citools Documentation

Release 0.1

Centrum Holdings

September 20, 2015

Ultimate goal of Continuous Integration tools is to provide “integration button”, magic key hit that will do everything needed for product to build & deploy. While there are some tools available, none of those fullfills our real-world needs.

Basic idea is simple: use setup.py in similar fashion as Makefiles or ant, but take advantage of setuptools plugin system to provide globally available commands, easily configurable for every project.

We are making some assumptions about project structure and needs. If you need to configure something more, let us know, or just fork us at github and send us Your pull request.

Licensed under BSD, this library is maintained by Ella team from Centrum Holdings. For feedback, ideas, bug reports and friends, let us know in mailing list.

CHAPTER 1

On (continuous) versioning

Idea is simple: if you should be able to deploy software any time, every revision must have a release number. Thus, we have a “stable” version prefix, which we assume to be set by tag. Last digit in version is build number, thus number of commits since last tag [1]. Number of digits in your version is arbitrary, but must be at least three (two for version prefix, like projectname-1, and one for build number).

Then, version must be replaced in all files needed. We’re now rewriting in following form in following places:

1. VERSION in $project/__init__.py is set to version tuple (not string). We’re assuming layout as in our django-base-library (which is actually not much about Django).
2. __versionstr__ (if found) in setup.py is replaced to string (not tuple). This is for libraries that must not import library itself and set version to $library.__versionstr__ dynamically
3. TODO: debian ,)

[1] (TODO: Following is actually not yet supported; we’re now assuming only version setting tags) “Last tag” means “last tag that is setting project version”. We support other tags, so either you must use $projectname-$version tags, or pass version_regexp argument to setuptools.setup in setup.py, which must be in form TODO

"Meta" packages

There may be need for creating "meta" packages, that do not contain any code, but just specifies which packages should be bundled together (and in which version). Citools supports this behaviour by automatic computation of metapackage version and specifying exact version (especially for debian).

You must only specify repositories You depend upon in arguments for setuptools.setup (only git supported for now)::

> **dependency_repositories = [** "ssh://server/my/library1", "http://github.com/myuser/library/",
>
>]

By running `setup.py compute_meta_version_git`, library version will be computed as it's own version + sum of all versions of all child libraries.

Specification via setup.py argument will be deprecated, you should use command line arguments instead.

CHAPTER 3

(Django) web environment

CHAPTER 4

Build process

CHAPTER 5

Testing

CHAPTER 6

Working with databases

Downloading and restoring backup:

```
[backup]
realm=backuprealm
username=blah
password=xxx
uri=https://my.backup.server.cz/my/dir/backup_archive.tar.gz
    tempdir=/if/you/need/bigger/tempdir/to/unpack/archive/

[database]
file=my/database/backup/db.sql
name=dbname
username=buildbot
password=xxx

citools -c /etc/$project/citools.ini db_restore
```

Supported backup formats are `.tar.gz|bz2`, `.sql.gz|bz2` and plain `.sql`. When you use tar archive with more databases, you can restore more databases at once, if you define more sections with prefix `database` in your ini config file:

```
[backup]
realm=myrealm
username=blah
password=xxx
uri=https://my.backup.server.cz/my/dir/backup_archive.tar.gz
tempdir=/if/you/need/bigger/tempdir/to/unpack/archive/

[database_first]
file=path/in/tarfile/database_first.sql
name=database_first
username=firstman
password=""

[database_second]
file=path/in/tarfile/database_second.sql
name=database_second
username=secondman
password=xxx
```

when run as setup.py db_restore, /etc/$project/citools.ini is the default

CHAPTER 7

Distribution and Deployment