

Framework Practice Exercise

1. Log into `letskodeit.teachable.com` using "`test@email.com`" and "`abccabc`" or use your own credentials. Please do not change the default credentials of test account. I will have to again reset it.
2. Search for "JavaScript" or any other course in the search box
3. Select the course you searched to go to the course detail page
4. Click the "Enroll in Course for \$Amount" button
5. Scroll to the bottom of the page
6. Enter some invalid card details
7. Click "Enroll in Course" button
8. Verify the error message

Automation Steps

Create Page Class by following these steps

Create a new package under "pages" with name "courses"
It should create a `__init__.py` file also

Create a new Python file, give it a name
"`register_courses_page.py`"

This class should inherit from `BasePage`

The initial declaration including constructor should look like this:

```
import utilities.custom_logger as cl
import logging
from base.basepage import BasePage

class RegisterCoursesPage(BasePage):

    log = cl.customLogger(logging.DEBUG)

    def __init__(self, driver):
        super().__init__(driver)
        self.driver = driver
```

Declare Locators for these elements:

```
_search_box = ""
_course= ""
_all_courses= ""
_enroll_button = ""
_cc_num = ""
_cc_exp = ""
_cc_cvv = ""
_submit_enroll = ""
_enroll_error_message = ""
```

Then write methods to perform these actions:

def enterCourseName(self, name):

def selectCourseToEnroll(self, fullCourseName):

def enterCardNum(self, num):

def enterCardExp(self, exp):

def enterCardCVV(self, cvv):

def clickEnrollSubmitButton(self):

def enterCreditCardInformation(self, num, exp, cvv):

Hint: Call all three methods to enter card details

def enrollCourse(self, num="", exp="", cvv=""):

Hint:

- Click on the enroll button
- Scroll down
- Enter credit card information
- Click Enroll in course button

def verifyEnrollFailed(self):

Verify the element for error message is displayed, not just present.

You need to verify if it is displayed

Hint: The element is not instantly displayed, it take some time to display

You need to wait for it to display

Create a Test Class by following these steps

Create a new package under "tests" with name "courses"

It should create a `__init__.py` file also

Create a new Python file, give it a name

"register_courses_tests.py"

Do everything similar to LoginTests

Examples:

- Import page class, TestStatus class
- Use fixtures

It should look like this:

```
from pages.courses.register_courses_pages import RegisterCoursesPage
from utilities.teststatus import TestStatus
import unittest
import pytest

@pytest.mark.usefixtures("oneTimeSetUp", "setUp")
class RegisterCoursesTests(unittest.TestCase):

    @pytest.fixture(autouse=True)
    def classSetup(self, oneTimeSetUp):
        self.courses = RegisterCoursesPage(self.driver)
        self.ts = TestStatus(self.driver)
```

@pytest.mark.run(order=1)

def test_invalidEnrollment(self):

- Call required methods from the page class to perform the test
- Enter course name
- Select course
- Enroll in course
- Verify error message
- Test Status.markFinal()