Testing Modules

Odoo provides support for testing modules using unittest.

To write tests, simply define a tests sub-package in your module, it will be automatically inspected for test modules. Test modules should have a name starting with test\_ and should be imported from tests/\_\_init\_\_.py, e.g.

your\_module

|-- ...

`-- tests

|-- \_\_init\_\_.py

|-- test\_bar.py

`-- test\_foo.py

and \_\_init\_\_.py contains:

from . import test\_foo, test\_bar

Warning

test modules which are not imported from tests/\_\_init\_\_.py will not be run

Changed in version 8.0: previously, the test runner would only run modules added to two lists fast\_suite and checks in tests/\_\_init\_\_.py. In 8.0 it will run all imported modules

The test runner will simply run any test case, as described in the official [unittest documentation](https://docs.python.org/2/library/unittest.html), but Odoo provides a number of utilities and helpers related to testing Odoo content (modules, mainly):

***class* odoo.tests.common.TransactionCase(*methodName='runTest'*)**

TestCase in which each test method is run in its own transaction, and with its own cursor. The transaction is rolled back and the cursor is closed after each test.

**browse\_ref(*xid*)**

Returns a record object for the provided [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier)

Parameters

**xid** -- fully-qualified [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), in the form *module*.*identifier*

Raise

ValueError if not found

Returns

BaseModel

**ref(*xid*)**

Returns database ID for the provided [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), shortcut for get\_object\_reference

Parameters

**xid** -- fully-qualified [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), in the form *module*.*identifier*

Raise

ValueError if not found

Returns

registered id

***class* odoo.tests.common.SingleTransactionCase(*methodName='runTest'*)**

TestCase in which all test methods are run in the same transaction, the transaction is started with the first test method and rolled back at the end of the last.

**browse\_ref(*xid*)**

Returns a record object for the provided [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier)

Parameters

**xid** -- fully-qualified [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), in the form *module*.*identifier*

Raise

ValueError if not found

Returns

BaseModel

**ref(*xid*)**

Returns database ID for the provided [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), shortcut for get\_object\_reference

Parameters

**xid** -- fully-qualified [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), in the form *module*.*identifier*

Raise

ValueError if not found

Returns

registered id

***class* odoo.tests.common.SavepointCase(*methodName='runTest'*)**

Similar to SingleTransactionCase in that all test methods are run in a single transaction *but* each test case is run inside a rollbacked savepoint (sub-transaction).

Useful for test cases containing fast tests but with significant database setup common to all cases (complex in-db test data): setUpClass() can be used to generate db test data once, then all test cases use the same data without influencing one another but without having to recreate the test data either.

***class* odoo.tests.common.HttpCase(*methodName='runTest'*)**

Transactional HTTP TestCase with url\_open and phantomjs helpers.

**browse\_ref(*xid*)**

Returns a record object for the provided [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier)

Parameters

**xid** -- fully-qualified [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), in the form *module*.*identifier*

Raise

ValueError if not found

Returns

BaseModel

**phantom\_js(*url\_path*, *code*, *ready='window'*, *login=None*, *timeout=60*, *\*\*kw*)**

Test js code running in the browser - optionnally log as 'login' - load page given by url\_path - wait for ready object to be available - eval(code) inside the page

To signal success test do: console.log('ok')

To signal failure do: console.log('error')

If neither are done before timeout test fails.

**ref(*xid*)**

Returns database ID for the provided [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), shortcut for get\_object\_reference

Parameters

**xid** -- fully-qualified [external identifier](https://www.odoo.com/documentation/10.0/glossary.html#term-external-identifier), in the form *module*.*identifier*

Raise

ValueError if not found

Returns

registered id

By default, tests are run once right after the corresponding module has been installed. Test cases can also be configured to run after all modules have been installed, and not run right after the module installation:

**odoo.tests.common.at\_install(*flag*)**

Sets the at-install state of a test, the flag is a boolean specifying whether the test should (True) or should not (False) run during module installation.

By default, tests are run right after installing the module, before starting the installation of the next module.

**odoo.tests.common.post\_install(*flag*)**

Sets the post-install state of a test. The flag is a boolean specifying whether the test should or should not run after a set of module installations.

By default, tests are *not* run after installation of all modules in the current installation set.

The most common situation is to use [TransactionCase](https://www.odoo.com/documentation/10.0/reference/testing.html#odoo.tests.common.TransactionCase) and test a property of a model in each method:

class TestModelA(common.TransactionCase):

def test\_some\_action(self):

record = self.env['model.a'].create({'field': 'value'})

record.some\_action()

self.assertEqual(

record.field,

expected\_field\_value)

# other tests...

**Running tests**

Tests are automatically run when installing or updating modules if [--test-enable](https://www.odoo.com/documentation/10.0/reference/cmdline.html#cmdoption-odoo-bin--test-enable) was enabled when starting the Odoo server.

As of Odoo 8, running tests outside of the install/update cycle is not supported.