

Генерация отчетов Allure

В windows powershell вводим команды:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

```
iex (New-Object net.webclient).downloadstring('https://get.scoop.sh')
```

```
scoop install allure
```

убедимся в том, что всё в порядке командой allure

В Pycharm в settings добавляем два пакета: allure-pytest и allure-python-commons

```
import allure
from allure_commons.types import AttachmentType
```

в base_page добавим импорты:

```
from selenium.webdriver import Keys, ActionChains
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import WebDriverWait as wait
```

и новую функцию:

```
def element_is_visible(self, locator, timeout=10):
    return wait(self.browser,
timeout).until(EC.visibility_of_element_located(locator))
В LoginPageLocators
```

Добавим локатор

```
PROFILE = (By.XPATH, '//*[@id="app"]/header/div')
```

В файле login_page напишем функцию, которая возвращает нам искомый элемент -Profile

```
def find_profile(self):
    profile = self.browser.find_element(*LoginPageLocators.PROFILE)
    return profile
```

И теперь в самом тесте напишем ассерт:

```
profile = page.find_profile()
assert profile
```

теперь перед тестом напишем декоратор с описанием фичи, которую мы тестируем:

```
@allure.feature('user_login')
```

И ещё один декоратор, который конкретизирует данный метод:

```
@allure.story('Вводим валидный email и пароль')
```

Добавляем декоратор, который проставляет критичность теста.

Они бывают: blocker critical normal minor trivial

```
@allure.severity('blocker')
```

Добавляем отслеживание шагов:

```
with allure.step('Делаем скриншот')
    allure.attach(browser.get_screenshot_as_png(), name='result8',
attachment_type=AttachmentType.PNG)
```

Запускаем наш тест в терминале командой с указанием папки, куда будут падать отчеты:

```
pytest .\tests\test_login_page.py --alluredir results
```

Теперь идем опять в powershell, переходим в наш проект

```
cd .\PycharmProjects\
```

```
cd имя проекта
```

```
allure serve results
```

