QTFeet Documentation

Release 0.1.0

hugosenari

December 19, 2016

Contents

Contents:

CHAPTER 1

"QTFeet"

"QTFeet, DBus instrospection tool, DFeet clone writed with QT"

- Free software: BSD license
- Documentation: http://qtfeet.rtfd.org.

1.1 Features

DONE

- Nothing

TODO

- Plugin System (all this app are plugin)
- Show available buses in dbus connections
- Show available interfaces in bus
- Show available methods in interface
- Show available properties in interface
- Show available signals in interface
- Show communications events (dbus-monitor)
- Convert interfaces to code
- Call methods in interface
- Get properties in interface (shortcut to call GetProperties)

1.2 Some Answers

Why clone D-Feet?

I like D-Feet, is useful to me. I Have nothing to do in my vacation and do something completely new is very hard, then I choose clone something to learn and tests some concepts.

Why QT?

Why not? QT is now GPL, cross platform... This is my first QT attempt, then is something new to learn.

How looks (or will look) architecture?

To be honest I'm trying create something like Eclipse Framework that in the end can used to create aplications for other uses.

- QT (for UI and some libs like DBus)
- iPOPO (python implementation of OSGi or something close to this)

CHAPTER 2

Installation

At the command line:

```
$ easy_install qtfeet
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv qtfeet
$ pip install qtfeet
```

CHAPTER 3

Usage

To use “QTFeet” in a project:

```
import qtfeet
```

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/hugosenari/qtfeet/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" 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

"QTFeet" could always use more documentation, whether as part of the official "QTFeet" 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/hugosenari/qtfeet/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 *qtfeet* for local development.

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

```
$ git clone git@github.com:your_name_here/qtfeet.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 qtfeet
$ cd qtfeet/
$ 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 qtfeet 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 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/hugosenari/qtfeet/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

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

CHAPTER 5

Credits

5.1 Development Lead

- hugosenari <hugosenari@gmail.com>

5.2 Contributors

Thomas Calmant author of QT-iPOPO (http://ipopo.coderxpress.net/tutorials/qt.html)

CHAPTER 6

History

6.1 0.1.0 (2013-08-11)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search