

Python Argentina -PyAfipWS

Continuous Integration, Deployment & Fixes with python

Hi, I wanted to work with the PyAfipWS project of Python Argentina.
This project which will allow me to practice with webservices developments.

Summary:

PyAfipWs is a Open Source project to provide interfaces, tools and multiplatform applications to access web services mainly electronic bill (among others) of the AFIP - Argentina.

Multiplatform and compatible with other languages (Visual Basic, Visual Fox Pro, Delphi, PHP, .Net, Java, etc.) and tools by command line.

The web service uses the SOAP communication protocol, which regulates the exchange of messages in XML format

Main Project Site: <https://github.com/reingart/pyafipws>

About Me:

- Guillermo Nicolas Sandoval
- From Buenos Aires, Argentina (GMT-3)
- National University of Moreno
- Engineering student in Electronics with orientation in Networks

Code contribution:

★ #65

Skills:

Python, C, Django, Virtualenv, CMD, Bash, CSS, Html, SQL.

Project Goals:

Initial work has been started in the py3k branch, but there are many rough edges and enhancements opportunities.

The idea is to maintain the same the bases and stability of the initial project (Python2) in its version of Python 3, update port the corresponding changes.

To be able to achieve the bases for the implementation of the continuous integration.

As it's multi-platform, used by a wide user base of developers (specially from other programming languages), special care must be taken for backward compatibility.

Minimum expected results:

- Stabilize Python3
- Review library support (m2crypto)
- Fix Python3 runtime unicode
- Unit tests 70 % coverage - test in all weeks
- Continuous Integration , Deployment & PyPI
- WebService Simulator for testing

Time work structure in weeks:

- 1. May 27-31:** Stabilize Python3 - 1 stage
 - Compare versions
- 2. June 3:** Stabilize Python3 - 2 stage
 - Basic code testing - update
- 3. June 10:** Stabilize Python3 - 3 stage
 - Test requirements
- 4. June 17:** Stabilize Python3 - 4 stage
 - Multiplatform check
- 5. June 24:** Review library support (m2crypto)
- 6. July 1:** Continuous Integration & Deployment with Travis CI - 1 stage
 - Setup
- 7. July 8:** Continuous Integration & Deployment with Travis CI - 2 stage
 - Integration tests / fixtures
- 8. July 15:** Generate Unit Test
- 9. July 22 :** WebService Simulator for testing
- 10. July 29 :** Fix Python3 runtime unicode
- 11. August 5 :** Checking code

12. August 12 : Complete tests/documentation - deployment to PyPI

13. Final week -August 19: Submit Project