
postalcodes-ni Documentation

Release 1.1.0

Oscar Cortez

Mar 28, 2019

Contents:

1	postalcodes-ni	1
1.1	Quickstart	1
1.2	Usage	1
1.3	Running Tests	2
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	postalcodes_ni	5
3.1	postalcodes_ni package	5
4	Contributing	7
4.1	Types of Contributions	7
4.2	Get Started!	8
4.3	Pull Request Guidelines	9
4.4	Tips	9
4.5	Deploying	9
5	History	11
5.1	1.0.0 (2019-03-27)	11
6	Indices and tables	13
	Python Module Index	15

CHAPTER 1

postalcodecodes-ni

Python package for handle Nicaragua postal codes

1.1 Quickstart

Install `postalcodecodes-ni`:

```
$ pip install postalcodes-ni
```

Then use it in your project:

```
from postalcodes_ni import (
    get_department_by_iso, get_department_by_postal,
    get_municipality_by_name, get_municipality_by_postal
)
```

This package uses data from those websites:

- [ISO 3166-2:NI](#)
- [Postal Codes](#)

1.2 Usage

`postalcodecodes_ni` exposes the following methods:

- `get_department_by_iso` for get all the municipalities in a department using the ISO code - Ex **MN** for **Managua** department.
- `get_department_by_postal` for get all the municipalities in a department using the postal code - Ex **10000** for **Managua** department.

- `get_municipality_by_name` for get an expecific municipality using his name - Ex **Altagracia** for the **Rivas** department.
- `get_municipality_by_postal` for get an expecific municipality using his postal code - Ex **48800** for the municipality of **Altagracia** from **Rivas** department.

1.3 Running Tests

Does the code actually work?

```
$ source env/bin/activate
(env) $ pip install -r requirements_dev.txt
(env) $ make test-all
```

2.1 Stable release

To install postalcodes-ni, run this command in your terminal:

```
$ pip install postalcodes_ni
```

This is the preferred method to install postalcodes-ni, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for postalcodes-ni can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/oscarcm/postalcodes-ni
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/oscarcm/postalcodes-ni/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


3.1 postalcodes_ni package

3.1.1 Submodules

3.1.2 postalcodes_ni.exceptions module

exception postalcodes_ni.exceptions.ISOCodeError

Bases: Exception

Thrown when ISO code doesnt exists

exception postalcodes_ni.exceptions.PostalCodeError

Bases: Exception

Thrown when postal code doesnt exists

3.1.3 postalcodes_ni.postalcodes module

postalcodes_ni.postalcodes.get_all_municipalities_by_iso(*iso_code: str*) → list

Returns all the municipalities for a department based on the ISO code

postalcodes_ni.postalcodes.get_all_municipalities_by_postal(*postal_code: int*) → list

Returns all the municipalities for a department based on the postal code

postalcodes_ni.postalcodes.get_municipality_by_name(*name: str*) → tuple

Returns a single record for a municipality based on the name

postalcodes_ni.postalcodes.get_municipality_by_postal(*postal_code: int*) → dict

Returns a single record for a municipality based on the postal code

3.1.4 Module contents

Top-level package for postalcodes-ni.

`postalcode_ni.get_all_municipalities_by_iso` (*iso_code: str*) → list

Returns all the municipalities for a department based on the ISO code

`postalcode_ni.get_all_municipalities_by_postal` (*postal_code: int*) → list

Returns all the municipalities for a department based on the postal code

`postalcode_ni.get_municipality_by_name` (*name: str*) → tuple

Returns a single record for a municipality based on the name

`postalcode_ni.get_municipality_by_postal` (*postal_code: int*) → dict

Returns a single record for a municipality based on the postal code

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/oscardmcm/postalcodes-ni/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

postalcodes-ni could always use more documentation, whether as part of the official postalcodes-ni docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/oscarcm/postalcodes-ni/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *postalcodes_ni* for local development.

1. Fork the *postalcodes-ni* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/postalcodes-ni.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv postalcodes_ni
$ cd postalcodes-ni/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 postalcodes_ni tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/oscarcmcm/postalcodes-ni/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_postalcodes_ni
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

5.1 1.0.0 (2019-03-27)

- First release on PyPI.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `postalcodes_ni`, [6](#)
- `postalcodes_ni.exceptions`, [5](#)
- `postalcodes_ni.postalcodes`, [5](#)

G

`get_all_municipalities_by_iso()` (in module `postal-codes_ni`), 6
`get_all_municipalities_by_iso()` (in module `postal-codes_ni.postalcodes`), 5
`get_all_municipalities_by_postal()` (in module `postal-codes_ni`), 6
`get_all_municipalities_by_postal()` (in module `postal-codes_ni.postalcodes`), 5
`get_municipality_by_name()` (in module `postalcodes_ni`), 6
`get_municipality_by_name()` (in module `postal-codes_ni.postalcodes`), 5
`get_municipality_by_postal()` (in module `postal-codes_ni`), 6
`get_municipality_by_postal()` (in module `postal-codes_ni.postalcodes`), 5

I

`ISOCodeError`, 5

P

`PostalCodeError`, 5
`postalcodes_ni` (module), 6
`postalcodes_ni.exceptions` (module), 5
`postalcodes_ni.postalcodes` (module), 5