2016

odoo

EXPERIENCE

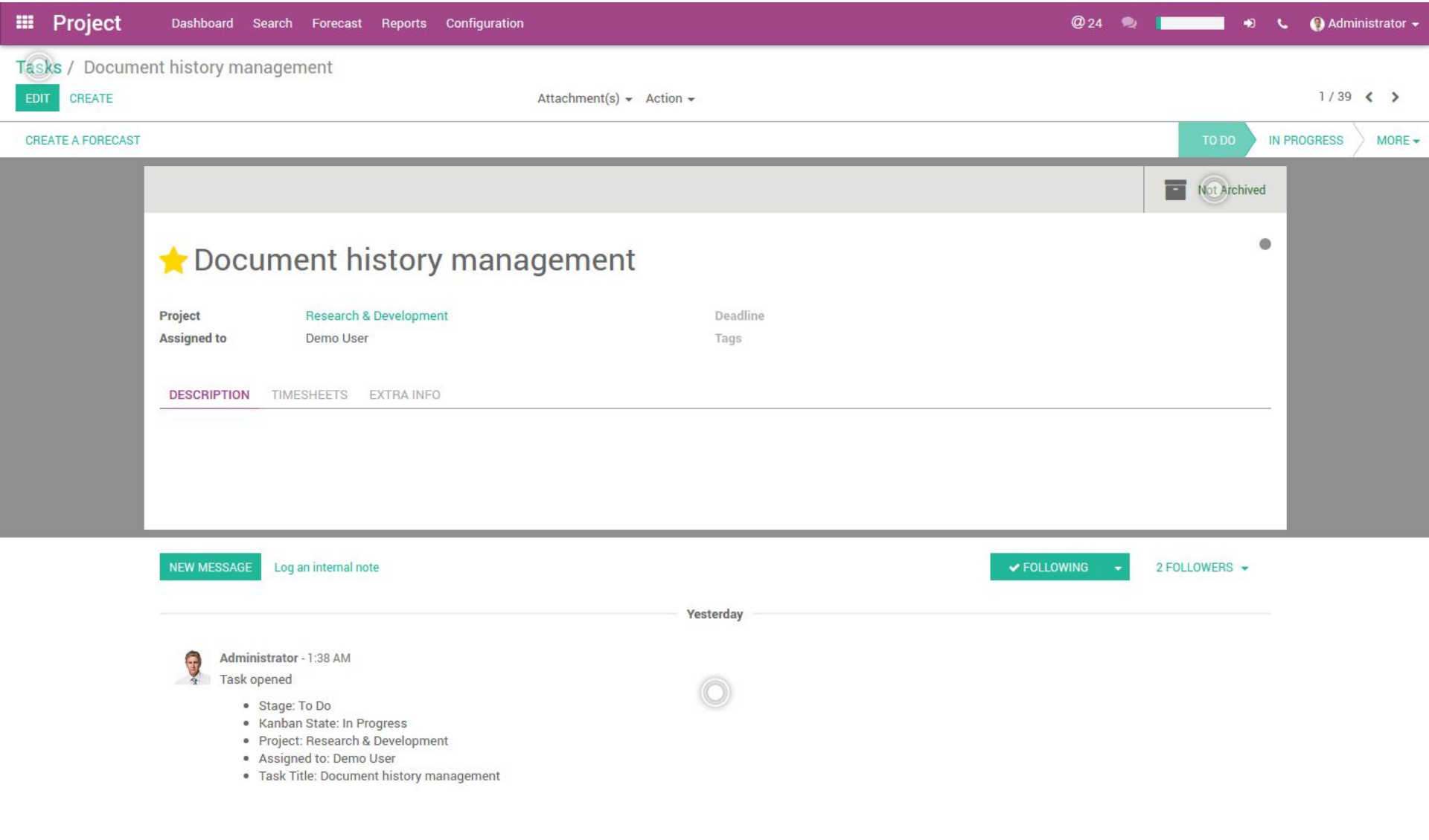
The new way to develop automated tests & tours

Aaron BOHY• R&D Developer - Framework team

1. Introduction
2. Demo
3. How to write a tour?
4. Advanced features

1 Introduction

Odoo v9


Project
Dashboard
Search
Forecast
Reports
Configuration
24
Administrator
Tasks / Document history management
EDIT
CREATE
Attachment(s)
Action
1 / 39
CREATE A FORECAST
TO DO
IN PROGRESS
MORE
Not Archived
Document history management
Project
Research & Development
Deadline
Assigned to
Demo User
Tags
DESCRIPTION
TIMESHEETS
EXTRA INFO
NEW MESSAGE
Log an internal note
FOLLOWING
2 FOLLOWERS
Yesterday
Administrator - 1:38 AM
Task opened
Stage: To Do
Kanban State: In Progress
Project: Research & Development
Assigned to: Demo User
Task Title: Document history management

Introduction

- Motivation: improve user onboarding
- Notion of sequence of tips instead of isolated and unrelated tips
- Tours can be executed automatically for testing purposes
- Allow to test flows and UI
- Available in the backend, the website, the POS, iframes...

Demo

3 How to write a tour?

Define and register your tour

- Define a new JS module
- Import the **web_tour.tour** module
- Register your tour with its name and an array of steps

```
odoo.define('my_addon.tour', function(require) {
"use strict";

var core = require('web.core');
var tour = require('web_tour.tour');

var _t = core._t;

var steps = []; /* Array of steps **/

tour.register('my_tour', steps);

});
```

Define your steps

A step is an object with the following keys:

- **trigger** (mandatory)
  - css selector of the element on which to attach the tip
- **extra_trigger**
  - additional css selector that must match a DOM element for the tip to be displayed
- **content**
  - the **html** content of the tip
- **position**
  - right (default), left, top or bottom
- **width**
  - default: 270px

```
var steps = [{
    trigger: '.o_app[data-menu-xmlid="sales_team.menu_base_partner"]',
    content: _t("Ready to boost your sales? Your <b>sales pipeline</b> can be found here."),
    position: 'bottom',
}];
```

Automated tests

Define a python module to execute your tour with unittest:

- Define a **tests** subpackage in your addon
- Test modules should have a name starting with **test_**
- Import your module in **tests/__init__.py**
- Write a test that runs your tour from a given url, and logged in as a given user

```
import odoo.tests

@odoo.tests.common.at_install(False)
@odoo.tests.common.post_install(True)
class TestUi(odoo.tests.HttpCase):
    def test_01_my_tour(self):
        self.phantom_js("/web",
                        "odoo.__DEBUG__.services['web_tour.tour'].run('my_tour')",
                        "odoo.__DEBUG__.services['web_tour.tour'].tours.my_tour.ready",
                        login="admin")
```

Advanced features

Tour options

```
odoo.define('my_addon.tour', function(require) {
"use strict";

var tour = require('web_tour.tour');

var options = {};
var steps = []; /* Array of steps **/

tour.register('my_tour', options, steps);

});
```

Optional second argument with the following keys:

- url
  - the url to load before starting the tour
- skip_enabled
  - set to *true* to add a link to skip the tour in its tips
- test
  - set to *true* if the tour is dedicated to tests
- wait_for
  - a deferred that must be resolved for the tour to be ready

Helpers to write your selectors

- **:containsExact(*value*)**
  - matches if the content of the element is exactly *value*
  - e.g. ‘span:containsExact(OdooExperience)’
    - ✔ OdooExperience
    - ✘ OdooExperience 2016
- **:containsExactCase(*value*)**
  - like containsExact but case sensitive
- **:containsRegex(*value*)**
  - like containsExact but with a regular expression
- **:propValueContains(*value*)**
  - matches if the element’s **value** property contains to *value*

Automatic steps: run key

Defines the operation to perform when the tour is executed automatically

Use some predefined helpers:

```
var steps = [{
    trigger: ".o_list_editable .o_form_required input",
    run: "text Ipad",
}];
```

- **text(*value*)**
  - writes *value* in the element (handles **select** elements as well)
- **click**
  - clicks on the element
- **drag_and_drop(*to*)**
  - drags and drops the element inside the element matching the given css selector
- **keydown(*keycodes*)**
  - simulates a keydown event on the element for the given *keycodes*
- **auto**
  - default, calls **click** or **text** according to the element's type

Or directly set a function's prototype to perform any DOM manipulation

Put a tip on a menu

✘ Avoid poor selectors like
- ‘.o_app:nth-child(2)’
- ‘.o_app:contains("CRM")’

✓ Write a selector based on the **xmlid**:
- ‘.o_app[data-menu-xmlid="sales_team.menu_base_partner"]’

By default, menu xmlids are not loaded in the webclient

⇒ Set the flag **load_xmlid** to *true*:

```xml
<?xml version="1.0" encoding="utf-8"?>
<odoo>
    <record id="sales_team.menu_base_partner" model="ir.ui.menu">
        <field name="load_xmlid" eval="True"/>
    </record>
</odoo>
```

2016

odoo

EXPERIENCE

Thank you. Questions?

#odooexperience