

تست برنامه https://github.com/Lemoncode/react-by-sample/tree/master/20_ErrorBoundaries

برای برنامه ذکر شده 2 تست زیر نوشته شده است:

1. تست تگ details

هنگامی که تگ details باز میشود، یک attribute به شکل `open=""` به آن اضافه میشود و هنگامی که بسته شود، این attribute حذف میشود
بنابر این با هنگام تست، وجود این attribute را بررسی میکنیم

ابتدا element موجود در webdriver را میگیریم (detailsTag) و بر روی آن کلیک میکنیم.
سپس attribute با نام open را از detailsTag دریافت میکنیم. در صورتی که None برگردانده شود یعنی tag بسته است. از طرفی ابتدا این تگ بسته بوده و با کلیک بر روی آن باید باز شود بنابر این در صورتی که مقدار گرفته شده توسط get_attribute برابر None بود یعنی تست پاس نشده است.

```
def first_test(self):
    print('Running Test#1')
    detailsTag = self.driver.find_element(By.XPATH,
'/html/body/div/div/div/details')
    detailsTag.click()
    self.assertTrue(detailsTag.get_attribute('open') is not None)
    print('Test#1 passed successfully')
```

2. تست ورودی name و نمایش آن

با هر حرفی که در textfield موجود وارد میشود، نام نمایش داده شده جلوی Hello user: باید آپدیت شود
برای تست آن، ابتدا element مربوط به textfield و label را میگیریم و textfield را خالی میکنیم

```
nameLabel = self.driver.find_element(By.XPATH, '/html/body/div/div/h2')
textField = self.driver.find_element(By.XPATH, '/html/body/div/div/input')
textField.clear()
```

یک فلگ با نام charactersValid با مقدار اولیه True در نظر میگیریم.
سپس رشته Johann_Sebastian_Bach را حرف به حرف به textField اضافه میکنیم و حروف اضافه شده در هر مرحله را در یک رشته با نام currentTarget ذخیره میکنیم. رشته nameLabel را دریافت میکنیم و 12 کاراکتر اول (Hello user:) و 2 کاراکتر آخر (!) را حذف میکنیم. در صورتی که مقدار رشته بدست آمده با currentTarget برابر نبود تست نباید پاس شود، پس فلگ مربوط به درستی تست (charactersValid) را برابر False قرار میدهیم و باقی حروف را بررسی نمیکنیم.

```
targetValue = 'Johann_Sebastian_Bach'
currentTarget = ''
```

```
charactersValid = True
for character in tqdm(targetValue):
    textField.send_keys(character)
    currentTarget += character
    tmpLabel = nameLabel.text[12:]
    tmpLabel = tmpLabel[:-2]
    if tmpLabel != currentTarget:
        charactersValid = False
        break
    time.sleep(0.1)
```

در صورتی که charactersValid برابر True شود تست پاس میشود.

```
self.assertTrue(charactersValid)
print('Test#2 passed successfully')
```