
Python bridge for IceNLP Documentation

Release 0.1.0

Sverrir Á. Berg

Sep 08, 2019

Contents:

| | | |
|----------|-----------------------------------|-----------|
| 1 | Python bridge for IceNLP | 1 |
| 1.1 | Features | 1 |
| 1.2 | Example | 1 |
| 1.3 | Credits | 2 |
| 2 | Installation | 3 |
| 2.1 | Stable release | 3 |
| 2.2 | From sources | 3 |
| 3 | Usage | 5 |
| 4 | icenlp_bridge | 7 |
| 4.1 | icenlp_bridge package | 7 |
| 5 | Contributing | 9 |
| 5.1 | Types of Contributions | 9 |
| 5.2 | Get Started! | 10 |
| 5.3 | Pull Request Guidelines | 11 |
| 5.4 | Tips | 11 |
| 5.5 | Deploying | 11 |
| 6 | Credits | 13 |
| 6.1 | Development Lead | 13 |
| 6.2 | Contributors | 13 |
| 7 | History | 15 |
| 7.1 | 0.1.0 (2019-09-08) | 15 |
| 8 | Indices and tables | 17 |
| | Python Module Index | 19 |
| | Index | 21 |

Simplify using IceNLP (Icelandic NLP library written in Java)

- Free software: Apache Software License 2.0
- Documentation: <https://icenlp-bridge.readthedocs.io>.

1.1 Features

- Run Icelandic NLP from python using a local IceNLP server and this library

1.2 Example

Start the local IceNLP server. For example using docker:

```
docker run -it --rm -p 1234:1234 sverrirab/icenlp
```

Install locally using:

```
pip install icenlp_bridge
```

Your code might look like this:

```
>>> from icenlp_bridge import parse
>>> print(parse('Vá hvað þetta var einfalt!'))
```

```
[InjP Vá au ] { *SUBJ [NP hvað fshen ] } { *SUBJ> [NP þetta fahen ] } [VPb var
sfg3ep ] { *COMP< [AP einfalt lhensf ] } ! !
```

You can also pipe files into `icenlp_bridge`

```
cat texti.txt | python -m icenlp_bridge
```

For connection options see `python -m icenlp_bridge -help` and `init` documentation.

1.3 Credits

This package is built on the excellent [IceNLP](#) library. Information on the docker image can be found on [DockerHub](#).

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install Python bridge for IceNLP, run this command in your terminal:

```
$ pip install icenlp_bridge
```

This is the preferred method to install Python bridge for IceNLP, 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 Python bridge for IceNLP can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/sverrirab/icenlp_bridge
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/sverrirab/icenlp_bridge/tarball/master
```

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

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


CHAPTER 3

Usage

To use Python bridge for IceNLP in a project:

```
from icenlp_bridge import parse
print(parse('Vá hvað þetta var einfalt!'))
```

*[InjP Vá au] { *SUBJ [NP hvað fshen] } { *SUBJ> [NP þetta fahen] } [VPb var sfg3ep] { *COMP< [AP einfalt lhensf] } !!*

4.1 icenlp_bridge package

4.1.1 Submodules

4.1.2 icenlp_bridge.bridge module

```
icenlp_bridge.bridge.icenlp_client (message: str) → Awaitable[str]
icenlp_bridge.bridge.init (hostname: Optional[str] = None, port: Optional[int] = None) → NoReturn
    Initialize IceNLP connection
icenlp_bridge.bridge.parse (text: str) → str
    Parse text with IceNLP
icenlp_bridge.bridge.writeData (writer: Any, data: bytes) → NoReturn
icenlp_bridge.bridge.writeHeader (writer: Any, num_packets: int) → NoReturn
```

4.1.3 Module contents

Top-level package for Python bridge for IceNLP.

```
icenlp_bridge.parse (text: str) → str
    Parse text with IceNLP
icenlp_bridge.init (hostname: Optional[str] = None, port: Optional[int] = None) → NoReturn
    Initialize IceNLP connection
```


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

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/sverrirab/icenlp_bridge/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

Python bridge for IceNLP could always use more documentation, whether as part of the official Python bridge for IceNLP docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/sverrirab/icenlp_bridge/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 :)

5.2 Get Started!

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

1. Fork the *icenlp_bridge* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/icenlp_bridge.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 icenlp_bridge
$ cd icenlp_bridge/
$ 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 icenlp_bridge 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.

5.3 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 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/sverrirab/icenlp_bridge/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_icenlp_bridge
```

5.5 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.

6.1 Development Lead

- Sverrir Á. Berg <sab@keilir.com>

6.2 Contributors

None yet. Why not be the first?

7.1 0.1.0 (2019-09-08)

- First release on PyPI.

CHAPTER 8

Indices and tables

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

i

`icenlp_bridge`, [7](#)

`icenlp_bridge.bridge`, [7](#)

I

`icenlp_bridge (module)`, 7
`icenlp_bridge.bridge (module)`, 7
`icenlp_client () (in module icenlp_bridge.bridge)`,
7
`init () (in module icenlp_bridge)`, 7
`init () (in module icenlp_bridge.bridge)`, 7

P

`parse () (in module icenlp_bridge)`, 7
`parse () (in module icenlp_bridge.bridge)`, 7

W

`writeData () (in module icenlp_bridge.bridge)`, 7
`writeHeader () (in module icenlp_bridge.bridge)`, 7