PyJson2C Documentation

Release 0+untagged.8.g2fdccfe.dirty

jmramosr

Jul 02, 2019

Contents

Contents:

CHAPTER 1

PyJson2C

Travis CI (Linux Build and Integration Tests)	
Coveralls coverage support	
Documentation Status	

SonarCloud			

Json translator into C code made with Python

- Free software: GPLv3
- Documentation: https://PyJson2C.readthedocs.org.

1.1 Features

- Parses Json files and translates them into C files or headers.

1.1.1 ROADMAP CHECKLIST

- **Architect the repository**
  - **Create the Github repository ✓**
    - Create a complete python project in Github
  - **SonarCloud ✓**
    - Link SonarCloud with the project
  - **Coveralls**

        - Link Coveralls with the project
    - **Travis ✓**
        - Link Travis with the project
    - **ReadTheDocs**
        - Link ReadTheDocs with the project
        - Generate the ReadTheDocs link to the project
        - Generate the documentation
- **TDD**
    - **System tests**
        - Generate a c or h file from a json file, returning a valid file, with custom spacing.
    - **Integration tests**
        - Create Custom Spacing Options
        - Create Environment
        - Create Language extensions
        - Create Documentation
        - Create Character sets
        - Create Standard libraries access
    - **Unit tests**
        - Create from json elements, valid C snippets
        - Create Identifiers
        - Create Types
        - Create Constants
        - Create Declarations and definitions
        - Create Initialisation
        - Create Arithmetic type conversions
        - Create Pointer type conversions
        - Create Expressions
        - Create Control statement expressions
        - Create Control flow
        - Create Switch statements
        - Create Functions
        - Create Pointers and arrays
        - Create Structures and unions
        - Create Preprocessing directives
- **Wishlist**
    - Create a good Python codebase to be useful to users

  - Create a complete Json translator into C code
  - C_Ansi, C99, C11

`Emojigend (emoji legend):`

means I think the part does not need nothing more

means I think the part could need something more

means a complete chapter

means an incomplete chapter

✓ means a complete section

means an incomplete section

means a complete item

means an incomplete item

CHAPTER 2

Installation

At the command line:

```
$ pip install PyJson2C
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv PyJson2C
$ pip install PyJson2C
```

CHAPTER 3

Usage

To use PyJson2C in a project:

```python
import PyJson2C
```

CHAPTER 4

Contributing

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/jmramosr/PyJson2C/issues.

If you are reporting a bug, please include:

- 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" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

4.1.4 Write Documentation

PyJson2C could always use more documentation, whether as part of the official PyJson2C 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/jmramosr/PyJson2C/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 *PyJson2C* for local development.

1. Fork the *PyJson2C* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:jmramosr/PyJson2C.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 PyJson2C
   $ cd PyJson2C/
   $ 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 PyJson2C tests
   $ python setup.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 2.7, 3.3, 3.4, 3.5, 3.6 and for PyPy. Check https://travis-ci.org/jmramosr/PyJson2C/pull_requests and make sure that the tests pass for all supported Python versions.

CHAPTER 5

Credits

5.1 Maintainer

- jmramosr <https://github.com/jmramosr>

5.2 Contributors

None yet. Why not be the first? See: CONTRIBUTING.rst

CHAPTER 6

History

Pre-alpha. Setting up the environment.

6.1 HISTORY CHECKLIST

- **Architect the repository**
  - **Create the Github repository ✓**
    - Create a complete python project in Github
  - **SonarCloud**
    - Link SonarCloud with the project
  - **Coveralls**
    - Link Coveralls with the project
  - **Travis**
    - Link Travis with the project
  - **ReadTheDocs**
    - Link ReadTheDocs with the project
    - Generate the ReadTheDocs link to the project
    - Generate the documentation

CHAPTER 7

Indices and tables

- genindex
- modindex
- search