
bradata Documentation

Release 0.0.post0.dev82+ngac7a25e

odanoburu

Jan 25, 2018

1	Contents	3
1.1	bradata package	3
1.1.1	Subpackages	3
1.1.2	Submodules	5
1.1.3	bradata.connection module	5
1.1.4	bradata.utils module	6
1.1.5	Module contents	6
1.2	Contributing	6
1.2.1	general guidelines	6
1.2.2	code guidelines	6
1.2.3	git workflow	7
1.2.4	contributors	7
1.3	tutorial for beginners	7
1.3.1	step-by-step	7
1.3.2	how-to's	8
1.4	License	8
1.5	Developers	8
1.6	Changelog	9
1.6.1	Version 0.1	9
2	Indices and tables	11
	Python Module Index	13

This is the documentation of **bradata**.

bradata means to make easily available **all** Brazilian government data as a Python package.

it should be as simple as:

```
import bradata  
  
bradata.inep.enem.get()
```

and you should have all ENEM microdata in your /bradata_download directory.

check our source code at [github](#).

1.1 bradata package

1.1.1 Subpackages

bradata.agencias package

Submodules

bradata.agencias.infraero module

`bradata.agencias.infraero.get` (*year='2015'*)

Get all statistics xls files from Infraero website for a given year :param year: string year, from 2017 to 2012
:return: links: A list of all the links downloaded

Module contents

bradata.cgu package

Submodules

bradata.cgu.cgu module

`bradata.cgu.cgu.get_ceaf` (*date=None*)

gets CEAF (Cadastro de Expulsões da Administração Federal, <http://www.transparencia.gov.br/servidores/SaibaMaisPunicoes.asp>) data. it converts the csv encoding to utf8. :param date: a string in YYYY-mm-dd format or a datetime object with year, month, and day attributes. if not provided, will get current day (be careful if on other timezone than Brasília). input can be constructed by importing datetime module and typing `datetime.date(1994, 07, 18)`. :return: downloads csv to directory `bradata.__download_dir__`

`bradata.cgu.cgu.get_ceis` (*date=None*)

gets CEIS (cadastro de empresas inidôneas e suspensas, <http://www.portaldatransparencia.gov.br/ceis>) data. it converts the csv encoding to utf8. :param date: a string in YYYY-mm-dd format or a datetime object with year, month, and day attributes. if not provided, will get current day (be careful if on other timezone than Brasília). input can be constructed by importing datetime module and typing `datetime.date(1994, 07, 18)`. :return: downloads csv to directory `bradata.__download_dir__`

`bradata.cgu.cgu.get_cepim` (*date=None*)

gets CEPIM (Cadastro de Entidades sem Fins Lucrativos Impedidas, <http://www.portaldatransparencia.gov.br/cepim>) data. it converts the csv encoding to utf8. :param date: a string in YYYY-mm-dd format or a datetime object with year, month, and day attributes. if not provided, will get current day (be careful if on other timezone than Brasília). input can be constructed by importing datetime module and typing `datetime.date(1994, 07, 18)`. :return: downloads csv to directory `bradata.__download_dir__`

`bradata.cgu.cgu.get_cgu_data` (*date, cadastro, freq, consulta=None*)

gets some CGU data at <http://www.portaldatransparencia.gov.br/>. it is wrapped by helper functions that make the library more discoverable. it converts the csv encoding to utf8.

Parameters `date` – a string in YYYY-mm-dd format or a datetime object with year,

month, and day attributes. if not provided, will get current day (be careful if on other timezone than Brasília). input can be constructed by importing datetime module and typing `datetime.date(1994, 07, 18)`. :param `cadastro`: this is the database to be fetched (e.g., ‘ceis’) :param `consulta`: usually the same as in `cadastro`, but sometimes the internal API calls it something else, as in the case of CEAF. :param `freq`: ‘d’ for daily, ‘m’ for monthly, ‘y’ or ‘a’ for annually. :return: downloads csv to directory `bradata.__download_dir__`

`bradata.cgu.cgu.get_cnep` (*date=None*)

gets CNEP (Cadastro Nacional de Empresas Punidas, <http://www.portaldatransparencia.gov.br/cnep>) data. it converts the csv encoding to utf8. :param date: a string in YYYY-mm-dd format or a datetime object with year, month, and day attributes. if not provided, will get current day (be careful if on other timezone than Brasília). input can be constructed by importing datetime module and typing `datetime.date(1994, 07, 18)`. :return: downloads csv to directory `bradata.__download_dir__`

`bradata.cgu.cgu.get_diarias` (*date=None*)

gets pagamentos de diárias pagas aos servidores e colaboradores eventuais (<http://www.portaltransparencia.gov.br/despesasdiarias/>) data. it converts the csv encoding to utf8. :param date: a string in YYYY-mm format or a datetime object with year and month attributes. if not provided, will get current day (be careful if on other timezone than Brasília). input can be constructed by importing datetime module and typing `datetime.date(1994, 07, 18)`. :return: downloads csv to directory `bradata.__download_dir__`

Module contents

bradata.tse package

Submodules

bradata.tse.candidatos module

class `bradata.tse.candidatos.Candidatos`

Bases: `object`

Download, organize and pre-process candidatos data from TSE

<http://www.tse.jus.br/eleicoes/estatisticas/repositorio-de-dados-eleitorais>

download (*type=None, year=None*)

Download a certain type of data from a year in the Candidatos option

You can also get several years or types, just pass a list

Types can be:

- candidatos
- bens
- legendas
- vagas

This method covers the following years: 2016, 2014

So, to download candidatos data from 2014, just put `download(type='candidatos', ano=2015)`

Parameters

- **type** – str or list with the type of the data
- **year** – str or int or list with a year

Returns: Saves data to a local data file as `../bradata/tse/[state]/candidatos_[type]_[year].csv`

bradata.tse.utils_tse module

`bradata.tse.utils_tse.aggregate_tse` (*path, type, year*)

`bradata.tse.utils_tse.download_headers` ()

`bradata.tse.utils_tse.unzip_tse` (*result, current_path*)

Module contents

class `bradata.tse.Tse`

Bases: `object`

Gets content from infraero website. It provides a mapping to content types. `statistics` This is the preferred (and only supported) way to get access to those classes and their methods. You can initialize your connection class by:

```
camara = bradata.Infraero()
```

and you'll be ready to use the API on your Python project.

1.1.2 Submodules

1.1.3 bradata.connection module

class `bradata.connection.Connection`

Bases: `object`

Class that handle connections

perform_request (*url, nr_tries=5, binary=False*)

Perform a request handling exception and server errors printing status :param url: string :param nr_tries: int :return: dict :: status : ok/error, content: xml/url, [error_type, error_desc] if error

1.1.4 bradata.utils module

1.1.5 Module contents

1.2 Contributing

note: nothing here is set in stone. if you think something here is misguided, speak to the maintainers.

1.2.1 general guidelines

- **OPEN-SOURCE:** this is an open-source project. therefore, everything in it should be open-source (scripts, documentation, file formats, etc).
- **LANGUAGE:** this project's language is English, even if most of our contributors are Brazilian and we're working with Brazilian data. Our purpose is to make this project welcoming of international contributors and maybe even spread its idea abroad.
- **STANDARDS:** whenever possible use (or convert things to) the international standard. for most data, this will mean changing the encoding from latin1 to UTF-8 and changing the date format from DD/MM/YYYY to YYYY-MM-DD. standardizing will make it easier to work with several databases together. if you find something that should be an exception, open an issue or talk to the coordinators.
- **ATTRIBUTION:** please be aware when employing third-party software: check if their license is compatible with your use. (if unsure, ask). **always** attribute someone else's work to them. similarly, when you complete any work, you must attribute it to yourself under an open-source license. check [here](#) if unsure about a license, or just pick the MIT license which is our default. all files contributed must be prefixed by their license and author in a comment.
- **DOCUMENTATION:** all code must be thoroughly documented. undocumented code or incomprehensible code will not be accepted. choose clarity over performance unless you absolutely have to pick the latter. (hint: [you almost never will.](#))
-

1.2.2 code guidelines

file structure

in the bradata package, every smodule is an institution (data provider). at its directory, its `__init__.py` should contain the functions and classes that are to be available to the public, and **nothing else**. that's because the preferred way for a user to use the bradata package is to explore what it has to offer by tab-completion available at ipython and jupyter notebook, as the package is projected to have a number of functions greater than what a user would like to memorize.

importing only the public functions in the `__init__.py` file prevents the namespace from being crowded with private objects:

```
import bradata.tse as tse
tse.get_candidatos()
```

submodules should be divided by similarity or proximity, for instance `bradata/cgu/_cadastros.py` has functions to get three different databases, but as the code to get them is mostly the same they reside together. (the three functions are actually only one function and two wrappers, to prevent writing more code than we need to). if the

submodule is not meant to be called by the user, it should start with an underscore (`_`), so that it doesn't pollute the namespace.

1.2.3 git workflow

so you've forked the repo and added some nice functionality, or correct some bug. thank you very much! but before we can accept your work, you must follow a few simple procedures:

- document every function, class, module, etc. you create or change, preferably using [google-style docstrings](#). if you are implementing some tricky part, we'd appreciate if you wrote a tutorial or some kind of extensive documentation. we autogenerate documentation using sphinx, and you may write in `.md` or `.rst`, but please write.
- always `git pull [source-repo] master` before making a pull request!
- if you created a new public module or submodule, import it in the `__init__.py` of the main package.
- add your name to the *Developers*;

1.2.4 contributors

contributors are listed under *Developers*. only people who have had a pull request accepted are listed as contributors.

1.3 tutorial for beginners

1.3.1 step-by-step

let's review the steps to start contributing:

- Fork the project to your account.
- Choose a path in your computer to store the project, go to it.
- Clone the fork that you have just done to this path using the terminal command

```
git clone https://github.com/YOUR-USERNAME/bradata
```
- At this point, you should have an exact copy of the latest version of the project on your machine.

Congratulations! Now you have a version of the repository in your machine. If you want to contribute and help to build this incredible project, keep reading!

- do your modifications.
- now you must check if your version is up-to-date with the original repository:

```
git pull https://github.com/labFGV/bradata
```
- if you have a merge conflict, you must solve it before committing your work.
- now you stage and commit your work:

```
git add YOUR_FILES
git commit -m "YOUR COMMIT MESSAGE"
```

- now you push the changes to your repo: `git push origin master`
- finally, you go to <https://github.com/labFGV/bradata> and complete your pull request.

1.3.2 how-to's

google and stackoverflow are your best friends, but:

- git:
 - dudler's simple guide: [en](#), [pt-br](#)
 - [official docs](#)
 - [oh, sh*t, git!](#)
 - [github guides & help](#)
 -
- markdown:
 - [learn in 60 seconds](#)
 - [syntax specs](#)
 - [syntax philosophy and some specs](#)
 -
- Restructured text
 - 'Restructured text and sphinx <http://thomas-cokelaer.info/tutorials/sphinx/rest_syntax.html>'
 -
-

1.4 License

Copyright 2017 AUTHORS

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.5 Developers

- [odanoburu <bcclaro+bradata@gmail.com>](mailto:odanoburu@bcclaro+bradata@gmail.com)
- [Joao Carabetta <joao.carabetta@gmail.com>](mailto:Joao.Carabetta@gmail.com)
-

1.6 Changelog

1.6.1 Version 0.1

- **add TSE module**
 - candidatos
 - bens dos candidatos
- **add CGU module**
 - CEIS
 - CEPIM
 - CNEP

CHAPTER 2

Indices and tables

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

b

bradata, 6
bradata.agencias, 3
bradata.agencias.infraero, 3
bradata.cgu, 4
bradata.cgu.cgu, 3
bradata.connection, 5
bradata.tse, 5
bradata.tse.candidatos, 4
bradata.tse.utils_tse, 5
bradata.utils, 6

A

aggregate_tse() (in module bradata.tse.utils_tse), 5

B

bradata (module), 6

bradata.agencias (module), 3

bradata.agencias.infraero (module), 3

bradata.cgu (module), 4

bradata.cgu.cgu (module), 3

bradata.connection (module), 5

bradata.tse (module), 5

bradata.tse.candidatos (module), 4

bradata.tse.utils_tse (module), 5

bradata.utils (module), 6

C

Candidatos (class in bradata.tse.candidatos), 4

Connection (class in bradata.connection), 5

D

download() (bradata.tse.candidatos.Candidatos method),
4

download_headers() (in module bradata.tse.utils_tse), 5

G

get() (in module bradata.agencias.infraero), 3

get_ceaf() (in module bradata.cgu.cgu), 3

get_ceis() (in module bradata.cgu.cgu), 3

get_cepim() (in module bradata.cgu.cgu), 4

get_cgu_data() (in module bradata.cgu.cgu), 4

get_cnep() (in module bradata.cgu.cgu), 4

get_diarias() (in module bradata.cgu.cgu), 4

P

perform_request() (bradata.connection.Connection
method), 5

T

Tse (class in bradata.tse), 5

U

unzip_tse() (in module bradata.tse.utils_tse), 5