

---

# **Destral Documentation**

***Release 0.20.1***

**GISCE-TI, S.L.**

January 16, 2017



<b>1</b>	<b>Destral testing library</b>	<b>3</b>
<b>2</b>	<b>Running tests</b>	<b>5</b>
<b>3</b>	<b>Configuring destral</b>	<b>7</b>
3.1	Variables definition . . . . .	7
<b>4</b>	<b>Configuring OpenERP</b>	<b>9</b>
4.1	Useful variables . . . . .	9
<b>5</b>	<b>API</b>	<b>11</b>
5.1	destral.openerp . . . . .	11
5.2	destral.patch . . . . .	11
5.3	destral.testing . . . . .	12
5.4	destral.transaction . . . . .	12
5.5	destral.utils . . . . .	12
5.6	destral.cover . . . . .	13
<b>6</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



Release v0.20.1

Contents:



---

## Destral testing library

---

A library to do tests with our OpenERP Server (v5)

---

**Note:** The API is not stable, and can be changed.

---

Code as if the next guy to maintain your code is a homicidal maniac who knows where you live.

—Kathy Sierra and Bert Bates





---

## Running tests

---

To run your tests call *destral* from the command line. Options to the *destral* command are:

```
Usage: destral [OPTIONS]

Options:
  --enable-coverage
  --report-coverage
  --dropdb/no-dropdb
  --requirements/no-requirements
  -m, --modules TEXT
  -t, --tests TEXT
  --help                Show this message and exit.
```

where *-m* are the modules to be tested and *-t* the tests to run. For the *-t* parameter is needed to pass full test class and method. Eg: `SomeTestClass.test_method`

If no specific tests are defined into the module, *destral* will test:

- Correct installation of the module.
- All the views defined in the module are ok.
- Definition of access rules for all the models of the module

If you have run with the *--enable-coverage* option a *.coverage* file will be generated with the results and you can see the report executing:

```
$ coverage report
```

If you want to reuse the database you can use *--no-dropdb* and this will keep the database after running all the tests. Then you can make *destral* use this database exporting the environment variable *OPEN-ERP\_DB\_NAME=database\_name*

If you are running the *destral* in local environment with all the requirements installed and you want to speed up the tests you can use the option *--no-requirements* to skip the requirements installation.



---

## Configuring destr

---

All the configuration can be made with environment variables with prefix *DESTRAL\_*

### 3.1 Variables definition

***DESTRAL\_MODULE:*** Module to test

***DESTRAL\_USE\_TEMPLATE:*** If we want to use a template when creating a database for test, if this variable is set to *True* a database with name *base* must exists and will be used to create a temporally database for this test.



---

## Configuring OpenERP

---

All configuration for OpenERP server can be made with environment variables with prefix *OPENERP\_* and will overwrite the defaults in *tools.config*.

### 4.1 Useful variables

***OPENERP\_DB\_NAME***: A database to use, if this is set then the test framework assumes that the module is already installed and this database will not be deleted after the tests.

***OPENERP\_UPDATE***: If you want to update some modules during the initialization of OpenERP service Eg value:  
*OPENERP\_UPDATE*={'base': 1, 'crm': 1}

---

**Note:** When developing is useful to have a database initialized and use *OPENERP\_DB\_NAME* variable to speed up the test.

---



## 5.1 destr.al.openerp

```
destral.openerp.DEFAULT_USER = 1
    Default user id
```

**class** destr.al.openerp.OpenERPService (\*\*kwargs)  
OpenERP Service.

**create\_database** (template=True)  
Creates a new database.

**Parameters** **template** – use a template (name must be *base*) (default True)

**db\_name**  
Database name.

**drop\_database** ()  
Drop database from *self.db\_name*

**install\_module** (module)  
Installs a module

**Parameters** **module** – Module to install

```
destral.openerp.patched_pool_jobs (*args, **kwargs)
    Patch pool jobs
```

## 5.2 destr.al.patch

**class** destr.al.patch.PatchNewCursors  
Util to patch creation of new cursor.

This will always return the cursor created by Transaction

**class** destr.al.patch.PatchedConnection (connection, cursor)  
Patched connection wrapper to return the same cursor.

This is useful when some method inside a testing method creates new cursors.

**Parameters**

- **connection** – Original connection
- **cursor** – Original cursor

**cursor** (*serialized=False*)

Wrapped function to return the same cursor

## 5.3 destral.testing

**class** `destral.testing.OOTestCase` (*methodName='runTest'*)

Base class to inherit test cases from for OpenERP Testing Framework.

**database**

Database used in the test.

**require\_demo\_data** = **False**

Require demo data to run the tests.

`destral.testing.get_spec_suite` (*module*)

Get spec suite to run Mamba :param module: Module to get the spec suite :return: suite

`destral.testing.get_unittest_suite` (*module, tests=None*)

Get the unittest suit for a module

`destral.testing.run_spec_suite` (*suite*)

Run Spec suite :param suite: mamba Runner :return:

`destral.testing.run_unittest_suite` (*suite*)

Run test suite

## 5.4 destral.transaction

**class** `destral.transaction.Singleton` (*mcs, name, bases, dict\_*)

Metaclass for singleton pattern.

**Copyright** Tryton Project

**class** `destral.transaction.Transaction`

Transaction object

**get\_context** ()

Loads the context of the current user

**start** (*database\_name, user=1, context=None*)

Start a new transaction

**Parameters**

- **database\_name** – Database name
- **user** – User id
- **context** – Context to be used

**stop** ()

Stop the transaction.

## 5.5 destral.utils

`destral.utils.update_config` (*config, \*\*kwargs*)

Updates config dictionary from keyword arguments.



`destral.utils.detect_module(path)`

Detect if a path is part of a openerp module or not

**Parameters** `path` – to examine

**Returns** None if is not a module or the module name

`destral.utils.module_exists(module)`

Check if a python module exists.

This is used to check if a module have its own tests defined, Eg: *addons.module\_name.tests*

**Parameters** `module` – Module name to check

**Returns** True if exists or False if not

`destral.utils.get_dependencies(module, addons_path=None, deps=None)`

Get all the dependencies of a module without database

Using `__terp__.py` files and is used to check requirements.txt in the dependencies.

**Parameters**

- **module** – Module to find the dependencies
- **addons\_path** – Path to find the modules

**Returns** a list of dependencies.

`destral.utils.find_files(diff)`

Return all the files implicated in a diff

`destral.utils.install_requirements(module, addons_path)`

Install module requirements and its dependencies

## 5.6 destral.cover



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## d

`destral.cover`, [13](#)  
`destral.openerp`, [11](#)  
`destral.patch`, [11](#)  
`destral.testing`, [12](#)  
`destral.transaction`, [12](#)  
`destral.utils`, [12](#)



## C

`create_database()` (destral.openerp.OpenERPService method), 11  
`cursor()` (destral.patch.PatchedConnection method), 12

## D

`database` (destral.testing.OOTestCase attribute), 12  
`db_name` (destral.openerp.OpenERPService attribute), 11  
`DEFAULT_USER` (in module destral.openerp), 11  
`destral.cover` (module), 13  
`destral.openerp` (module), 11  
`destral.patch` (module), 11  
`destral.testing` (module), 12  
`destral.transaction` (module), 12  
`destral.utils` (module), 12  
`detect_module()` (in module destral.utils), 13  
`drop_database()` (destral.openerp.OpenERPService method), 11

## F

`find_files()` (in module destral.utils), 13

## G

`get_context()` (destral.transaction.Transaction method), 12  
`get_dependencies()` (in module destral.utils), 13  
`get_spec_suite()` (in module destral.testing), 12  
`get_unittest_suite()` (in module destral.testing), 12

## I

`install_module()` (destral.openerp.OpenERPService method), 11  
`install_requirements()` (in module destral.utils), 13

## M

`module_exists()` (in module destral.utils), 13

## O

`OOTestCase` (class in destral.testing), 12  
`OpenERPService` (class in destral.openerp), 11

## P

`patched_pool_jobs()` (in module destral.openerp), 11  
`PatchedConnection` (class in destral.patch), 11  
`PatchNewCursors` (class in destral.patch), 11

## R

`require_demo_data` (destral.testing.OOTestCase attribute), 12  
`run_spec_suite()` (in module destral.testing), 12  
`run_unittest_suite()` (in module destral.testing), 12

## S

`Singleton` (class in destral.transaction), 12  
`start()` (destral.transaction.Transaction method), 12  
`stop()` (destral.transaction.Transaction method), 12

## T

`Transaction` (class in destral.transaction), 12

## U

`update_config()` (in module destral.utils), 12