????????

Локаторы: способ поиска элементов на странице. Поэтому они являются методами page

В VSC Ctrl+Click по методу выводит код метода.

Локатор	Описание
page.get_by_role('link', name="Docs")	Поиск элемента по роли name - текст link <a> heading <h> radio, checkbox, button

page.get_by_label("Email address")	Для выделения элементов, у которых есть привязанная метка. Например
	`<div>`
	`<label for="exampleInputEmail1" class="form-label mt-4">Email address</label>`
	`<input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">`
	`<small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>`
	`</div>`

page.get_by_placeholder("Enter email")	Поиск элементов по placeholder
page.get_by_text("Something", exact=False)	Поиск по тексту. Exact=False ищет вхождение.
page.get_by_alt_text(text)	Поиск по атрибуту alt у изображений
page.get_by_title(text)	Атрибут title

Локатор	Описание
page.locator(text)	Поиск по CSS. Можно использовать tagname, classname, id, attribute/value Примеры^ css=h1 footer <tagname>.<classname> button.btn-outline-sucess <tagname>#<idname> button#BtnGroupDrop1 <tagname>[attribute] input[readonly] <tagname>[attribute=somevalue] input[value='correct value']
	Поиск по иерархии элементов. Если через пробелы-то вложенные элементы. Если через точку - то у элемента несколько классов. Но они не обязательно непосредственно вложенные. nav.bg-dark a.nav-link.active Для непосредственного вложения: nav.bg-dark > a.nav-link.active
	Называются sudo классами, Класс и текст в теге. Для вхождения: h1:text('Navbars') Для полного соответствия: h1:text-is('Navbars') div.dropdown-menu:visible Для определения по номеру вхождения, когда их много :nth-match(button.btn-primary, 4)
	XPath Абсолютный путь: xpath=/html/head/title С любого начала: xpath=//h1/h2 С указанием атрибута xpath=//h1[@id='navbars']
	Функции XPath Для поиска по тексту, точно: //h1[text()='Headling1'] Для поиска по тексту, содержит: //h1[contains(text(), 'Headling1')] Для поиска по тексту, содержит: //h1[contains(@class, 'btn')]
Множественные условия	Поиск родительского элемента page.get_by_label("Email address").locator("..") Фильтрация page.get_by_role("heading").filter(has_text="First") По дочернему элементу page.locator("div.form-group").filter(has=page.get_by_label("Password"))

Доступ к iframe

```
'''Test for auth module'''
import pytest
```

```
from playwright.sync_api import Browser, Page, expect

AUTH_URL = "https://wood.bobrobotirk.ru/auth"

@pytest.fixture
def page_and_auth(browser: Browser):
    context = browser.new_context(
        storage_state="playwright/.auth/vk.json"
        )
    page = context.new_page()
    yield page
    context.close()

def test_first(page_and_auth: Page):
    page_and_auth.goto(AUTH_URL)
    vkframe = page_and_auth.frame(url=lambda url: "id.vk.com" in url)

    if vkframe:
        authbutton = vkframe.get_by_role("button", name="Продолжить как")
        expect(authbutton, "Кнопка Продолжить как...
отсутствует").to_be_visible(timeout=20000)
        authbutton.click()
        back_button = page_and_auth.get_by_role("button", name="Авторизация успешна")
        expect(back_button, "Сервис не произвел авторизацию").to_be_visible()
        back_button.click()
    else:
        assert False, "VK фрейм не найден."
```

Revision #5
Created 17 April 2025 09:59:45 by Admin
Updated 24 June 2025 14:28:25 by Admin