



Coverage for my-modules/aspire360_measures/tests/test_aspire360.py : 95%

147 statements 140 run 7 missing 0 excluded

```

1  # -*- coding: utf-8 -*-
2  # Part of Odoo. See LICENSE file for full copyright and licensing details.
3
4  from odoo.tests import *
5  from odoo.tests import HttpCase, common, tagged
6  from odoo import _
7  from odoo import http
8  from odoo.http import request
9  # from odoo.addons.survey.tests import common
10
11 # Tests for model
12 class TestModelA(common.TransactionCase):
13     print(" IN TESTING TEST MODEL A FROM ASPIRE360")
14
15     # Test 1: Check Odoo res user to make sure that unique ID is intact after install
16     @tagged('-standard', 'aspire360')
17     def test_assert_unique_uid_at_installation(self):
18         print("In test_assert_unique_uid_at_installation test")
19         uid = self.uid
20         print("Testing uid is: ", uid)
21         print("Odoo user uid is: ", request.env.context.get ('uid'))
22         record = self.env['res.users'].search([('id', '=', request.env.context.get ('uid'))])
23         # Check that uid created is unique
24         self.assertEqual(len(record),1)
25
26     # Test 2: Ran at initial install. Checks to make sure that user doesn't already exist in entrepreneur or VC
27     @tagged('-standard', 'aspire360')
28     def test_assert_no_entrepreneur_or_vc_entry_at_installation(self):
29         print("In test_assert_no_entrepreneur_or_vc_entry_at_installation")
30         check_entrepreneur = self.env['aspire360.entrepreneurs'].search([])
31         check_vc = self.env['aspire360.venturecapitalists'].search([])
32         if len(check_entrepreneur) == 0 and len(check_vc) == 0:
33             #Check that no survey entrey associated
34             num_surveys = self.env['survey.user_input'].search([('aspire_entrepreneur', '=', request.env.context.get ('uid'))])
35             if self.assertEqual(len(num_surveys),0):
36                 print("Passed test: test_assert_no_entrepreneur_or_vc_entry_at_installation!!")
37         else:
38             self.assertTrue(True)
39
40     # Test 3: Assert that assigned Entrepreneurs for surveys actually exist
41     @tagged('-standard', 'aspire360')
42     def test_assert_survey_entrepreneur_exists(self):
43         print("In test_assert_survey_entrepreneur_exists")
44         records = self.env ['survey.user_input'].search([('aspire_entrepreneur', '=', request.env.context.get ('uid'))])
45         if len(records) > 0:
46             check_entrepreneur = self.env['aspire360.entrepreneurs'].search([('user_id', '=', request.env.context.get ('uid'))])
47             self.assertEqual(len(check_entrepreneur),1)
48             #TODO: Update many2many relationfields for testing
49             self.assertTrue(True)
50
51     # Test 4: Assert that assigned followed VCs for entrepreneurs actually exist
52     @tagged('-standard', 'aspire360')
53     def test_assert_vc_followed_exists(self):
54         print("In test_assert_vc_followed_exists")
55         #TODO: Update many2many relationfields for testing
56         # Change below to grab all entrepreneurs where this exists
57         # followed_entrepreneurs = self.env['aspire360.entrepreneurs'].search([('users_id', '=', request.env.context.get ('uid'))])
58         # for ennentrepreneur in followed_entrepreneurs:
59             venturecapitalists = self.env['aspire360.venturecapitalists'].search([])
60             venturecapitalists.update_entrepreneur(self.uid,2)
61             self.assertTrue(True)
62
63     # Test 5: Assert nthat Readiness to Fundraise survey exists
64     @tagged('-standard', 'aspire360')
65     def test_assert_fundraise_survey_exists(self):
66         print("In test_assert_fundraise_survey_exists")
67         surveys = self.env['survey.survey'].search([('title', '=', 'Readiness to Fundraise Assessment')])
68         num_surveys = len(surveys)
69         self.assertEqual(num_surveys,1)
70
71     # Test 6: Assert that Readiness to Fundraise survey exists

```

```

72 | @tagged('-standard', 'aspire360')
73 | def test_assert_sell_survey_exists(self):
74 |     print("In test_assert_sell_survey_exists")
75 |     surveys = self.env['survey.survey'].search([('title', '=', 'Readiness to Sell Assessment')])
76 |     num_surveys = len(surveys)
77 |     self.assertEqual(num_surveys,1)
78 |
79 | # Test 7: Creating entrepreneur profile in odoo model
80 | @tagged('-standard', 'aspire360')
81 | def test_create_entrepreneur(self):
82 |     new_record = {'name':'Testing User', 'user_id':999999}
83 |     entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
84 |     before_len = len(entrepreneurs)
85 |     print("before length is: ", before_len)
86 |     entrepreneurs.create(new_record)
87 |     entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
88 |     after_len = len(entrepreneurs)
89 |     print("before length is: ", before_len)
90 |     self.assertEqual(before_len+1,after_len)
91 |     # pass
92 |
93 | # Test 8: Creating vc profile in odoo model
94 | @tagged('-standard', 'aspire360')
95 | def test_create_venturecapitalist(self):
96 |     new_record = {'name':'Testing User', 'user_id':999998}
97 |     venturecapitalists = self.env['aspire360.venturecapitalists'].search([])
98 |     before_len = len(venturecapitalists)
99 |     print("before length is: ", before_len)
100 |     venturecapitalists.create(new_record)
101 |     venturecapitalists = self.env['aspire360.venturecapitalists'].search([])
102 |     after_len = len(venturecapitalists)
103 |     print("before length is: ", before_len)
104 |     self.assertEqual(before_len+1,after_len)
105 |     # pass
106 |
107 | # Test 9: Test updating user profile Odoo
108 | @tagged('-standard', 'aspire360')
109 | def test_entrepreneur_edit_profile(self):
110 |     entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
111 |     data = {
112 |         "company_name": "Aspire360",
113 |         "industry": "Administrative Services",
114 |         "employees": "1-10 employees",
115 |         "funding_stage": "Pre-seed"
116 |     }
117 |     entrepreneurs.edit_profile(data,2)
118 |     entrepreneurs = self.env['aspire360.entrepreneurs'].search([('company_name', '=', 'Aspire360')])
119 |     self.assertTrue(len(entrepreneurs)>0)
120 |
121 | # Tests for controller
122 | """ --- Entrepreneur Role --- """
123 | class WebsiteEntrepreneurTests(HttpCase, TransactionCase):
124 |     #Test: Test controller methods for new entrepreneur setup works
125 |     @tagged('-standard', 'aspire360')
126 |     def test_e_creation(self):
127 |         # url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/"
128 |         # self.url_open(url)
129 |         url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/e"
130 |         self.url_open(url)
131 |         entrepreneurs = self.env['aspire360.entrepreneurs'].search([('user_id', '=', self.uid)])
132 |         self.assertEqual(len(entrepreneurs),1)
133 |
134 | #Test 10: Testing controller index as entrepreneur
135 | @tagged('-standard', 'aspire360')
136 | def test_e_index(self):
137 |     #authenticate:
138 |     self.authenticate("admin","admin")
139 |     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures"
140 |     print("URL is: ", url)
141 |     self.assertTrue(self.url_open(url))
142 |
143 | # Test edit_profile controller form access works
144 | @tagged('-standard', 'aspire360')
145 | def test_controller_entrepreneur_edit_profile(self):
146 |     self.authenticate("admin","admin")
147 |     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/entrepreneur_edit_prof:
148 |     self.assertTrue(self.url_open(url))

```

```

149
150 # Test submitting entrepreneur info to controller for profile update
151 @tagged('-standard', 'aspire360')
152 def test_controller_entrepreneur_submit_info(self):
153     self.authenticate("admin", "admin")
154     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/submit_info"
155     data = {
156         "company_name": "Aspire360",
157         "industry": "Administrative Services",
158         "employees": "1-10 employees",
159         "funding_stage": "Pre-seed"
160     }
161     self.url_open(url, data=data)
162     entrepreneurs = self.env['aspire360.entrepreneurs'].search([('company_name', '=', 'Aspire360')])
163     self.assertTrue(len(entrepreneurs)>0)
164
165 # Test entrepreneur can start fundraise survey
166 @tagged('-standard', 'aspire360')
167 def test_controller_entrepreneur_take_fundraise(self):
168     self.authenticate("admin", "admin")
169     prev_num_surveys = len(self.env['survey.user_input'].search([]))
170     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/survey/fundraise"
171     self.url_open(url)
172     after_num_surveys = len(self.env['survey.user_input'].search([]))
173     self.assertEqual(prev_num_surveys+1, after_num_surveys)
174
175 # Test entrepreneur can start sell survey
176 @tagged('-standard', 'aspire360')
177 def test_controller_entrepreneur_take_sell(self):
178     self.authenticate("admin", "admin")
179     prev_num_surveys = len(self.env['survey.user_input'].search([]))
180     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/survey/sell"
181     self.url_open(url)
182     after_num_surveys = len(self.env['survey.user_input'].search([]))
183     self.assertEqual(prev_num_surveys+1, after_num_surveys)
184
185 """ --- VC Role --- """
186 class WebsiteVentureCapitalistTests(HttpCase, TransactionCase):
187     #Test: Test controller methods for new entrepreneur setup works
188     @tagged('-standard', 'aspire360')
189     def test_v_creation(self):
190         # url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/"
191         # self.url_open(url)
192         url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/v"
193         self.url_open(url)
194         venturecapitalists = self.env['aspire360.venturecapitalists'].search([('user_id', '=', self.uid)])
195         self.assertEqual(len(venturecapitalists), 1)
196
197 #Test 11: Testing controller index as venturecapitalist
198 @tagged('-standard', 'aspire360')
199 def test_v_index(self):
200     #authenticate:
201     self.authenticate("phananh1096@gmail.com", "testuser123")
202     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures"
203     print("URL is: ", url)
204     self.assertTrue(self.url_open(url))
205
206 # Test email controller works
207 @tagged('-standard', 'aspire360')
208 def test_controller_vc_email(self):
209     self.authenticate("phananh1096@gmail.com", "testuser123")
210     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/email"
211     self.assertTrue(self.url_open(url))
212
213 # Test email helper for Initial contact
214 @tagged('-standard', 'aspire360')
215 def test_controller_vc_email_helper(self):
216     self.authenticate("phananh1096@gmail.com", "testuser123")
217     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/submit_email"
218     data = {
219         "email_recipient": "phananh1096@gmail.com",
220         "email_sender": "pn2363@columbia.edu",
221         "email_template": "Initial reachout"
222     }
223     self.assertTrue(self.url_open(url, data=data))
224
225 # Test email helper for Follow-up

```

```
226 | @tagged('-standard', 'aspire360')
227 | def test_controller_vc_email_helper(self):
228 |     self.authenticate("phananh1096@gmail.com","testuser123")
229 |     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/submit_email"
230 |     data = {
231 |         "email_recipient": "phananh1096@gmail.com",
232 |         "email_sender": "pn2363@columbia.edu",
233 |         "email_template": "Follow-up"
234 |     }
235 |     self.assertTrue(self.url_open(url, data=data))
236 |
237 | # Test controller for search
238 | @tagged('-standard', 'aspire360')
239 | def test_controller_vc_search(self):
240 |     self.authenticate("phananh1096@gmail.com","testuser123")
241 |     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/search"
242 |     data = {
243 |         "company_name": "Aspire360",
244 |         "industry": "Administrative Services",
245 |         "employees": "1-10 employees",
246 |         "funding_stage": "Pre-seed"
247 |     }
248 |     self.assertTrue(self.url_open(url, data=data))
249 |
250 | # Test controller for view fundraise survey/profile
251 | @tagged('-standard', 'aspire360')
252 | def test_controller_vc_view_fundraise(self):
253 |     self.authenticate("phananh1096@gmail.com","testuser123")
254 |     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/display_fundraise"
255 |     data = {
256 |         "user_id": 2
257 |     }
258 |     self.assertTrue(self.url_open(url, data=data))
259 |
260 | # Test controller for view sell survey/profile
261 | @tagged('-standard', 'aspire360')
262 | def test_controller_vc_view_sell(self):
263 |     self.authenticate("phananh1096@gmail.com","testuser123")
264 |     url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/display_sell"
265 |     data = {
266 |         "user_id": 2
267 |     }
268 |     self.assertTrue(self.url_open(url, data=data))
269 |
270 | # print("{} tests completed, {} tests passed.".format(test_count, test_passed))
271 |
272 | # test_assert_unique_uid_at_installation()
```