

Dist, Docs, and Testing with OSS on GitHub

Craig Kelly

November 16, 2015

Hello

I'm Craig, a Research Developer for the Institute for Intelligent Systems at the University of Memphis (<http://www.memphis.edu/iis/>).

Our mission depends on an interdisciplinary approach that brings together researchers from many different research areas in the cognitive sciences, including biology, communication sciences and disorders, computer science, education, engineering, linguistics, philosophy, physics, and psychology.

Why Does This Matter?

Grad students!

They generally aren't experienced coders. In some cases they are just learning to program. And they're generally more concerned with DOE than making the database work. So...

What is GLUDB?

(And why should you care?)

- Simplified for the demands of working academics and grad students
- Annotate and go!
- Active Record (-ish)
- Documented-oriented data store with support for indexing
- Includes optional functionality for object change history and backups
- Allows easily moving from dev to server to cloud
- Supports sqlite, MongoDB, Google Cloud Datastore, and Amazon DynamoDB

Is GLUDB for you?

Probably not.

If you want a good, solid ORM then use SQLAlchemy. Or Django's ORM is you're Django'ing. Especially if you care about migrations.

GLUDB can be useful for rapid iteration, hobby projects, or when you need a nice “object store”

Dist, Docs, and Testing

- Dist via PiPy (pipy.python.org)
- Docs via Read The Docs (readthedocs.org)
- Testing via Travis CI

Overview

- Free account!
- Pulls from your repo automatically when you push changes
- Supports different doc types
- Has a nice template, and provides downloads

How to Implement

- You write docs in reStructuredText or Markdown
- You push the docs to your GitHub repo
- RTD pull the docs, builds them, and shows them on the web

Working Example

- We used Markdown - you should probably use Sphinx
- See our mkdocs.yml and our docs directory
- Published at <http://gludb.readthedocs.org/en/latest/>

Overview

- Continuous Integration for free! (for open source repos on GitHub)
- Pulls from your repo automatically when you push changes
- Multiple environments (e.g. test both Python 2 and 3)
- Supports lots of languages
- Provides lots of services for testing
(<http://docs.travis-ci.com/user/database-setup/>)
- Allows for container customization if what need isn't in the box

How to Implement

- You have tests to run!
- You log in to travis-ci.org with your GitHub account
- You set up your repository
- You create a .travis.yml file
- When you push to your repository your tests are run

Working Example

- See our `travis.yml` and tests
- We run tests for both Python 2.7 and 3.4
- We use the Travis CI MongoDB service
- We use a manual process to run backing services that we need using supervisor. See `start_ci_services.sh` (See next slide)

Test Services Startup Script

- Uses npm to install the dynalite server
- Uses wget to download and extract the gcd test server
- Uses virtualenv and pip to set up a Python 2.7 environment for supervisor and our S3 test server
- We even run a Python script as a service (s3server.py) that we copied from Facebook and modified!
- This is all very cool, but also check out our sleep hack :)

Overview

Do you want to do `pip install my-cool-package?`

Then you need to be on PyPI (<https://pypi.python.org/pypi>)

Not to be confused with the PyPy project at pypy.org

How to Implement

- You get your account at pypi.python.org
- You create a setup script for your package
- You upload to PyPI

Working Example

- First we need a setup script - setup.py (and setup.cfg) that includes dependencies. (IMPORTANT: your requirements.txt file will not get used)
- It also helps to have a wrapper script like build.sh
- You can handle uploads using twine
- If you like Markdown for your README (who doesn't), you can use pandoc for conversion
- Note that you probably won't need the extras

Don't Panic

RTD and Travis have excellent documentation (and PyPI's isn't bad).

They're also well-covered at StackOverflow and the usual places

Thank You!

Thanks for listening