
django-discover-jenkins **Documentation**

Release 0.1.0

Lincoln Loop

September 28, 2015

1	Installation	3
1.1	Configuration	3
1.2	Usage	3
2	Settings	5
3	Tasks	7
3.1	CoverageTask	7
3.2	PyLintTask	8
3.3	JSHintTask	8
3.4	SlocCountTask	9
3.5	Pep8Task	9
3.6	Flake8Task	9
4	Why?	11
5	What's Changed?	13
6	Who?	15

A streamlined fork of django-jenkins designed to work with the default test command and the discover runner.

Contents:

Installation

From PyPI:

```
pip install django-discover-jenkins
```

Due to a bug in the coverage library you have to use this specific version:

```
pip install coverage==3.5
```

1.1 Configuration

Add `discover_jenkins` to your `INSTALLED_APPS` and set `TEST_RUNNER` to the `DiscoverCIRunner` that `discover_jenkins` provides:

```
INSTALLED_APPS = (  
    ...  
    'discover_jenkins',  
    ...  
)  
  
TEST_RUNNER = 'discover_jenkins.runner.DiscoverCIRunner'
```

Even though `discover_jenkins` doesn't use app names to discover tests, it does use them to handle tasks like coverage and pylint. Add your desired apps to setting called `TEST_PROJECT_APPS`:

```
TEST_PROJECT_APPS = (  
    'my_project.my_app',  
    'my_project.my_other_app',  
)
```

1.2 Usage

Run Django's test management command with the `--jenkins` option:

```
python manage.py test --jenkins
```

If you have not specified a different directory, the output will go to a directory called "reports" under your current working directory. You can use this output in Jenkins to measure your results.

Settings

- TEST_TASKS

List of tasks handled by the test runner when the `--jenkins` option is active.

Default value:

```
TEST_TASKS = (  
    'discover_jenkins.tasks.run_pylint.PyLintTask',  
    'discover_jenkins.tasks.with_coverage.CoverageTask',  
)
```

- TEST_OUTPUT_DIR

The directory that the reports generated by your tasks are saved to.

Default value:

```
TEST_OUTPUT_DIR = 'reports'
```

- TEST_PROJECT_APPS

The list of apps to run tasks from. This is not used for test discover, but it is used with tasks like coverage and pylint.

Default value:

```
TEST_PROJECT_APPS = ()
```


3.1 CoverageTask

`discover_jenkins.tasks.with_coverage.CoverageTask`

Reports test coverage across your apps. Uses the `TEST_PROJECT_APPS` setting.

3.1.1 Settings

- `TEST_COVERAGE_WITH_MIGRATIONS`

Measure test coverage in your South migrations.

Default value:

```
TEST_COVERAGE_WITH_MIGRATIONS = False
```

- `TEST_COVERAGE_REPORT_HTML_DIR`

Directory to which HTML coverage report should be written. If not specified, no HTML report is generated.

Default value:

```
TEST_COVERAGE_REPORT_HTML_DIR = ''
```

- `TEST_COVERAGE_MEASURE_BRANCH`

Measure branch coverage.

Default value:

```
TEST_COVERAGE_MEASURE_BRANCH = True
```

- `TEST_COVERAGE_EXCLUDE_PATHS`

File paths to exclude. Can be `myapp/admin.py` or `myapp/management/*`

Default value:

```
TEST_COVERAGE_EXCLUDE_PATHS = []
```

- `TEST_COVERAGE_RCFILE`

Specify configuration file. Please note if you set the `TEST_COVERAGE_EXCLUDE_PATHS` setting, coverage will ignore your `coverage.rc` file. So if you want to customize coverage settings only use this file and not the other settings.

Default value:

```
TEST_COVERAGE_RCFILE = 'coverage.rc'
```

3.2 PyLintTask

`discover_jenkins.tasks.run_pylint.PyLintTask`

Runs pylint across your apps. Uses the `TEST_PROJECT_APPS` setting.

3.2.1 Settings

- `TEST_PYLINT_RCFILE`

Specify configuration file.

Default value:

```
TEST_PYLINT_RCFILE = 'pylint.rc'
```

3.3 JSHintTask

`discover_jenkins.tasks.run_jshint.JSHintTask`

Runs jshint across your apps. Uses the `TEST_PROJECT_APPS` setting.

3.3.1 Settings

- `TEST_JSHINT_CHECKED_FILES`

If provided, check only the specified files.

Default value:

```
TEST_JSHINT_CHECKED_FILES = None
```

- `TEST_JSHINT_RCFILE`

Specify configuration file.

Default value:

```
TEST_JSHINT_RCFILE = None
```

- `TEST_JSHINT_EXCLUDE`

Python `fnmatch` patterns for excluded files.

Default value:

```
TEST_JSHINT_EXCLUDE = []
```

3.4 SlocCountTask

`discover_jenkins.tasks.run_sloccount.SlocCountTask`

Run sloccount across your apps. Uses the `TEST_PROJECT_APPS` setting.

3.5 Pep8Task

`discover_jenkins.tasks.run_pep8.Pep8Task`

Run pep8 across your apps. Uses the `TEST_PROJECT_APPS` setting.

3.6 Flake8Task

`discover_jenkins.tasks.run_flake8.Flake8Task`

Run flake8 across your apps. Uses the `TEST_PROJECT_APPS` setting.

Why?

The overall goal is to run tests on Jenkins the same way you do on your local machine. This project is designed to take advantage of `django-discover-runner`, which is the default test runner in Django 1.6. Instead of using a setting to list which apps should be tested, or accepting Django-specific test labels, it uses the official test discovery feature of the new `unittest2`.

Also, the original `django-jenkins` project doesn't take advantage of improvements to testing introduced in Django 1.4. Special management commands are no longer needed, as test runners themselves can add options that are handled by the built-in `test` command.

What's Changed?

- **It doesn't override the suite construction.** Use the included test runner based on the `django-discover-runner`, or use the included mixin to add Jenkins functionality to your own runner. Your test suite will be built the same way on Jenkins as it is on your local machine.
- **No management commands are provided.** Since Django 1.4, the built in 'test' command has allowed the test runner to define additional options that the management command will accept.
- **It doesn't use signals.** Instead of using the event/callback style of signals and using `inspect.getmembers` to connect everything, the test runner simply checks for key methods on each task the same way Django's request handler checks for methods on middleware.
- **Not everything works yet.** Only a subset of the `django-jenkins` tasks have been ported at this point. I'd love to accept your pull requests to add more of them.

Who?

First and foremost, the authors of [django-jenkins](#) are responsible for the basis of most of this code. This project just took it apart and put it back together again in a different way, then fixed or reworked some things. Thank you to the contributors of that project!

For the full list of original authors and for the contributors to this project, see the `AUTHORS.md` file.