude and latitude

Parameters

x, y [float] web mercator coordinates in meters.

truncate [bool, optional] Whether to truncate or clip inputs to web mercator limits.

Returns

LngLat

mercantile.**neighbors**(**tile*, ***kwargs*)

The neighbors of a tile

The neighbors function makes no guarantees regarding neighbor tile ordering.

The neighbors function returns up to eight neighboring tiles, where tiles will be omitted when they are not valid e.g. Tile(-1, -1, z).

Parameters

tile [Tile or sequence of int] May be be either an instance of Tile or 3 ints, X, Y, Z.

Returns

list

Examples

```
>>> neighbors(Tile(486, 332, 10))
[Tile(x=485, y=331, z=10), Tile(x=485, y=332, z=10), Tile(x=485, y=333, z=10),␣
↪Tile(x=486, y=331, z=10), Tile(x=486, y=333, z=10), Tile(x=487, y=331, z=10),␣
↪Tile(x=487, y=332, z=10), Tile(x=487, y=333, z=10)]
```

mercantile.**parent**(**tile*, ***kwargs*)

Get the parent of a tile

The parent is the tile of one zoom level lower that contains the given "child" tile.

Parameters

tile [Tile or sequence of int] May be be either an instance of Tile or 3 ints, X, Y, Z.

zoom [int, optional] Determines the *zoom* level of the returned parent tile. This defaults to one lower than the tile (the immediate parent).

Returns

Tile

Examples

```
>>> parent(Tile(0, 0, 2))
Tile(x=0, y=0, z=1)
>>> parent(Tile(0, 0, 2), zoom=0)
Tile(x=0, y=0, z=0)
```

mercantile.**quadkey**(**tile*)
Get the quadkey of a tile

Parameters

tile [Tile or sequence of int] May be be either an instance of Tile or 3 ints, X, Y, Z.

Returns

str

mercantile.**quadkey_to_tile**(*qk*)
Get the tile corresponding to a quadkey

Parameters

qk [str] A quadkey string.

Returns

Tile

mercantile.**simplify**(*tiles*)
Reduces the size of the tileset as much as possible by merging leaves into parents.

Parameters

tiles [Sequence of tiles to merge.]

Returns

list

mercantile.**tile**(*lng*, *lat*, *zoom*, *truncate=False*)
Get the tile containing a longitude and latitude

Parameters

lng, lat [float] A longitude and latitude pair in decimal degrees.

zoom [int] The web mercator zoom level.

truncate [bool, optional] Whether or not to truncate inputs to limits of web mercator.

Returns

Tile

mercantile.**tiles**(*west*, *south*, *east*, *north*, *zooms*, *truncate=False*)
Get the tiles overlapped by a geographic bounding box

Parameters

west, south, east, north [sequence of float] Bounding values in decimal degrees.

zooms [int or sequence of int] One or more zoom levels.

truncate [bool, optional] Whether or not to truncate inputs to web mercator limits.

Yields

Tile

Notes

A small epsilon is used on the south and east parameters so that this function yields exactly one tile when given the bounds of that same tile.

mercantile.**ul**(**tile*)

Returns the upper left longitude and latitude of a tile

Parameters

tile [Tile or sequence of int] May be be either an instance of Tile or 3 ints, X, Y, Z.

Returns

LngLat

Examples

```
>>> ul(Tile(x=0, y=0, z=1))
LngLat(lng=-180.0, lat=85.0511287798066)
```

```
>>> mercantile.ul(1, 1, 1)
LngLat(lng=0.0, lat=0.0)
```

mercantile.**xy_bounds**(**tile*)

Get the web mercator bounding box of a tile

Parameters

tile [Tile or sequence of int] May be be either an instance of Tile or 3 ints, X, Y, Z.

Returns

Bbox

Notes

Epsilon is subtracted from the right limit and added to the bottom limit.

mercantile.**minmax**

Minimum and maximum tile coordinates for a zoom level

Parameters

zoom [int] The web mercator zoom level.

Returns

minimum [int] Minimum tile coordinate (note: always 0).

maximum [int] Maximum tile coordinate (2 ** zoom - 1).

Raises

InvalidZoomError If zoom level is not a positive integer.

Examples

```
>>> minmax(1)
(0, 1)
>>> minmax(-1)
Traceback (most recent call last):
...
InvalidZoomError: zoom must be a positive integer
```

Subpackages

mercantile.scripts package

Module contents

Mercantile command line interface

mercantile.scripts.**configure_logging**(*verbosity*)
Configure logging level

Parameters

verbosity [int] The number of -*v* options from the command line.

Returns

None

mercantile.scripts.**iter_lines**(*lines*)
Iterate over lines of input, stripping and skipping.

mercantile.scripts.**normalize_input**(*input*)
Normalize file or string input.

1.4 Command line interface

```
$ mercantile --help
Usage: mercantile [OPTIONS] COMMAND [ARGS]...

  Command line interface for the Mercantile Python package.

Options:
  -v, --verbose  Increase verbosity.
  -q, --quiet    Decrease verbosity.
  --version      Show the version and exit.
  --help         Show this message and exit.

Commands:
  bounding-tile  Print the bounding tile of a lng/lat point, bounding box, or
                                GeoJSON objects.

  children       Print the children of the tile.
  neighbors      Print the neighbors of the tile.
  parent         Print the parent tile.
  quadkey        Convert to/from quadkeys.
  shapes         Print the shapes of tiles as GeoJSON.
```

(continues on next page)

(continued from previous page)

```
  tiles          Print tiles that overlap or contain a lng/lat point, bounding
                                 box, or GeoJSON objects.
```

1.4.1 Examples

`mercantile shapes` generates GeoJSON from tiles and `mercantile tiles` performs the reverse operation.

```
$ mercantile shapes "[2331, 1185, 12]" | mercantile tiles 12
[2331, 1185, 12]
```

If you have geojsonio-cli installed, you can shoot this GeoJSON straight to geojson.io for lightning-fast visualization and editing.

```
$ echo "[-105, 39.99, -104.99, 40]" \
| mercantile tiles 14 \
| mercantile shapes --collect \
| geojsonio
```

`mercantile parent` and `mercantile children` traverse the hierarchy of Web Mercator tiles.

```
$ mercantile parent "[2331,1185,12]" | mercantile children
[2330, 1184, 12]
[2331, 1184, 12]
[2331, 1185, 12]
[2330, 1185, 12]
```

`mercantile quadkey` will convert to/from quadkey representations of tiles.

```
$ mercantile quadkey "[486, 332, 10]"
0313102310

$ mercantile quadkey 0313102310
[486, 332, 10]
```

1.4.2 bounding-tile

The bounding-tile command writes the input's bounding tile, the smallest mercator tile of any resolution that completely contains the input.

```
$ mercantile bounding-box --help
Usage: mercantile bounding-tile [OPTIONS] [INPUT]

  Print the Web Mercator tile at ZOOM level bounding GeoJSON [west, south,
  east, north] bounding boxes, features, or collections read from stdin.

  Input may be a compact newline-delimited sequences of JSON or a pretty-
  printed ASCII RS-delimited sequence of JSON (like
  https://tools.ietf.org/html/rfc8142 and
  https://tools.ietf.org/html/rfc7159).

  Example:

  echo "[-105.05, 39.95, -105, 40]" | mercantile bounding-tile
```

(continues on next page)

(continued from previous page)

```
  [426, 775, 11]

Options:
  --seq / --lf  Write a RS-delimited JSON sequence (default is LF).
  --help        Show this message and exit.
```

Note that when the input crosses longitude 0 or latitude 0, or any such tile boundary, the bounding tile will be at a shallow zoom level.

```
$ echo "[-1, 1, 1, 2]" | mercantile bounding-tile
[0, 0, 0]
$ echo "[-91, 1, -89, 2]" | mercantile bounding-tile
[0, 0, 1]
```

Compare these bounding tiles to the one for a similarly size input box shifted away from the zoom=1 tile intersection.

```
$ echo "[-92, 1, -91, 2]" | mercantile tiles bounding-tile
[31, 63, 7]
```

1.4.3 children

```
$ mercantile children --help
Usage: mercantile children [OPTIONS] [INPUT]

  Takes [x, y, z] tiles as input and writes children to stdout in the same
  form.

  Input may be a compact newline-delimited sequences of JSON or a pretty-
  printed ASCII RS-delimited sequence of JSON (like
  https://tools.ietf.org/html/rfc8142 and
  https://tools.ietf.org/html/rfc7159).

  Example:

  echo "[486, 332, 10]" | mercantile children
  [972, 664, 11]
  [973, 664, 11]
  [973, 665, 11]
  [972, 665, 11]

Options:
  --depth INTEGER  Number of zoom levels to traverse (default is 1).
  --help           Show this message and exit.
```

1.4.4 neighbors

The neighbors command writes out the tiles adjacent to the input tile.

```
$ mercantile neighbors --help
Usage: mercantile neighbors [OPTIONS] [INPUT]

  Takes [x, y, z] tiles as input and writes adjacent tiles on the same zoom
  level to stdout in the same form.
```

(continues on next page)

(continued from previous page)

```
  There are no ordering guarantees for the output tiles.

  Input may be a compact newline-delimited sequences of JSON or a pretty-
  printed ASCII RS-delimited sequence of JSON (like
  https://tools.ietf.org/html/rfc8142 and
  https://tools.ietf.org/html/rfc7159).

  Example:

  echo "[486, 332, 10]" | mercantile neighbors
  [485, 331, 10]
  [485, 332, 10]
  [485, 333, 10]
  [486, 331, 10]
  [486, 333, 10]
  [487, 331, 10]
  [487, 332, 10]
  [487, 333, 10]

Options:
  --help  Show this message and exit.
```

1.4.5 parent

The parent command writes out the tiles that contain the input tiles.

```
$ mercantile parent --help
Usage: mercantile parent [OPTIONS] [INPUT]

  Takes [x, y, z] tiles as input and writes parents to stdout in the same
  form.

  Input may be a compact newline-delimited sequences of JSON or a pretty-
  printed ASCII RS-delimited sequence of JSON (like
  https://tools.ietf.org/html/rfc8142 and
  https://tools.ietf.org/html/rfc7159).

  Example:

  echo "[486, 332, 10]" | mercantile parent
  [243, 166, 9]

Options:
  --depth INTEGER  Number of zoom levels to traverse (default is 1).
  --help           Show this message and exit.
```

1.4.6 quadkey

The quadkey command converts between [x, y, z] arrays and quadkey strings.

```
$ mercantile parent --help
Usage: mercantile quadkey [OPTIONS] [INPUT]
```

(continues on next page)

(continued from previous page)

```
  Takes [x, y, z] tiles or quadkeys as input and writes quadkeys or a [x, y,
  z] tiles to stdout, respectively.

  Input may be a compact newline-delimited sequences of JSON or a pretty-
  printed ASCII RS-delimited sequence of JSON (like
  https://tools.ietf.org/html/rfc8142 and
  https://tools.ietf.org/html/rfc7159).

  Examples:

  echo "[486, 332, 10]" | mercantile quadkey
  0313102310

  echo "0313102310" | mercantile quadkey
  [486, 332, 10]

Options:
  --help  Show this message and exit.
```

1.4.7 shapes

The shapes command writes tile shapes to several forms of GeoJSON.

```
$ mercantile shapes --help
Usage: mercantile shapes [OPTIONS] [INPUT]

  Print tiles as GeoJSON feature collections or sequences.

  Input may be a compact newline-delimited sequences of JSON or a pretty-
  printed ASCII RS-delimited sequence of JSON (like
  https://tools.ietf.org/html/rfc8142 and
  https://tools.ietf.org/html/rfc7159).

  Tile descriptions may be either an [x, y, z] array or a JSON object of the
  form

        {"tile": [x, y, z], "properties": {"name": "foo", ...}}

  In the latter case, the properties object will be used to update the
  properties object of the output feature.

          Example:

                  echo "[486, 332, 10]" | mercantile shapes --precision 4 --bbox
                  [-9.1406, 53.1204, -8.7891, 53.3309]

Options:
  --precision INTEGER       Decimal precision of coordinates.
  --indent INTEGER          Indentation level for JSON output
  --compact / --no-compact  Use compact separators (',', ':').
  --geographic              Output in geographic coordinates (the default).
  --mercator                Output in Web Mercator coordinates.
  --seq                     Write a RS-delimited JSON sequence (default is
                                                        LF).
```

(continues on next page)

(continued from previous page)

```
  --feature                Output as sequence of GeoJSON features (the
                                                    default).

  --bbox                   Output as sequence of GeoJSON bbox arrays.
  --collect                Output as a GeoJSON feature collections.
  --extents / --no-extents Write shape extents as ws-separated strings
                                                    (default is False).

  --buffer FLOAT           Shift shape x and y values by a constant number
  --help                   Show this message and exit.
```

1.4.8 tiles

With the tiles command you can write descriptions of tiles intersecting with a geographic point, bounding box, or GeoJSON object.

```
$ mercantile tiles --help
Usage: mercantile tiles [OPTIONS] [ZOOM] [INPUT]

  Lists Web Mercator tiles at ZOOM level intersecting GeoJSON [west, south,
  east, north] bounding boxen, features, or collections read from stdin.
  Output is a JSON [x, y, z] array.

  Input may be a compact newline-delimited sequences of JSON or a pretty-
  printed ASCII RS-delimited sequence of JSON (like
  https://tools.ietf.org/html/rfc8142 and
  https://tools.ietf.org/html/rfc7159).

  Example:

  $ echo "[-105.05, 39.95, -105, 40]" | mercantile tiles 12
  [852, 1550, 12]
  [852, 1551, 12]
  [853, 1550, 12]
  [853, 1551, 12]

Options:
  --seq / --lf  Write a RS-delimited JSON sequence (default is LF).
  --help        Show this message and exit.

        $ echo "[-105, 39.99, -104.99, 40]" | mercantile tiles 14
        [3413, 6202, 14]
        [3413, 6203, 14]
```

When supplying GeoJSON as input, you may need to first compact with the help of `jq`

```
$ cat input.geojson | jq -c . | mercantile tiles 14
```

1.5 Changes

1.5.1 1.2.1 (2021-04-21)

- A missing comma in the `__all__` list caused the neighbors and parent method to drop out of the module documentation (#135).

1.5.2 1.2.0 (2021-04-19)

- Copyright holder and date (Mapbox, 2021) have been updated.
- CLI help and documentation has been updated.

1.5.3 1.2b1 (2021-04-14)

There have been no changes since 1.2a1.

1.5.4 1.2a1 (2021-04-12)

Project infrastructure changes:

- Default GitHub branch is now "main".

Deprecations and future warnings:

- The Tile constructor in mercantile 2.0 will refuse to make tiles with X and Y indexes outside of the range 0 <= value <= 2 ** zoom. It will also require indexes to be integers.

New features:

- Moved the `coords()` function in mercantile/script/__init__.py to `_coords()` in mercantile/__init__.py to support `geojson_bounds` (#119).
- Add `geojson_bounds` to get the bounding box of a GeoJSON geometry, feature, or collection (#119).
- Add `minmax` to get minimum and maximum x and y tile coordinates.
- Add `neighbors` function and command to get adjacent tiles (#118).

1.5.5 1.1.6 (2020-08-24)

- In some cases tile() could return a Tile with float x or y attributes (#115). This is a new bug in 1.1.5 and breaks some user code on Python 2.7 and is now fixed.

1.5.6 1.1.5 (2020-06-16)

- A bug in `simplify()` has been fixed and the algorithm has been improved (#111).
- Implementation of `tile()` has been simplified and corrected (#114).

1.5.7 1.1.4 (2020-04-28)

- Change a list comprehension to a generator expression in simplify().
- Change DeprecationWarning introduced in 1.1.3 to a UserWarning to increase visibility.
- Ensure symmetric InvalidLatitudeErrors at both poles (#106).

1.5.8 1.1.3 (2020-04-13)

- Warn about deprecation of support when mercantile is imported with Python versions < 3. Mercantile 2.0 will not be compatible with Python 2.7.
- The bounding tile of the bounds of a tile is now that same tile (#100).

1.5.9 1.1.2 (2019-08-05)

- fid of 0 is now allowed by the `feature()` function (#85).
- Passing the bounds of a tile to `tiles()` now yields exactly that tile only (given the correct zoom), resolving #84 and #87.
- QuadKeyError derives from ValueError again, resolving issue #98, but only until version 2.0. A deprecation warning explaining this is raised from quadkey_to_tile just before QuadKeyError is raised.
- Format source using black.

1.5.10 1.1.1 (2019-07-01)

- Update tests to work with pytest 5.

1.5.11 1.1.0 (2019-06-21)

- A zoom keyword argument has been added to both `children()` and `parent()`, allowing the user to specify the zoom level for each (#94).
- A new `simplify()` function merges child to parent tiles, upwards, producing the shortest sequence of tiles that cover the same space (#94).
- The mercantile module now raises only exceptions deriving from MercantileError. Such errors indicate improper usage of the mercantile module. The occurance of a builtin exception indicates a bug in mercantile.

1.5.12 1.0.4 (2018-06-04)

- Added missing docstrings (#80).

1.5.13 1.0.3 (2018-05-17)

- Support a single zoom value passed to `tiles()` as advertised (#78).

1.5.14 1.0.2 (2018-05-08)

- The `xy` function returns `float(inf)` and `float(-inf)` for y at the North and South pole, respectively, instead of raising an exception (#76).

1.5.15 1.0.1 (2018-02-15)

- Corrected an error in the `bbox` parameter description in the `bounding_tile` docstring (#73).
- Corrected an error in the geojson.io example in the CLI docs: the proper usage is `mercantile shapes --collect` (#71, #72).
- Add missing `--version` option to mercantile command.
- Python 3.6 has been added to the Travis build matrix.

1.5.16 1.0.0 (2017-12-01)

Thanks to all contributors (see AUTHORS.txt), users, and testers, Mercantile 1.0.0 is ready. Share and enjoy!

1.5.17 1.0b2 (2017-11-27)

- Add `tiles` to `__all__` and sort that list. This sorts the classes and functions in the API docs.

1.5.18 1.0b1 (2017-11-21)

- Documentation: overhauled API reference docs based on output of sphinx-apidoc.

1.5.19 1.0a1 (2017-11-16)

- New feature: the `feature` function returns a GeoJSON feature for a tile (#46).

1.5.20 0.11.0 (2017-10-17)

- New feature: the `lnglat` function is the inverse of `xy` (#62).
- New feature: the –bounding-tile option of mercantile-tiles has been made into a new mercantile-bounding-tile command (#43).
- API change: the –bounding-tile and –with-bounds options of mercantile-tiles have been removed.

1.5.21 0.10.0 (2017-05-26)

- API change: `InvalidLatitudeError` is raised by `tile` when Y cannot be computed.
- New feature: `xy_bounds` to get Spherical Mercator bounds for tile (#60).
- New feature: `Bbox` class with `left`, `bottom`, `top`, `right` properties (#60).
- Bug fix: prevent `tiles` from returning tiles with invalid indexes (#47).

1.5.22 0.9.0 (2016-05-20)

- Refactoring: new `normalize_input` and `iter_lines` functions for use in the CLI (#58).
- Refactoring: the coarse `try/except` blocks have been removed from within CLI subcommands, `sys.exit` calls are removed, and `sys.stdout.write` calls have been replaced by `click.echo` (#58).
- Refactoring: many PEP 8 changes and new tests to bring the project to 100% coverage (#58).
- New feature: functions and subcommand for converting between tiles and quadkeys (#50, #51, #56, #57).
- Bug fix: correct output when a point is given to mercantile-tiles (#48, #49).
- Bug fix: more consistent interface for tile arguments (#53).

1.5.23 0.8.3 (2015-08-24)

- Fix error in lng/lat truncation. If lat was > 90, the *lng* was set to a wrong value.

1.5.24 0.8.2 (2014-10-29)

- Add tiles() function (#38).
- Split antimeridian crossing bounds in tiles() (#40).

1.5.25 0.8.1 (2014-10-22)

- Emulate JS >>> operator so we get same results as tilebelt (#36).

1.5.26 0.8 (2014-10-22)

- Streamlining of sequence related options (#35).
- Add customization of output shape ids (#33).

1.5.27 0.7.1 (2014-10-21)

- Make lng/lat truncation optional and off by default (#29).

1.5.28 0.7 (2014-10-21)

- Add customization of output shape properties (#30).

1.5.29 0.6.1 (2014-10-13)

- Guard against lng/lat values off the globe (#27).

1.5.30 0.6 (2014-09-27)

- Add bounding_tile function and tests (#25).
- Add –bounding-tile option to tiles command.

1.5.31 0.5.1 (2014-09-25)

- Let mercantile tiles accept point input as well as bbox or GeojSON.

1.5.32 0.5 (2014-09-24)

- Add mercantile parent and children commands (#17).
- Fix numerical precision bug in roundtripping shapes/tiles (#19).
- Compute bbox if input GeoJSON doesn't provide one (#23).

1.5.33 0.4 (2014-08-21)

- Add option for RS-deliminted JSON sequences.
- Transparent handling of file, stream, and text input (#11).
- Add buffering option (#13).
- Add –extents option to mercantile shapes (#14, #16).
- Round coordinates to proper precision.

1.5.34 0.3 (2014-08-19)

- Add mercator output option for shapes (#9).

1.5.35 0.2.1 (2014-08-19)

- Feature collections as option for shapes command.

1.5.36 0.2 (2014-08-19)

- Added tile() function (#2).
- Add mercantile script (#3).
- Added shapes command (#6).

1.5.37 0.1 (2014-03-26)

- Added mercantile.tool script for use with geojsonio-cli.

1.6 License

1.7 Authors

Mercantile is written by:

- Sean Gillies
- Matthew Perry
- Patrick M Young
- Felix Jung
- Amit Kapadia
- Daniel J. H
- Brendan Ward
- Sam Murphy
- dnomadb
- Andrew Harvey
- Jacob Wasserman
- James Gill
- Jeremiah Cooper
- Michal Migurski
- Pratik Yadav
- Rohit Singh

- Stefano Costa
- drnextgis
- jqtrde

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

m

Index