

Doing The
Jitterbug

Cross
Language
Continu-
ous
Integra-
tion for
Git

@dukeleto

Doing The Jitterbug

Cross Language Continuous Integration for Git

Jonathan "Duke" Leto



What is Continuous Integration (CI) ?

Continually and automatically testing code, usually with relevant notifications.

Why is CI Useful?

- Know the exact commit that broke something
- Automate testing many different combinations of different versions of libraries, languages, OS's, browsers, etc...
- Quickly identifies tests that only pass on the authors machine due to implicit assumptions

What problems does Jitterbug solve?

- People forgetting to run the test suite
- People forgetting to notify others when they see breakage
- Not having a visual interface to which commits passed and which failed a test suite

Current Jitterbug Features

- Extremely Memory Efficient
- Integrates seamlessly with Github post-receive hooks
- Can autodetect test suites in many languages
- Highly customizable YAML configuration file
- Email+RSS notifiers
- Supports custom build/test scripts
- Pretty web interface

What does Jitterbug look like?

Dashboard

Repositories (4)

jitterbug

Cross Language Continuous Integration for Git

last updated at 2011-04-29T19:29:33-07:00 by Jonathan "Duke" Leto

parrot

Parrot Virtual Machine

last updated at 2011-04-29T18:46:29-07:00 by Peter Lobsinger

parrot-embed

Embed Parrot in Perl 5

last updated at 2011-03-16T20:53:06-07:00 by Jonathan "Duke" Leto

test

This Universe For Testing Purposes Only

What does Jitterbug look like? (2)

Dashboard / jitterbug / build history 

"Cross Language Continuous Integration for Git"

2011-04-29

Make the command-line arg of the builder --config to match
jitterbug_db



Jonathan "Duke" Leto

2011-04-29T19:29:33-07:00

perl-v5.10.1 **PASS**

commit 7d4f2cd6db48e527f4822fe318ff-
on refs/heads/master
built in 14 seconds

Add more docs about the build_process config key



Jonathan "Duke" Leto

2011-04-29T19:46:18-07:00

perl-v5.10.1 **PASS**

What does Jitterbug look like? (3)

Create a document Expand all Collapse all Forward all Turn on highlighting Turn off highlighting Translate cs

[Cxgn-devel] [jitterbug] FAIL sgn @ 5768ba81 make broken-image tests work with relative links too [cxgn](#) | [X](#)

☆ jitterbug@sgn.cornell.edu to rmb32, cxgn-devel [show details](#) 11:29 AM (11 hours ago) [Reply](#)

Jitterbug Web Interface : <http://jitterbug.sgn.cornell.edu:8090>
sgn : <http://jitterbug.sgn.cornell.edu:8090/project/sgn>
Failing Commit Diff: <https://github.com/solgenomics/sgn/commit/5768ba8129e627cb365a011ee81f844f57cacc11>

Test Summary Report

t/SGN/Controller/Organism.t	(Wstat: 65280 Tests: 5 Failed: 0)
Non-zero exit status: 255	
Parse errors: No plan found in TAP output	
t/cview/map.t	(Wstat: 512 Tests: 90 Failed: 2)
Failed tests: 89-90	
Non-zero exit status: 2	
t/image/add_image.t	(Wstat: 65280 Tests: 0 Failed: 0)
Non-zero exit status: 255	
Parse errors: No plan found in TAP output	

Current Features

- Integrates seamlessly with Github post-receive hooks
- Can run tests for Perl 5/6, Parrot, Ruby and Makefile-based projects
- Highly customizable YAML configuration file
- Email notifier
- Supports custom build/test scripts
- Pretty web interface

Getting Jitterbug

- `git clone git://github.com/franckcuny/jitterbug.git`
- `cd jitterbug`
- `perl Build.PL`
- `cpanm --installdeps . # cpanminus to install deps`
- OR
- `./Build installdeps # install deps`
- `./Build test # run tests`

Will release to CPAN Real Soon Now

Doing the Jitterbug

- `vi config.yml # Customize YAML config`
- `perl jitterbug.pl -p 8080 # Web interface`
- `perl scripts/jitterbug_db --config config.yml --deploy`
- `perl scripts/builder.pl -c config.yml`
- Add `http://example.com:8080/hook/` as a Github post-receive URL

Customizing Jitterbug

Interesting config file options...

- `reuse_repo` - Reuse git repos instead of recloning
- `stack_tasks` - Test every commit, not just current HEAD
- `perlbrew` - use perlbrew to test under many versions/builds of Perl
- `email_on_pass` - whether to send emails for PASSing test suites

Customizing Jitterbug (2)

SQLite by default, but any database can be used

```
plugins:
```

```
  DBIC:
```

```
    schema:
```

```
      skip_automake: 1
```

```
      pkg: "Jitterbug::Schema"
```

```
      connect_info:
```

```
        - dbi:SQLite:dbname=jitterbug.db
```

Customizing Jitterbug (3)

Per-repo config

```
jitterbug :  
  options :  
    perlbrew : 1  
    email_on_pass : 0  
    branches :  
      parrot: master|smoke me  
    projects :  
      rakudo :  
        builder: ./scripts/custom build.sh
```

Future Goals

- Jitterbug wants to support running tests in many more languages, including
 - Python
 - Javascript
 - PHP
- Code coverage
- Graphic visualization

Get involved!

- Install Jitterbug and test your projects
- Join #dancer on irc.perl.org for help
- Submit a pull request for an issue!
<https://github.com/franckcuny/jitterbug>
- <http://lumberjaph.net/jitterbug>

Cross Language Continu- ous Integra- tion for Git

@dukeleto

Thanks!

- Franck Cuny
- Dancer team

Stalk Me

- <http://dukeleto.pl>
- <http://linkedin.letto.net>
- <http://twitter.com/dukeleto>
- <http://identi.ca/letto>
- dukeleto on irc.perl.org and Freenode